INTERACTIVE SEGMENTATION OF THE HIPPOCAMPUS FROM MAGNETIC RESONANCE IMAGES USING DEFORMABLE SHAPE TEMPLATES

by

William Vann Hasty Jr.

A thesis submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Master of Science in the
Department of Computer Science

Charlotte

1998

Approved by:

_____
Dr. Kalpathi R. Subramanian

_____
Dr. Mohammad T. Mostafavi

_____
Dr. John P. Brockway

_____
Dr. Thomas P. Weldon

ABSTRACT

William Vann Hasty Jr.
Interactive Segmentation of the Hippocampus from Magnetic Resonance Images Using Deformable Shape Templates (Under the direction of Dr. KALPATHI RAMAN SUBRAMANIAN)

It is of interest to physicians to be able to measure the hippocampus of the brain in order to study the relationship between its size and certain pathological conditions. The hippocampus is known to be smaller in patients with temporal lobe epilepsy and Alzheimer's disease. It is of clinical interest to measure large numbers of hippocampi with a reliable semi-automatic techniques. Magnetic resonance imaging provides a non-invasive means to study anatomical structures of the brain, and is particularly good at capturing sharp images of soft tissues such as the hippocampus. Traditional segmentation techniques fail with respect to segmentation of the hippocampus due to its small size, variability of shape, and incomplete boundaries. Orthogonal curves, a form of deformable shape templates, provide a means to segment organs with incomplete gradient boundaries from the surrounding tissues.

We use orthogonal curves to search for the organ boundaries by applying *a priori* knowledge of its shape and by using the external image energy to segment its most likely boundary. This semi-automatic method allows us to search for the gradient boundary of the organ while avoiding local gradient maxima that would yield inaccurate results. Once the boundaries of the region of interest are located, its area is readily measured. We have developed a system that can rapidly construct search spaces, orthogonal curves, and perform segmentations of medical images. The system accurately locates the boundaries of the hippocampus given a good prior model of the shape. The operator can scale

images, construct templates, refine local parameters, segment, and measure the hippocampus, all within a single framework. Our system provides for the study and mensuration of the hippocampus while avoiding the error commonly produced during manual segmentations. Our experiments show that the system proposed here yields segmentations that are more reliable and more reproducible than manual segmentations. With the aid of a database of pre-computed templates, our semi-automatic segmentation system can segment the hippocampus faster, and more accurately than manual segmentations.

ACKNOWLEDGEMENTS

I wish to thank my thesis and academic advisor Dr. K.R. Subramanian for his guidance and expertise during the completion of this thesis. His time and experience have been an invaluable resource, without which this work could not have been completed. I also wish to thank Dr. John Brockway for the motivation behind this work as well as his expertise. Presbyterian Hospital of Charlotte, North Carolina has generously provided all of the MRI data used in this project. I extend my thanks to Dr. Taghi Mostafavi and Dr. Tom Weldon for serving on my thesis committee, and the Department of Computer Science for providing the computing facilities. Special thanks go to James Robinson for invaluable outside perspectives and suggestions during the lifecycle of this work.

DEDICATION

To my wife Wendy T. Hasty,

my brother Justin Hasty,

my parents Bill Hasty and Brenda Leslie,

and my step father Mike Leslie

for all of their unyielding support

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

Anatomic changes in the structure of the various structures in the brain have been associated with many chronic conditions such as aging, Alzheimer's disease, autism, and schizophrenia. It is possible to detect these anatomical changes by analyzing medical imaging data such as MRI scans of the heads of patients. Some disorders affect the structure of the brain enough that the changes caused by the affliction can be detected within the individual without outside comparison. The hippocampus is a good example of a brain organ that can show this type of dramatic change.

The hippocampus is an organ of the brain that has an elongated structure extending over the floor of the horn of each lateral ventricle of the brain. It can be found in the medial portion of the temporal lobe immediately above the lateral ventricle. The formation is widest at its most anterior point, and grows thinner as it moves posteriorly and superiorly toward the corpus callosum. The role of the hippocampus is not completely understood, but strong correlations have been made between hippocampal damage and short term memory loss [27]. See figure 1.1 for three-dimensional views of the hippocampus. The shape and volume of the hippocampus are of interest in the detection and prediction of some neurological disorders such as intractable temporal lobe.

epilepsy, and dementia of the Alzheimer's type. In the case of temporal lobe epilepsy, structural abnormalities within the hippocampi can be detected during the analysis of MRI images of the regions. Unilateral or asymmetric atrophy of the hippocampus correlates with hippocampal sclerosis, location of seizure onset, and outcome after temporal lobectomy for correction of epilepsy [5].

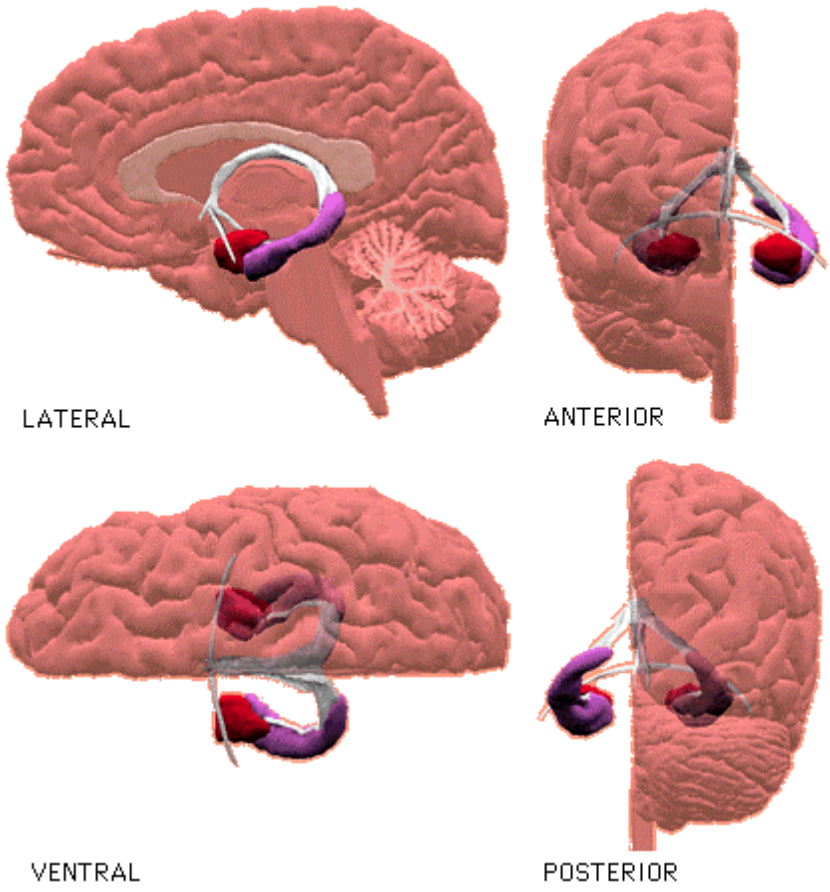

Figure 1.1 The hippocampus and surrounding structures. (Reproduced from http://neurocog.psy.tufts.edu/hippocampus_and_amygdala.htm, NeuroCognition Laboratory, Department of Psychology, Tufts University.)

The attraction of an automatic system for the segmentation of the hippocampus is that large numbers of patients and many different organs could be quickly and automatically measured to produce volumes of data about the "normal" and "abnormal"

shapes and volumes of organs. However, automatic systems are very difficult to apply in the case of small organs with very little intensity gradient separating them from surrounding organs, organs such as the hippocampus. To measure the organs from imaging data, the expert operator has in the past been forced to use time-consuming and error-prone techniques such as slice tracing and region painting in order to segment a region of interest from the surrounding tissues [6]. All of the methods currently available for computer aided segmentation that are capable of segmenting delicate structures such as the hippocampus are supervised to some degree. They may require seed points, construction of prior knowledge of the shape of the object, or expert tuning of the parameters to the technique. Semi-automatic image processing techniques, combined with judicious manual interaction, can provide a reliable segmentation system for segmentation of delicate brain structures from MR data.

Although there exist many different imaging modalities used to collect data such as: Computed Tomography (CT), X-ray, Positive Emission Tomography (PET), Single Photon Emission Computed Tomography (SPECT), ultrasound, and Magnetic Resonance Imaging (MRI), we have chosen to use data collected with the MRI modality because of the high quality of the images of the hippocampus. MRI is much better for these small delicate tissues in the brain because of its high spatial resolution and excellent discrimination of soft tissues [3]. MRI measures the spatial distribution of the relaxation times and proton densities of tissues in the presence of magnetic excitement and relaxation periods [4]. The MRI data that we use for experimentation is relatively clean and clear, and free of serious artifacts.

Due to the small gradient changes that delineate the boundaries between the hippocampus and its surrounding tissues, traditional techniques that simply use edges or grow regions are unreliable. Edge based techniques uncover very faint and incomplete edges that are difficult to connect. Region based techniques tend to bleed out beyond the edges of the true boundary of the region of interest, because of the lack of complete gradient boundaries. We have based our method on a method first proposed by Hemant Tagare called deformable 2-D template matching using orthogonal curves [7]. Orthogonal curves take advantage of *a priori* knowledge of the shape of the organ to be segmented while they perform a search for its actual boundary. The orthogonal curves define a search space that is orthogonal to the expected shape of the organ of interest. An energy function is evaluated for the possible combinations of boundary positions within the search space that may represent the actual boundary of the organ of interest [7]. This technique offers us the ability to search for the edges of the organ in the image while applying some external tension to the resultant boundary so the segmented region resembles both the previously known general shape and size of the organ. Orthogonal curves provide the base algorithm for the segmentation system developed for this body of research.

Chapter 2

SEGMENTATION TECHNIQUES

2.1 Introduction

Image segmentation is the process of subdividing an image into constituent parts or objects. Segmentation is traditionally divided into three categories: global knowledge based segmentation; edge based segmentation; and region based segmentation. Global knowledge based segmentation employs the global properties of the image to separate objects, and is usually only of use in very simple situations, or as a preprocessing step to further segmentation.

Edge based segmentation techniques represent a large family of algorithms that employ transitions in image intensity to locate the boundaries of objects. They rely on edge detecting operators to locate discontinuities in image intensity, but the resulting gradient image alone does not provide enough information to yield a segmentation. Separate methods must be employed to link edges together into sets to form object boundaries within the image. When prior knowledge of the object of interest is employed in conjunction with edge based segmentation techniques; reasonable segmentations can often be obtained.

Region based techniques differ from edge based techniques in that they attempt to directly construct regions of interest rather than simply separating regions with edges. Region based techniques are often employed in the presence of noise, where edge based

techniques may not find enough solid edge information with which to properly work. Region based techniques attempt to use local homogeneity in the image to form large areas of similar intensity. The two approaches, edge based and region based, are often used in combination with each other to yield segmentations that are more acceptable than either technique would yield alone [1]. Segmentation remains one of the most daunting tasks in computer vision, and generally requires very specialized algorithms or combinations of algorithms to solve different problems. Each new segmentation problem may require an existing technique, a combination of existing techniques, or an entirely new technique. Automatic, or unsupervised segmentation, is one of the most difficult tasks in the image processing pipeline [2]. Most approaches to difficult segmentation problems must be supervised to some degree, whether they require operator supplied prior knowledge of the object of interest, operator parameter tuning for the technique, or just an expert eye to determine degenerate cases of the technique.

2.2 Edge Based Segmentation

Edge detection is the search for meaningful discontinuities in image intensity when considered in the area of image processing. Segmentation based upon edge detection is useful when the areas under consideration are fairly homogeneous. Detection of edges involves a local derivative operator that must be convolved with the entire image. Convolution is the process of moving the operator about the image in a scanline fashion, and replacing the pixel beneath the operator, or mask, with the sum of the products of the mask terms and the pixels beneath it. A mask is shown in figure 2.1.

Z_1	Z_2	Z_3
Z_4	Z_5	Z_6
Z_7	Z_8	Z_9

Figure 2.1 A mask operator.

The operator will score high when the sum of products is high and will score low when the sum of products is low.

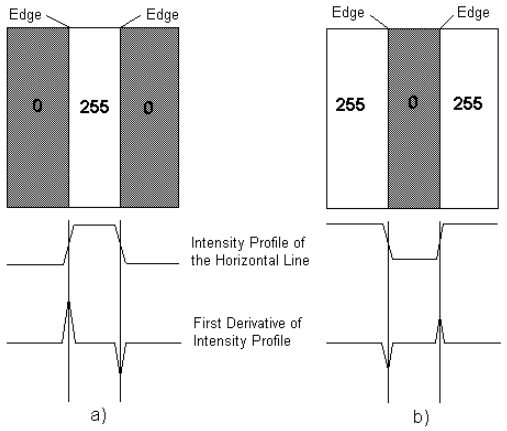

Figure 2.2 Edge detection using the first derivative operator. *Figure a) shows a white line and figure b) shows a dark line. The derivative finds the edge.*

Edge detection techniques use some derivative or finite differences operator to detect discontinuities in image intensity. These discontinuities can be expressed in terms of the first derivative of the image. See figure 2.2 for a description of edge detection using the first derivative. Some well-known first derivative operators are the Sobel, Prewitt, and central difference operators. These operators are most often expressed in the form of masks that are convolved with the image. The x and y derivatives are usually computed separately. Considering the mask in figure 2.1, the different operators can be characterized as follows (Note: We use the central difference operator for gradient computations in our method):

Sobel:

$-Z_1$	0	Z_3
$-2Z_4$	0	$2Z_6$
$-Z_7$	0	Z_9

Gx

$-Z_1$	$-2Z_2$	$-Z_3$
0	0	0
Z_7	$2Z_8$	Z_9

Gy

Prewitt:

$-Z_1$	0	Z_3
$-Z_4$	0	Z_6
$-Z_7$	0	Z_9

Gx

$-Z_1$	$-Z_2$	$-Z_3$
0	0	0
Z_7	Z_8	Z_9

Gy

Central Differences:

0	0	0
$-Z_4$	0	Z_6
0	0	0

Gx

0	$-Z_2$	0
0	0	0
0	Z_8	0

Gy

Edge detection alone does not yield a segmentation. For a true segmentation, the edges must be linked together using some sort of edge linking and boundary detection. Edge linking involves chaining edges together based upon some criteria. A common approach is to consider the pixels in a local neighborhood using some mask to determine similarities. Similarly valued pixels are linked together. This is known as local edge linking. Global Edge Linking can be accomplished using the Hough transform for line detection, graph theoretic techniques, and many other approaches [2].

These edge based approaches work well for strong edges and simple shapes. Without the presence of strong or complete edges, these techniques falter. If the object has relatively strong gradients and has some well-known geometric shape, then these techniques work well. The hippocampus has neither strong nor complete edges, and no apparent analytically consistent shape.

2.3 *Region Growing Segmentation Techniques*

Region growing constructs regions directly as opposed to detecting edges and attempting to find regions from sets of edges. Region growing techniques are generally

better than edge detection in noisy images where edges are difficult to find or are inconsequential. Some measure of region homogeneity is used as a criterion to subdivide the image into separate regions. The measure can be texture, intensity, model, or any combination of these measures [1]. Region growing begins with a seed pixel or a group of seed pixels that comprise the original region. If the intensity of each neighbor falls with some tolerance of the intensity value of the region, then it is merged with the region. This operation proceeds recursively, considering each neighboring pixel in turn for membership into the region [8]. Figure 2.3 illustrates the progression of the region growing algorithm from a single seed pixel.

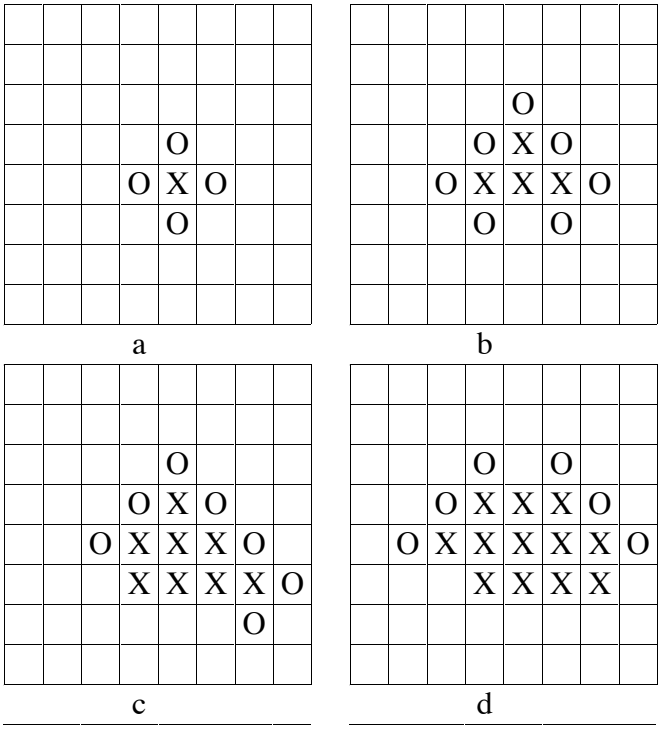

Figure 2.3 Region Growing Progression: *a) The X indicates the seed pixel, and the o's represent the neighboring pixels using a 4 neighbor operator; b) The newly added X's indicate the neighbors that were initiated into the region, and the O's represent the neighbors currently under consideration; c) & d) The progression continues. (Reproduced from [8]).*

2.4 Split and Merge Segmentation Techniques

As region growing builds regions from the bottom up, growing larger and larger regions, region split and merge constructs regions from the top down. Region split and merge subdivides large regions into successively smaller regions based upon some homogeneity criteria. Split and merge operations have a very natural underlying representation in a quadtree. A quadtree is a tree in which each node has exactly four descendants. In this situation, if a uniform splitting approach is taken, the four children divide the parent's region into four equally partitioned regions. Figure 2.4 illustrates two views on a quadtree as it is used to partition a region [2].

