University of California
Los Angeles

Segmentation of 3D MR Images of the Brain Using a PCA Atlas and Nonrigid Registration

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Science in Computer Science

by

Gautam Prasad

2010

The thesis of Gautam Prasad is approved.

David Shattuck

Zhuowen Tu

Stefano Soatto

Demetri Terzopoulos, Committee Chair

University of California, Los Angeles

2010

Table of Contents

List of Figures

List of Tables

Acknowledgments

I am grateful to Demetri Terzopoulos and David Shattuck for being my thesis supervisors. The valuable discussions I had with them were critical to the ideas in this thesis. Their careful and detailed readings of the text helped me to learn about organization and style in writing.

I am thankful to Zhuowen Tu and Stefano Soatto for comprising the rest of my thesis committee. Their comments on the significance of this work were very enlightening.

I am thankful to Alex Vasilescu for her technical assistance with PCA methods and discussions about the methods being used in this thesis.

I am appreciative of Owen Carmichael for giving me my first opportunity to conduct medical imaging research and for a wealth of timely and indispensable advice and help throughout my undergraduate and graduate work.

I am indebted to Arthur Toga and the Laboratory of Neuro Imaging (LONI) for supporting me during the past three years and providing an excellent research environment that has exposed me to so much in medical imaging.

I am grateful to Ilya Eckstein, Shantanu Joshi, and Anand Joshi for their thoughtful advice and discussions pertaining to research and graduate school.

I am thankful to Wenona Colinco and Verra Morgan for all their assistance with administrative issues and paperwork within the Computer Science Department.

I am thankful to Nina Gupta for her friendship and love, who along with Shatrugana and Sitara have provided great companionship.

I am extremely grateful to my family: my parents, Manoj Prasad and Punita Sinha; my grandparents, Sheo Prasad, Kamini Prasad, Krityunjai Sinha, and

Kanti Sinha; and uncle Sanjai Sinha. Their love and support throughout my life and education has made life enjoyable.

Abstract of the Thesis

Segmentation of 3D MR Images of the Brain Using a PCA Atlas and Nonrigid Registration

by

Gautam Prasad

Master of Science in Computer Science

University of California, Los Angeles, 2010

Professor Demetri Terzopoulos, Chair

This thesis presents a method for the automatic segmentation of the brain in magnetic resonance (MR) images of the human head. The method identifies brain areas of interest, including the gyri and other subcortical structures, that were manually delineated in a set of labeled training images. Principal components analysis (PCA) is applied to the training ensemble in order to learn a PCA atlas subspace, which is a dimensionality-reduced linear subspace of labeled brain images. We employ this subspace to segment and label previously unseen subject images. This is accomplished by finding the PCA atlas closest to an input subject image through orthogonal projection of the latter into the subspace. The PCA atlas is then nonrigidly registered to the subject image and the nonrigid transformation is used to transfer the labels from the former to the latter, thereby segmenting the subject image. Our method is compared with alternative methods and the results are validated using overlap and distance metrics.

CHAPTER 1

Introduction

The *segmentation* of brain images involves localizing and delineating the brain's different anatomical structures within three-dimensional (3D) tomographic images of the head. Brain image segmentation is important for longitudinal studies investigating how medications and other interventions affect the shapes and volumes of brain tissues, for surgical planning and for diagnosing disease. Traditionally, the various parts of the brain are segmented by trained experts who manually identify and delineate regions of interest. This process can be very difficult and time consuming. It requires the design of complex protocols for identifying brain structures. The results from different manual segmentations using the same protocol may be inconsistent, which raises issues if a set of images in a study are delineated by different individuals. With the advent of non-invasive medical imaging devices that can easily acquire large quantities of tomographic data, manual segmentation becomes impractical for many studies. An alternative to manual segmentation is to develop computer algorithms that can automatically segment anatomical structures of interest in images of the brain.

The automatic 3D brain image segmentation algorithm developed in this thesis involves computing a 3D *atlas* consisting of labeled voxels, from among a space of possible atlases learned from an ensemble of manually segmented training images, that is most suitable for segmenting an unlabeled subject image. This optimal atlas is then nonrigidly registered to the subject image. Finally,

the nonrigid transformation is used to transfer the labels from the atlas to the subject image, thus segmenting the subject image.

In greater detail, our procedure is as follows: Registration is the process of establishing correspondences between two images by finding a transformation of a source image into the coordinate system of a target image. First a set of training data comprising brain intensity images and their corresponding labels are mutually registered to minimize the variance across the data. This ensures that the variation in the data is attributed mostly to the differences in the shapes and textures of the anatomical structures of the brain and not to their locations, orientations, and sizes. Principal components analysis (PCA) is then performed on the intensity and label training data to learn a dimensionally-reduced linear subspace that is capable of generating labeled images that interpolate the training data. We refer to this subspace as a *PCA atlas subspace*. Then, an orthogonal projection is performed to find the point in the PCA atlas subspace that is closest to the subject image. The point is used to generate an optimal intensity image with its associated labels, which we call the *PCA atlas*. A nonrigid registration algorithm is then used to deform the input subject image to match the intensity image component of the PCA atlas and the resulting nonrigid transformation is finally used to map the associated labels over to the subject image in order to complete the segmentation.

We apply our algorithm to 3D magnetic resonance (MR) images of the brain.

1.1 Applications of Brain MR Image Segmentation

The results from brain MR image segmentation can provide a wealth of information that is important in understanding and dealing with the brain. The insight

that segmentation affords about the shape and volume of different brain structures helps in understanding how the brain changes over time. It enables us to gain an understanding of the effects of disease and medication on the brain and can enable us to detect the onset of disease through the classification of these changes. Segmentation also allows us to reconstruct three-dimensional models of the structures in the brain that can provide assistance to surgical planning and guidance. The following is a survey of some of the literature that makes use of the results from brain MR image segmentation, emphasizing its importance in understanding the brain.

The segmentation of a 3D MR image of the head provides a delineation of all the voxels that make up a particular anatomical region. From this, a model of the shape and the volume of the region can be computed. A great deal of research has focused on the shape and volume changes that occur in the brain over periods of time. Giedd et al. (1999) examined a range of individuals from ages four to twenty and scanned them every two years with a maximum of five scans. The white and gray matter in these images was segmented using artificial neural networks and automatic registration to a template image in which these areas have been manually identified. The authors of this study were able to calculate the volumes of the white and gray matter to understand how they change over time. The white matter grew linearly over all regions of the brain, while the gray matter had a nonlinear growth that was specific to different areas of the brain. This discovery gave insight into how the different parts of the brain develop over time.

Resnick et al. (2003) examined a group of healthy adults aged fifty-nine to eighty-five years old and acquired MR head scans every two years for a total of three scans. The extracranial tissue, cerebellum, and brainstem regions were

manually removed from the images by a trained expert. Then the locations of the white matter, gray matter, and cerebrospinal fluid (CSF) were found using an adaptive Bayesian segmentation algorithm. From this, the volumes of the white matter, gray matter, and CSF in different areas of the brain were calculated. The results showed that there was a significant decline in the volume of white and gray matter and an increase in the CSF filled ventricle volumes for healthy elderly adults.

A related research thread examines how the brain is affected by disease over time. Thompson et al. (2003) compared sets of elderly individuals with and without Alzheimer's disease. Two MR scans of the head were acquired two years apart for each subject, and a series of image processing steps were applied to measure how the brain changes across the set of images over time. The images were registered to a standard brain imaging template and were each mapped into the same space. A Gaussian mixture model was then used to classify each image into white matter, gray matter and CSF. A three-dimensional cortical surface was extracted from this segmentation and further analysis was performed to study the changes in shape and tissue distribution over time. The analysis showed that cortical atrophy was significantly greater in brains affected by Alzheimer's disease compared to those of control subjects. This study also allows an understanding of where and when certain parts of the brain undergo atrophy, which can be used as a biological marker for the disease.

The work presented in (Grimson et al., 1997) reports on an algorithm for segmenting structures in whole head MR images and how that information can assist in surgical planning. Their "augmented reality" method enables images of the segmentation of a patient's MR image to be projected onto the patient's head so that surgeons can visualize and localize the blood vessels of critical functional

brain regions. The authors also describe methods for tracking the surgeons' instruments in relation to the segmented MR images. This exposes the surgeon to a three-dimensional reconstruction of the different segmented regions in relation to the location of their instruments within the brain. It shows the surgeons structures near the area in which they are working, which would otherwise be very difficult to see or locate. Gering et al. (2001) built on this surgical planning framework by creating a language to easily specify different anatomical structures and their locations in the scenes created from the imaging data. That work also contributed a procedure for fusing multi-modal imaging data to visualize what occurs during surgery and to plan an optimal trajectory through the brain. Additional information about the application of brain MR image segmentation to surgical planning is available in the literature (Nowinski, 2001; Kikinis et al., 1996; Jolesz et al., 2001).

1.2 Thesis Contributions and Overview

This thesis takes a machine learning approach that combines principal components analysis, affine registration, and nonlinear registration to automatic image segmentation, applied to the segmentation of 3D MR images of the brain. In particular, we develop a novel method that

1. efficiently learns a PCA atlas subspace from a training ensemble of manually segmented intensity and label image data,

2. identifies the particular PCA atlas within the learned subspace that is optimally customized to segmenting an input subject image of interest, and

3. nonrigidly registers this PCA atlas to the unlabeled subject image in order to complete the segmentation.

Furthermore, we apply our method to a significant clinical dataset of 3D MR images of the brain and compare it against competing methods using a variety of quantitative metrics, demonstrating its superior performance.

The remainder of this thesis is organized as follows: Chapter 2 reviews relevant prior work in general segmentation methods, methods specific the brain image segmentation, and reviews algorithms for image registration. Chapter 3 describes how we apply PCA to learn the atlas subspace from an intensity and label training ensemble. Chapter 4 describes how, through projection, we can compute a PCA atlas that is optimally adapted to segmenting the unlabeled subject image. Chapter 5 gives an overview of the training and segmentation algorithms. Chapter 6 presents the application of our method to clinical MR brain image data and reports the results of our empirical evaluation. In Chapter 7 we discuss various aspects of the method presented in this thesis and future work that can be pursued to improve our algorithm.

CHAPTER 2

Previous Work

The literature on image segmentation is enormous. Most algorithms model how pixels within a group are related to each other and then find segments that best fit this model across and within segments. The following is a description of some of the many methods. These algorithms are designed for general use in image processing applications. This differs from the situation of identifying anatomical regions in MR images in that often the voxel-based information is not enough in itself to distinguish the different parts of the brain. In those cases, additional information, such as labeled training data, is needed to find an accurate segmentation.

k-means is a clustering technique that segments or groups data into k different mutually exclusive parts. Over all of the segments, it minimizes the following within-segment sum-of-squares error:

$$e = \sum_{i=1}^{k} \sum_{j \in S_i} |\mathbf{x}_j - \boldsymbol{\mu}_j|^2, \tag{2.1}$$

where S_i represents one of the k subsets, $\mathbf{x}_j$ is a vector representing a piece of data in the jth segment, and $\boldsymbol{\mu}_j$ represents the centroid of the jth group. This approach is similar to estimating a mixture of Gaussians model of the data. Recent work on the algorithm is discussed in (Arthur and Vassilvitskii, 2007).

