GEOMETRIC APPROACHES TO GROUP-WISE IMAGE ANALYSIS

By

YUCHEN XIE

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2012

© 2012 Yuchen Xie

To my parents

ACKNOWLEDGMENTS

This dissertation is not possible without the help, support and guidance from many people. First, I would like to thank my advisor, Dr. Baba C. Vemuri, who introduced me to the field of computer vision and medical image analysis. His insight, encouragements and guidance have helped me through many obstacles during my graduate study. I feel so fortunate to have the opportunity to work with him. My deep gratitude also goes to Dr. Jeffrey Ho, who closely worked with me in most of my projects and provided numerous invaluable suggestions. I am also thankful to Dr. Anand Rangarajan, Dr. Arunava Banerjee and Dr. Sergei Shabanov for being on my thesis committee and being very generous of their time and support.

I feel very privileged to spend time with many wonderful people in the Lab of Computer Vision, Graphics and Medical Imaging (CVGMI). I am heartily grateful to Guang Cheng who has been extremely helpful and patient. I deeply appreciate Ting Chen for her friendship and support throughout my graduate study. I want to take this opportunity to thank Meizhu Liu, Wenxing Ye, Bing Jian, Santhosh Kodipaka, Angelos Barmpoutis, Ritwik Kumar, Dohyung Seo, Sile Hu, Qi Deng, Yan Deng, Yuanxiang Wang and other CVGMI lab members. We have shared so much joy together. I am also thankful to my wonderful friends in CISE and UF: Ying Xuan, Tao Li, Lu Chen, Yan Li, Ming Zhang, Zhe Wang, Lei Yang, Xiaoke Qin, Kun Li and Yu Chen for their help and support.

I am grateful to Dr. Xiaodong Tao for giving me the opportunity to work with him as an intern in GE Global Research and for his continued help. I also appreciate the opportunity given to me by Dr. Debargha Mukherjee to work at Google in summer 2011.

Finally and most importantly, I thank my parents for their unconditional love and support. My gratitude to them is beyond what I can express in words. This dissertation is dedicated to them.

The research for this dissertation was financially supported by UF Alumni Fellowship, NIH grants NS046812, EB007082 and NS066340 to Dr. Baba C. Vemuri, and NSF grant IIS 0916001 to Dr. Jeffrey Ho and Dr. Baba C. Vemuri.

TABLE OF CONTENTS

page

ACKNOWLEDGMENTS 4

LIST OF TABLES 8

LIST OF FIGURES 9

ABSTRACT 11

CHAPTER

1 INTRODUCTION 13

2 MULTIPLE IMAGE ATLAS CONSTRUCTION 16

2.1 Motivation 16
2.2 Related Work 20
2.3 Theoretical Framework for Atlas Construction 22
2.3.1 Rotational Invariance and the Geometry of the quotient space **Q** . 23
2.3.2 Graph representation of the manifold **M** 24
2.3.3 Estimating mean from pairwise geodesic distances 27
2.3.4 Computing the image atlases 29
2.4 Experiments 30
2.4.1 Experimental Results on COIL-20 31
2.4.2 Experimental Results on Synthetic Images 33
2.4.3 Multiple atlases for OASIS data 35
2.4.4 Atlases for male and female subjects 41
2.4.5 Atlases for dementia patients and healthy elderly subjects 43

3 A GEOMETRIC FRAMEWORK FOR TENSOR FIELDS ANALYSIS 45

3.1 Motivation and Related Work 45
3.2 Statistical Analysis in the Space of Tensor Fields 48
3.2.1 Geometry of Tensor Fields 49
3.2.2 Statistics on the Space of Tensor Fields 50
3.3 Tensor Fields Classification 53
3.4 Experimental Results 55
3.4.1 Deformation tensor fields 55
3.4.2 Diffusion tensor fields 57

4 NONNEGATIVE FACTORIZATION OF DIFFUSION TENSOR IMAGES 61

4.1 Motivation 61
4.2 Preliminaries 63
4.3 Algorithm and implementation details 65
4.3.1 Optimization with respect to the basis matrix **W** 65

4.3.2 Optimization with respect to coefficient matrix **H** 68
4.3.3 Diffusion tensor image segmentation 68
4.4 Experiments 70
4.4.1 Diffusion tensor image segmentation 70
4.4.2 Nonnegative factorization with multiple tensor fields 72

5 DICTIONARY LEARNING ON RIEMANNIAN MANIFOLDS 75

5.1 Motivation 75
5.2 Related Work 78
5.3 A Brief on Riemannian Manifold 78
5.4 Dictionary Learning on Riemannian Manifolds 79
5.5 Symmetric Positive Definite Matrix Space 83
5.6 Square-root Density Function Space 85
5.7 Experiments 87
5.7.1 Brodatz dataset 88
5.7.2 OASIS dataset 89