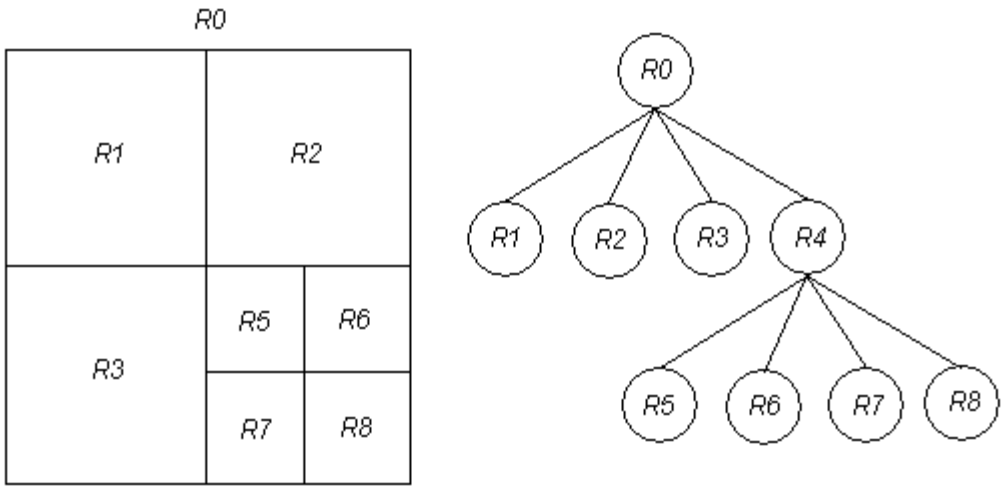

Figure 2.4 Split and merge hierarchies – quadtrees. *The left illustration shows the spatial relationship of the quadtree representation on the right.*

The initial region for the split and merge region can be the whole image or some sub-image. The initial region is evaluated for homogeneity. If it does not meet the homogeneous condition, it is split into four underlying regions and they are recursively split if they are not homogeneous. The stopping criteria for each subtree is met when the leaves are homogeneous or the leaves consist of a single pixel. It is possible for adjacent

regions to be homogeneous after the initial tree construction. If this is the case, a merging pass is made on the tree to merge all adjacent homogeneous regions. The merging operations are performed on the same underlying data structure that the split operations are performed upon, the quadtree. Figure 2.6 illustrates a possible progression of the split and merge algorithm.

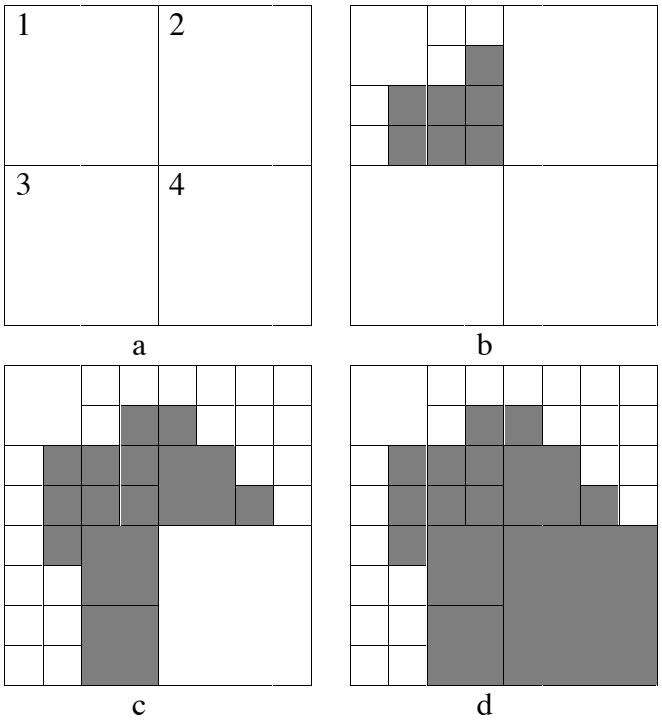

Figure 2.5 Split and merge progression*: a) The initial region is inhomogeneous so it is split into four sub-regions; b) Region 1 is recursively sub-divided until the sub-regions are homogeneous; c) Region 2 then 3 are sub-divided. Note the large areas of homogeneity that were not split; d) All of region 4 is homogeneous so there is no split of that node*.

Region growing algorithms have the problem of growing "holes" in their segmented regions. If a pixel falling within an otherwise homogeneous region has an intensity above or below the threshold due to noise, artifact, or some other extraneous

reason, then the region will grow around the pixel leaving a “hole”. Both techniques suffer from the over segmentation and under segmentation problems depending upon the precision in tuning the parameters [1]. A combination of the two techniques can overcome some of the problems inherent in each of them. A combination of the region growing and the edge based approaches can be found in [9], and a combination of the region growing and the split and merge approaches can be found in [8].

2.5 Template Matching Segmentation

Template matching involves finding the areas of an image that correlate highly with some *a priori* knowledge of the shape to be segmented. The Hough transform, first used to locate collinear points within an image, is a versatile algorithm that has been applied to a variety of problems. The transform can be generalized to locate shapes that do not have simple analytical representations. If the shape to be found, the shape in the template, is vectorized with respect to some reference point, the method can detect any shape [9]. The Hough transform approach to template matching has the advantage of recognizing partially occluded objects, or slightly deformed shapes. It can also be used to measure the similarity between a template and a detected object on the basis of size. It is insensitive to image noise and can find several occurrences of the template shape in an image [1]. The Hough transform has been used with high degrees of success in industrial applications when the sought after shape is known in advance and has a relatively high correlation to the actual occurrence of the shape in the image. Since the hippocampus changes frequently in shape and size, we decided that the Hough transform was too rigid to be used to successfully segment anatomy like the hippocampus. For a full development of the generalized Hough Transform, see [10] and [1].

2.6 Statistical Techniques

Statistical approaches to segmentation are most often employed at the global level and thus are usually not applicable to segmentation problems involving very subtle intensities and boundaries. However, they are being used to segment and classify large structures in medical data. Very often, they are used to segment white matter, gray matter, and cerebral spinal fluid (CSF). These are tissues that have fairly uniform intensities and can often be separated based upon their intensities alone.

In medical images, segmentations on broad categories of tissues can be performed by combining statistical classification methods with morphological image processing operations. When dealing with MRI data, conventional intensity based tissue segmentation techniques can often be disturbed by intrascan inhomogeneities introduced by the modality itself. These areas visually appear homogeneous, but in fact there may be sufficient variations within each tissue that cause the boundaries between even very distinctly different tissue classes to blur [3]. For techniques that statistically determine the range of intensities attributable to a specific tissue class, intersection of the ranges between two different tissue classes causes many problems. The noise introduced by the modality can be accurately classified in three categories. Thermal or electronic noise is mostly white noise, tends to model gaussian noise, and can be removed with filtering techniques that do not blur edges and structural information. The second type of noise is radio frequency (RF) noise. It results from inconsistencies in the magnetic fields, the sensitivity of the RF receiver, and from the actual magnetization of the object being scanned. RF noise causes inconsistencies in the intensity of objects of otherwise uniform density. Magnetic field noise introduces slowly changing intensity gradients where there

would otherwise be none. Low pass and homomorphic filtering techniques have been used to correct this type of noise. Because statistical techniques use intensity to classify tissues, the RF and the magnetic noise cause them the most problems [11]. A reasonable example of one approach to the problem can be found in [12]. Grimson et. Al. use statistical classification to label the anatomy from MRI image data. They then apply morphological operations such as dilation and erosion to separate equal intensitied but logically separate tissues that had been erroneously labeled identically.

2.7 Techniques that Employ Variability and Structure

With the exception of template matching, the techniques discussed thus far use only the "low level" information that is actually present in the image data to perform their segmentations. Another class of techniques cross the "low level" image data boundary and incorporate "high level" information about the object of interest to perform their segmentations. This "high level" information is most often prior knowledge of the general shape to be segmented. For most of the difficult medical image segmentation problems currently under investigation, the high level of inter-subject anatomical variability makes most traditional techniques by themselves unacceptable. Traditional template matching cannot accommodate the high shape variability either. The most promising techniques utilize the structure of prior knowledge of the shape, and the freedom to deform the prior shape knowledge in order to find a balance to accurately represent the true boundaries of the desired object.

2.7.1 Snakes: Active Contour Models

A snake is a spline that moves through an image or volume in order to minimize some energy function. The snake attempts to minimize an energy function that

incorporates external constraints and external image forces that pull it toward edges, lines, or any other selected image feature. Snakes are also active rather than passive. They continuously attempt to minimize their energy function and exhibit dynamic behavior. Due to the internal forces imposed them by their basis functions and the external forces imposed by the image, these splines appear to slither while they search for a minimum. It is due to this behavior that they are called snakes. Snakes are not global operators; rather they rely on an operator or some other mechanism to place them near the object of interest. Snakes are the first example explored here of a general model that will be developed more completely: deforming a model to an image feature by minimizing an energy function [13].

The snake first proposed by Kass, Witkin, and Terzopoulos in [13] is a controlled continuity spline. The internal continuity control of the spline imposes an internal energy piecewise continuity constraint. The external image forces drive the snake toward local gradient maxima, and the internal constraints hold the snake to a desired order of continuity. The energy function in [13] is parameterized as:

$$E^*_{snake} = \int_0^1 E_{snake}(v(s))\partial s$$

$$E^*_{snake} = \int_0^1 E_{\text{int}}(v(s)) + E_{image}(v(s)) + E_{con}(v(s))\partial s$$

where E_{int} represents the internal energy of the spline due to tension, E_{image} is the image gradient force, and E_{con} is the external constraint forces. E_{int} is composed of a first order term $\alpha(s)$ that makes the snake act like a membrane, and a second order term $\beta(s)$ that makes the snake act like a thin plate. Setting $\beta(s)$ to 0 at some point on the snake allows

the snake to become second-order discontinuous and develop a corner. The term E_{image} can be expanded to yield:

$$E_{image} = w_{line}E_{line} + w_{edge}E_{edge} + w_{term}E_{term}$$

which reflects the three external energy components: lines, edges, and terminations. The weight w_{line} determines how much influence the line attraction plays in the overall external image energy, and the sign determines whether the attraction is toward dark or light lines. The weight w_{edge} determines how much the image gradient E_{edge}, defined as $E_{edge} = -|\nabla I(x,y)|^2$, contributes to the overall external image energy and its sign indicates light to dark or dark to light transitions. (Note $I(x,y)$ is the intensity function of the image). The term $w_{term}E_{term}$ gives weight to the terminations of lines and corners, and a high value attracts the snake more to them. The last term of the energy function E_{con} allows the snake to seek a fixed position or another snake. The authors of [13] introduce the concept of a spring that anchors itself from the snake to some other entity. Creating a spring between point x_1 on the snake and point x_2 elsewhere adds $-k(x_1 - x_2)^2$ to the constraint energy E_{con}.

Snakes provide a powerful framework for a semi-automatic boundary finding mechanism. The general form presented above has been specialized for closed boundaries for segmentation applications. Work has also been done to deform multiple snakes in an area of interest, and split and merge the deformed snakes to generate improved segmentations [14]. Snakes can be finely tuned for specific tasks, and they are not overly difficult to implement. Snakes have been used as the basis for many deformable contour

finding techniques. The formulation of the snakes energy function as presented in [13] and summarized above is a useful overview to many of the energy functions used in other deformable shape techniques. However, snakes do not offer the level of semi-autonomous segmentation that is desirable in a hippocampal segmentation system.

2.7.2 Deformable Templates

Deformable Templates first appeared in the 1980's as global shape models. Global shape models represent images in terms of some structure criteria in the form of a template, and represent the variations on the images by probabilistic transformations on the template. The transformations on the templates are affine in nature, and can be represented by combinations of translation, scale, and rotation. This approach assumes that the template is highly similar to the shape in the image being matched, and only global rigid affine transformations are required to match the coarse features between the template and the image [15]. As discussed previously, the high degree of inter-study anatomical variability in brain images makes this approach too restrictive. In order to account for high local variability in an image, the transforms on the template must be of higher order. These high order transformations allow the global shape model to converge on the appropriate areas of the image. In areas of high correlation with the image only small transformations are needed, and in areas of low correlation more extensive transformations are needed to make the template converge on the objects of interest in the image. Some user defined fiducial markers and landmarks are sometimes defined as boundary conditions for these templates, but are not required. It is only under the condition of very high similarity between template and image that linear transformations can yield results as accurate as non-linear optimizations. For segmentation of anatomical

scans, non-linear transformations on global shape models, or deformable templates, is clearly the method of choice [16].

A deformable atlas (template) is a mathematically defined template, or a probabilistic measure of normal anatomical variation in brain structure [17]. The atlas is normally pre-registered and labeled, and its transformation to the image yields the transformed pre-registered atlas and the registered and labeled image. A textbook of neuroanatomy consists of two types of images: measured images; and labeled images. The measured images are a vector function defined on an ideal coordinate system local to the textbook. The vector of values includes measures of the intrinsic composition of tissues associated with the one or more imaging modalities represented in the textbook. The labeled images include symbolic information such as location of any organs to segment and any prominent features such as: white matter tracts, gray matter nuclei, Brocas's areas, and etc [16]. A textbook consists of sets of images that include information for many slices of data, while an atlas generally represents one data slice or series with one modality. Once the atlas or textbook is defined it is individualized by estimating the transformations that morph the template from the atlas to the actual study image. After the transformation is understood, the labels can be mapped from the atlas to the study image, yielding a segmented labeled study image [18]. The construction of any such textbook is not a trivial process. The collections of segmented, registered, and labeled normal images are time consuming to produce. Projects such as NML's Visible Human project for the construction of digital libraries that catalog anatomical information for entire humans motivate the large collections and descriptions of images that comprise textbooks [16]. For a description of the construction of one such textbook, see [19] where

they describe their process of filtering, semi-automatic segmentation, manual correction of the segmentation, and region of interest definition (labeling). In [19] is also found a truncated description of some sample landmark structures normally defined in textbooks.

The textbook is a vector mapping of the coordinate system $\Omega \subset \Re^3$ according to the transformation $\mathrm{T} : \Omega \rightarrow T$ with range space τ , assumed to be an M-fold product of spaces $T_1 x T_2 x ... x T_M$, where each component corresponds to a different feature of the tissue. The triple (Ω, T, T) mathematically defines the anatomical textbook.

A set of transformations $h \in \mathrm{H}$ is defined on the ideal coordinate system where H is the set of homomorphic maps $h : \Omega \rightarrow \Omega$. The topological maps are generated from translation groups applied to points $\vec{x} \in \Omega$:

$$\vec{x} = (x_1, x_2, x_3) \rightarrow (x_1 - u_1(\vec{x}), x_2 - u_2(\vec{x}), x_3 - u_3(\vec{x})).$$

The vector field $\vec{u}(\vec{x}) = (u_1(\vec{x}), u_2(\vec{x}), u_3(\vec{x}))$ is called the displacement field. The maps constructed from the transformations allow for dilation, contraction, and warping of the ideal coordinates of the template to the study coordinates at a local level. The set of anatomies generated with the textbook (Ω, T, T) becomes $\{\mathrm{T} \circ h : h \in H\}$. In other words, T composed with *h* where *h* is in the set *H*.

The information in a textbook is transformed into the coordinates of the study through the transformation $h \in \mathrm{H}$ on Ω that registers the studies $\{S_n\}_{n=1}^{N}$ with the textbook. Registration between the template and study is performed using a squared-error distance measure for MRI images, defined as:

$$\frac{1}{2\sigma^2} \int_{\Omega} \sum_{n=1}^{N} | T_n(\vec{x} - \vec{u}(\vec{x})) - S_n(\vec{x}) |^2 \, \partial \vec{x}$$

which is consistent with the Gaussian models of noise in the MR imaging modality. The distance == 0 if and only if the two are equal [16].

2.7.2.1 Elastic Models vs. Fluid Models

Anatomy is usually smooth and thus calls for smooth transformations from an atlas to a study. Since the 1980's, researchers have used linear models such as the membrane or Laplacian model, the thin-plate or bi-harmonic model, and the linear elastic model to constrain deformations in images. Of the aforementioned techniques, the linear elastic model has seen more application [20]. It is interesting to note that the membrane and the thin-plate models were employed in the formulation of the internal energy of Terzopoulos's snakes [13]. To ensure that the transformations from template space to study space are smooth, elastic mechanics have been employed to constrain the deformations. A full description of the use of the kinematics of elastic solids to deform the templates can be found in [18] and a full mathematical development of the technique can be found in [21]. In short, these transformations based on the theory of elasticity develop restoring forces proportional to the square of the deformed distance. This is a sound model for minor transformations, but to accommodate the high variability present in anatomical data this model has proved too restrictive. The elastic model prevents the textbook from being fully deformed to accommodate the study. Another model, the viscous fluid model, allows the restorative forces to relax over time, allowing the textbook to be more full deformed to match the studies [22]. The viscous fluid transformation (VFST) that is used is modeled by the partial differential equation (PDE):

$$\mu\nabla^2\vec{v} + (\mu + \lambda)\vec{\nabla}(\vec{\nabla}^T\vec{v}) = \vec{b}$$

where $\vec{v} = [v_1(\vec{x},t), v_1(\vec{x},t), v_1(\vec{x},t)]^T$ is the instantaneous velocity deformation field $\vec{u}$. The gradient operator is defined by $\vec{\nabla} = \left[\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \frac{\partial}{\partial x_3}\right]^T$ and $\nabla^2 = \vec{\nabla}^T\vec{\nabla}$ is the Laplacian. The $\nabla^2\vec{v}$ term is the viscous term. It constrains the neighboring particles of the displacement field to deform with approximately the same velocity by spatially smoothing the velocity field. The mass source term allows the structures to grow and shrink in mass, and is denoted by $\vec{\nabla}(\vec{\nabla}^T\vec{v})$ in the PDE. The viscosity coefficients are the μ, λ coefficients. More detailed descriptions of the application of the fluid dynamic PDE above to the transformation of textbooks can be found in [18] and [22]. A complete mathematical development of the technique can be found in [21]. An Eulerian reference frame is used to track the deformation. It places reference points at each of the pixels of the study to observe the deformation of the coordinate system of the textbook. As the textbook deforms in space over time, the points (particles in the viscous fluid mechanical system) in the textbook flow through the reference points. At any point in time, the original position of the point can be determined. Because the transformation takes place in an Eulerian reference frame, the velocity *v* of a particle can be related to its displacement, *u*, by $v = \frac{\partial u}{\partial t} + v \cdot \nabla u$. The $v \cdot \nabla u$ term provides for non-linear particle trajectory through observation points. The system is modeled successively at many discrete time points until the energy norm associated with the body force (the gradient of the likelihood):

$$b = \frac{1}{\sigma^2}\sum_{k=1}^{M}(T_k(x - u(x)) - S_k(x))\nabla T_k(x - u(x))$$

approaches zero. This technique yields the *maximum a posteriori* (MAP) transformation that deforms the textbook into the study [22].