Mean shift is another clustering algorithm that groups data together based on

a non-parametric density estimation of the data. The following formula estimates the density at a particular point $\mathbf{x}$:

$$\mathbf{d}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} K(\mathbf{x} - \mathbf{x}_i), \tag{2.2}$$

where K is usually a Gaussian kernel and n is the total number of data points. The points in the data will be grouped together based on the peak or mode in the density to which they are closest. This will usually produce an over-segmentation of the image, and some other criteria are used to combine the smaller segments. A discussion of the mean-shift algorithm can be found in (Comaniciu and Meer, 2002).

Another widely used method is that of active contour models, or "snakes" (Kass et al., 1988). Classical snakes represent 2D parametric contours in the image plane. The geometry of the model, usually represented as splines, is constrained by physically derived elastic forces—internal forces that govern the smoothness of the model and external forces derived from the image. Formally, snakes are parametric contours $\mathbf{v}(s) = [x(s), y(s)]^{\mathrm{T}}$, where x and y denote the coordinate functions of the contour parameterized by $s \in [0, 1]$, that automatically evolve to minimize the functional

$$\mathcal{S}(\mathbf{v}) = \int_0^1 w_1(s)\left|\frac{\partial \mathbf{v}}{\partial s}\right|^2 + w_2(s)\left|\frac{\partial^2 \mathbf{v}}{\partial s^2}\right|^2 ds + \int_0^1 P(\mathbf{v}(s))ds, \tag{2.3}$$

where $w_1(s)$ determines the tension and $w_2(s)$ determines the rigidity of the snake. The scalar potential $P(x, y)$ is an image-derived function whose minima will correspond to features of interest in the image. For image intensity edges, this potential could correspond to the gradient of the image intensity convolved with a Gaussian. Snakes are a particularly popular instance of Deformable Models

that have been formulated in an arbitrary number of dimensions and have been applied to numerous segmentation tasks including medical imaging. A survey of many of their medical imaging applications can be found in (McInerney and Terzopoulos, 1996).

The region competition method presented by Zhu et al. (1995) uses a probabilistic model to segment an image into M mutually exclusive regions. It assumes that each region is generated from the same distribution, but with a different set of parameters α_i. This is used to create an energy function for the full segmentation. The algorithm operates by initially seeding N regions throughout the image. It then picks a boundary between the regions, and with that boundary fixed it optimizes the parameters of each region. Then, the boundary is moved to minimize the energy with all the α_i parameters fixed. This is iterated until convergence. The regions are then merged if their merger decrease the energy function. Both the boundary optimization and region merging procedures are repeated until the energy function cannot decrease further. This algorithm allows segmentation when the number of regions is unknown and enables regions to compete over boundaries in the images that they would model better. This approach was further improved by allowing each region to be modeled by a set of differently-parameterized distributions. Data Driven Markov Chain Monte Carlo is required to perform inference in this more complicated setting. The approach is described in (Tu and Zhu, 2002).

The normalized cut method treats an image as a graph and segmentation as a graph partitioning problem (Shi and Malik, 2000). A 2D image is treated as a weighted undirected graph $G = (V, E)$, where the pixels represent the vertices and there is an edge between every pair of vertices, i and j, and the weight $W(i, j)$ between the two pixels being a measure of their similarity. The goal is to split

the image into two groups, A and B, by removing a set of edges connecting the two. The weight of this cut is formalized by

$$\text{cut}(A, B) = \sum_{a \in A, b \in B} W(a, b), \tag{2.4}$$

where a and b are vertices in the segments A and B respectively. Minimizing this cut will give a partition that separates vertices that are least similar. The problem with this measure is that an optimal solution could be such that one of the segments contains only a single vertex. The normalized cut approach weights this measure as

$$NC(A, B) = \frac{\text{cut}(A, B)}{\text{assoc}(A, V)} + \frac{\text{cut}(A, B)}{\text{assoc}(A, V)}, \tag{2.5}$$

where $\text{assoc}(A, V)$ is the number of edges connecting the vertices in A to all the vertices V in the graph. This prevents partitioning into one large and one small segment, as that would not minimize $NC(A, B)$. The minimization problem can be solved efficiently in the form of a generalized eigenvalue problem. Each of the segments can be further split into two pieces until some maximum number of desired segments is obtained.

2.1 Brain Image Segmentation

There has been a great deal of work on the segmentation of medical images of the brain. Most approaches rely on training data to learn the parameters of a model of the different structures in the brain. This is necessary because of the complex protocols used to specify where anatomical structures of interest appear in the brain images and because in many instances different areas do not have

a natural, easily recognizable boundary. The methods used to segment normal structures in the brain typically differ from those used to segment tumors and abnormalities in the brain because the former assume that the model created from the training data provides a good representation of the brain's anatomy in general and they do not rely on features that are easily differentiated from normal brain appearance and structure.

The methods used for segmenting normal brain images can roughly be categorized into voting methods, atlas based methods, and machine learning classification methods.

Arno et al. (2005) registered every image in a set of manually labeled training image data to a subject image. The labels are transferred to the subject image from each of the images in the labeled training set and the most frequent label at each voxel is used as the final labeling. This process can be computationally expensive if the set of labeled images is very large because it needs to be performed for every subject image. Aljabar et al. (2007) use a similar approach, but they use a similarity criterion to select a subset of images from the training set that are similar to the subject image. This method relies heavily on the quality of the algorithm used to register the training data to the subject image. Compared with the other approaches, it retains the training data set in its entirety instead of creating a compact representation.

The work in (Fischl et al., 2002) uses a probabilistic atlas to segment a novel subject image. It poses the problem in a Bayesian setting where the objective is to maximize the probability of a labeling and transformation of a subject image into the atlas space. This method allows the incorporation of prior information such as the spatial structure of the different anatomical areas and restrictions on the possible transformations of the subject image into the atlas space. It also

incorporates Markov Random Fields (MRFs) to model the spatial relationship between the voxels of different anatomical parts.

Babalola et al. (2008) used 3D volumetric Active Appearance Models (AAM) to segment images of the brain. They start with a global AAM to obtain the location of the different structures in the brain and then refine these locations with AAMs trained specifically for each area. They then use linear regression functions at each voxel in the image to compute the probability that it lies within the domain of a certain anatomical area. The AAMs that they use rely on a grid of corresponding points in the data that are transformed to produce different shapes and a set of shape-free volumetric intensities that are transformed to the grid shape for optimizing the locations of the AAM. This differs from (Fischl et al., 2002) in that the structural information is learned for each specific structure or label as a whole instead of only the relationship between a voxel and its neighborhood found in MRFs.

The method described by Tu et al. (2008) uses a statistical hybrid discriminative / generative method to segment images. The discriminative component learns from thousands of features in the images to create an appearance model that can classify a voxel by assigning a particular label using local information from the image in a probabilistic boosting tree (PBT) framework. The generative component uses PCA to create a global shape model of the different shapes. The components are used together to create an energy function that is minimized to find a final segmentation.

The method presented in this thesis relates to (Fischl et al., 2002) in that the goal is to create an atlas that is tuned to the subject image, but we assume that once such an atlas is found, an additional high performance nonrigid registration will be performed to transform the subject image into the atlas space. PCA is

used to create a linear subspace representing the training data. A subject image is registered to the mean training image and then orthogonally projected into the PCA atlas subspace to find the atlas image.

2.2 Image Registration

Image registration is the process of bringing a subject image into correspondence with a target image. It involves finding a transformation of the subject image that transforms its voxel locations such that they align as closely as possible to the voxels in the target image with respect to some measure of the similarity between corresponding voxels in the two images. Registration is frequently used to bring different modalities of imaging data into correspondence. Examples are the registration of computed tomography (CT) and functional magnetic resonance (fMR) images into the same space as MR images in order to gain more insight from the different imaging modalities.

The algorithm developed in this thesis utilizes image registration in three different places. The first is in registering all of the training data together. The second is to register the subject image to the mean intensity training image. The third is in registering the subject image to the PCA atlas.

A registration algorithm is composed of four different parts. First, a metric is required that evaluates the quality of the registration. It measures the similarity of the two images after the subject image has been transformed into the target image space. Depending on the application, this measure may need to be robust to the similarity of two images that are of two different modalities.

Second, a transformation model is needed to specify what deformations are permissible in the transformation of the subject image. It can be as simple

as a global rigid-body transformation of the subject image to the much more complicated situation of a nonrigid transformation, or deformation. Nonrigid transformations are usually constrained to prevent deformations that fold onto themselves or create other drastic changes. The deformation field specifies the displacement for every voxel in the image. A common way of regulating the deformations is to constrain the Jacobian determinant to be positive at every voxel.

The third critical component is the optimization scheme used to find a transformation that optimizes the metric. For example, this could be a randomized search scheme for an optimal value, a gradient descent method, or the Levenberg-Marquardt optimization method.

Fourth, when an image has been transformed, an interpolation method is required to resample the subject image at locations that are displaced from the original voxel grid. For example, the interpolation method can be a nearest neighbor method that looks for the voxel that is closest to the point being sampled or something more complex, such as trilinear interpolation among the 8 voxels that surround the displaced voxel.

2.2.1 Rigid and Affine Registration

A critical aspect of a registration algorithm is the transformation model that it uses. This specifies what possible transformations or deformations a subject image can undergo. This also dictates the method used to optimize the registration and how many parameters are in the model. Rigid registration uses a three-dimensional transformation model that takes into account translation and rotation. It is represented using six parameters, three for translation and three for rotation. An affine transformation model allows translation, rotation, stretching,

and shearing. This is represented using a twelve-parameter model. Both rigid and affine registration are linear and have a matrix representation.

In this thesis, we use an algorithm developed by the Oxford Centre for Functional MRI of the Brain (FMRIB), called FMRIB's linear image registration tool (FLIRT). It is described in (Jenkinson et al., 2002). The algorithm performs affine registration between two images of the same or different modalities. It uses a correlation ratio as the default metric, which has the following form:

$$\frac{1}{\sigma_S^2} \sum_b \frac{n_b}{N} \sigma_{S_b}^2. \tag{2.6}$$

Let the sets S and T be intensities from the voxels in the subject and target images respectively. σ_S^2 is the variance of the set S. The intensities in the set T are binned into B different bins containing n_b elements in bin b. $N = \sum_b n_b$ and S_b is the set of intensities at the voxel locations in the subject image that correspond to the voxel locations that fell into bin b in the target image. This metric is suitable for intramodal and intermodal images. The implementation of the algorithm is capable of using mutual information, normalized cross correlation, normalized mutual information, and least squared differences as the metric. Its default is to use trilinear interpolation. It optimizes in a multiresolution framework by combining local and global searching through the parameter space and then refining possible solutions in an approach similar to simulated annealing.