6 CONCLUSIONS 91

REFERENCES 93

BIOGRAPHICAL SKETCH 100

LIST OF TABLES

Table page

2-1 Means and standard deviations of subject ages for two classes partition 37

2-2 Means and standard deviations of subject ages for three classes partition . . . 39

3-1 Tensor Fields Classification on OASIS 57

5-1 The texture classification accuracy for different methods on the Brodatz dataset. 88

5-2 The classification accuracy for different methods on the OASIS dataset. 89

LIST OF FIGURES

Figure page

2-1 Intrinsic mean template construction. . . . 18
2-2 Semi-supervised graph partition. . . . 24
2-3 Samples of input duck images. . . . 31
2-4 Experimental results for duck images. . . . 32
2-5 Experimental results for five objects from COIL-20. . . . 32
2-6 Experimental results on synthetic images. . . . 34
2-7 Camera positions of input Bunny images. . . . 34
2-8 The histogram of age for all subjects in OASIS data set. 40 bins are used. . . . 35
2-9 Sample images from youth, middle-aged and elderly subjects. . . . 36
2-10 Two classes partition. . . . 37
2-11 Two atlases for OASIS data. . . . 38
2-12 Three classes partition. . . . 39
2-13 Three atlases for OASIS data. . . . 40
2-14 Atlases across different ages for subjects in OASIS data set. . . . 41
2-15 Female atlases generated by our semi-supervised algorithm. . . . 41
2-16 Male atlases generated by our semi-supervised algorithm. . . . 42
2-17 Volume ratios of CSF, GM and WM in atlases for female subjects and male subjects. . . . 42
2-18 Axial view of atlases for old subjects and dementia patients. . . . 43
2-19 Coronal view of atlases for old subjects and dementia patients. . . . 43
3-1 Example on synthetic tensor fields. . . . 46
3-2 The space of tensor fields. . . . 48
3-3 The log map on the unit sphere. . . . 53
3-4 Statistical Analysis for deformation tensor fields from old age group. . . . 55
3-5 Comparisons with voxel-based method. . . . 56
3-6 Five samples from input rat brain diffusion tensor images. . . . 58

3-7 Statistical analysis for diffusion tensor images. 58
3-8 Diffusion tensor image reconstruction. 59
3-9 Reconstruction errors between the test tensor field and its projections on the geodesic submanifolds of different dimensions. 59
4-1 Segmentation experiment on synthetic tensor field. 70
4-2 Diffusion tensor image segmentation. 71
4-3 Nonnegative factorization on multiple synthetic tensor fields. 72
4-4 Nonnegative factorization on real diffusion tensor images. 74
5-1 Dictionary learning on Riemannian Manifold. 80
5-2 Samples of Brodatz textures used in our experiments. 88

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

GEOMETRIC APPROACHES TO GROUP-WISE IMAGE ANALYSIS

By

Yuchen Xie

August 2012

Chair: Baba C. Vemuri
Major: Computer Engineering

Group-wise analysis of a set of images is a fundamental problem in computer vision and medical image analysis. A structured representation of the image collection is extremely useful in many important applications such as image registration, segmentation, classification and recognition. In this dissertation, we present a set of techniques which perform group-wise image analysis based on the geometric structure of the data, and we investigate the problems and solutions in two different contexts: low-dimensional subspaces embedded in an Euclidean and non-Euclidean high-dimensional ambient spaces. In particular, image atlas construction problem can be formulated as computing the mean template by using the intrinsic geometric structure of the image dataset. In contrast, statistical analysis of tensor images such as diffusion tensor images (DTI) and conductivity tensor images (CTI) requires the ambient space to be non-Euclidean.

We will start by presenting a novel method for constructing multiple image templates for a set of 2D or 3D images via an intrinsic geometric approach. Next, we will present a statistical framework for analyzing a group of tensor-valued images. Furthermore, we will focus on finding a small number of bases to represent the tensor-valued images, and we will present one approach based on nonnegative factorization in this dissertation. Finally, we will propose a dictionary learning framework for Riemannian manifolds. All of

the above mentioned methods have been experimentally validated and tested on both synthetic and real image data sets.

CHAPTER 1
INTRODUCTION

Finding an effective representation to model a collection of data is one of the central topics in statistical analysis. It has also become immensely important in various computer vision and medical imaging applications. For instance, Eigenface performs the principal component analysis (PCA) [68] to find a low-dimensional linear subspace for modeling a large set of face images and uses it in face recognition. Active shape model (ASM) [14] and active appearance model (AAM) [13] build a statistical model of object shape and appearance during the training phase. The model can be used as a prior in landmarks alignment, object segmentation and tracking. In medical image analysis, the first order statistics (mean) of images from different subjects is usually considered as an unbiased template (atlas) that can be useful in medical diagnosis, particularly with associated clinical information.