These techniques take an extended period of time to perform their segmentations, but they can perform several simultaneous segmentations at once. Published in [22], table 2.1 yields some timing results from the application of the viscous fluid technique. The segmentations were performed on a 4K DECmpp 12000Sx with MP-2 processors. It is a massively parallel machine that is a remarketed version of the Maspar computer. The Circle to "C" entry reflects the extended time that the technique can take when the transformation must map transform textbook a very long distance from it's original state. The long running times of the technique reflect the time involved in solving the PDEs to a satisfiable resolution. The problem of long convergence time of the algorithm is compounded by the difficulty of the implementation, and those two criteria made this technique impractical for our application. Figure 2.6 shows the transformation computed for the circle to c transformation. Figure 2.7 shows some sample data from a textbook.

Experiment	Number of Voxels in Study	Time
Circle to "C"	128 x 128	3.45 minutes
2-D neuroanatomy textbook to study	256 x 265	1.41 minutes
3-D neuroanatomy textbook to study	256 x 256 x 64	88 minutes (estimated)

Table 2.1 Execution times for the fluid transformation algorithm

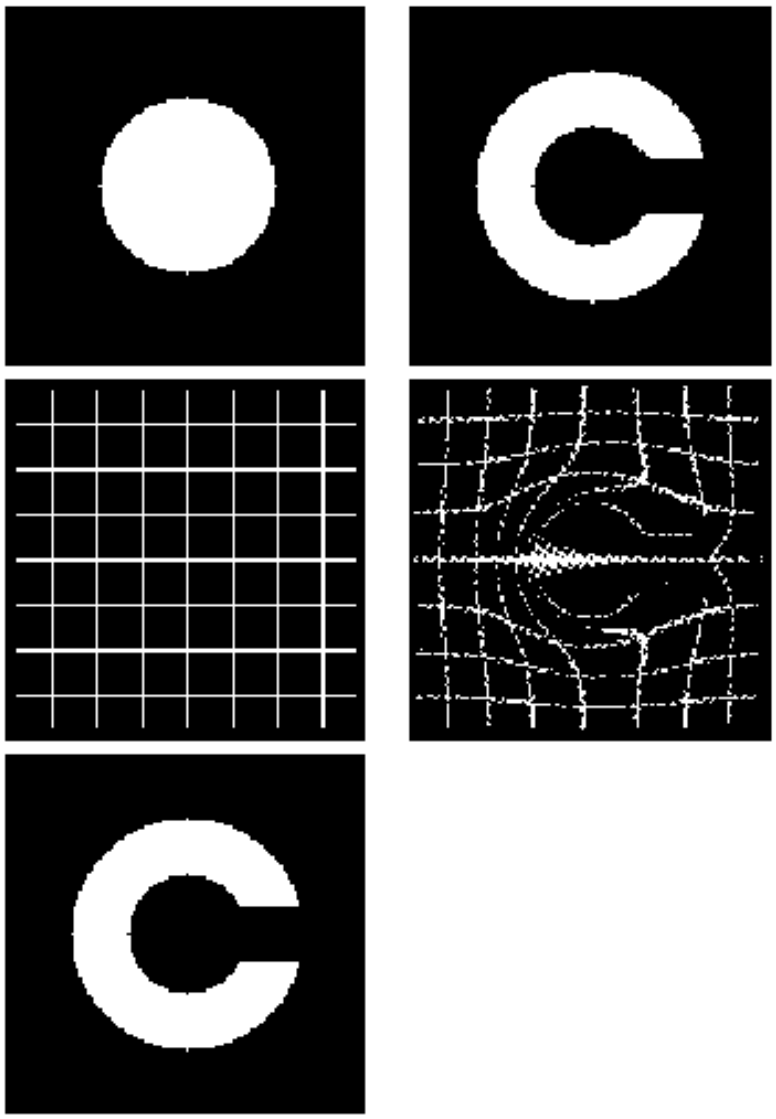

Figure 2.6 A sample fluid transformation. *The top left panel shows a circular textbook of a circle. The middle left shows the regular rectilinear grid of the textbook space. The bottom left and top right panels show the study, a white "C" on a black background. The middle right panel shows the minimizing transformation applied to the regular rectilinear grid of the textbook. Note the preservation of the topology of the textbook from the middle left panel to the transformed middle right panel. (Reproduced from [22])*

Table 2.2 shows the running times, number of iterations of the PDE, the number of simulated time steps, and the total time to complete the transformations for the experiment. It is reproduced from [18].

Image Size	Number of PDE Iterations	Number of Time Steps	Time (sec)	Total Time (sec)
64 x 64	250	200	33	
128 x 128	250	40	27	85.3
256 x 256	250	8	24	

Table 2.2 Viscous fluid transformation times. Note the greatly reduced number of time steps for the larger images. (Reproduced from [18])

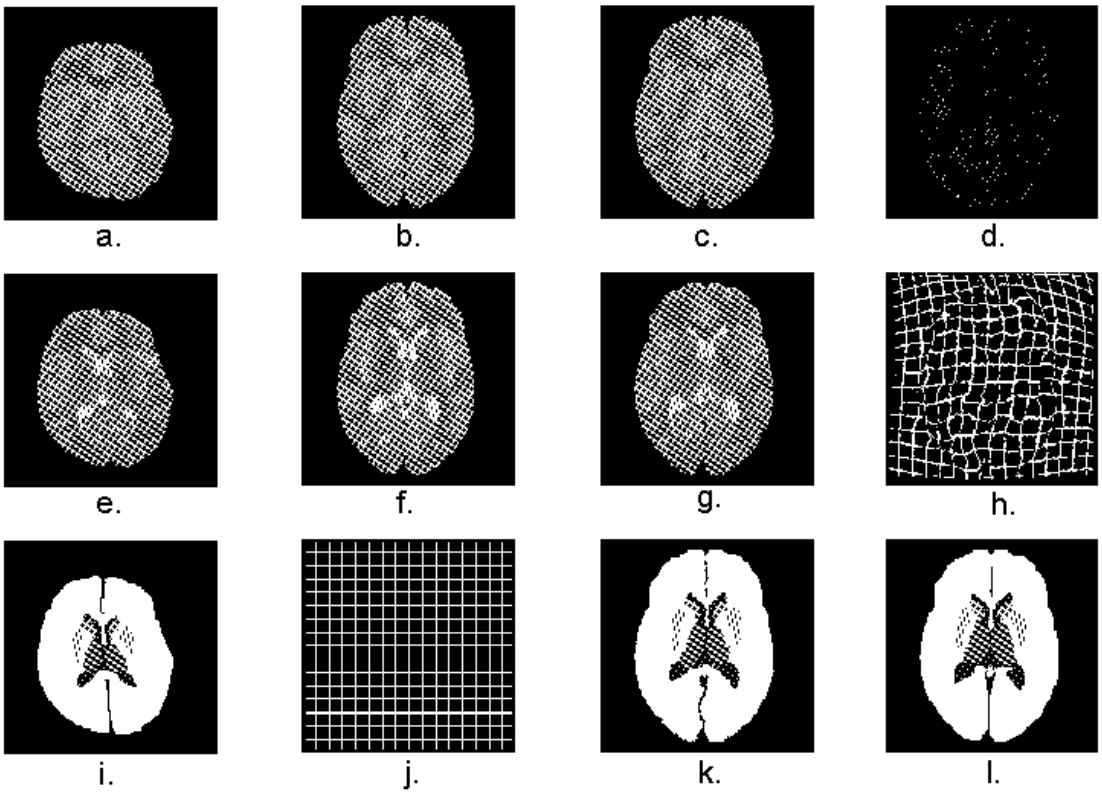

Figure 2.7 Views of the viscous fluid template technique. *Panels a), e), and i) show three components of a 2-dimensional anatomical textbook. a) is a MR spin density image, e) is a T2 weighted image, and i) is a major nuclei segmentation. b) is a study image in the spin density MR protocol, and f) is a T2 weighted study scan. Panel j) shows the regular rectilinear grid of the textbook. c) shows the result of applying the estimated transformation the spin density study, and g) the T2 study, and the resulting segmented and labeled study in i). Panel d) shows the magnitude difference between the T2 textbook image and the T2 study image. Panel h) illustrates the result of the transformation to the textbook grid, and l) illustrates a manual segmentation. (Reproduced from [22])*

The deformable textbook approach has been applied specifically to the problem of hippocampal segmentation. It has been used in studies because of the clinical importance of automatic and semiautomatic segmentation and mensuration of it with respect to the temporal lobe epilepsy problem. The published applications of hippocampal segmentation employing deformable textbooks use high resolution 256 x 256 images. They can be found in [23] and [24]. The entire volume of the hippocampus was not segmented and measured. The studies only used one slice of the MR scans from each patient on which to perform their segmentations. The slice chosen from each image was comparable, and chosen from a sagittal series of images. The most lateral sagittal slice where the amagdyla was clearly separated from the anterior hippocampus by cerebral spinal fluid (CSF) was chosen for each segmentation. In this slice the hippocampus is clearly separated anteriorly from surrounding tissues by CSF (which appears black in most T1 weighted MR protocols). Posteriorly the lateral ventricle, another well defined structure, separates the hippocampus from surrounding tissue. Inferiorly, a strip of white tissues separates the hippocampus from the fusiform gyri and the parahippocampal gyri. Superiorly, CSF separates the hippocampus from surrounding tissue [24]. These very distinct boundaries cannot be considered to be an average case for the segmentation of the hippocampus. It appears in many slices, and an accurate measurement of its volume would have to consider all of the slices in which it is visible. Figure 2.8 shows the location of the hippocampus in the chosen slices, and a scaled image of it. Note the prominent edges bounding the hippocampus. Figure 2.9 shows the results of segmentations on the template and on the template deformed to fit the study. Figure 2.10

shows the resulting transformations after applying the same template to five different study subjects.

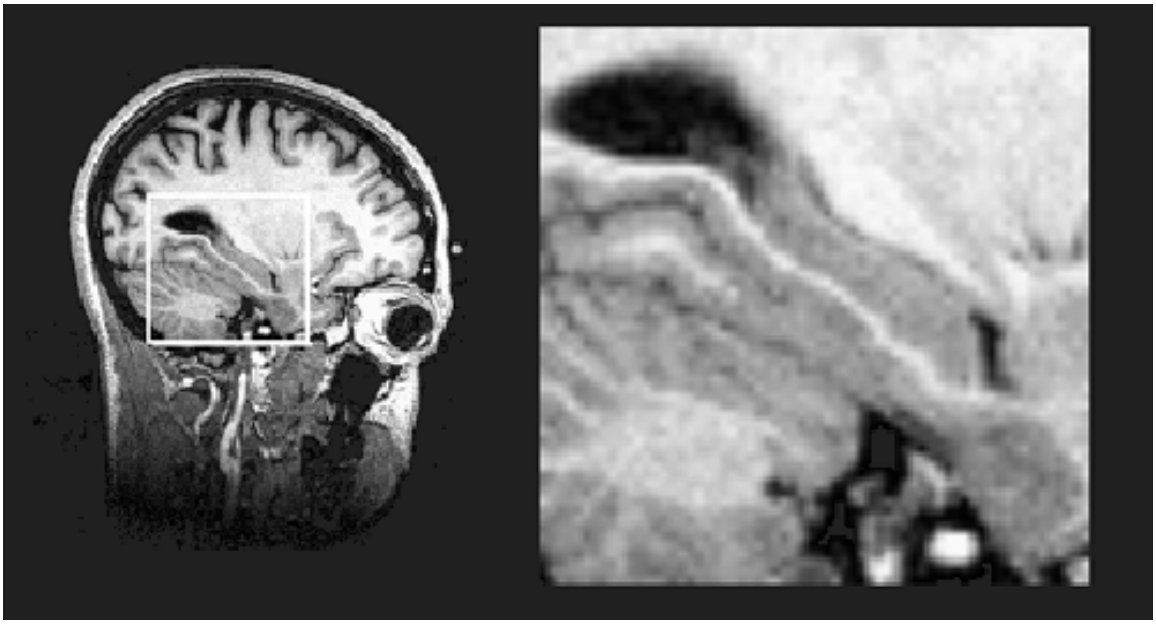

Figure 2.8 Location of the hippocampus in a sagittal MRI slice. *The right view is a scaled image of the area of interest on the left denoted by the white bounding box. (Reproduced from [24].)*

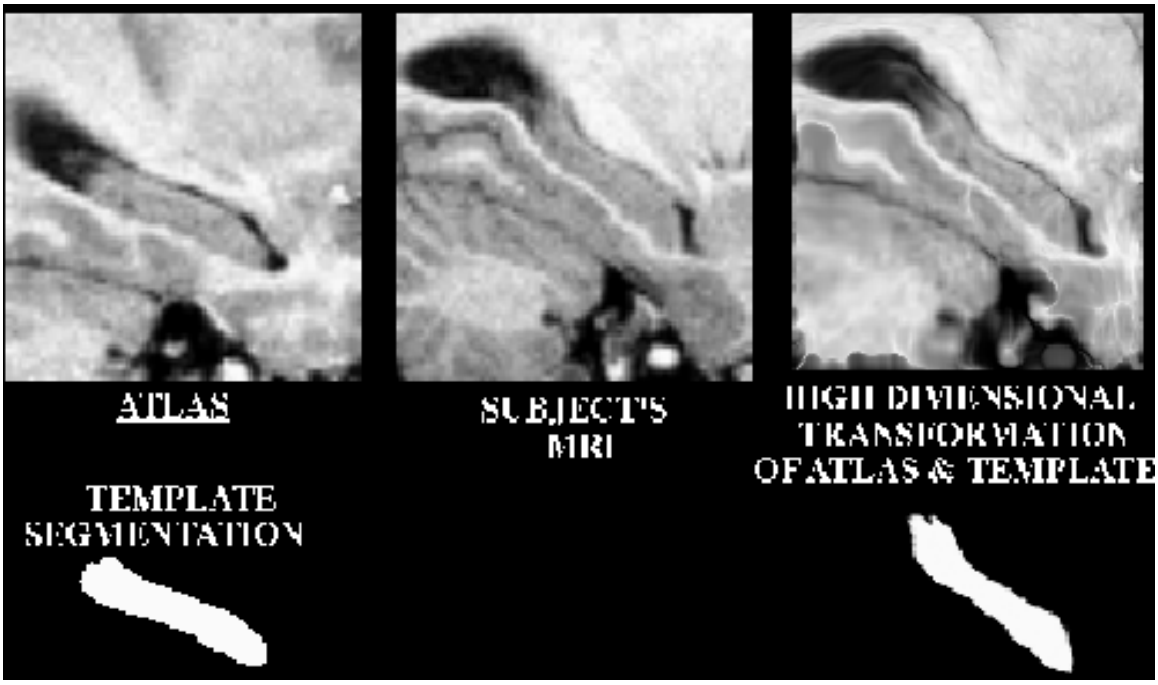

Figure 2.9 Segmentation of the template and the template after deformation to a study hippocampus. *(Reproduced from [24].)*