2.2.2 Nonlinear Registration

Nonlinear registration algorithms provide many more degrees of freedom to deform a subject image into the target image space. In most cases constraints need to be imposed in order to avoid unrealistic deformations. Klein et al. (2009)

compare fourteen nonlinear registration algorithms. The results of that study show that the Automated Registration Tool (ART), developed by Ardekani et al. (2005), is one of the top performing algorithms. In this thesis we use the ART algorithm for nonlinear registration between two images. For a given subject image the algorithm produces a deformation field that contains a displacement vector for each voxel. This means that for an image of dimension $256 \times 124 \times 256$, there are three parameters for each voxel making 24,379,392 parameters that need to be optimized. The metric used in the algorithm is the normalized cross-correlation

$$\frac{\mathbf{s}^{\mathrm{T}}\mathbf{H}\mathbf{t}}{\sqrt{\mathbf{s}^{\mathrm{T}}\mathbf{H}\mathbf{t}}}, \tag{2.7}$$

where $\mathbf{H}$ is a centering matrix, meaning that it is symmetric, idempotent ($\mathbf{H} = \mathbf{H}^2$), and removes the mean of what it multiplies, and where $\mathbf{s}$ and $\mathbf{t}$ are vectors of voxel intensities in a neighborhood surrounding a particular voxel of interest in the subject and target images, respectively.

2.2.3 Groupwise Registration

A traditional method used to register images into a common space is to select a registration target and register all images to it. This target image can be derived from a specific brain, such as the Talairach and Tournoux atlas (Talairach and Tournoux, 1988), or from an average of brains registered to a common space (e.g., MNI-305 or ICBM152). The target image may also be a member of the set needing to be mutually registered. The problem with these methods when applied to PCA is that the mean image should represent a point that requires an approximately equal amount of deformation to conform to each of the training images. This is desirable so that each of the training images are generated using coefficients that are an equal distance from the null coefficient vector, which

would generate the mean image. If the mean is preselected without respect to the training data, then there is no guarantee that this will be true. Also, the target image chosen may not be able to account for the anatomical variety in the set of images and it may be difficult to register images into that space. An alternative is to register the set of images together in a groupwise fashion (Witkin et al., 1987), which avoids relying on a specific target image.

Studholme and Cardenas (2004) introduced a method to register images in a groupwise manner by optimizing the joint probability distribution of all the images together. Density estimation is used to calculate the deformations for all the images where the sum of all the deformation needs to be zero. Bhatia et al. (2004) performed registration groupwise by using a normalized mutual information metric that looks at pairwise comparisons combined with 1D cubic B-splines. The methods described by Twining et al. (2005) relies on information theoretic methods to create a cost function taking into account similarities and deformation to register the images.

The method used in this thesis takes advantage of the Insight Toolkit (ITK) implementation of the algorithm described in (Balci et al., 2007), which is an extension of the work in (Zollei, 2006). It minimizes a voxelwise entropy measure across all images. Given a set of images $\{I_1, ..., I_N\}$ a common reference frame X_R can be defined so that a set of transforms satisfy

$$\{T_n : x_n = T_n(x_R), n = 1, ..., N\}, \tag{2.8}$$

where $x_R \in X_R$ is a point in the reference frame and $x_n \in X_N$ is a point in the space of a particular image. The transforms map locations in the reference frame to their corresponding locations in the images being registered. The entropy

measure across the images is

$$f = -\sum_{v=1}^{V} \frac{1}{N} \sum_{i=1}^{N} \log \frac{1}{N} \sum_{j=1}^{N} G_\sigma \left(I_i(T_i(x_v)) - I_j(T_j(x_v)) \right), \tag{2.9}$$

where G_σ is a Gaussian kernel. The voxel locations through the set of images are assumed to be independent and identically distributed samples and can thus be evaluated as the product of the individual voxel locations across the images. This entropy measure will become small as the difference across the voxels is reduced and it is minimized to create a groupwise registration of the images. To improve efficiency, this measure is computed over a random sample of the voxel locations. It uses a gradient descent line search optimizer to optimize the registration and trilinear or nearest neighborhood interpolation. The entropy measure is used first to register the images groupwise using an affine transformation. This is followed by a nonrigid registration using B-splines to further deform the images into groupwise correspondence.

CHAPTER 3

Learning the PCA Atlas Subspace From Training Data

Learning a PCA atlas subspace from manually segmented and labeled training data involves the application of principal components analysis.

3.1 Principal Components Analysis (PCA)

Principal Components Analysis (PCA) is a method that computes the mutually orthogonal directions of maximum variance in a collection of d-dimensional data. These d directions form an orthogonal basis that spans the data space. PCA is usually used for dimensionality reduction, by projecting the data on a subset $\tilde{d} \leq d$ of the basis vectors. The subset of vectors are termed the *principal components*, and they account for a large percentage of the variance. They thus represent each d-dimensional data point as a linear combination of the $\tilde{d}$ principal components and, hence, by the $\tilde{d}$-dimensional vector of coefficients of the linear combination.

Among its many uses, PCA can be used to compress data, to classify data, to visualize data trends in a lower-dimensional space, etc. Its application in the context of the image segmentation method proposed in this thesis is to capture within a compact representation the intensity and shape variation evident in the training data.

To apply PCA to image analysis, each of the n images in the training data set is vectorized, yielding the set of vectors $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$. Then the mean vector $\boldsymbol{\mu}$ of this set is subtracted from each vector in order to center the data at the origin, thus yielding the $d \times n$ data matrix

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 - \boldsymbol{\mu} & \mathbf{x}_2 - \boldsymbol{\mu} & ... & \mathbf{x}_n - \boldsymbol{\mu} \end{bmatrix} \tag{3.1}$$

The covariance matrix of the data is the $d \times d$ matrix $\mathbf{X}\mathbf{X}^{\mathrm{T}}$. Then, the eigenvalues and eigenvectors of the covariance matrix are computed. The eigenvectors are the principal directions. The eigenvalues give a measure of the variance of the data in the corresponding principal directions.

Because the dimension d of each vector $\mathbf{x}$ may be in the order of ten million elements or larger when processing high-resolution 3D image data, the covariance matrix could be huge. Computing the eigenvectors and eigenvalues of such a matrix is prohibitive, so rather than doing so from $\mathbf{X}\mathbf{X}^{\mathrm{T}}$, they may be computed from the (presumably much smaller) $n \times n$ matrix $\mathbf{X}^{\mathrm{T}}\mathbf{X}$. The eigenvectors will be the same for both matrices, while the eigenvalues for $\mathbf{X}\mathbf{X}^{\mathrm{T}}$ are calculated by multiplying the eigenvectors of $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ by $\mathbf{X}$. This can be shown as follows:

$$\begin{aligned} \mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{e} &= \lambda\mathbf{e}, \\ \mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{e} &= \lambda\mathbf{X}\mathbf{e}, \\ \mathbf{X}\mathbf{X}^{\mathrm{T}}(\mathbf{X}\mathbf{e}) &= \lambda(\mathbf{X}\mathbf{e}), \end{aligned} \tag{3.2}$$

where $\mathbf{e}$ represents an eigenvector and λ represents an eigenvalue. This method and its application is discussed in (Cootes et al., 1995) and (Trucco and Verri, 1998).

It is also assumed that, for natural image data, $\mathbf{X}$ will have rank n. Thus,

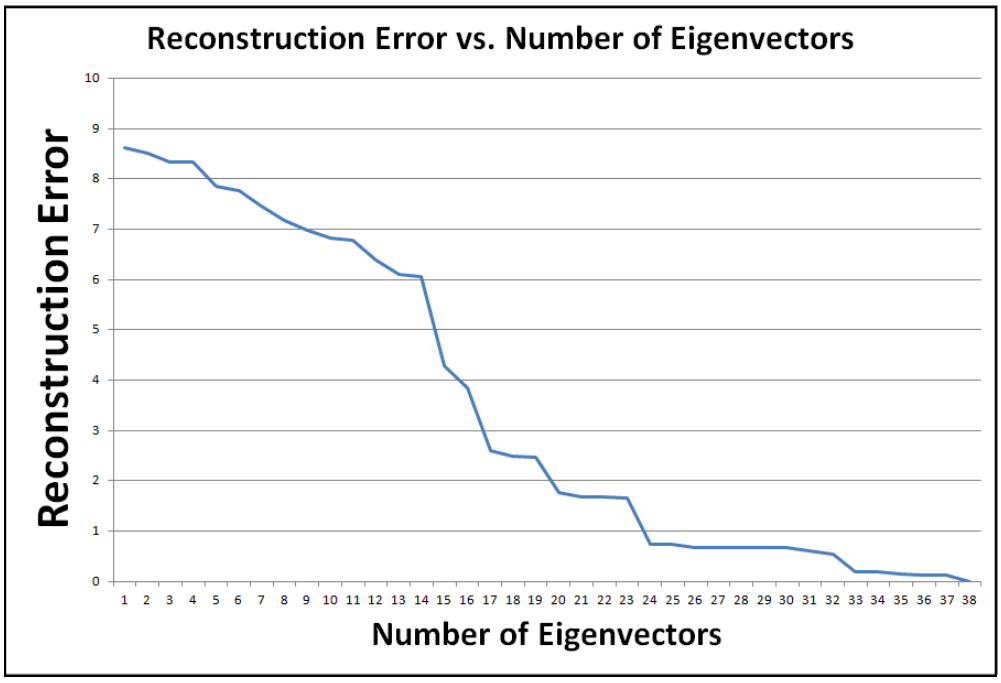


Figure 3.1: PCA Error Graph

$\mathbf{X}\mathbf{X}^{\mathrm{T}}$ will also be rank n and will have no zero eigenvalues. We can assume that $\mathbf{X}$ is full rank because we are working with a set of real world image data where the images do not have a linear relationship with each other. Each image used to create $\mathbf{X}$ is represented by the mean vector plus a vector of the n coefficients of the linear combination of each of the eigenvectors:

$$\mathbf{x} \approx \boldsymbol{\mu} + \mathbf{X} \sum_{i=1}^{n} c_i \mathbf{e}_i, \tag{3.3}$$

where c_i is the coefficient weighting the eigenvector $\mathbf{e}_i$. If most of the variance in the data is accounted for by just a few principal eigenvectors associated with the largest eigenvalues, then each image may be well represented by a vector that is significantly smaller than n, though n will probably be much smaller than the dimension of the images d. The representation of the data by a subset of the basis will not reconstruct the data exactly, and there is a trade-off between the

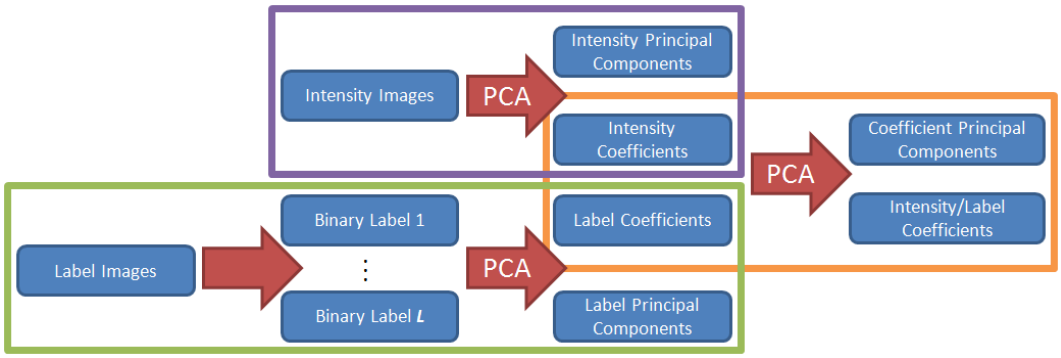


Figure 3.2: PCA Processing of Label and Intensity Images

compactness of the representation and the error that it incurs. Figure 3.1 shows this relationship. It displays the sum of squared differences error between a training image and its PCA representation as the number of eigenvectors (principal components) is increased. The eigenvectors are ordered in decreasing order of their corresponding eigenvalues. As the number of eigenvectors increases, those that are added later account for less of the variance than the earlier ones.