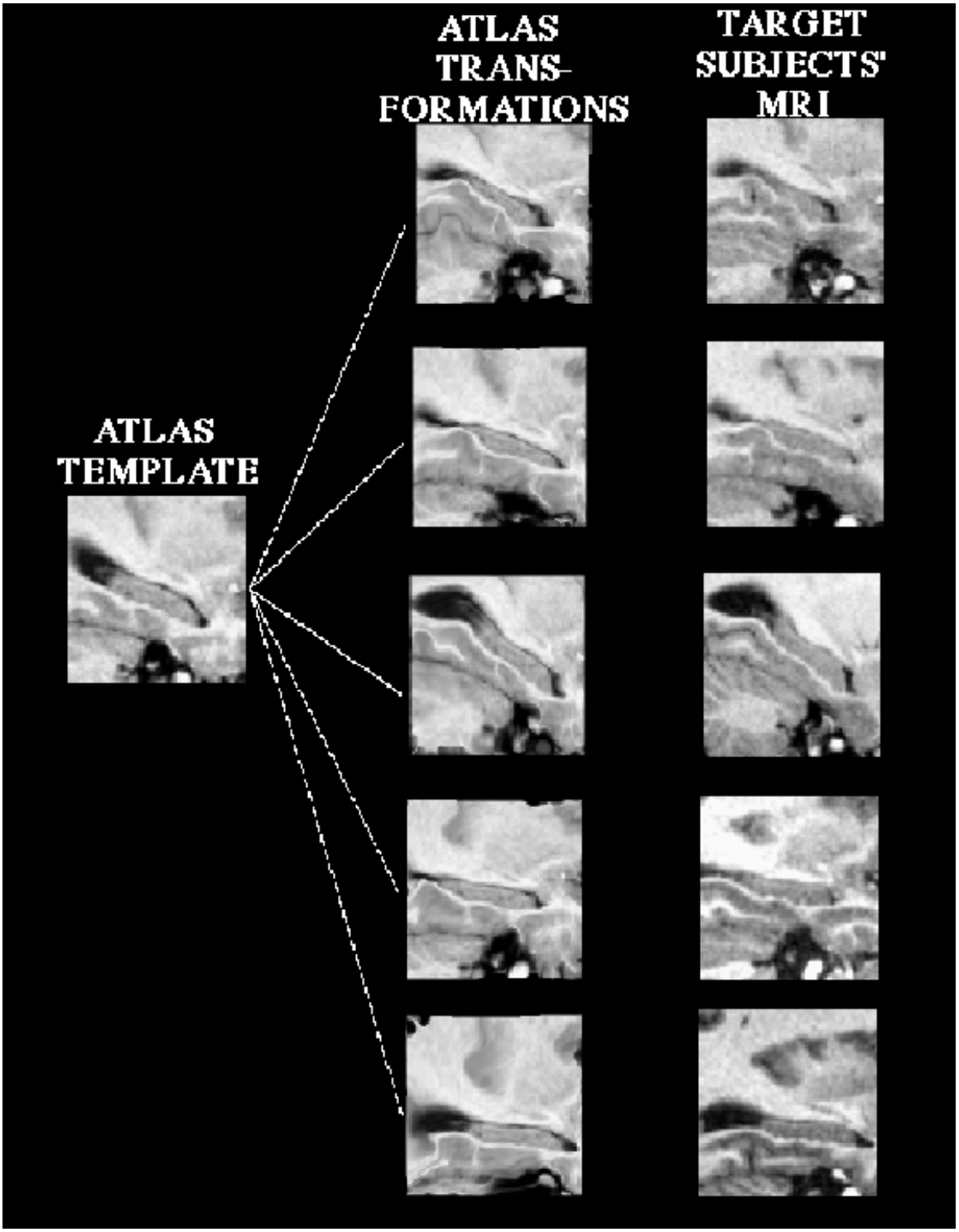

Figure 2.10 Five resulting transformed templates after applying the same template to five different studies. *(Reproduced from [24].)*

Chapter 3

DEFORMABLE TEMPLATES USING ORTHOGONAL CURVES

Deformable textbooks and atlases use the entire subject image space as the optimization space. Every pixel in the image is used in the calculation of the energy function at each time step. To compute the deformations, partial differential equations must also be solved at each time step. The running times the individual time steps are understandably long. Deformable templates using orthogonal curves restrict the search space and simplify the energy function to greatly improve the segmentation time for an organ. The construction of the template requires more human interaction, but the time savings for the deformation stage are orders of magnitude greater.

3.1 Introduction

Recalling the snake technique for closed boundary segmentation, the initial snake is the deformable template. It is parameterized as a curve $C : t \rightarrow (x(t), y(t))$, and energy is associated with the curve. The solutions to the internal and external energy functions are computed to determine minimums that represent the potential boundaries of the region of interest. The energy function associated with the curve is of the form:

$$E(C) = \int_0^1 \Gamma(x, y, \dot{x}, \dot{y}, ..., \overset{n}{x}, \overset{n}{y}) \partial t$$

where the individual terms of $\Gamma(...)$ are the partial energy functions that comprise the internal and external energy measures. For the purpose of boundary finding, the

parameterization is unimportant. The energies are independent of the parameterization. If the optimal deformation is relatively small, than it only needs to be normal to the template. If the template contains a sufficiently high degree of prior shape information the optimal deformation will be acceptably small. For a formal explanation see [7].

This technique generates curves that are perpendicular to the template and do not intersect each other. The curves define the search space for the optimal set of energy minimizing points and the template is restricted to movement along the orthogonal curves. The optimal deformed template is defined by the set of points on the orthogonal curves that minimize the energy function over the orthogonal curve space. Since the range of the energy function is restricted to the orthogonal curve space, the number of degrees of freedom for the search is reduced to one for each curve, the parameterization of the curve along itself.

The prior knowledge of the shape is expressed by the shape of the template, and preserved by energy functions that evaluate the distance with respect to the expected shape, and the smoothness of the shape. Distance is Euclidian between the intersection of each orthogonal curve with the template, and smoothness is the change in distance along the deformed curve. The distance, smoothness, and external energy are the partial functions that comprise the energy function, where the external energy function is a measure of gradient. Figure 3.1 illustrates the relationship between the orthogonal curves, the template, and the deformed curve or boundary.

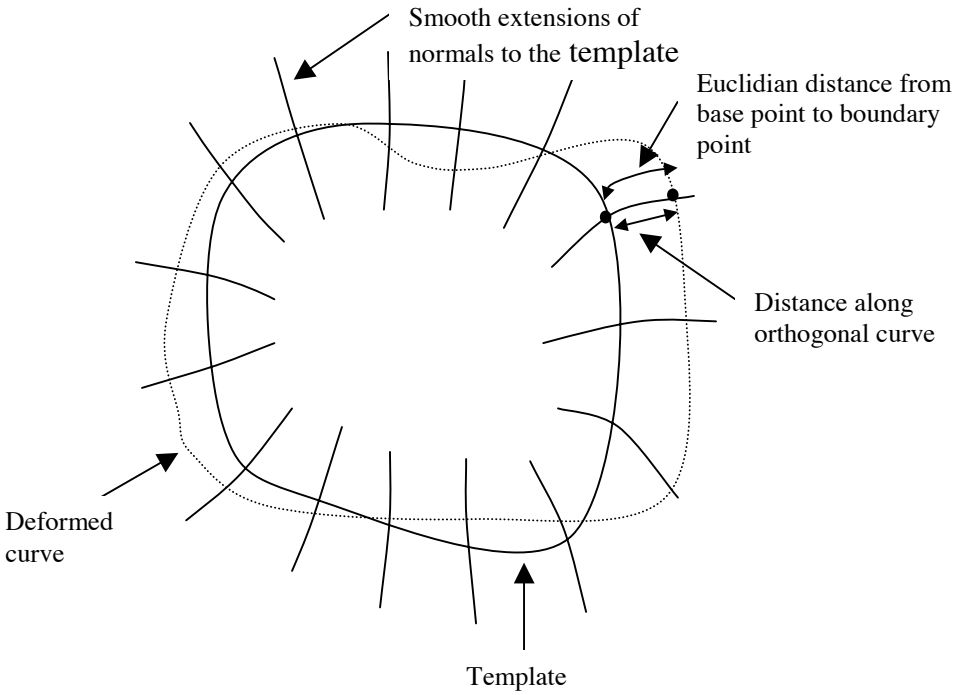

Figure 3.1 Smooth orthogonal curves.

3.2 Orthogonal Curves

These templates are closed curves that are guaranteed to be non-self intersecting. If a template curve is parameterized with θ as $C:\theta \rightarrow (x(\theta), y(\theta))$, and the space in $\Re^2$ over which the template can be deformed is bounded on the interior of curve *C* by a non-self intersecting closed curve C_{in}, and on the exterior of *C* by a non-self intersecting closed curve C_{out}, then the subspace formed by space between C_{in} and C_{out} defines the allowable deformable space for the template *C*. See figure 3.2 for an illustration of the relationship between C_{in} and C_{out}.

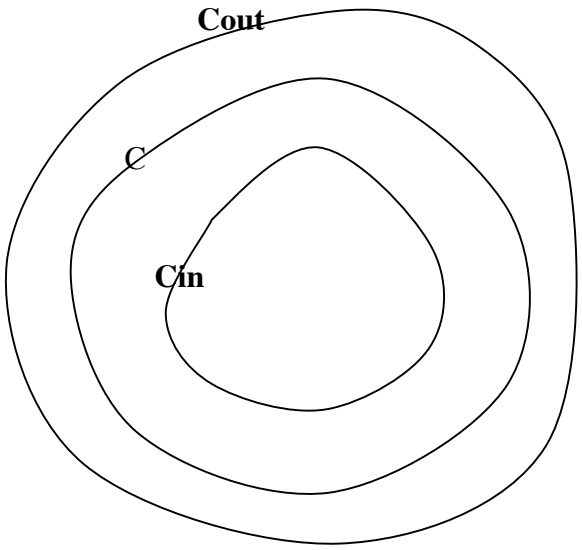

Figure 3.2 Relationship between C, C_{in} and C_{out}.

If R_θ is an orthogonal curve passing through the base point $(x(\theta), y(\theta))$ on the template, the following assumptions are made:

1) Each R_θ begins on the interior bounding curve C_{in}, orthogonally intersects the base point on C, and terminates at C_{out}.
2) Each R_θ has a continuous tangent.
3) No two orthogonal curves intersect.

If R_θ is parameterized by its arc length r (which is zero at the base point on the intersection of R_θ and C, *and increases as* R_θ proceeds outwards, and decreases as R_θ proceeds inwards), then the parameterized form of R_θ can be written in the coordinate system of the template as:

$$R_\theta : r \rightarrow \begin{pmatrix} x(\theta) + \Delta x(r) \\ x(\theta) + \Delta x(r) \end{pmatrix}$$

where the terms $\Delta x_\theta(r)$ and $\Delta y_\theta(r)$ give the position of R_θ with respect to the base point and $(\Delta x_\theta(0), \Delta y_\theta(0)) = (0,0)$.

If the transformed template C^* is arrived at by moving template points along the orthogonal curves, then C^* is formalized by:

$$C^* : \theta \rightarrow \begin{pmatrix} x(\theta) + \Delta x(r(\theta)) \\ x(\theta) + \Delta x(r(\theta)) \end{pmatrix}$$

where $r(\theta)$ gives the displacements of the template points along the orthogonal curves. Since the template is placed on the image manually by translation and rotation, the set of curves in the image obtained by deforming the template can be formalized as:

$$C^* : \theta \rightarrow \begin{pmatrix} x_I(\theta) \\ y_I(\theta) \end{pmatrix} = R\begin{pmatrix} x(\theta) + \Delta x(r(\theta)) \\ x(\theta) + \Delta x(r(\theta)) \end{pmatrix} + T$$

where $(x_I(\theta), y_I(\theta))$ are the coordinates of C^* in the image space, R is the 2x2 rotation matrix, and T is the translation vector. See figure 3.3 for an example of orthogonal curves and the curves C, C_{in}, and C_{out}.

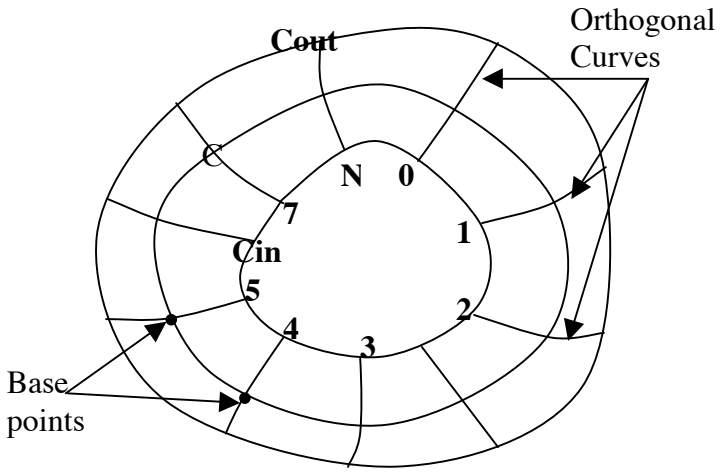

Figure 3.3 Curves C, C_{in}, and C_{out} with orthogonal curves.

For the application of template matching using orthogonal curves, it is necessary to use their discrete formulation. In the original work [7] the curves were discretized by sampling the template curve uniformly along its arc length an *N* base points. From these base points, the orthogonal curves were traced uniformly inwards and outwards and each was sampled uniformly at 2*M*+1 points. The set of discrete points can therefore be represented by the convenient notation $x_{i,j}, i = 1,...,N$, and $j = -M,...,M$ where *I* indicates the orthogonal curve that the point belongs to and j refers to the location of the point along the orthogonal curve. The deformed curve is discretized as an *N*-sided polygon whose i^{th} vertex is constrained to fall on the i^{th} orthogonal curve. The polygon can be expressed as $P(v_1, v_2,...,v_N)$. Figure 3.4 illustrates the relationship between discrete orthogonal curves, base points, and the polygonal boundary forming the deformed curve.

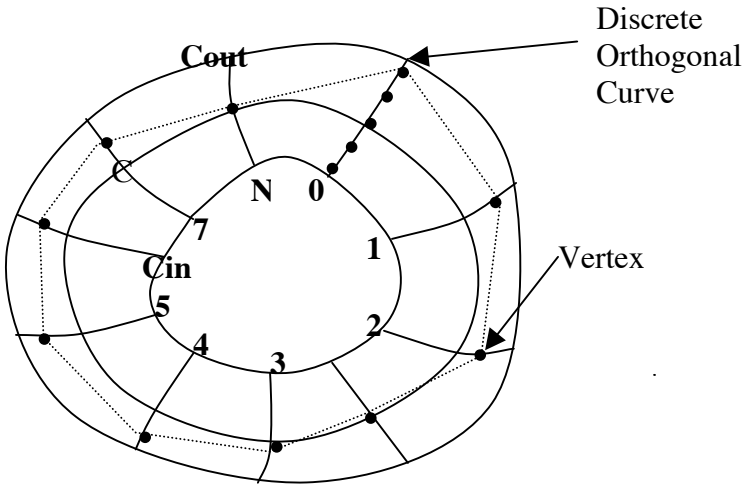

Figure 3.4 Discrete orthogonal curves, base points, and the vertices that form the boundaries of the deformed curve.

The orthogonal curves exist if there exists a function $\phi(x, y)$ whose range is the subspace of $\Re^2$ between C_{in} and *C* such that:

$\phi(x, y) = c_1$, on C_{in}

$\phi(x, y) = c_2$, on C

with $c_1 > c_2$, and $\phi(x, y)$ has well defined gradients. If the function $\phi(x, y)$ is defined as Laplace's equation:

$$\frac{\partial^2 t}{\partial^2 y} + \frac{\partial^2 t}{\partial^2 y} = 0$$

and it is solved for the space between C_{in} and C, then the gradients of $\phi(x, y)$ can be determined using the central differences operators across the results of $\phi(x, y)$. Starting at the base points and following the trajectory of the gradient of the function $\phi(x, y)$, the orthogonal curves are formed by the resulting path. By construction, the path is orthogonal to C. By repeating the same procedure between C_{in} and C_{out}, and joining the resulting orthogonal curves at the base points, the entire set of orthogonal curves can be obtained that intersect C, C_{in}, and C_{out}. For a more complete description of the existence and relevance of orthogonal curves, see [7].

3.3 Energy Functions

The energy function of a deformed curve has three components, distance, smoothness, and external energy functions. The distance and smoothness together compose the internal energy representation. The distance energy $I_1(P)$ is a measure of the scale of the deformed shape with respect to the original template shape. The distance energy integrates the Euclidian displacements of the vertices of the deformed curve from their base points and is characterized by:

$$I_1(P(v_1,...,v_2)) = \sum_{k=1}^{N} D(v_k - x_{k,0}).$$

The smoothness energy $I_2(P)$ measures the distances in successive vertices that form the deformed boundary. It is a measure of how close the shape of the deformation is to the original template shape, and is characterized by:

$$I_2(P(v_1,...,v_2)) = \sum_{k=1}^{N} | D(v_k - x_{k,0}) - D(v_{k+1} - x_{k+1,0}) | .$$

The external energy $X(P)$ is the net component of the image gradients along the orthogonal outward pointing normals to the polygon constructing the deformed curve. It is characterized by:

$$X(P(v_1,...,v_2)) = \delta \sum_{k=1}^{N} L_k \int_0^1 n_k \cdot \nabla I(\alpha v_k + (1-\alpha) v_{k+1}) \partial \alpha$$

where n_k is the orthogonal outwards pointing normal and L_k is the length of the line segment joining the vertices v_k and v_{k+1}. The integrand is the dot product between the image gradient and the orthogonal outwards pointing normal. The coefficient δ determines the sign of the gradient transition (-1 for light to dark transitions, and +1 for dark to light transitions). The entire energy function can be expressed together in the form:

$$E(P(v_1,...,v_2)) = -\rho_1 X(P(v_1,...,v_2)) + \rho_2 I_1(P(v_1,...,v_2)) + \rho_3 I_2(P(v_1,...,v_2))$$

where the ρ terms are non-negative weights use to tune the function for particular behavior with different data sets [7].

3.4 Optimization

The optimal deformed curve will minimize the energy function *E(P)*. An exhaustive search for the actual minimization of the function would be too computationally expensive, so another search algorithm must be used. If the energy function *E(P)* is reformulated to stress the relationships between successive vertices in the boundary, it can take the form:

$$E(P(v_1,...,v_N)) = \sum_{k=1}^{N} \phi(v_k, v_{k+1})$$

where:

$$\phi(v_k, v_{k+1}) = -\rho_1 \delta \sum_{k=1}^{N} L_k \int_0^1 n_k \cdot \nabla I(\alpha v_k + (1-\alpha) v_{k+1}) \partial \alpha$$
$$+ \rho_2 \sum_{k=1}^{N} D(v_k - x_{k,0})$$
$$+ \rho_3 \sum_{k=1}^{N} | D(v_k - x_{k,0}) - D(v_{k+1} - x_{k+1,0}) |$$

With *E(P)* in this form, dynamic programming can be used to obtain a search space minimum solution. Note that dynamic programming will not guarantee all unique search space energy minima if there exist multiple solution sets. The dynamic programming procedure proposed in [7] is outlined as follows:

1) For every v_3, tabulate the v_2 that minimizes partial energy ε_1.