3.2 Efficiently Computing the PCA Atlas Subspace

The training data from which we learn the PCA atlas subspace comprises an intensity image and a corresponding label image for each manually segmented brain scan. We would like a single vector of coefficients to represent each intensity/label image pair in the PCA atlas subspace. The subspace is computed efficiently by performing multiple PCAs. First, a PCA is performed on the intensity images to create an intensity subspace while a separate PCA is performed on the label images to create a label subspace. Then a final PCA is performed in order to combine the subspaces into the desired PCA atlas subspace. Figure 3.2 presents an overview of how the training data is processed to create the components for the intensity images, label images, and the final combined intensity/label coefficients

of the PCA atlas subspace.

For the purposes of the following mathematical development, let us assume that there are N training images.

3.2.1 The Intensity Subspace

The intensity space is created using the N intensity images in the training image set. Each three-dimensional training intensity image is vectorized as follows:

$$\mathbf{a}_n = \begin{bmatrix} a_1 & a_2 & ... & a_K \end{bmatrix}^{\mathrm{T}}, \tag{3.4}$$

where K is the number of voxels in the image and voxel a_k takes a value in $[0, 1]$. The PCA method detailed above is used to compute B eigenvectors where $B \leq N$ that form a basis of the intensity space. Thus, each image in the training set is represented by a coefficient vector $\mathbf{c}_n$, where $n \in \{1, ..., N\}$. The length B of $\mathbf{c}_n$ can be chosen to be smaller than N and still yield a good approximation of the training data depending on the variance associated with each of the eigenvectors in the chosen subset. Length B can be adjusted to reduce computation time at the cost of precision. A training intensity image $\mathbf{a}_n$ is reconstructed by

$$\mathbf{a}_n = \boldsymbol{\mu}_a + \mathbf{E}_a \mathbf{c}_n, \tag{3.5}$$

where $\boldsymbol{\mu}_a$ is the mean of the intensity training data and $\mathbf{E}_a$ is a matrix whose B columns are the eigenvectors found from performing PCA on the intensity training data.

3.2.2 The Label Subspace

The label space is created using the N training label images that correspond to the N intensity training images. Each training label image is vectorized as follows:

$$\mathbf{l}_n = \begin{bmatrix} l_1 & l_2 & ... & l_K \end{bmatrix}^{\mathrm{T}}, \tag{3.6}$$

where K is the number of voxels in the image, and $l_k \in \{g_1, g_2, ..., g_P\}$ is the label value at the k-th voxel. There are P different label identifiers, each denoted by g_p.

Binary images are computed for each label, yielding N binary images for each of the P labels. Then PCA is used to create a separate basis for each label using its respective N binary images. The binary image $\mathbf{b}_n^p$ for a particular training image n and a particular label identification number g_p is generated as

$$\mathbf{b}_n^p = \boldsymbol{\mu}_l^p + \mathbf{E}_l^p \mathbf{h}_n^p, \tag{3.7}$$

where $\boldsymbol{\mu}_l^p$ is the mean binary image, $\mathbf{E}_l^p$ is the eigenvector matrix, and $\mathbf{h}_n^p$ are the coefficients for training image n, all with respect to label identification number g_p. $B \times P$ coefficients are needed to generate all P binary images. These binary images are combined to create a complete label image that has a corresponding intensity image. B being the number of eigenvectors preserved. These coefficients are combined into a single vector

$$\mathbf{h}_n = \begin{bmatrix} \mathbf{h}_n^{1\,\mathrm{T}} & \mathbf{h}_n^{2\,\mathrm{T}} & ... & \mathbf{h}_n^{P\,\mathrm{T}} \end{bmatrix}^{\mathrm{T}}. \tag{3.8}$$

3.2.3 Combining the Intensity and Label Spaces

The coefficients that result from orthogonally projecting a training intensity image onto the intensity subspace and the coefficients that result from projecting a training label image onto the label subspace are combined into one vector

$$\mathbf{s}_n = \begin{bmatrix} \mathbf{c}_n{}^{\mathrm{T}} & \mathbf{h}_n{}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}. \tag{3.9}$$

Then, a PCA is performed on this set of N combined coefficients, one for each training image. This will generate a basis for the combined coefficients, and a coefficient vector of size B in this space will generate the coefficients for both the intensity and label spaces.

3.3 Implementation Details

In practice, almost sixty percent of the voxels in the intensity images used as training images are not located within the brain and are not important to the segmentation process. Therefore a mask is used to suppress all irrelevant voxels in the training intensity images. When an image is vectorized, only the unmasked voxels are used. This reduces the computational burden substantially. When a complete image needs to be generated, the missing voxels are constructed using information from the pre-computed mask. The same issue occurs with the binary images used to create the label PCA space. In this case even more of the voxels in the image are not relevant to the construction of the basis and the use of a mask is crucial.

When the binary images are created, the value 1 represents "on", while 0 represents "off". When a binary image is generated using the PCA basis, some

voxels will have values between 0 and 1. Thus, when the P binary images are generated for creating a full label image, the value at each voxel location across the P images is compared and the largest value with the corresponding label identification number g_p is used to label that voxel. Uniform random noise in the range $[0, 10^{-10}]$ is also added to every voxel in the image to differentiate between two labels that may have the same value between 0 and 1.

3.4 PCA Atlas Subspace GUI

A graphical user interface (GUI) was developed using MATLAB to explore and visualize the PCA atlas subspace. It works with the data described in Section 6.1. It allows a user to interactively synthesize three-dimensional intensity images along and their associated labels. These PCA atlas images are generated using combined label-intensity coefficients and the PCA basis described earlier in this section. The GUI allows a user to vary the coefficients corresponding to the ten eigenvectors with the largest eigenvalues using sliders. Figure 3.3 shows all the coefficients at zero, which yields the mean PCA atlas. Figure 3.4 shows four examples of different coefficients that were selected to synthesize different intensity and label images. Forty training images were used to create the PCA atlas subspace in this example, but only the coefficients associated with the 10 eigenvectors with the largest eigenvalues are retained in the GUI, while the 30 less significant ones are set to zero.

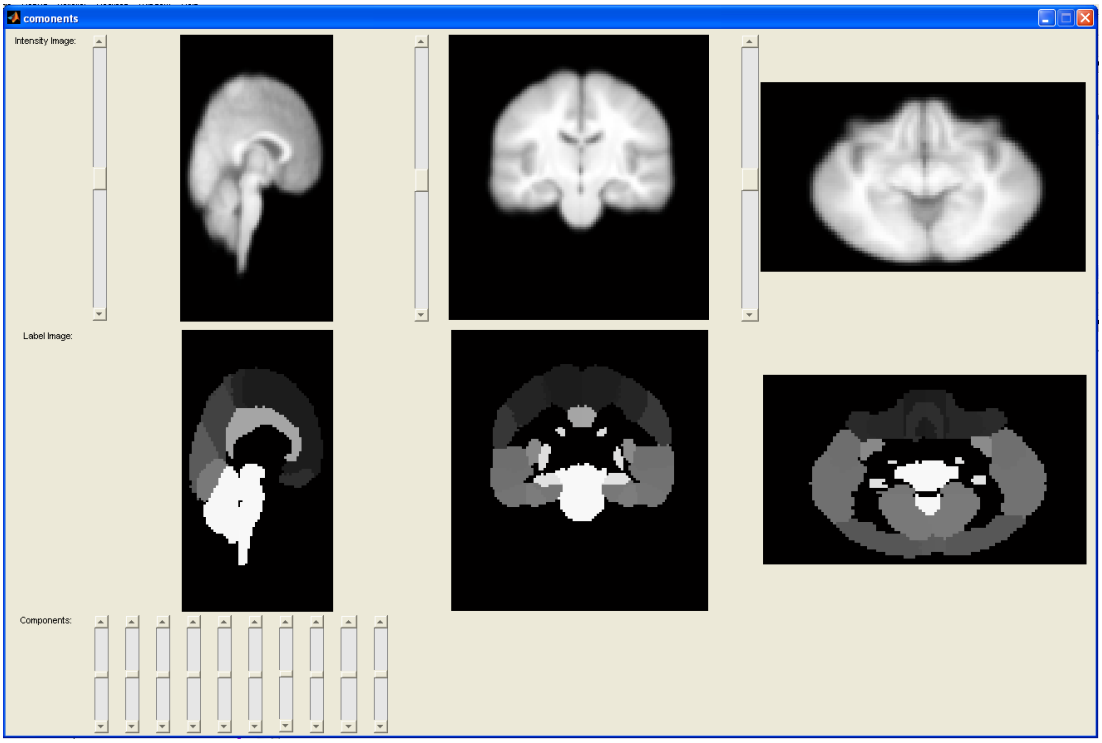
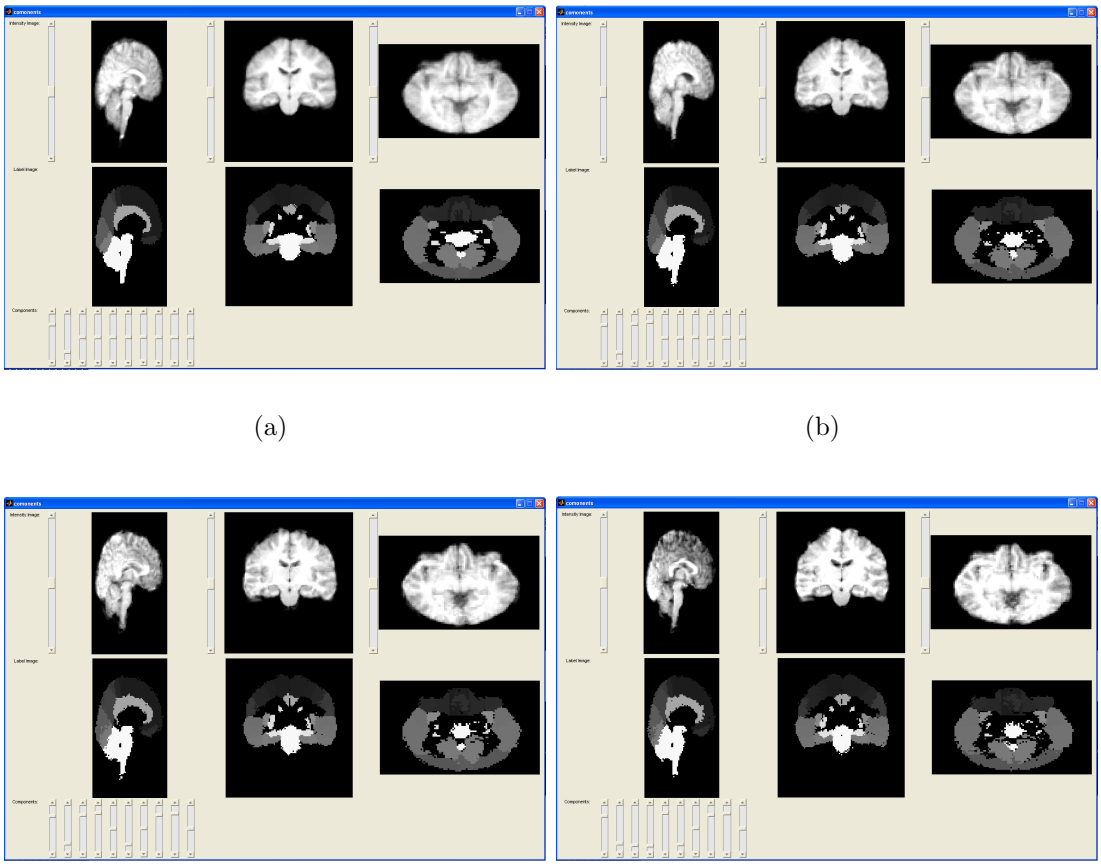


Figure 3.3: PCA Mean Image and Labels



(a) (b)

(c) (d)

Figure 3.4: PCA Intensity and Label Examples

CHAPTER 4

Computing the Optimal PCA Atlas

To obtain an atlas with which to segment an unlabeled subject image, the subject image is orthogonally projected into the PCA atlas subspace created from the intensity and label image training data, yielding the coefficients that represent the PCA atlas that is closest to the subject image. The label and intensity images of this optimal PCA atlas are then synthesized from those coefficients. This step is performed after the subject image has been affinely registered to the mean intensity training image because the subject image may not be in the same space, meaning there could be some discrepency in rotation, scaling, or translation.

4.1 Orthogonal Projection Into the Subspace

Subject image $\mathbf{a}_s$ is orthogonally projected on the intensity subspace by subtracting the mean intensity and multiplying the result by the inverse eigenvector matrix whose columns are the eigenvectors from the PCA. This gives a set of coefficients

$$\mathbf{c}_{as} = \mathbf{E}_a{}^{+}(\mathbf{a}_s - \boldsymbol{\mu}_a). \tag{4.1}$$

Where the + superscript denotes the matrix pseudoinverse. This is required because the matrix may not be square. These coefficients are then projected on the label/intensity coefficient subspace.

The label/intensity coefficient subspace is represented by an eigenvector matrix that is split into two parts

$$\mathbf{E}_c = \begin{bmatrix} \mathbf{E}_{ac} \\ \mathbf{E}_{lc} \end{bmatrix}. \tag{4.2}$$

$\mathbf{E}_{ac}$ generates the intensity coefficients and $\mathbf{E}_{lc}$ generates the label coefficients when $\mathbf{E}_c$ is multiplied by a set of label/intensity coefficients. The mean of the intensity coefficients is subtracted from the intensity coefficients of the subject image, $\mathbf{c}_{as}$, and the result multiplied by the inverse of the $\mathbf{E}_{ac}$ matrix:

$$\mathbf{c}_s = {\mathbf{E}_{ac}}^{+}(\mathbf{c}_{as} - \boldsymbol{\mu}_{ac}). \tag{4.3}$$

Vector $\mathbf{c}_s$ is used to generate both the intensity and label PCA atlas images. The PCA atlas intensity image is generated by

$$\mathbf{a}_p = \mathbf{E}_a \mathbf{E}_{ac} \mathbf{c}_s. \tag{4.4}$$

The PCA atlas label image is generated by

$$\mathbf{l}_p = \mathbf{E}_l \mathbf{E}_{lc} \mathbf{c}_s. \tag{4.5}$$

CHAPTER 5

Image Segmentation Using the PCA Atlas

To segment a subject image in an efficient manner, our algorithm combines affine registration, nonlinear registration, and a PCA intensity/label model learned from a data set with intensity images and their corresponding labels. All experiments reported in this thesis use brain MR image data. However, the algorithm can be generalized to any modality if a set of intensity and corresponding label data is available for training.

5.1 Training the Algorithm

Algorithm 1 outlines the steps used to train the PCA-registration segmentation algorithm. Our implementation uses the ITK implementation of the minimum entropy registration algorithm described in (Balci et al., 2007) (see Section 2.2.3) to find an affine registration of the training intensity data groupwise for Step 1. It applies the computed transformations to the label training images. The remainder of the algorithm goes through the steps to create a PCA subspace of the registered training intensity and label images.

Algorithm 1: Training Algorithm

Data: Intensity and label training images
Result: Intensity, label, and intensity/label coefficient PCA subspaces.

1 Affinely register the intensity training images groupwise;
2 Apply transformations from the registration to the respective label data;
3 Compute the PCA basis for the registered intensity training data;
4 Compute the binary images from the registered training labels;
5 Compute the PCA basis for the binary images;
6 Project the binary label and intensity training images onto their respective bases;
7 Compute the PCA basis of the coefficients of the training data from the projection;

Algorithm 2: Segmenting a Subject Image

Data: Subject intensity image. Intensity, label, and coefficient PCA bases.
Result: Labels for the subject image.

1 Affinely register the subject image to the mean intensity image of the training data;
2 Orthogonally project the subject image into the PCA subspace to find label and intensity coefficients;
3 Use the coefficients to generate the intensity and label images of the PCA atlas;
4 Register the subject image again using a nonlinear registration algorithm to the PCA atlas;
5 Reverse transform the PCA atlas's label image to the original subject image space for the final labeling;

5.2 Segmenting a Subject Image

Algorithm 2 outlines the steps used to segment a subject image. It assumes that the results from the training algorithm have been computed. In our implementation, Step 1 is computed using FLIRT. An affine registration algorithm is used in this step because the PCA subspace was created using images that were co-registered as a group using an affine registration algorithm. Thus, this step serves to project the subject image into the space where the PCA bases were created.

Step 4 uses the ART algorithm for the nonlinear registration (see Section 2.2.2). For brain MR data, the subject image would need to be preprocessed so that it contains only brain voxels. A skull-stripping algorithm, such as brain surface extractor (BSE) (Shattuck et al., 2001) or brain extraction tool (BET) (Smith, 2002) is needed to remove voxels that represent extraneous tissues around the brain, such as cerebrospinal fluid, skull, and skin. Then inhomogeneity correction, such as the BFC algorithm, is required to remove changes in brightness of the voxels caused by distortions from the MRI scanner, patient anatomy and position, and static field inhomogeneity.

CHAPTER 6

Experiments and Results

We evaluated our method against two other algorithms using a set of 40 manually delineated MR images. The data included manual labelings of 56 brain structures, which were used as the gold standard. In validating our PCA-registration algorithm, we used a leave-one-out strategy to test each of the 40 images in the training set by selecting one of them to serve as a subject image while using the remaining thirty-nine images as training data.

6.1 Data

The training images used to create the PCA space were the 40 labeled MR images from the LONI Probabilistic Brain Atlas (LPBA40)[1] (Shattuck et al., 2008). The images consist of forty whole human head MR images. These images have been manually labeled by experts who delineated fifty-six different structures in each image. The images were then skull-stripped by using the information from the labels combined with morphological operations. The skull-stripped images were then inhomogeneity corrected using the Bias Field Corrector (BFC) developed by Shattuck et al. (2001). The skull-stripped, inhomogeneity-corrected images along with their labels were used to learn the PCA atlas subspace and also used as test data. The images are of dimension $256 \times 124 \times 256$ with a voxel resolution

[1] http://www.loni.ucla.edu/Atlases/LPBA40

of $0.86 \times 1.50 \times 0.86$ mm^3.

6.2 Competing Algorithms

The two other algorithms used for comparison were a voting algorithm and the LONI Brain Parser. The voting algorithm used the 40 images in the training set using a leave-one-out strategy. Of the forty images in the LPBA40 data set, one served as a subject image while the other 39 served as training data. This was repeated 40 times so that each image was employed as a subject image. Once a subject image was selected, each of the 39 other images was nonrigidly registered to it using the ART algorithm. The registration was performed using only the intensity information, as it is assumed that the subject image does not have any labels. The ART algorithm computed deformations for each of the registrations that are used to register all of the 39 training images to the subject image. Then the mode, or most common label across all of the registered training labels for a voxel in the subject image space, was chosen as its label classification. The label selection was implemented in MATLAB. This method is similar to that described by Arno et al. (2005), though their work focuses on how the number of labeled atlases affects the segmentation results.

The LONI Brain Parser[2] method uses a statistical hybrid generative / discriminative method to segment the images and it is described by Tu et al. (2008). It was trained on the 40 LPBA40 training images and then tested on the same 40 images; thus, it should have an advantage relative to our method when segmenting those images.

[2] http://www.loni.ucla.edu/Software/BrainParser

6.3 Metrics

We used overlap metrics as well as a distance metric in our experiments.

6.3.1 Overlap Metrics

The following overlap metrics that were used to validate the results of the segmentation are described in (Shattuck et al., 2009). They consist of Jaccard Similarity, Dice Coefficient, Sensitivity, and Specificity measures. Each metric is a combination of the true positive (TP), false positive (FP), true negative (TN), and false negative (FN) counts of the voxels in the label image resulting from the final segmentation computed by an algorithm, compared to the manually traced gold standard labels. The metrics are defined as follows:

$$\text{Jaccard Similarity} = \frac{|\text{TP}|}{|\text{FP}| + |\text{TP}| + |\text{FN}|} \tag{6.1}$$

$$\text{Dice Coefficient} = \frac{2|\text{TP}|}{|\text{FP}| + 2|\text{TP}| + |\text{FN}|} \tag{6.2}$$

$$\text{Sensitivity} = \frac{|\text{TP}|}{|\text{TP}| + |\text{FN}|} \tag{6.3}$$

$$\text{Specificity} = \frac{|\text{TN}|}{|\text{TN}| + |\text{FP}|}. \tag{6.4}$$

A score of 1.0 indicates that the two regions overlap exactly.

6.3.2 Distance Metric

The Hausdorff distance measure (Huttenlocher et al., 1993; Babalola et al., 2009) computes the distance between two sets of voxel locations in an image. In our work, this is the distance between the segmentation results and the gold standard voxel locations for a particular label value. Given two sets of voxels, A and B,

Method	Jaccard	Dice	Sensitivity	Specificity	Hausdorff Distance
PCA Atlas	0.5975±0.0196	0.7442±0.0167	0.7998±0.0157	0.9994±0.0000	9.4502±0.5696
Voting	0.6106±0.0444	0.7472±0.0432	0.7457±0.0457	0.9995±0.0001	10.9123±2.2904
LONI Brain Parser	0.5656±0.0410	0.7164±0.0384	0.6716±0.0441	0.9996±0.0001	18.5963±1.6975

Table 6.1: Metric Results

the minimum distance from any voxel location a in A to all voxel locations in the set B is represented as d_a^{AB}. The corresponding smallest distance from any voxel location b in B to the set A is represented as d_b^{BA}. The Haussdorff distance is the maximum of the maximum smallest distance in each direction over all voxel locations in each set, and it is expressed as

$$\max\{\max\{d_a^{AB}\}, \max\{d_b^{BA}\}\} \tag{6.5}$$

In practice, finding the minimum distance in each direction over all the possible voxels in the two sets A and B is very time consuming when the images and the label sets are large. To facilitate the computation, the border voxels of each set are used in the computation instead of the complete set. The border is found by eroding the set of voxels by a diamond shaped kernel and then taking the set difference as the set of border voxels.