2) For every v_4, tabulate the v_3 that minimizes the partial energy $\varepsilon_2 = \phi(v_3, v_4) + \overset{\min}{v_2, v_3} \varepsilon_1$. Where $\overset{\min}{v_2, v_3} E_1$ is the minimum value of ε_1 with respect to v_2 for a given v_3.

3) For every v_5, tabulate the v_4 that minimizes $\varepsilon_3 = \phi(v_5, v_4) + \overset{\min}{v_3, v_2} \varepsilon_2$.

4) For every v_N, tabulate the v_{N-1} that minimizes the partial energy

$$\varepsilon_{N-2} = \phi(v_N, v_{N-1}) + \min_{v_{N-2}, v_{N-3}} \varepsilon_{N-3}.$$

5) For every v_1, repeat the calculations of the partial energy minimums to find the global minimum value, and identify a minimizing set.

For the version presented in [7] the cost of the minimizing function is $O(M^3 N)$. This is not as prohibitive as the upper bound seems because M and N tend not to be excessively high.

Chapter 4

A SYSTEM FOR THE CONSTRUCTION OF ORTHOGONAL CURVES AND THE SEMI-AUTOMATIC SEGMENTATION OF IMAGES

4.1 Introduction

We have constructed an application to serve as a framework to explore the use of orthogonal curves to segment the hippocampus from its surrounding tissues. The application allows the user to construct the curves C, C_{in} ,and C_{out} interactively. It allows the orthogonal curves R_θ to be automatically generated from the curves C, C_{in} , and C_{out} . It then provides the user dialog boxes to interactively choose the parameters ρ_1 , ρ_2 , ρ_3 , and δ , as formulated by:

$$\phi(v_k, v_{k+1}) = -\rho_1 \delta \sum_{k=1}^{N} L_k \int_0^1 n_k \cdot \nabla I(\alpha v_k + (1-\alpha) v_{k+1}) \partial \alpha$$
$$+ \rho_2 \sum_{k=1}^{N} D(v_k - x_{k,0})$$
$$+ \rho_3 \sum_{k=1}^{N} | D(v_k - x_{k,0}) - D(v_{k+1} - x_{k+1,0}) |$$

for the energy optimization step. The resulting deformed template, or segmentation, is displayed in the application superimposed over the original image. Its area can then be measured, and the segmentation, the curves, and the orthogonal curves can be saved for later reference and use. In addition to the core functionality, the application also provides dialog boxes to aid in batch processing of the data files, renumbering and renaming data files, and the preprocessing of the images. The user interface is implemented in the Java

programming language utilizing the widget set provided by version 1.2 of the Abstract Window Toolkit (AWT). Some of the processing of the application is handled by Java's main thread, but most of the computation is performed in a separate Java thread. Threads are spawned that call corresponding C++ object methods compiled natively. The C++ code is invoked through the Java Native Interface (JNI), and compiled into dynamic shared libraries. This organization combines the ease and convenience of the Java programming language and packages with the speed of the natively compiled C++ code. Figure 4.1 shows the applications main widget. Figure 4.2 shows a block diagram of the system.

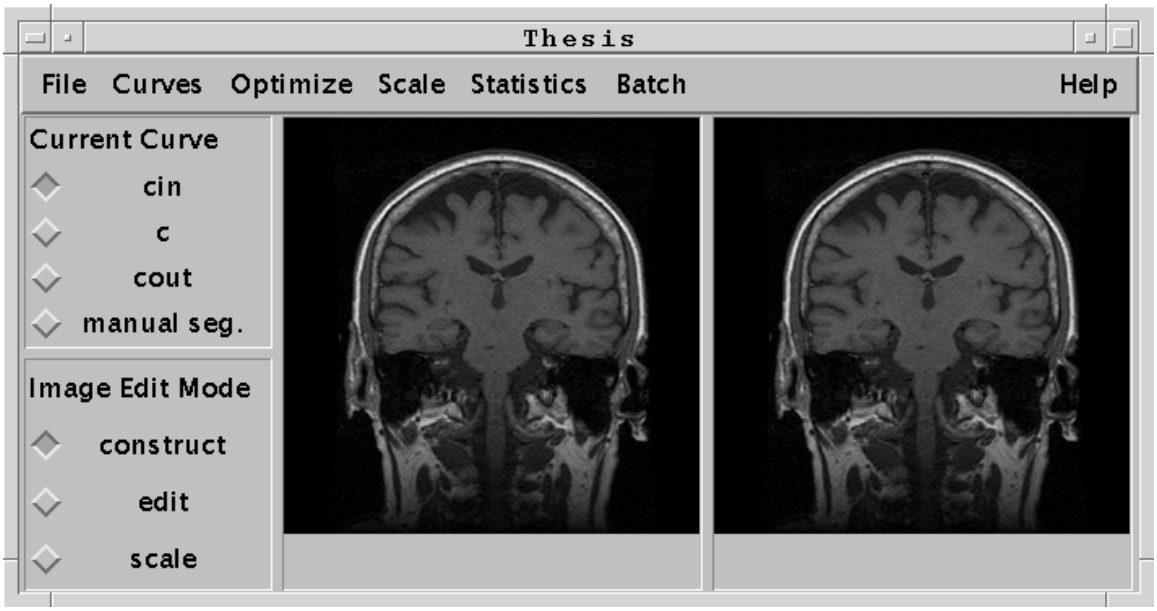

Figure 4.1 Main application widget. *The left panel displays the current curves (none shown here) and current orthogonal curves (none shown here) and the right panel displays the current segmentation (none shown here).*

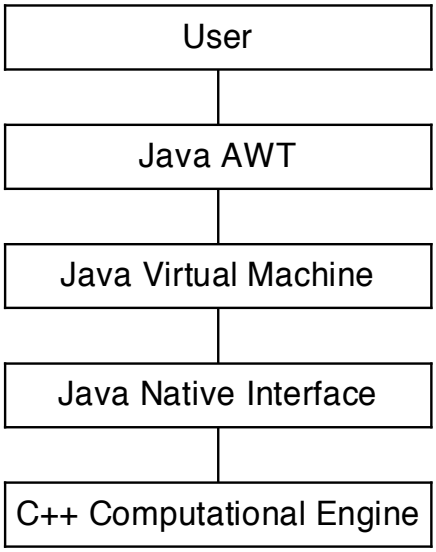

Figure 4.2 Application organization.

4.2 Preprocessing

The preprocessing stage of this segmentation pipeline consists of a semi-automatic scaling operation. When the scale option is selected from the "Image Edit Mode" group of radio buttons on the left of the main application window, the user can draw a uniformly square bounding box by dragging the mouse with the left button depressed. The interior of the bounding box identifies the region to be scaled. The scaling operation itself is a native function.

A bi-cubic spline interpolation is performed to scale the region of to an operable size. This operation is implemented in C++ using the cardinal spline implementation of the Visualization ToolKit (VTK) [28]. This is a one-dimensional spline. The two dimensional transformation is performed by creating splines on each of the rows of the region to be scaled. Then the splines are sampled at the new resolution to create *N* new splines where *N* will be the number of columns in the original and the scaled image. When the splines are evaluated at the resolution of the width of the original image, the

scaled image is reproduced. Since the cardinal spline family is first derivative continuous, the edge pixels and derivatives are reproduced for the boundary conditions. The bi-cubic spline interpolation has a smoothing effect on the data, and no other smoothing is performed in this pipeline. It would be valuable to utilize the statistical noise suppression method used by Christensen et. al. in future applications.

There is a menu option "Scale: Make Scale Persist" that leaves the scale bounding box intact after the scaling operation is performed. This allows entire sequences of images to be scaled using the same bounding box. The "Scale: Cease Scale Persist" menu item ends the persistence of the bounding box, and it will disappear after the next scaling operation. The "Scale: Image Scale" menu option invokes the scaling algorithm on the region of interest in the current bounding box. Figure 4.2 shows the scale bounding box in place. All of the data used in our experiments has been scaled using this method. The typical scaling factor is 450%-650%.

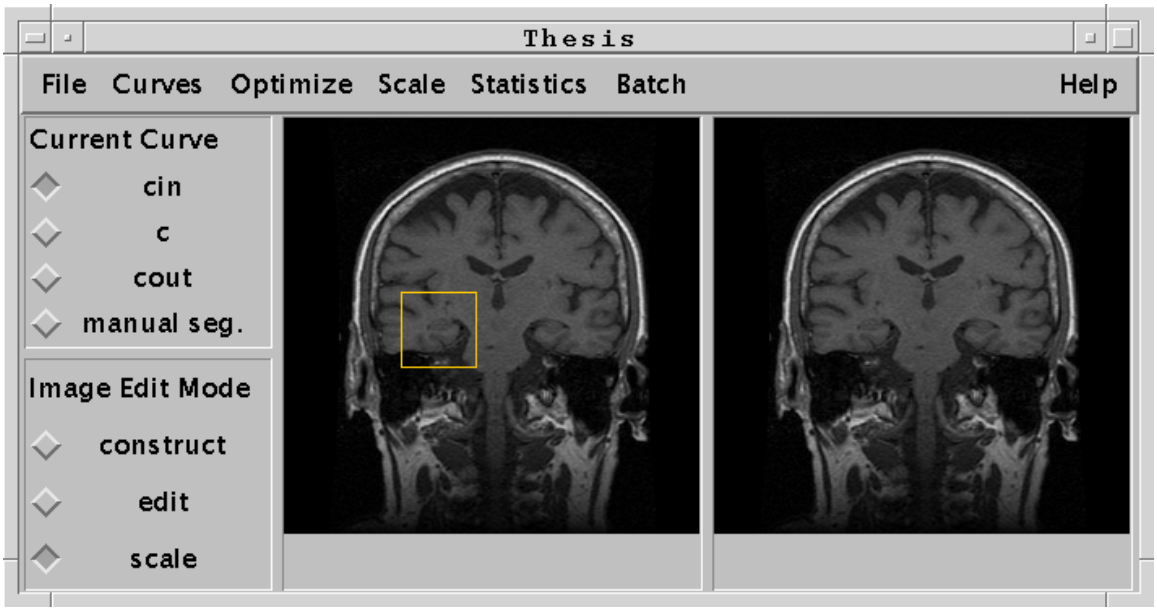

Figure 4.3 Image scale bounding box. *The box in the image above identifies the region that the scaling operation will act upon.*

4.3 Boundary Curves Construction

The boundary curve construction is the next step in the pipeline if there does not already exist a set of boundary curves that are sufficient for the shape to be segmented. The boundary curves are C, C_{in}, and C_{out}, and are constructed one at a time. To construct the boundary curves the user selects the "Construct" radio button from the "Image Edit Mode" radio group on the lower left of the main application window (Default). The radio buttons in the "Current Curve" radio group box determine which curve the operations will be applied to.

If C_{in} is the current curve and the current "Image Edit Mode" is "Construct", then the user uses mouse clicks to plot vertices in a bounding polygon of the desired shape. The polygon is always closed. This operation can be repeated for each of the bounding curves to generate a complete set of polygons representing the curves C, C_{in}, and C_{out} simply by changing the current curve. By changing the "Image Edit Mode" to "Edit", the user may alter the currently plotted points in the current curve by picking them and dragging the mouse. Figure 4.3 displays the result of the construction of the three bounding curves.

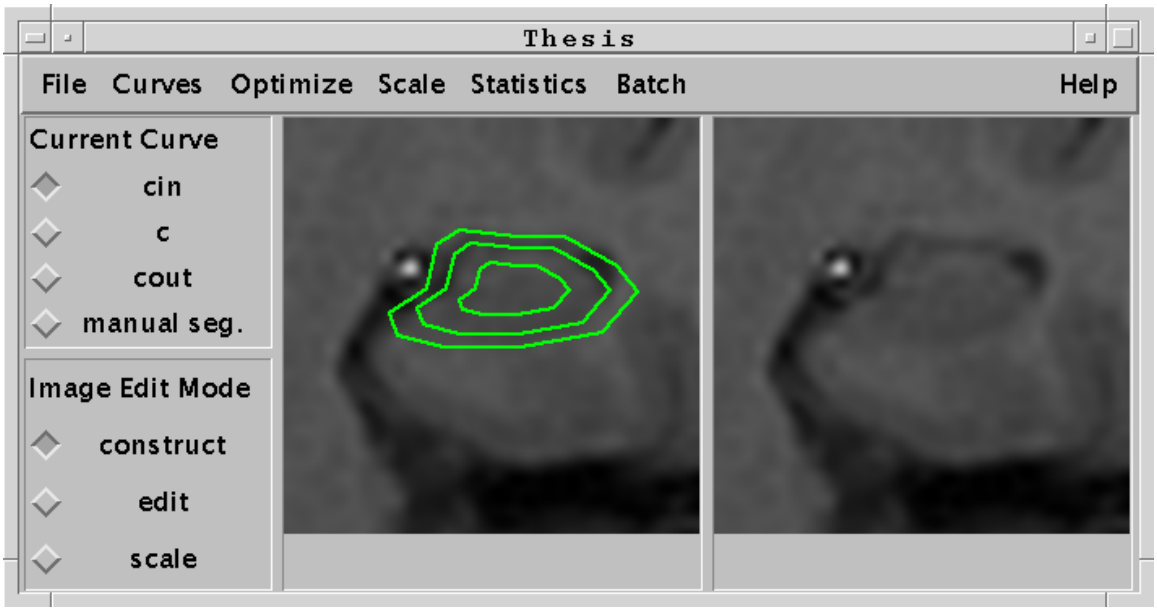

Figure 4.4 Bounding curves C, C_{in}, and C_{out} that define the template.

There also exists a "Curve Construction Dialog" that can be invoked from the menu item Curves: Invoke Curves Construction Box. This utility allows the user to construct the C_{in} curve, and scale it to the correct inner bounding size. The other curves can then be generated by scaling duplicates of the inner curve. This allows the operator to construct C, C_{in}, and C_{out} curves that are very similar. In our experimentation, the more similar the curves C, C_{in}, and C_{out} were, the more uniform our orthogonal curves were. The dialog also allows the user to delete curves. Each of the entered scale factors is a function of the current C_{in} curve. Shown in figure 4.4 is the "Construct Curves Dialog".

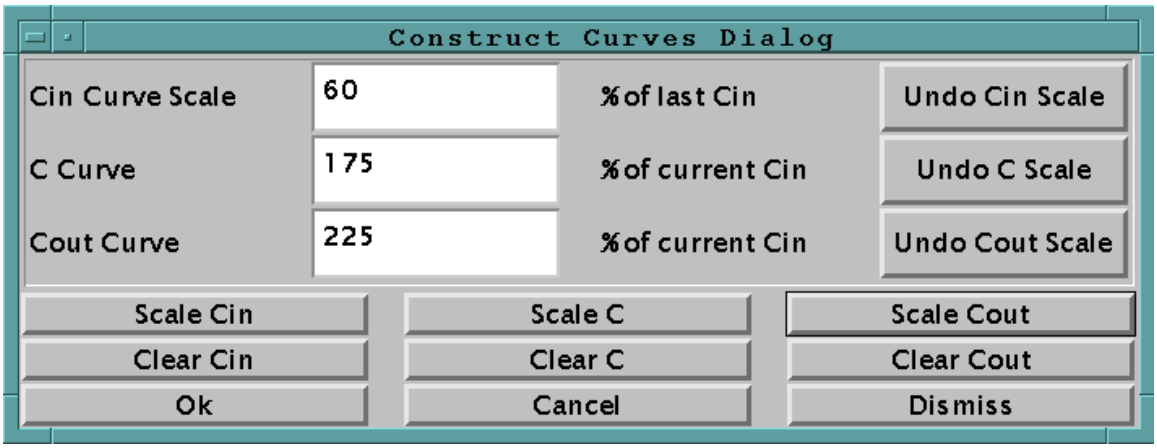

Figure 4.5 “Construct Curves” dialog box.

4.4 Generations of Orthogonal Curves

Once the application has a complete set of curves, it can compute the orthogonal curves. The polygons that form the curves are discretized into points by using the “Curves: Discretize Curves” menu item. Next, the invocation of the Curves: Make O Curves menu item will display a dialog box titled “Make O Curves Dialog”. This dialog is designed to give the user control over the generation of the orthogonal curves. Here the operator can enter the percentage of curves to generate, and the number of iterations to evaluate the PDE that determines the trajectory of he orthogonal curves. The percentage refers to the number of the discrete points that fall on the discrete curve C_{in} that will generate orthogonal curves. If the percentage is 100% then every point will start an orthogonal curve. If it is 10%, then every 10^{th} point will generate an orthogonal curve. Once the user has chosen the “Ok” button, the native C++ function to generate the orthogonal curves is invoked with the appropriate parameters. The “Make O Curves Dialog” box is displayed in figure 4.5.

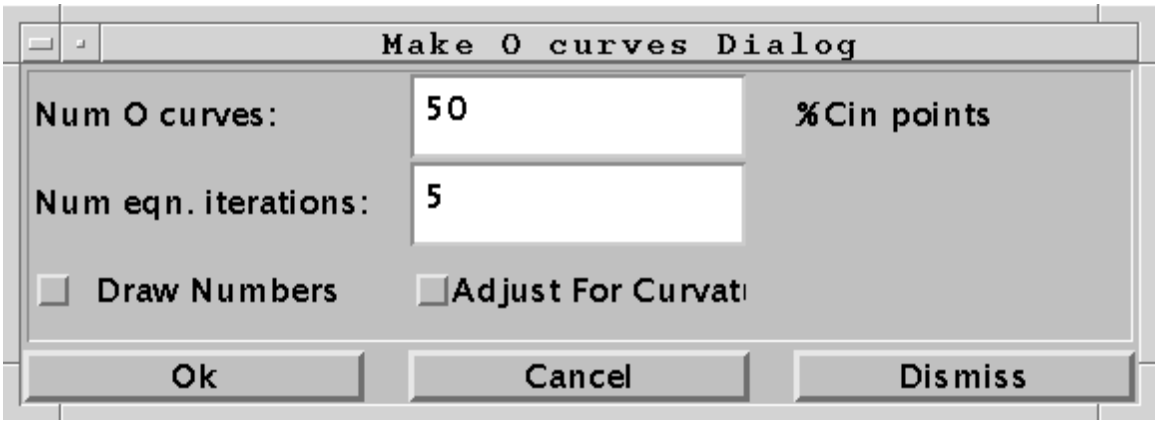

Figure 4.6 “Make O Curves Dialog” box.