6.4 Results Tables

Table 6.1 shows results from applying the three different segmentation algorithms: The PCA-registration method, the voting method, and the LONI Brain Parser. Each of the metrics was calculated for each of the 56 labels on each of the 40 test images. These results were then averaged across the different labels and then across the test images, yielding the results in the table (mean + standard deviation).

Tables 6.2, 6.3, and 6.4 show the results averaged over the data sets for our method (PCA Atlas), the voting method, and the LONI Brain Parser. The grey boxes in each of the tables indicate the particular method that achieved the best score for that measure.

Examining the averages across the forty test images, the LONI Brain Parser had a higher value for the Jaccard and Dice measures than the other two algorithms on the left inferior gyrus, the right angular gyrus, and the left precuneus. It also had a consistently higher value for the Specificity measure across all of the different anatomical regions that were segmented. Since the other overlap measures were lower for the Brain Parser, the fact that the Specificity was so high may be because it did not include enough of the boundary in each of the regions. It could also mean that it rejected too many voxels from the structure. The voting method had the overall highest scores for the Jaccard and Dice measures and, in particular, had better performance than the other two algorithms in all the metric categories for the left fusiform gyrus. In 14 of the 56 areas, the voting method achieved better scores than the other two algorithms in all but one metric measure.

Our method achieves a better overall result in the Sensitivity and Hausdorff distance measures. In 16 of the 56 areas it achieved better measures than the other two methods in all but one measure. It performed better on gyri that were located in the anterior, superior, and posterior of the brain. The voting method performed better with larger structures such as the cingulate gyrus and brainstem as well as smaller structures found in the brain such as the putamen and hippocampus. Our PCA method also seemed to cope better in regions with many adjacent gyri in close proximity, whereas the voting method was better with larger areas and areas with relatively sparse regions.

Region	Jaccard	Dice	Sensitivity	Specificity	Hausdorff Distance
L superior frontal gyrus	0.712792	0.831978	0.866102	0.998673	9.692282
R superior frontal gyrus	0.700557	0.823605	0.856137	0.998612	10.888745
L middle frontal gyrus	0.672906	0.803954	0.871948	0.998458	11.268219
R middle frontal gyrus	0.664498	0.797543	0.857354	0.998470	11.857572
L inferior frontal gyrus	0.609115	0.755713	0.829139	0.999227	10.684132
R inferior frontal gyrus	0.625044	0.767898	0.823930	0.999259	10.893739
L precentral gyrus	0.593943	0.743496	0.781037	0.999096	8.972878
R precentral gyrus	0.596230	0.745532	0.776929	0.999126	9.166451
L middle orbitofrontal gyrus	0.581004	0.732554	0.764327	0.999677	8.340211
R middle orbitofrontal gyrus	0.576053	0.729151	0.760620	0.999670	8.566678
L lateral orbitofrontal gyrus	0.503968	0.667401	0.706743	0.999735	9.545252
R lateral orbitofrontal gyrus	0.490486	0.653351	0.689254	0.999753	9.308912
L gyrus rectus	0.565781	0.720911	0.724615	0.999902	6.692414
R gyrus rectus	0.590600	0.740833	0.751003	0.999897	5.979427
L postcentral gyrus	0.540117	0.698488	0.739011	0.999114	10.875179
R postcentral gyrus	0.537880	0.698004	0.733088	0.999159	10.876567
L superior parietal gyrus	0.610466	0.757122	0.814690	0.999075	11.478200
R superior parietal gyrus	0.580074	0.732648	0.829965	0.998923	10.957222
L supramarginal gyrus	0.543592	0.701850	0.747353	0.999455	11.854969
R supramarginal gyrus	0.517858	0.678347	0.741617	0.999422	11.225909
L angular gyrus	0.512760	0.673397	0.725012	0.999219	13.400234
R angular gyrus	0.524947	0.685082	0.736968	0.999185	13.230522
L precuneus	0.560333	0.716709	0.761758	0.999611	8.998374
R precuneus	0.581279	0.733887	0.793133	0.999616	8.556240
L superior occipital gyrus	0.488755	0.653771	0.672003	0.999656	12.501387
R superior occipital gyrus	0.476805	0.641897	0.684086	0.999605	12.136082
L middle occipital gyrus	0.566697	0.720266	0.756372	0.999145	12.057395
R middle occipital gyrus	0.566473	0.720762	0.790711	0.999042	12.383315
L inferior occipital gyrus	0.552355	0.708562	0.713232	0.999643	9.806255
R inferior occipital gyrus	0.555188	0.711750	0.715881	0.999648	9.918910
L cuneus	0.525893	0.687175	0.707756	0.999768	9.324926
R cuneus	0.526709	0.686532	0.772635	0.999701	8.885530
L superior temporal gyrus	0.650053	0.786858	0.850857	0.999083	10.181602
R superior temporal gyrus	0.664863	0.798311	0.841000	0.999208	10.788142
L middle temporal gyrus	0.566209	0.721232	0.769410	0.999032	13.821012
R middle temporal gyrus	0.595898	0.745744	0.778516	0.999142	12.747502
L inferior temporal gyrus	0.560568	0.717173	0.759191	0.999197	12.269397
R inferior temporal gyrus	0.581699	0.734011	0.782656	0.999167	11.405524
L parahippocampal gyrus	0.610582	0.756944	0.787509	0.999795	7.412518
R parahippocampal gyrus	0.599976	0.748473	0.773667	0.999787	7.795251
L lingual gyrus	0.602026	0.747873	0.828893	0.999502	8.247752
R lingual gyrus	0.638689	0.777805	0.839301	0.999567	7.859232
L fusiform gyrus	0.611662	0.757277	0.784561	0.999628	9.932300
R fusiform gyrus	0.628098	0.769996	0.778077	0.999670	10.281410
L insular cortex	0.666046	0.799000	0.922064	0.999709	4.553200
R insular cortex	0.644132	0.782410	0.923967	0.999704	4.812814
L cingulate gyrus	0.616513	0.761667	0.796397	0.999594	8.230087
R cingulate gyrus	0.606571	0.754214	0.793898	0.999567	8.382270
L caudate	0.616566	0.761573	0.875873	0.999857	6.214841
R caudate	0.615949	0.761030	0.887008	0.999859	5.373920
L putamen	0.638358	0.778520	0.921826	0.999806	5.852025
R putamen	0.644151	0.782333	0.921171	0.999810	7.703920
L hippocampus	0.610517	0.757151	0.880389	0.999831	5.070776
R hippocampus	0.624292	0.767694	0.880912	0.999829	4.717766
cerebellum	0.823884	0.903208	0.971920	0.997405	9.179919
brainstem	0.789495	0.882174	0.947097	0.999350	6.052942

Table 6.2: PCA Mean Region Results

Region	Jaccard	Dice	Sensitivity	Specificity	Hausdorff Distance
L superior frontal gyrus	0.723767	0.838750	0.849444	0.998919	12.137237
R superior frontal gyrus	0.716343	0.833588	0.840000	0.998910	13.114236
L middle frontal gyrus	0.683542	0.810680	0.810366	0.998959	13.048929
R middle frontal gyrus	0.678911	0.807193	0.810803	0.998914	13.712247
L inferior frontal gyrus	0.634223	0.773686	0.778296	0.999487	11.677897
R inferior frontal gyrus	0.622534	0.763123	0.755243	0.999485	11.334471
L precentral gyrus	0.484236	0.633635	0.627026	0.998989	12.748230
R precentral gyrus	0.453115	0.606130	0.605297	0.998923	12.882180
L middle orbitofrontal gyrus	0.622257	0.763540	0.767390	0.999772	8.585295
R middle orbitofrontal gyrus	0.617847	0.760891	0.771864	0.999757	8.350426
L lateral orbitofrontal gyrus	0.552005	0.708425	0.690068	0.999837	9.824250
R lateral orbitofrontal gyrus	0.512180	0.672881	0.628887	0.999858	8.971915
L gyrus rectus	0.612163	0.756525	0.737975	0.999932	6.258452
R gyrus rectus	0.633144	0.772880	0.770307	0.999926	5.812882
L postcentral gyrus	0.434601	0.587804	0.558562	0.999223	14.099164
R postcentral gyrus	0.408743	0.563922	0.533328	0.999190	13.983447
L superior parietal gyrus	0.556056	0.704349	0.721987	0.999109	14.218304
R superior parietal gyrus	0.539259	0.693193	0.720794	0.998995	14.687505
L supramarginal gyrus	0.514577	0.668149	0.649829	0.999565	14.362728
R supramarginal gyrus	0.504165	0.660224	0.641158	0.999583	13.642907
L angular gyrus	0.465434	0.617921	0.617584	0.999288	16.512930
R angular gyrus	0.483538	0.643939	0.649569	0.999282	14.939041
L precuneus	0.554464	0.707743	0.709618	0.999667	11.868805
R precuneus	0.565814	0.716685	0.729749	0.999651	11.561882
L superior occipital gyrus	0.413742	0.572508	0.565408	0.999666	15.803370
R superior occipital gyrus	0.400113	0.559634	0.555124	0.999621	17.115829
L middle occipital gyrus	0.537925	0.690740	0.722123	0.999129	16.613628
R middle occipital gyrus	0.542587	0.696380	0.726612	0.999126	16.399083
L inferior occipital gyrus	0.536324	0.683110	0.653741	0.999776	10.995922
R inferior occipital gyrus	0.570695	0.716434	0.689606	0.999783	11.164366
L cuneus	0.505683	0.664888	0.677309	0.999777	16.413925
R cuneus	0.516368	0.674080	0.677865	0.999781	14.898427
L superior temporal gyrus	0.695325	0.818443	0.817113	0.999458	11.121614
R superior temporal gyrus	0.695856	0.818804	0.817505	0.999481	11.127415
L middle temporal gyrus	0.603225	0.750331	0.748772	0.999333	15.265366
R middle temporal gyrus	0.612659	0.757383	0.760437	0.999334	12.821224
L inferior temporal gyrus	0.611088	0.757054	0.756072	0.999484	12.207272
R inferior temporal gyrus	0.614081	0.758863	0.751244	0.999467	12.003975
L parahippocampal gyrus	0.658043	0.792791	0.791443	0.999867	7.343991
R parahippocampal gyrus	0.640200	0.779523	0.774728	0.999858	7.706724
L lingual gyrus	0.631806	0.765070	0.749558	0.999751	10.034694
R lingual gyrus	0.652529	0.779849	0.772876	0.999745	9.448630
L fusiform gyrus	0.667202	0.798645	0.800630	0.999741	9.563310
R fusiform gyrus	0.659260	0.791440	0.792009	0.999734	9.784716
L insular cortex	0.761747	0.864496	0.883379	0.999873	3.872961
R insular cortex	0.743345	0.852240	0.869554	0.999875	4.163772
L cingulate gyrus	0.657701	0.792143	0.803268	0.999708	8.494722
R cingulate gyrus	0.648280	0.784281	0.811811	0.999654	8.663810
L caudate	0.701569	0.822146	0.802079	0.999956	6.255313
R caudate	0.703240	0.823693	0.814325	0.999953	5.301520
L putamen	0.728026	0.841585	0.838632	0.999930	5.594183
R putamen	0.729981	0.843207	0.839252	0.999931	7.242599
L hippocampus	0.707747	0.828287	0.812041	0.999937	4.802703
R hippocampus	0.707886	0.828380	0.816703	0.999931	4.601887
cerebellum	0.899592	0.944718	0.961994	0.998728	10.361291
brainstem	0.866188	0.928009	0.929527	0.999757	5.563309