The discrete curves are passed from Java to C++ in flattened arrays. Once they are expanded back in C++, the generation can proceed. The method of the curve generation begins by constructing an image of the two curves C and C_{in}. C_{in} and its interior are represented in the image by real valued 0.0s, and the curve C and everything outside of it is represented by real valued 1.0s. all of the points between the two curves in the image are represented by 0.5s. This image is then input into a numerical method published in [25] for the solution of boundary value elliptical partial differential equations, Laplace’s equation:

$$\frac{\partial^2 t}{\partial^2 y} + \frac{\partial^2 t}{\partial^2 y} = 0$$

as previously discussed in section 3.3 is the elliptical equation that yields our result. The number of iterations parameter from the “Make O Curves Construction Box” indicates the number of iterations the method will perform as it successively refines its result. For most of our experiments, 5 iterations were sufficient. If the deformation is small, a parameter of 1 is sufficient. Once this image is produced, another analogous image is

created for the curves *C* and C_{out}. The image is solved with respect to the above equation, and the resultant images are kept.

From the solution images, gradient images are produced. The central difference operator, as described in section 2.1, is used to generate the x and y components of the gradient images. Boundary pixels are duplicated for boundary conditions. When the gradient images have been generated, the orthogonal curves can be formed.

According to the user defined percentage parameter, candidate points on the curve C_{in} are chosen to initiate the orthogonal curves. From the initial orthogonal curve points, each curve moves from pixel to pixel according to the gradient trajectory. The decision to choose one pixel over another is made by using a variation of the Liang-Barsky line-clipping algorithm. At each point the linearly interpolated gradient direction is determined, and a line is generated from that point through an imaginary 1x1 bounding box. The intersection of the line with the bounding box is computed, and that determines the location of the next decision position. The algorithm moves to the next position and makes the next decision. The real valued decisions are then clamped to pixel positions to determine the components of the discrete orthogonal curve. The curve terminates on the *C* curve. When it reaches the C curve, the gradient image of the solution of Laplace's equation between *C* and C_{out} is then used to make the trajectory decisions. The terminating condition of the growth of these curves along the gradient trajectory is the boundary C_{out}. This process is repeated for each orthogonal curve initial point. The result is a set of orthogonal curves that is passed back to Java and displayed over the image in the left frame of the main image. The main application window with the orthogonal

curves displayed on the left is illustrated in figures 4.6, 4.7, and 4.8. The three figures have curves generated with a percent parameter of 8, 25, and 50 percent respectively.

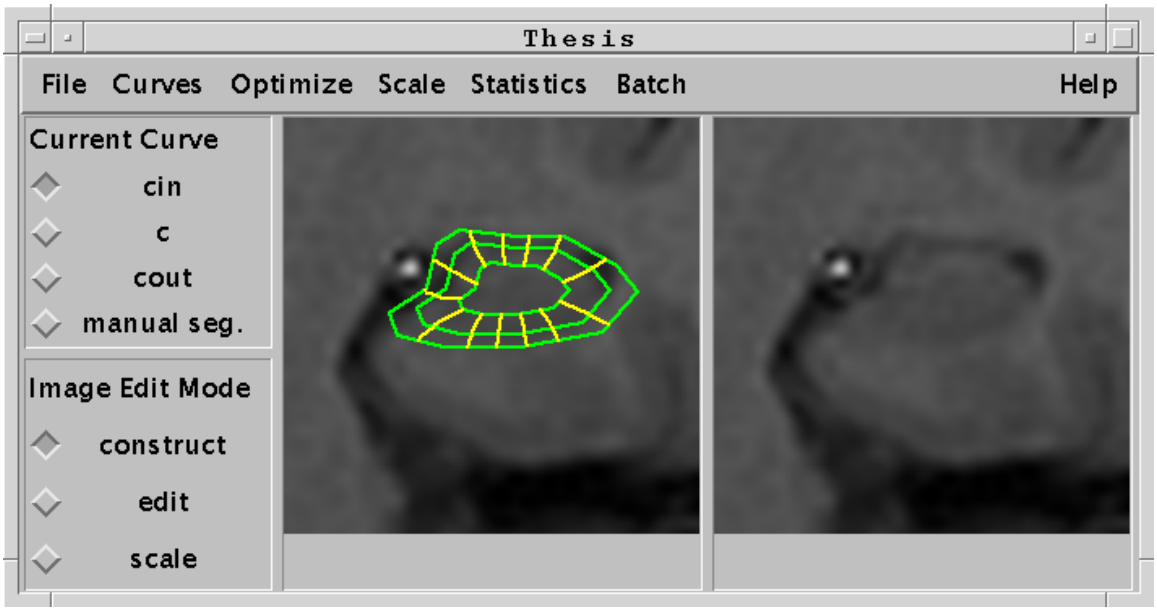

Figure 4.7 Orthogonal curves generated with 8% of the points on C_{in}.

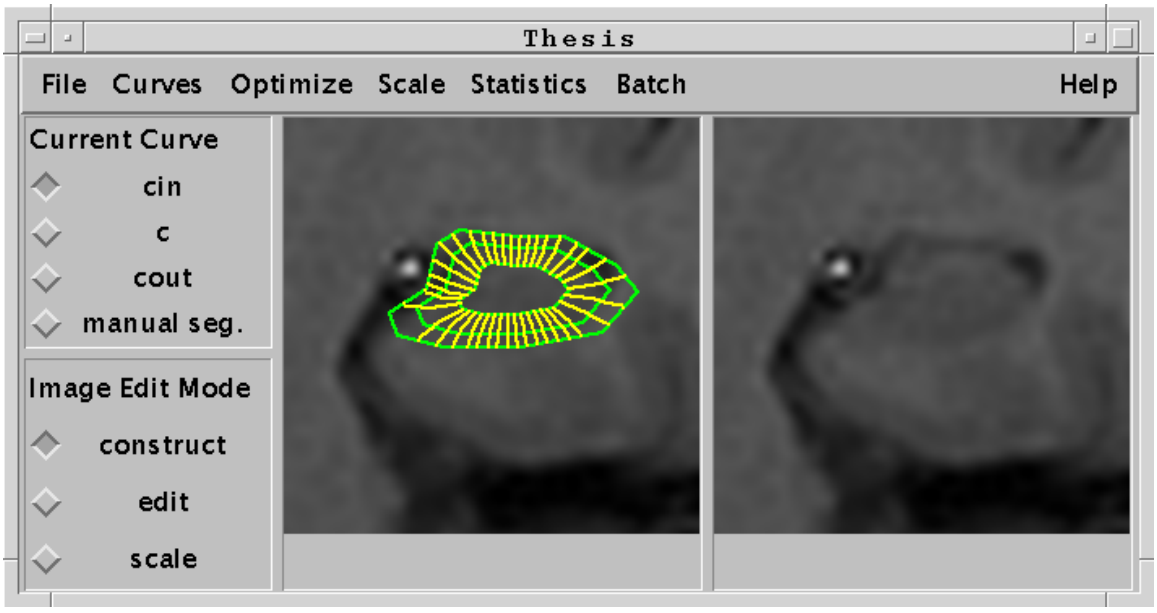

Figure 4.8 Orthogonal curves generated with 25% of the points on C_{in}.

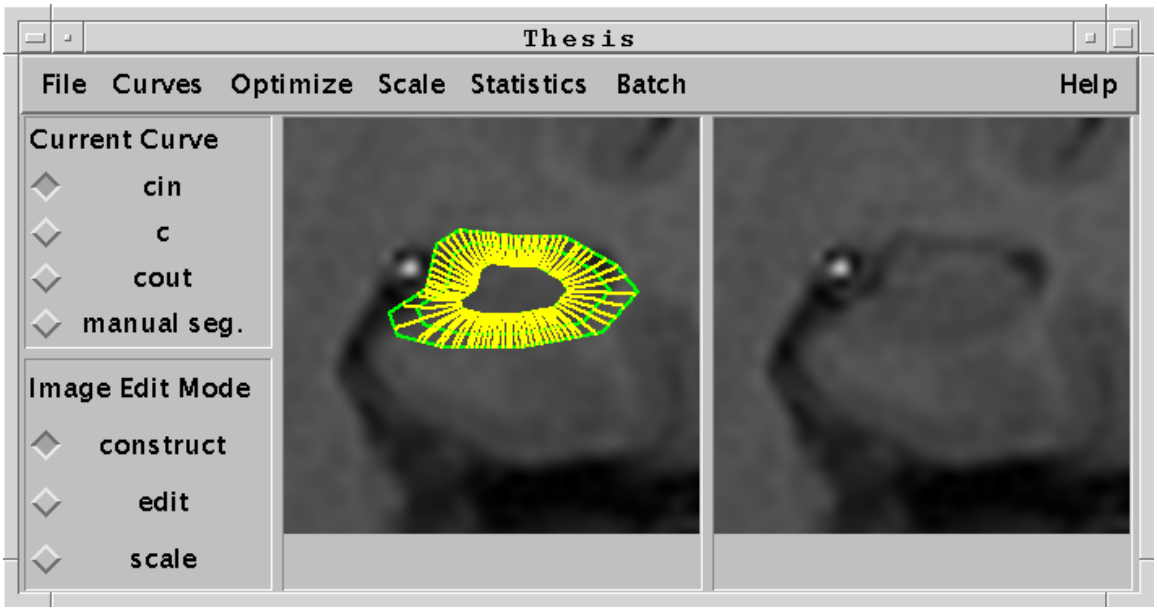

Figure 4.9 Orthogonal curves generated with 50% of the points on C_{in}.

4.5 Segmentation

Once the orthogonal curves have been generated, there exists a space for the template to deform across while attempting to minimize the energy formalized in 3.3. The operator should next designate the parameters for the optimization. The menu item "Optimize: Optimize" will invoke the "Energy Functions Coefficients" dialog box. This dialog box allows the user to enter the parameters ρ_1, ρ_2, ρ_3, and δ. They are named "External Energy Weight", "Distance Energy Weight", "Smoothness Energy Weight", and "Transition" respectively in the dialog box. The parameters ρ_1, ρ_2, ρ_3, and δ control the energy function in the manner described in 3.3-3.4. Once they have been designated, the "Ok" button will begin the optimization procedure. Figure 4.8 shows the "External Energy Coefficients" dialog box.

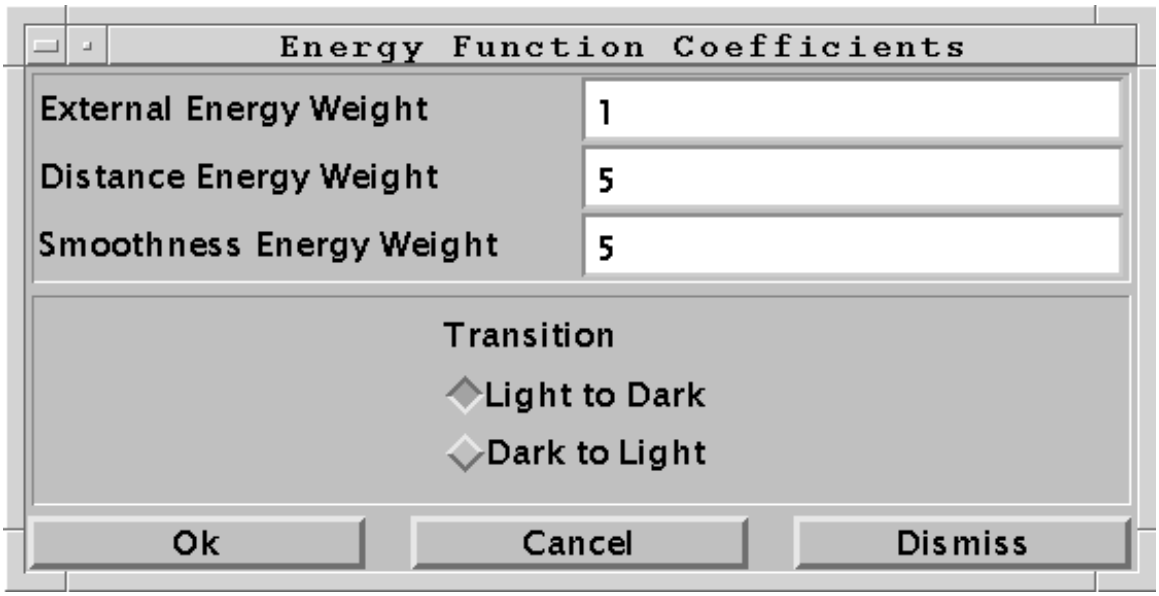

Figure 4.10 “Energy Function Coefficients” dialog box.

The optimization procedure is a native function that accepts its data and parameters from Java and executes natively in a Java thread. The dynamic programming procedure outlined in section 3.4 is used to optimize the segmentation. The Euclidian distance measure and the smoothness measure can be assumed to be implemented directly from the description in section 3.4. The external energy function computes the discrete line segment between each successive vertex using the midpoint line drawing algorithm [29]. For each discrete point on the line, the dot product between the unit normal orthogonal to the line and the image gradient is taken and summed. The sum becomes the external gradient energy for those two pixels. Once the minimum has been found, the resulting minimizing set of vertices $v_1,...,v_n$ that form the bounding polygon $P(v_1,...,v_N)$ are returned to the main Java application. The resulting boundary, or segmentation, is displayed in the right image panel of the main application window as

well as in the left image panel over the set of orthogonal curves. Figures 4.10 and 4.11 shows the results of segmentations using 8% and 25% of the points on C_{in} respectively.

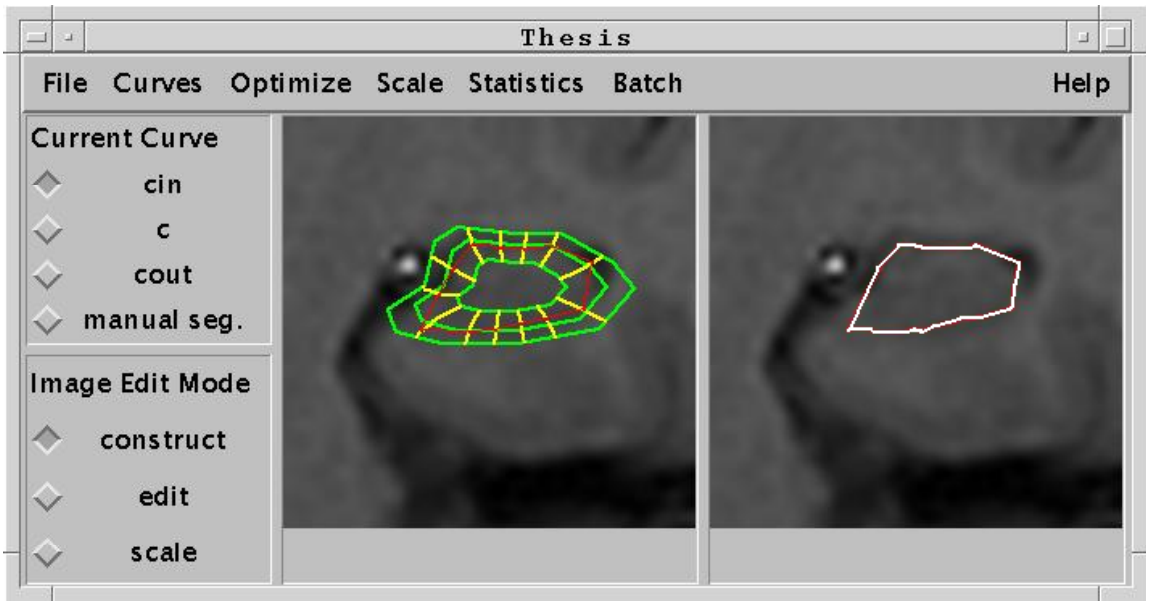

Figure 4.11 Result of a segmentation using 8% of the points on C_{in}.

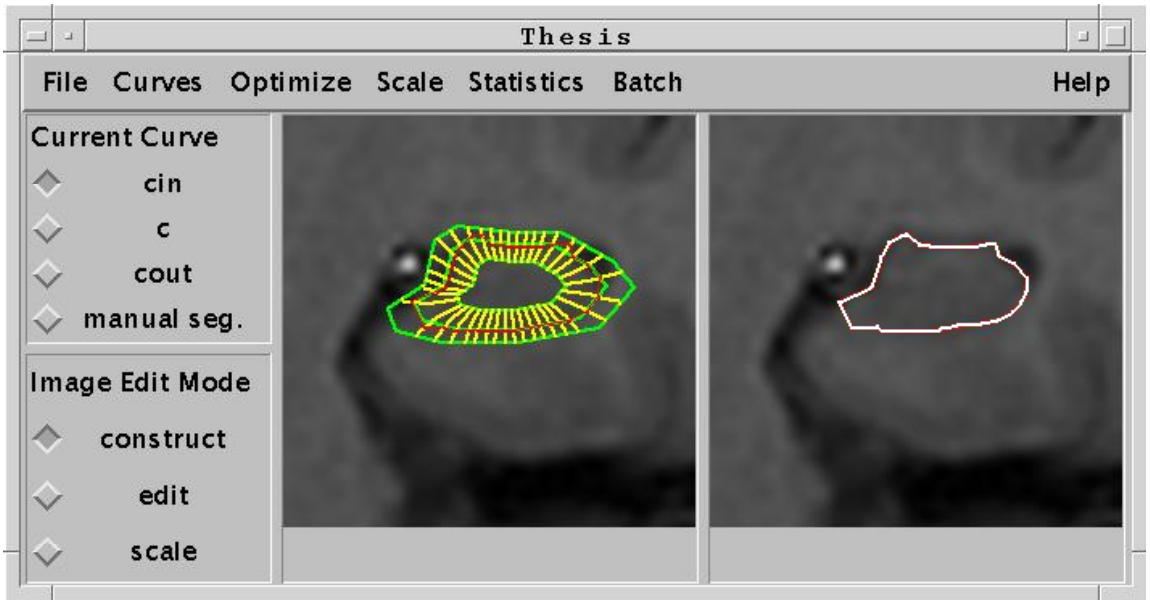

Figure 4.12 Result of a segmentation using 25% of the points on C_{in}.

Manual segmentations can be performed by choosing “Manual Seg.” Radio button from the “Current Curve” radio group in the upper left corner of the main application window. This will set the current curve mode to manual. Constructing and editing is performed in the same manner as the boundary curves were constructed.

4.6 Measurement

Once a segmentation has been performed and displayed in the right panel of the application window, it can be measured. A dialog box entitled “Calculate Area” can be invoked using the “Statistics: Calculate Area for Segmentation” menu item. The user clicks on the “Calculate Area” button to invoke the Java implementation of the area measurement algorithm. The measurement of the area of the polygon is done analytically on the polygon as opposed to pixel counting. A technique found in [26] uses the formula:

$$Area = \frac{1}{2}\begin{bmatrix} x_1 & x_2 & \dots & x_n & x_1 \\ y_1 & y_2 & \dots & y_n & y_1 \end{bmatrix}$$

to compute the area of a simple polygon. The orthogonal curves guarantee that the polygon generated for the boundary will be simple. The resulting area is displayed in the text box in square pixel units. Figure 4.12 shows the “Compute Area” dialog box. The compute area dialog also measures the manual segmentations.

Figure 4.13 “Compute Area” dialog box.

4.7 Miscellaneous Utilities

The application allows the user to perform some tedious operations in batches. Some frequently needed operations include: the conversion of file formats from the raw machine output of the MRI machine to rgb format; and the renumbering of files so their order is correctly interpreted by the operating system in use. These operations were implemented so they can be performed in batches. The "Batch: Perform Batch Operations" menu item invokes a "Batch" dialog box. This dialog box allows the operator to select groups of files to operate on, and destination directories in which to place the processed files. It contains a "Status:" text area that the status of operations is posted into in its upper left corner. Figure 4.13 shows the "Batch" dialog box.

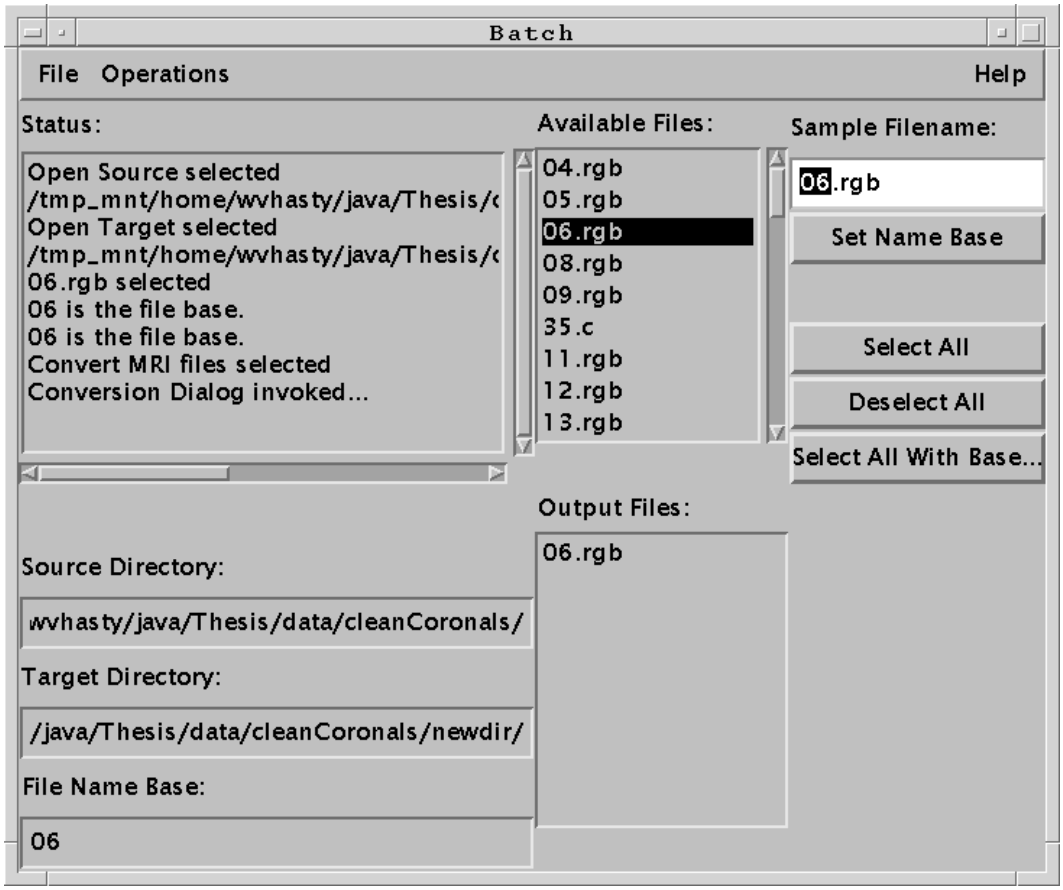

Figure 4.14 The "Batch" dialog box.

Chapter 5

EXPERIMENTATION AND RESULTS

The images used for the segmentations were acquired from a General Electric Signa Horizon Echo-Planar 1.5 Tesla scanner. They were acquired with the SPGR protocol using a 256x256 matrix, 2mm slices/0.0Sp (Skip Zero), and an unknown pixel size. The field of view (FOV) was 20x20, and the TR and TE parameters were variable. The subjects were normal, and were scanned through the volunteer research protocol of Dr. John Brockway at Presbyterian Hospital located in Charlotte, NC.

Data from two individuals were used, treating the left side and right side of the image scans as separate data sets. Only slices that our expert manual segmenter, Dr. John Brockway, selected as containing hippocampus were chosen for segmentation. Our data sets numbered 1 through 4 consisted of 5, 5, 11, and 8 slices respectively. The original size of each slice was 256x256, but the regions of interest were scaled up to facilitate the segmentation. The system described in chapter 4 was used to perform the semi-automatic batch scaling of the data sets. The same scaling window was used on all intra data set images without translation. The scaling method used was a bi-cubic spline interpolation based on the cardinal spline. The smoothing of the data was performed by the scaling operation itself. For data sets 1 and 2, a scale factor of 692% was used to enlarge the regions of interest. For data sets 3 and 4 a scale factor of 474% was used to enlarge the

regions of interest. Intra-rater error in the manual segmentations was measured by having the expert manual segmenter repeat each segmentation twice over a two-week period.

The semi-automatic segmentations were performed by first constructing a template that accurately represented the shape of the organ of interest. The templates were not constructed to duplicate the manual segmentations, rather they simply approximated the shapes. Once the template for an image was obtained it was applied to the image. The optimization step was performed with numerous combinations of parameters as we explored the nature of the algorithm, and it was determined that a set of three combinations of energy function parameters were most suitable for all of the data sets. These parameters are 1-3-3, 1-5-5, 1-7-7, where the three quantities respectively are the weights given to the external energy term, the distance energy term, and the smoothness energy term. See section 3.3 for a more complete description of these parameters. Initial experimentation yielded segmented boundaries that were caught in local gradient maxima and were not smooth when distance and smoothness parameters were less than three. Distance and smoothness weights greater than seven placed too much emphasis on the prior shape, and used too little gradient information. All of the transitions were considered to be light to dark transitions across the boundary of the hippocampus.

The results of the segmentations are listed in table 5.1. It shows the resulting areas from all of the manual and semi-automatic segmentations, as well as the scale factors for each image.

Data Set	Slice	Scale Percent	Image Width	Image Height	Study 1-3-3 Area	Study 1-5-5 Area	Study 1-7-7 Area	Manual 1 Area	Manual 2 Area
1	1	691.89	256	256	3145	3288	3250	2573	1576
1	2	691.89	256	256	1775	2088	2113	1750	2306
1	3	691.89	256	256	2616	2872	2877	2609	2495
1	4	691.89	256	256	4581	4551	4551	4596	3841
1	5	691.89	256	256	5695	5627	5534	4838	4580
2	1	691.89	256	256	2241	2586	2609	2318	2886
2	2	691.89	256	256	2245	2195	2198	2046	2678
2	3	691.89	256	256	3103	3066	3130	2633	2914
2	4	691.89	256	256	4900	4897	5012	4068	4590
2	5	691.89	256	256	4754	4748	4616	4031	4494
3	1	474.07	256	256	4192	4106	4046	4029	4233
3	2	474.07	256	256	3841	4094	3953	2959	2888
3	3	474.07	256	256	3363	3425	3405	3094	2222
3	4	474.07	256	256	2866	2948	2712	2606	1507
3	5	474.07	256	256	2661	2648	2607	1677	1690
3	6	474.07	256	256	1796	1925	1939	1328	1055
3	7	474.07	256	256	2214	2147	2050	1281	1107
3	8	474.07	256	256	2241	2158	2006	1191	900
3	9	474.07	256	256	1987	1922	1933	1185	852
3	10	474.07	256	256	2469	2408	2239	1626	871
4	1	474.07	256	256	310	310	416	316	433
4	2	474.07	256	256	920	948	961	734	701
4	3	474.07	256	256	1130	1069	1069	951	863
4	4	474.07	256	256	1752	1708	1632	1151	774
4	5	474.07	256	256	1662	1583	1659	1439	628
4	6	474.07	256	256	1771	1739	1725	1153	823
4	7	474.07	256	256	1530	1327	1218	756	518
4	8	474.07	256	256	1064	1086	1082	848	827

Table 5.1 Segmentation Results. Areas are measure in square pixels.

Table 5.2 lists the percent differences between all the slices segmented using external energy, distance energy, and smoothness energy parameters of 1,3,3 respectively.

Data Set	Slice	Study 1-3-3	Manual 1	Manual 2	Percent Difference between the Study and Manual 1	Percent Difference between the Study and Manual 2
1	1	3145	2573	1576	18.19	49.89
1	2	1775	1750	2306	1.41	29.92
1	3	2616	2609	2495	0.27	4.63
1	4	4581	4596	3841	0.33	16.15
1	5	5695	4838	4580	15.05	19.58
2	1	2241	2318	2886	3.44	28.78
2	2	2245	2046	2678	8.86	19.29
2	3	3103	2633	2914	15.15	6.09
2	4	4900	4068	4590	16.98	6.33
2	5	4754	4031	4494	15.21	5.47
3	1	4192	4029	4233	3.89	0.98
3	2	3841	2959	2888	22.96	24.81
3	3	3363	3094	2222	8.00	33.93
3	4	2866	2606	1507	9.07	47.42
3	5	2661	1677	1690	36.98	36.49
3	6	1796	1328	1055	26.06	41.26
3	7	2214	1281	1107	42.14	50.00
3	8	2241	1191	900	46.85	59.84
3	9	1987	1185	852	40.36	57.12
3	10	2469	1626	871	34.14	64.72
4	1	310	316	433	1.94	39.68
4	2	920	734	701	20.22	23.80
4	3	1130	951	863	15.84	23.63
4	4	1752	1151	774	34.30	55.82
4	5	1662	1439	628	13.42	62.21
4	6	1771	1153	823	34.90	53.53
4	7	1530	756	518	50.59	66.14
4	8	1064	848	827	20.30	22.27

Table 5.2 The percent differences between study 1-3-3 and the manual segmentations.

Table 5.3 lists the percent differences between all the slices segmented using external energy, distance energy, and smoothness energy parameters of 1,5,5 respectively.

Data Set	Slice	Study 1-5-5	Manual 1	Manual 2	Percent Difference between the Study and Manual 1	Percent Difference between the Study and Manual 2
1	1	3288	2573	1576	21.75	52.07
1	2	2088	1750	2306	16.19	10.44
1	3	2872	2609	2495	9.16	13.13
1	4	4551	4596	3841	-0.99	15.60
1	5	5627	4838	4580	14.02	18.61
2	1	2586	2318	2886	10.36	11.60
2	2	2195	2046	2678	6.79	22.00
2	3	3066	2633	2914	14.12	4.96
2	4	4897	4068	4590	16.93	6.27
2	5	4748	4031	4494	15.10	5.35
3	1	4106	4029	4233	1.88	3.09
3	2	4094	2959	2888	27.72	29.46
3	3	3425	3094	2222	9.66	35.12
3	4	2948	2606	1507	11.60	48.88
3	5	2648	1677	1690	36.67	36.18
3	6	1925	1328	1055	31.01	45.19
3	7	2147	1281	1107	40.34	48.44
3	8	2158	1191	900	44.81	58.29
3	9	1922	1185	852	38.35	55.67
3	10	2408	1626	871	32.48	63.83
4	1	310	316	433	1.94	39.68
4	2	948	734	701	22.57	26.05
4	3	1069	951	863	11.04	19.27
4	4	1708	1151	774	32.61	54.68
4	5	1583	1439	628	9.10	60.33
4	6	1739	1153	823	33.70	52.67
4	7	1327	756	518	43.03	60.96
4	8	1086	848	827	21.92	23.85

Table 5.3 The percent differences between study 1-5-5 and the manual segmentations.

Table 5.4 lists the percent differences between all the slices segmented using external energy, distance energy, and smoothness energy parameters of 1,7,7 respectively.

Data Set	Slice	Study 1-7-7	Manual 1	Manual 2	Percent Difference between the Study and Manual 1	Percent Difference between the Study and Manual 2
1	1	3250	2573	1576	20.83	51.51
1	2	2113	1750	2306	17.18	9.13
1	3	2877	2609	2495	9.32	13.28
1	4	4551	4596	3841	0.99	15.60
1	5	5534	4838	4580	12.58	17.24
2	1	2609	2318	2886	11.15	10.62
2	2	2198	2046	2678	6.92	21.84
2	3	3130	2633	2914	15.88	6.90
2	4	5012	4068	4590	18.83	8.42
2	5	4616	4031	4494	12.67	2.64
3	1	4046	4029	4233	0.42	4.62
3	2	3953	2959	2888	25.15	26.94
3	3	3405	3094	2222	9.13	34.74
3	4	2712	2606	1507	3.91	44.43
3	5	2607	1677	1690	35.67	35.17
3	6	1939	1328	1055	31.51	45.59
3	7	2050	1281	1107	37.51	46.00
3	8	2006	1191	900	40.63	55.13
3	9	1933	1185	852	38.70	55.92
3	10	2239	1626	871	27.38	61.10
4	1	416	316	433	24.04	4.09
4	2	961	734	701	23.62	27.06
4	3	1069	951	863	11.04	19.27
4	4	1632	1151	774	29.47	52.57
4	5	1659	1439	628	13.26	62.15
4	6	1725	1153	823	33.16	52.29
4	7	1218	756	518	37.93	57.47
4	8	1082	848	827	21.63	23.57

Table 5.4 The percent differences between study 1-7-7 and the manual segmentations.

Table 5.5 lists the percent differences between all the manually segmented slices in the first group of manual segmentations, and the slices manually segmented in the second group.

Data Set	Slice	Manual 1	Manual 2	Percent Difference 1 vs. 2	
1	1	2573	1576	38.748543	high
1	2	1750	2306	31.771429	
1	3	2609	2495	4.3694902	low
1	4	4596	3841	16.427328	
1	5	4838	4580	5.3327821	
2	1	2318	2886	24.503883	high
2	2	2046	2678	30.889541	
2	3	2633	2914	10.672237	low
2	4	4068	4590	12.831858	
2	5	4031	4494	11.485984	
3	1	4029	4233	5.0632911	
3	2	2959	2888	2.3994593	
3	3	3094	2222	28.183581	
3	4	2606	1507	42.171911	
3	5	1677	1690	0.7751938	low
3	6	1328	1055	20.557229	
3	7	1281	1107	13.583138	
3	8	1191	900	24.433249	
3	9	1185	852	28.101266	
3	10	1626	871	46.432964	high
4	1	316	433	37.025316	
4	2	734	701	4.4959128	
4	3	951	863	9.2534175	
4	4	1151	774	32.754127	
4	5	1439	628	56.358582	high
4	6	1153	823	28.620989	
4	7	756	518	31.481481	
4	8	848	827	2.4764151	low

Table 5.5 The percent differences between the two manual segmentation groups. The high and low labels to the right indicate the high and low percentages for each data set.

.

Table 5.6 lists the average percent differences between the areas derived by manual segmentation and semi-automatic segmentation by data set.

Data Set	Study	Average Percent Difference Between Study and Manual 1 Segmentations	Average Percent Difference Between Study and Manual 2 Segmentations	Average Percent Differences between Manual Segmentations 1&2
1	1-3-3	7.05	24.03	
1	1-5-5	12.42	21.97	19.33
1	1-7-7	12.18	21.35	
2	1-3-3	11.93	13.19	
2	1-5-5	12.66	10.04	18.08
2	1-7-7	13.09	10.08	
3	1-3-3	27.05	41.66	
3	1-5-5	27.45	42.42	21.17
3	1-7-7	25.00	40.97	
4	1-3-3	23.94	43.39	
4	1-5-5	21.99	42.19	25.31
4	1-7-7	24.27	37.31	

Table 5.7 Average percent differences between manual and semi-automatic segmentations by data set.