Table 6.3: Voting Mean Region Results

Region	Jaccard	Dice	Sensitivity	Specificity	Hausdorff Distance
L superior frontal gyrus	0.678743	0.808054	0.753098	0.999326	17.069348
R superior frontal gyrus	0.656778	0.792320	0.723673	0.999362	17.282361
L middle frontal gyrus	0.672356	0.803044	0.773569	0.999168	29.273995
R middle frontal gyrus	0.658671	0.793227	0.781069	0.998958	22.607256
L inferior frontal gyrus	0.636110	0.776630	0.780410	0.999470	13.228251
R inferior frontal gyrus	0.604309	0.751899	0.715032	0.999557	28.690861
L precentral gyrus	0.570494	0.724926	0.664507	0.999516	54.395750
R precentral gyrus	0.558411	0.714737	0.686981	0.999345	12.353161
L middle orbitofrontal gyrus	0.535188	0.694235	0.608764	0.999872	32.120356
R middle orbitofrontal gyrus	0.545058	0.703394	0.638090	0.999837	23.148568
L lateral orbitofrontal gyrus	0.479267	0.645062	0.589909	0.999843	16.574153
R lateral orbitofrontal gyrus	0.456399	0.622560	0.547078	0.999875	10.093706
L gyrus rectus	0.554474	0.711648	0.646522	0.999945	6.083291
R gyrus rectus	0.532776	0.691457	0.604104	0.999954	6.814861
L postcentral gyrus	0.511270	0.673389	0.611720	0.999515	14.761311
R postcentral gyrus	0.462263	0.629561	0.555988	0.999517	13.407234
L superior parietal gyrus	0.574400	0.728806	0.650314	0.999635	15.130718
R superior parietal gyrus	0.550334	0.708225	0.634895	0.999583	21.972708
L supramarginal gyrus	0.544695	0.701477	0.622686	0.999788	31.740162
R supramarginal gyrus	0.506434	0.668954	0.614908	0.999690	18.664096
L angular gyrus	0.492228	0.654937	0.581790	0.999658	18.939472
R angular gyrus	0.530550	0.690996	0.645568	0.999556	14.077258
L precuneus	0.572165	0.726179	0.746575	0.999648	24.893301
R precuneus	0.561450	0.717098	0.727721	0.999660	24.665593
L superior occipital gyrus	0.455392	0.619612	0.534642	0.999851	11.953124
R superior occipital gyrus	0.415086	0.581202	0.480320	0.999855	12.966847
L middle occipital gyrus	0.488718	0.652614	0.556426	0.999665	14.395611
R middle occipital gyrus	0.494044	0.658698	0.578075	0.999594	15.621224
L inferior occipital gyrus	0.473336	0.630992	0.550650	0.999789	12.081841
R inferior occipital gyrus	0.484961	0.641132	0.606433	0.999678	11.428252
L cuneus	0.508199	0.668576	0.626877	0.999853	17.824856
R cuneus	0.488342	0.652208	0.617406	0.999826	13.257395
L superior temporal gyrus	0.619336	0.763054	0.713458	0.999540	28.274902
R superior temporal gyrus	0.618017	0.762777	0.716168	0.999533	32.238738
L middle temporal gyrus	0.547221	0.704923	0.656755	0.999450	22.676242
R middle temporal gyrus	0.542005	0.701491	0.661062	0.999402	20.429677
L inferior temporal gyrus	0.525416	0.684424	0.649248	0.999475	43.423873
R inferior temporal gyrus	0.522779	0.683569	0.622011	0.999552	14.086580
L parahippocampal gyrus	0.578610	0.728195	0.678805	0.999884	14.311745
R parahippocampal gyrus	0.553277	0.705956	0.635450	0.999898	10.802467
L lingual gyrus	0.563192	0.712161	0.735183	0.999551	16.907396
R lingual gyrus	0.554125	0.705780	0.744865	0.999504	15.851577
L fusiform gyrus	0.574384	0.724035	0.729884	0.999638	14.599786
R fusiform gyrus	0.546420	0.699939	0.639211	0.999769	19.068377
L insular cortex	0.669949	0.801923	0.852205	0.999780	27.490853
R insular cortex	0.676771	0.805839	0.845952	0.999812	14.251284
L cingulate gyrus	0.584106	0.735887	0.734846	0.999655	40.698404
R cingulate gyrus	0.555581	0.710974	0.739240	0.999532	18.807962
L caudate	0.551473	0.707132	0.595558	0.999976	7.516682
R caudate	0.650945	0.786987	0.747466	0.999956	7.260142
L putamen	0.645613	0.784031	0.751637	0.999924	6.538608
R putamen	0.609282	0.754652	0.665459	0.999958	9.816367
L hippocampus	0.638602	0.778098	0.744393	0.999929	6.134519
R hippocampus	0.595932	0.744065	0.674819	0.999938	5.200453
cerebellum	0.742712	0.833347	0.801915	0.998509	24.948051
brainstem	0.755548	0.858797	0.817903	0.999728	22.538838

Table 6.4: LONI Brain Parser Mean Region Results

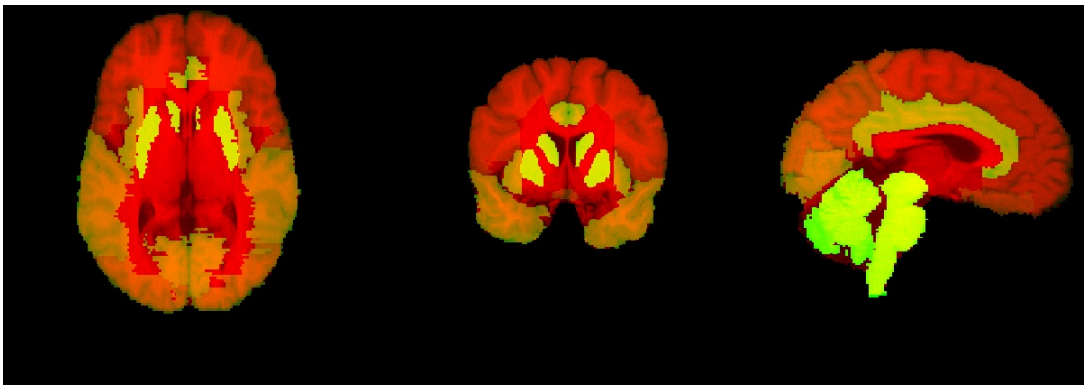


Figure 6.1: Manual Segmentation Results

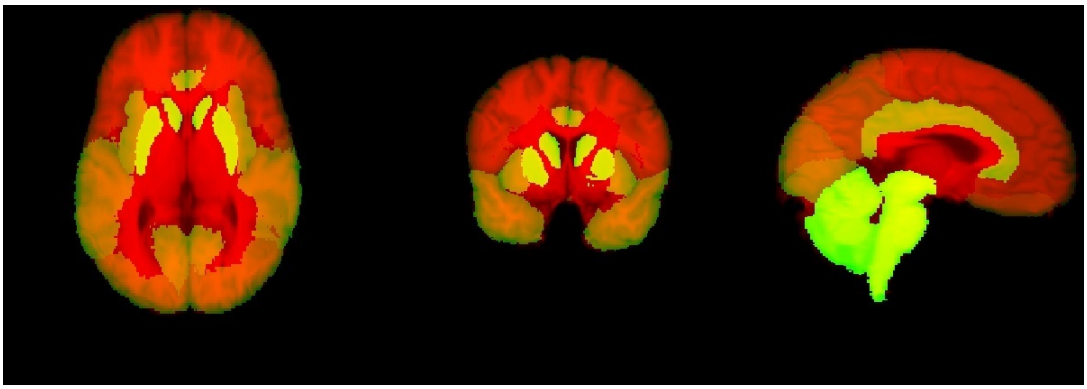


Figure 6.2: PCA-segmentation Results

6.5 Segmentation Images

Figure 6.1 shows horizontal, coronal, and sagittal slices of a typical brain image with the manually segmented, gold standard labels overlaid on the image in yellow. Figure 6.2 shows the same slices overlaid with the labels from the segmentation that was automatically computed by the algorithm that we developed in this thesis.

CHAPTER 7

Conclusion

Our novel segmentation algorithm based on PCA atlases operates by learning a combined statistical model of the training intensity images and their corresponding label images—the linear, PCA atlas subspace—finding the point at which an unlabeled subject image projects into this subspace, and nonlinearly registering the intensity and label images associated with that projected point—the PCA atlas—to a subject image in order to segment and label the image. The nonrigid registration is needed in general because there will be a mismatch between the intensity image component of the PCA atlas and the input subject image.

We tested our algorithm against the two competing methods. Its performance is similar to that of the voting strategy in all but the Sensitivity measure, meaning that it had the fewest false negatives or that it included more voxels in the boundary of the regions that were true positives than the other algorithms. It was able to describe the information from the training data in a very compact form, which can be advantageous when there is a huge amount of training data. In such a case, the voting strategy would have an exorbitantly long running time and would be impractical for computing timely image segmentations because it would need to register all the images together for each subject image. Our method would be able to store this information in a basis comprising a significantly smaller number of vectors than the number of training examples while maintaining a high level of accuracy. Once the statistical model is created, it can be used to find

the location of any subject image in its statistical space and not increase the segmentation running time as would the voting strategy.

Another important aspect of these algorithms is the time that they take to perform a complete segmentation. Since the voting procedure requires as many registrations as there are images in the training set, it took more than two hours to complete the segmentation using 39 training images in order to generate the results reported in Section 6. This is required for any segmentation using the voting algorithm, since any subject image is required to be in correspondence with all the training data and this cannot be computed beforehand. Our method took four hours to complete a segmentation using the full-resolution training data, including the offline training and online segmentation phases of the algorithm. The segmentation phase takes roughly 30 minutes to perform by itself. Also, once our algorithm has been trained, it can be applied to any subject image that is well represented by the training data. Similarly, the LONI brain parser, takes roughly the same amount of time for the actual segmentation after it has been trained. These times do not include the time to preprocess the image data, which may include skull-stripping and inhomogeneity correction.

7.1 Future Work

The performance of our method is dependent on how well it is able to capture information from the training data. In particular, when a model is built for the label images contained in the training set, the generation of labels from the basis is highly dependent on the thresholding that takes place. In its current implementation, the threshold is selected as the midpoint of the range of possible values that a generated label can take. This parameter may need to be optimized for better performance. Also, when deciding the ownership of a particular voxel,

random noise is added so as not to favor one over the other, but another strategy, such as splitting the voxels evenly between the number of competing regions, may be better. This is an issue that can be investigated in future work.

The nonrigid registration algorithm has a significant effect on the segmentation results. Our framework can be kept as is and different registration algorithms can be employed. It would be informative to evaluate quantitatively the performance of different nonrigid registration algorithms in order to determine how dependent our method is on them.

Another aspect of our method that requires further investigation is the way that the label data are analyzed. Currently, a binary image is created for each of the labels in the image space and a separate PCA space is created for each binary image space. One alternative to this would be to fit a mesh to a particular label and register the points on the mesh to each of the training label images. Then a PCA space can be created by processing the points in the mesh, instead of treating a binary image as a point in a high dimensional space. This would help to ease the computational burden of generating the labels and it would place more importance on the surface of the anatomical regions in question instead of the 3D volume structure. This notion can also be extended to the intensities. We could focus our attention only on the image intensity values that occur at the surface interface. PCA can be performed on these intensities to create a statistical model. Compared to using the entire voxel space, which contains useful intensity information, this alternative method would focus on relatively few intensities. The underlying idea is that if the model is positioned initially in approximately the correct location, it can use its learned knowledge to deform its surface parameters to better fit the intensities of a subject image. However, by considering intensity information only at the surface, the model may restrict its

intensity information so much that it will be unable to converge onto the correct boundary. This could be improved by using the gradient of the intensity image instead of the actual intensities. A neighborhood of intensities may be used to compute a smoothed gradient. This affords intensities sampled from these regions more information about the neighborhood of the mesh and will hopefully improve its ability to capture how the model changes as it approaches the correct boundary.

In the algorithm presented in this thesis, the intensity images and label images were affinely registered together as a group. Then a statistical model was generated for the corresponding intensities and labels. Another possible way to model the intensities would be to register them together nonlinearly to the same structural image. The structural images in our case are represented as label images. The mean label image can be used as a target to register the remaining training data. The transformation would thus put each training label image into close correspondence with the mean label image. These transformations can be applied to the intensity images. PCA of the intensity images can then be performed to account for the change texture. A separate PCA is performed on the unregistered label images so that its texture and structure are modeled separately. To generate a new image, a label image and texture image are synthesized. Then the texture image is transformed from the mean label shape to the synthesized label shape. This type of modeling of the data is similar to the Active Appearance Model (Cootes et al., 1998).

Another method that might improve the segmentation would be to create a separate statistical model for each of the regions being segmented. Each specific anatomical region would then have more freedom to adapt to fit a subject image, because it would be independent of the other models. A bounding box could be

created around each region to bound the area that the statistical model takes into account, or the masking method that we use can be adapted to help restrict the model to only the most essential voxels in the image.

Finally, we would like to develop potentially more powerful nonlinear generalizations of the linear (matrix) algebraic PCA atlas subspace method developed in this thesis, through the use of multilinear (tensor) algebra and the multilinear PCA (Vasilescu and Terzopoulos, 2003, 2007). This should make it possible to learn multilinear PCA (MPCA) atlas subspaces from more extensive training image datasets associated with multiple causal factors, such as disease type, age, and gender.

Bibliography

Aljabar, P., Heckemann, R., Hammers, A., Hajnal, J., and Rueckert, D. (2007). Classifier selection strategies for label fusion using large atlas databases. *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2007*, pages 523–531. 11

Ardekani, B., Guckemus, S., Bachman, A., Hoptman, M., Wojtaszek, M., and J., N. (2005). Quantitative comparison of algorithms for inter-subject registration of 3d volumetric brain mri scans. *Journal of Neuroscience Methods*, 142(1):67–76. 16

Arno, K., Brett, M., Satrajit, G., Jason, T., and Hirsch, J. (2005). Mindboggle: Automated brain labeling with multiple atlases. *BMC Medical Imaging*, 5. 11, 35

Arthur, D. and Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, page 1035. Society for Industrial and Applied Mathematics. 7

Babalola, K., Cootes, T., Twining, C., Petrovic, V., and Taylor, C. (2008). 3D brain segmentation using active appearance models and local regressors. *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2008*, pages 401–408. 12

Babalola, K., Patenaude, B., Aljabar, P., Schnabel, J., Kennedy, D., Crum, W., Smith, S., Cootes, T., Jenkinson, M., and Rueckert, D. (2009). An evaluation of four automatic methods of segmenting the subcortical structures in the brain. *NeuroImage*. 36

Balci, S., Golland, P., Shenton, M., and Wells, W. (2007). Free-form B-spline deformation model for groupwise registration. In *Medical image computing and computer-assisted intervention: MICCAI... International Conference on Medical Image Computing and Computer-Assisted Intervention*, volume 10, page 23. NIH Public Access. 17, 31

Bhatia, K., Hajnal, J., Puri, B., Edwards, A., and Rueckert, D. (2004). Consistent groupwise non-rigid registration for atlas construction. In *IEEE International Symposium on Biomedical Imaging: Nano to Macro, 2004*, pages 908–911. 17

Comaniciu, D. and Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619. 8

Cootes, T., Taylor, C., Cooper, D., Graham, J., et al. (1995). Active shape models-Their training and application. *Computer vision and image understanding*, 61(1):38–59. 20

Cootes, T. F., Edwards, G. J., and Taylor, C. J. (1998). Active appearance models. *Proceedings of the European Conference on Computer Vision*, 2:484–498. 46

Fischl, B., Salat, D., Busa, E., Albert, M., Dieterich, M., Haselgrove, C., van der Kouwe, A., Killiany, R., Kennedy, D., Klaveness, S., et al. (2002). Whole Brain Segmentation:: Automated Labeling of Neuroanatomical Structures in the Human Brain. *Neuron*, 33(3):341–355. 11, 12

Gering, D., Nabavi, A., Kikinis, R., Hata, N., ODonnell, L., Grimson, W., Jolesz, F., Black, P., and Wells III, W. (2001). An integrated visualization system for surgical planning and guidance using image fusion and an open MR. *Journal of Magnetic Resonance Imaging*, 13:967–975. 5

Giedd, J., Blumenthal, J., Jeffries, N., Castellanos, F., Liu, H., Zijdenbos, A., Paus, T., Evans, A., and Rapoport, J. (1999). Brain development during childhood and adolescence: A longitudinal MRI study. *Nature Neuroscience*, 2:861–862. 3

Grimson, W., Ettinger, G., Kapur, T., Leventon, M., Wells, W., and Kikinis, R. (1997). Utilizing segmented MRI data in image-guided surgery. *International Journal of Pattern Recognition and Artificial Intelligence*, 11(8):1367–1402. 4

Huttenlocher, D., Klanderman, G., and Rucklidge, W. (1993). Comparing images using the Hausdorff distance. *IEEE Transactions on pattern analysis and machine intelligence*, pages 850–863. 36

Jenkinson, M., Bannister, P., Brady, M., and Smith, S. (2002). Improved optimization for the robust and accurate linear registration and motion correction of brain images. *NeuroImage*, 17(2):825–841. 15

Jolesz, F., Nabavi, A., and Kikinis, R. (2001). Integration of interventional MRI with computer-assisted surgery. *Imaging*, 13:69–77. 5

Kass, M., Witkin, A., and Terzopoulos, D. (1988). Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331. 8

Kikinis, R., Gleason, P., Moriarty, T., Moore, M., Alexander III, E., Stieg, P., Matsumae, M., Lorensen, W., Cline, H., Black, P., et al. (1996). Computer-assisted interactive three-dimensional planning for neurosurgical procedures. *Neurosurgery*, 38(4):640. 5

Klein, A., Andersson, J., Ardekani, B., Ashburner, J., Avants, B., Chiang, M., Christensen, G., Collins, D., Gee, J., Hellier, P., et al. (2009). Evaluation of

14 nonlinear deformation algorithms applied to human brain MRI registration. *NeuroImage*, 46(3):786–802. 15

McInerney, T. and Terzopoulos, D. (1996). Deformable models in medical image analysis. In *Mathematical Methods in Biomedical Image Analysis, 1996., Proceedings of the Workshop on*, pages 171–180. 9

Nowinski, W. L. (2001). Computerized brain atlases for surgery of movement disorders. *Seminars in Neurosurgery*, 12:183–194. 5

Resnick, S., Pham, D., Kraut, M., Zonderman, A., and Davatzikos, C. (2003). Longitudinal magnetic resonance imaging studies of older adults: a shrinking brain. *Journal of Neuroscience*, 23(8):3295. 3

Shattuck, D., Mirza, M., Adisetiyo, V., Hojatkashani, C., Salamon, G., Narr, K., Poldrack, R., Bilder, R., and Toga, A. (2008). Construction of a 3D probabilistic atlas of human cortical structures. *NeuroImage*, 39(3):1064–1080. 34

Shattuck, D., Prasad, G., Mirza, M., Narr, K., and Toga, A. (2009). Online resource for validation of brain segmentation methods. *NeuroImage*, 45(2):431–439. 36

Shattuck, D., Sandor-Leahy, S., Schaper, K., Rottenberg, D., and Leahy, R. (2001). Magnetic resonance image tissue classification using a partial volume model. *NeuroImage*, 13(5):856–876. 33, 34

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905. 9

Smith, S. (2002). Fast robust automated brain extraction. *Human Brain Mapping*, 17(3):143–155. 33

Studholme, C. and Cardenas, V. (2004). A template free approach to volumetric spatial normalization of brain anatomy. *Pattern Recognition Letters*, 25(10):1191–1202. 17

Talairach, J. and Tournoux, P. (1988). *Co-planar stereotaxic atlas of the human brain: 3-dimensional proportional system: An approach to cerebral imaging*. Thieme. 16

Thompson, P., Hayashi, K., De Zubicaray, G., Janke, A., Rose, S., Semple, J., Herman, D., Hong, M., Dittmer, S., Doddrell, D., et al. (2003). Dynamics of gray matter loss in Alzheimer's disease. *Journal of Neuroscience*, 23(3):994. 4

Trucco, E. and Verri, A. (1998). *Introductory Techniques for 3-D Computer Vision*. Prentice Hall PTR, Upper Saddle River, NJ, USA. 20

Tu, Z., Narr, K., Dollár, P., Dinov, I., Thompson, P., and Toga, A. (2008). Brain anatomical structure segmentation by hybrid discriminative/generative models. *IEEE transactions on medical imaging*, 27(4):495. 12, 35

Tu, Z. and Zhu, S. (2002). Image segmentation by data-driven Markov chain Monte Carlo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 657–673. 9

Twining, C. J., Cootes, T., Marsl, S., Petrovic, V., Schestowitz, R., and Taylor, C. J. (2005). A unified information-theoretic approach to groupwise non-rigid registration and model building. In *In Proceedings of the International Conference on Pattern Recognition (ICPR*, pages 1–14. Springer. 17

Vasilescu, M. and Terzopoulos, D. (2003). Multilinear subspace analysis of image ensembles. In *Proceedings of 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 93–99. 47

Vasilescu, M. and Terzopoulos, D. (2007). Multilinear projection for appearance-based recognition in the tensor framework. pages 1–8. 47

Witkin, A., Terzopoulos, D., and Kass, M. (1987). Signal matching through scale space. *International Journal of Computer Vision*, 1(2):133–144. 17

Zhu, S., Lee, T., and Yuille, A. (1995). Region competition: unifying snakes, region growing, energy/Bayes/MDL for multi-band image segmentation. In *International Conference on Computer Vision*, page 416. Published by the IEEE Computer Society. 9

Zollei, L. (2006). *A Unified Information Theoretic Framework for Pair-and Group-wise Registration of Medical Images*. PhD thesis, Massachusetts Institute of Technology. 17