Table 5.7 is perhaps the most important for the interpretation of our results. Data sets 1 and 2 have the largest scaling factor, 692% versus 474% for data sets 3 and 4. This results in the hippocampus having larger image area in these images. The percent error between the semi-automatic segmentation and both of the manual segmentations is lower for the data sets with the larger hippocampus images present. From the percent differences it appears that there is more agreement between successive manual segmentations than between semi-automatic and manual segmentations. However this can be deceiving if the percentages are large positive and large negative numbers. Large errors may be being canceled out. The mean squared error of the areas tells a more

accurate story. Figure 5.8 shows the mean squared error of study versus both manual segmentations, and between the manual segmentations.

Data Set	Study	Mean Squared Error Study vs.1	Mean Squared Error Study vs.2	Mean Squared Error 1 vs. 2
1	1-3-3	212506.4	909837.6	390546.0
1	1-5-5	263836.8	944181.2	
1	1-7-7	229672.6	879933.0	
2	1-3-3	296276.6	160587.0	257572.4
2	1-5-5	296568.8	101031.6	
2	1-7-7	317631.0	109350.6	
3	1-3-3	545857.6	1241613.3	288541.1
3	1-5-5	565966.7	1284035.2	
3	1-7-7	452536.2	1079552.3	
4	1-3-3	188157.4	517379.5	123544.6
4	1-5-5	139602.8	432946.6	
4	1-7-7	118824.8	409759.9	

Table 5.8 The mean squared error of study versus both manual segmentations, and between the manual segmentations.

The automatic segmentation is at least as accurate as the manual segmentations. It is difficult to measure the error because it must be evaluated with respect to the manual segmentations. The manual segmentations vary widely with respect to their area and size. This only highlights the high variability in measurement accuracy from the same rater, and stresses the need for a good semi-automatic technique. The mean squared errors in table 5.8 suggest that the semi-automatic technique produces lower error with respect to one manual segmentation for each data set, than do the two manual segmentations with respect to each other.

Chapter 6

CONCLUSIONS

Orthogonal curve based deformable templates provide a quick segmentation technique that can escape moderate local gradient maxima with the right parameters. We have found that it is acceptable to apply the same parameters to most of the slices in a data set. But there exist some slices within each data set that need fine tuning for a variety of reasons. The technique must be supervised to detect these cases. Each segmentation must be visually inspected to assure its reliability, and adjusted to accommodate artifacts and unusually high gradients that may occur local to the area of the hippocampus.

The presence of a very strong gradient local to the object of interest, but not belonging to its boundary, will cause the template to deform improperly. The strong local gradient in essence "fools" the technique. In order to pull the template back into shape, the smoothness and distance parameters often have to be raised unnaturally high. Very often the result of overemphasizing the distance and smoothness parameters is a rubberband effect. The template will conform to most of the boundary of the shape with the exception of the strong gradient in its search space that does not belong to the hippocampus. It will naturally be highly attracted to the strong gradient. By incrementally increasing the smoothness and distance parameters, the template can eventually be

snapped out of this maxima; and it will conform to the actual boundary of the organ. The real trouble is that it often takes very high distance and smoothness parameters to snap the template back into shape. If the parameters are too exaggerated, then distance and smoothness energies will overpower the external image energy, and the deformed template simply becomes a duplicate of the original *C* curve. In this case, the technique reduces to manual segmentation. See figure 6.1 for a sequence of images that illustrate the rubberband phenomenon.

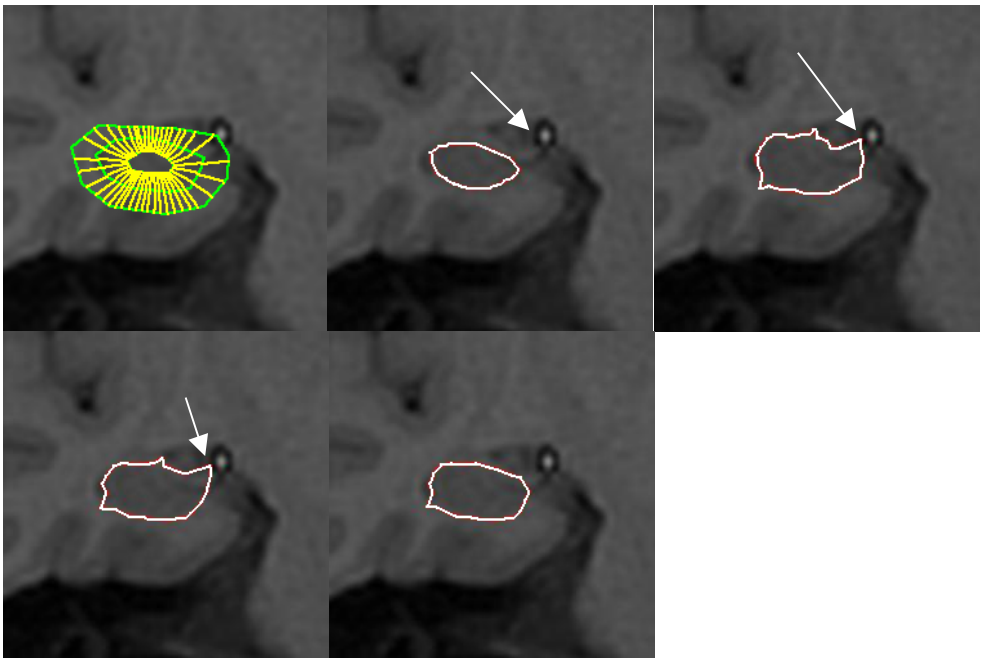

Figure 6.1 A strong gradient stretching the boundary. *The upper left frame contains the orthogonal curves. The upper middle frame shows a manual segmentation (Note the white arrow indicating the string gradient that will fool the technique). The upper right frame shows the automatic segmentation with smoothness and distance parameters of 5. The gradient has caught the vertex of the boundary nearest it while the rest of the boundary is relatively close to the hippocampus. The lower left frame shows the segmentation with smoothness and distance parameters of 7 (Note the boundary is still being stretched). In the lower right frame, the parameters at 10 have finally snapped the boundary into place around the shape.*

The orthogonal curves technique does not perform well when the organ has a large part of its boundary with a light to dark gradient transition and another sizeable part

demonstrating a dark to light transition. This has occurred in experimentation. The template behaves by skipping one of the transitions and seeking some other gradient marker. But when the incomplete gradient boundary is not large, the template will bridge the gradient gap with even modest distance and smoothness parameters. For the small gradient inconsistencies, this technique will correctly identify the boundary of the hippocampus. Even so, the large gradient inconsistencies are not easily or gracefully accommodated. The possibility of their presence demands supervision. A reformulation of the energy function may be able to solve this problem.

In a clinical environment, it is unreasonable to expect an expert rater to spend time constructing templates when manual segmentation is much quicker. But it is reasonable for an expert rater to select a template from a few well-known templates and place it over an organ of interest if the time for the automatic stage of the segmentation is very small (the search phase). This is a reasonable approach if the organ of interest displays a relatively high degree of inter-subject shape and size coherence. In the case of the hippocampus, we have seen little evidence that this is the case.

The execution time of other deformable template segmentation techniques is too prohibitive to be considered for implementation in a clinical environment. Even with large numbers of curves, up to 50%, and hippocampal images that have been scaled up 600% to fill a quarter of a 256x256 image, the orthogonal curves technique still can perform its segmentation on an Intel 200mhz Pentium II or an 180mhz MIPS r5000 O2 in less than 5 seconds. Combined with the time to select a template, this amount of time is reasonable for a clinical setting if it means that a higher degree of accuracy can be obtained during the measurement. If it is necessary to construct a new template, the

technique takes too much time for practical application in a clinical environment. A database of templates that can be transformed into family of templates may help speed up the template acquisition time to an acceptable rate.

This technique would perform well with organs that displayed some degree of inter-slice shape or size coherence. In that case, the same template could be applied to adjacent image slices. The hippocampus has neither inter-slice shape nor size coherence. In fact the organ frequently varies tremendously from slice to slice. This demands that a new template be selected or constructed for each image.

When the organ of interest is of relatively smooth shape, the smooth templates generate more uniform orthogonal curves than in the case of shapes with expected high degrees of curvature. The hippocampus appears in the data as both. In the anterior portion of the hippocampus (coronal view) the hippocampus has a large smooth shape. In the most posterior portion of the hippocampus, the organ begins to acquire a thin "S" shape that is very difficult to segment with orthogonal curves. If there is too much curvature and too many curves are generated, the risk of singular points in orthogonal curves arises and the segmentation is not possible. It is not reasonable for a manual rater to be expected to know and apply these restrictions about the application of the algorithm during a segmentation.

The orthogonal curves technique by itself is not reliable enough to be applied in a clinical environment. It requires careful curve construction that takes longer than manual segmentation. It requires fine-tuning of parameters, and constant supervision for the avoidance of degenerate boundaries. It also requires some knowledge of the nature of the orthogonal curves for the operator to avoid the singular points created by orthogonal

curve intersection. The extremely high inter and intra subject anatomical variability of the hippocampus makes the generation of a predefined database set of templates a less attractive approach, but its feasibility has yet to be determined. The most alluring facet of the orthogonal curves approach is the execution time with respect to the alternative deformable template approaches. The high shape variability of the hippocampus makes the orthogonal curves technique impractical if new templates must be constructed for each data slice. However, other brain structures with higher slice to slice coherence may be effectively segmented using orthogonal curves. The increased coherence may make it possible to apply the same template across many data slices. This would minimize the template construction time, and make the technique more clinically applicable. Organs with more well-defined gradients are also better candidates for segmentation with this technique.

Chapter 7

FUTURE WORK

There are several approaches that can be taken to improve this system. Since the major problems for clinical application lie in the: curve construction time, the possibility of strong non-hippocampal gradients local to the organ, the fine tuning of parameters, the nature of the orthogonal curves, and the high degree of anatomical variability; these are the natural place for extensions and research.

A database of orthogonal curve based templates with thumbnail identification images could greatly facilitate the selection of orthogonal curves. If the database contained a curve of the same approximate shape as the region of interest, the template could be scaled and translated to fit the image. This would for the most part eliminate the time needed for curve construction.

An improved energy function may be able to eliminate the possibility of grossly inaccurate boundaries by displaying more tolerance for certain types of artifacts and gradients. An improved energy function may be able to avoid the tuning of the parameters. If there were some pre-processing step that examined the gradients within the search space, it is possible that the technique could be self-tuning.

The orthogonal curves run the risk of intersection in areas of high concave curvature. If the number of curves were reduced in these areas, it would greatly reduce

the risk of singular points and make the algorithm more robust with respect to the curve construction. In areas of high convex curvature, radial error is introduced. The opposite approach can be taken in these cases. If the number of orthogonal curves is increased in these areas of high convex curvature, then the radial error can be reduced. This adaptive sampling of the orthogonal curves based on degree and inflection of curvature will most likely lead to templates with more uniform sampling coverage across the template space.

BIBLIOGRAPHY

[1] Milan Sonka, Vaclaw Hlavac, and Roger Boyle, "Image Processing, Analysis and Machine Vision". Chapman and Hall Computing, 1993.

[2] Raphale C. Gonzalez and Richard E. Woods, "Digital Image Processing". Addison-Wesley Publishing Company, 1992.

[3] W.M. Wells III, W.E.L. Grimson, R.Kikinas, and F.A. Jolesz, "Adaptive Segmentation of MRI Data". IEEE Transactions on Medical Imaging, Vol. 15, No. 4, 1993.

[4] J.C. Bezdek, L.O. Hall, L.P. Clark, "Review of MR Image Segmentation Techniques Using Pattern Recognition". Medical Physics, Vol. 20, No. 4, 1993.

[5] Edward A. Ashton, Kevin J. Parker, Michel J Berg, and Chang Wen Chen, "A Novel Volumetric Feature Extraction Technique with Applications to MR Images". IEEE Transactions on Medical Imaging, Vol. 16, No. 4, 1997.

[6] G. Sundaramoorthy, W. E. Higgins, J. Hoford, E. A. Hoffman, "A Graphical User Interface System for Automatic 3-D Medical Image Analysis". IEEE Symposium for Computer-based Medical Systems, 1992.

[7] Hemant D. Tagare, "Deformable 2-D Template Matching Using Orthogonal Curves". IEEE Transactions on Medical Imaging, Vol. 16, No.1, 1997.

[8] K.R. Subramanian, D.M. Lawrence, and M.T. Mostafavi,, "Interactive Segmentation and Analysis of Fetal Ultrasound Images". 8th Eurographics Workshop on Visualization in Scientific Computing, bologne Sur Mer, France, 1997.

[9] Rajaraman Rajamani, "A Template Matching Based Video Processing Technique to Remove Motion Artifacts". Thesis for Master of Science degree, University of North Carolina Charlotte, 1997.

[10] V. Koivunen and M. Pietikainen, "On Threshold Selection Using Clustering Criteria". IEEE Transactions on Systems, Man, and Cybernetics, Vol. 15, No. 5, 1985.

[11] Jagath C. Rajapakse, Jay N. Giedd, and Judith L. Rapoport, "Statistical Approach to Segmentation of Single-Channel Cerebral MR Images". IEEE Transactions on Medical Imaging, Vol. 16, No. 2, 1997.

[12] W.E. Grimson, G.J. Ettinger, T. Kapur, M.E. Leventon, W.M. Wells III, R. Kikinis, "Utilizing Segmented MRI Data in Image-Guided Surgery", International Journal of Pattern Recognition and Artificial Intelligence, 1996.

[13] Michael Kass, Andrew Witkin, and Demetri Terzopoulos, "Snakes: Active Contour Models". International Journal of Computer Vision, 321-331, 1988.

[14] Anthony Yezzi Jr., Satyanad Kichenassamy, Arun Kumar, Peter Oliver, and Allen Tannenbaum, "A Geometric Snake Model for Segmentation of Medical Imagery". IEEE Transactions on Medical Imaging, Vol. 16 No. 2, 1997.

[15] Ulf Grenander, Michael I. Miller, Gary E. Christensen, "Deformable Anatomical Databases Using Global Shape Models: A Position Paper for the 1992 Electronic Imaging of the Human Body Workshop", Proceedings of the Cooperative Working Group on Whole Body 3-D Electronic Imaging of the Human Body, 1992.

[16] Michael I. Miller, Gary E. Christensen, Yali Amit, and Ulf Grenander, "Mathematical Textbook of Deformable Neuroanatomies". Proceedings of the National Academy of Science, Vol.90, No. 24, 1993.

[17] J. Haller, G.E. Chhristensen, M.I. Miller, M. Gado, D.W.McKeel Jr., J.G. Csernansky, and M.W. Vannier, "A Comparison of Automated and Manual Segmentation of Hippocampus of MR Images". SPIE Proceedings of the Conference on Medical Imaging, 1995.

[18] Gary E. Christensen, Richard D. Rabbitt, and Michael I. Miller, "3D Brain Mapping Using a Deformable Neuroanatomy". Physics in Medicine and Biology, Vol. 39, 1994.

[19] Ron Kikinis, Martha E. Shenton, Dan V. Iosifescu, Robert W. McCarley, Pairash Saiviroonporn, Hiroto H. Hokama, Andre Robatino, David Metcalf, Cynthia G. Wible, Charra M. Partas, Robert M. Donnino, and Ferenc A. Jolesz, "A Digital Brain Atlas for Surgical Planning, Model-Driven Segmentation, and Teaching". IEEE Transactions on Visualization and Computer Graphics, Vol. 2, No. 3, 1996.

[20] G.E. Christensen, M.I.. Miller, J.L. Marsh, and M.W. Vannier, "Automatic Analysis of Medical Images Using a Deformable Textbook". Computer Assisted Radiology: Proceedings of the International Symposium on Computer and Communication Systems for Image Guided Diagnosis and Therapy, 1995.

[21] Gary E. Christensen, "Deformable Shape Models for Anatomy". Dissertation for Doctor of Science degree, Washington University, Sever Institute of Technology, 1994.

[22] Gary E. Christensen, Richard D. Rabbitt, and Michael I. Miller, "A Deformable Neuroanatomy Textbook Based on Viscous Fluid Mechanics". Proceedings of the 1993 Conference on Information Sciences and Systems, 1993.

[23] John W. Haller, Gary E. Christensen, Sarang Joshi, Michael I. Miller, and Micheal W. Vannier, "Digital Atlas Based Segmentation of the Hippocampus". Computer Assisted Radiology: Proceedings of the International Symposium on Computer and Communication Systems for Image Guide Diagnosis and Therapy, 1995.

[24] J. Haller, G.E. Christensen, M.I. Miller, M. Gado, D.W. McKeel Jr., J.G.Csernansky, and M.W.Vannier, "A Comparison of Automated and Manual Segmentation of Hippocampus MR Images". SPIE Proceedings of the Conference on Medical Imaging, 1995.

[25] William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery, "Numerical Recipes in C: The Art of Scientific Computing". Cambridge University Press, 1992.

[26] Jon Rokne, "The Area of a Simple Polygon". Graphics Gems I, p. 1, Academic Press, Inc., 1991.

[27] David G. Amaral, "Hippocampal Formation". Encyclopedia of Human Biology, Vol. 4, pp. 227-233, Academic Press 1991.

[28] Will Schroder, Ken Martin, and Bill Lorensen, "The Visualization toolkit: An Object Oriented Approach to 3D Graphics, 2nd Edition". Prentice-Hall, 1997.

[29] James D. Foley, Stephen Feiner, Andries Van Dame, Hughes, "Computer Graphics: Principles and Practice, 2nd Edition". Addison-Wesley, 1996.