The work presented in this dissertation aims to consider the underlying geometric structure of the data when modeling a set of images. In computer vision and medical imaging, images are often modeled as points in a high-dimensional vector space (image space). For example, a small image with the size 100×100 is considered as a point in a 10^4 dimensional image space after the usual rasterization. In recent years, manifold learning techniques such as Isomap [34], LLE [53] and Laplacian Eigenmaps [6] have gotten a lot of attention in computer vision and machine learning communities. These methods assume that the images of interest lie on an embedded non-linear space situated in a high-dimensional ambient space. Given this assumption, it's more appropriate to perform statistical analysis on the embedded non-linear space instead of the ambient image space. For instance, the mean of a set of face images should also be a face image instead of "some image" that remotely resembles a face, i.e., it falls outside the space of face images. Extending this idea to image template construction, we

propose a novel approach to construct multiple image atlases via the intrinsic averaging on the embedded space of the dataset.

Further extensions can be made by assuming that the ambient image space itself is a nonlinear manifold, and a diffusion tensor image (DTI) offers the first example. It uses a symmetric positive-definite matrix (SPD) within each voxel to model the diffusion of water molecules in tissues, and each diffusion tensor image can be considered as an SPD matrix field. However, the space of SPD matrices is not a vector space, and the statistical methods developed for vector space cannot be directly used for tensor images. To remedy this, we generalize three popular methods in this thesis: principal component analysis (PCA), nonnegative matrix factorization (NMF) [36] and dictionary learning to analyze a group of tensor valued data. Specifically, we consider a tensor image as a point in the product of SPD matrix spaces. The well-known PCA paradigm can be extended to the product space by first computing the Karcher mean of tensor images using an iterative method. The classes of tensor images are then represented by submanifolds spanned via a set of principal geodesics, and this approach allows us to directly extend the well-known classification framework such as Eigenface to tensor fields classification. Similar to PCA, NMF also represents the data as a linear combination of bases. Instead of the orthonormal constraints placed on the basis vectors in PCA, NMF requires elements in both coefficients and bases to be nonnegative. As the intensity values in images are nonnegative, NMF has been considered as more natural for image data than PCA. In our generalization of nonnegative factorization, each basis is a vector of positive semi-definite matrices, where nonnegativity essentially comes from the cone structure underlying the space of SPD matrices. This novel method offers an effective approach to simultaneously analyze a large number of diffusion tensor images, an essential task in group studies and other applications. Dictionary learning considers a model to represent the data as several low dimensional subspaces. The dictionary trained from a collection of images is a set

of basis images (generators) and each image sample can be effectively approximated by choosing only a small number of generators from the dictionary. This technique has been successfully used in image restoration, reconstruction and object recognition. In this thesis, we propose a general dictionary learning framework for data on Riemannian manifolds such as SPD matrix space and hypersphere.

The rest of this dissertation is organized as follows. Chapter 2 describes a multiple image atlas construction method. In Chapter 3 and Chapter 4, we generalize the PCA and NMF for tensor images analysis. In Chapter 5, we present a dictionary learning approach on Riemannian manifolds. Finally, we summarize the main contributions of this dissertation and discuss a few thoughts for future work in Chapter 6. Part of the work presented in this dissertation has been published in [80–82].

CHAPTER 2
MULTIPLE IMAGE ATLAS CONSTRUCTION

2.1 Motivation

Atlases as the informative representatives of a population of images have been widely used in many computer vision and medical imaging applications such as template-based image alignment [64], atlas-based image segmentation [5, 21, 65, 73] and statistical analysis across subjects [3, 30]. In addition, for heterogeneous or longitudinal data sets, multiple atlases are usually required to provide an informative and complete representation of the image data. For example, the ages of the subjects in Open Access Series of Imaging Studies (OASIS) database [43] range from 18 to 96. Since the structural difference in the brain across different age groups can be significant [44], it's not appropriate to construct a single representative atlas image for the OASIS data. Not surprisingly, algorithms and methods for computing multiple atlases from image collections have attracted considerable amount of attention recently, and in particular, the unbiased diffeomorphic atlas construction algorithm proposed in [32] has been influential for several notable recent developments such as the iCluster algorithm proposed in [54].

The iCluster algorithm computes multiple atlases by fitting a Gaussian mixture model to the input images. The resulting expectation-maximization algorithm (EM) iteratively groups the images into different clusters (the E-step) and computes a single atlas for images in each cluster by averaging the multiple aligned images, an approach that is similar in spirit to the unbiased diffeomorphic atlas construction algorithm. Figure 2-13 displays several atlas images produced by the iCluster method. A common and visible feature of these atlas images is their characteristic blurriness and the apparent absence of clear and sharp anatomical details. In short, with their characteristic lack of image fidelity, the computed atlases do not come across as real specimen of MRI images of human brains, a shortcoming that could be critical in some

applications. For instance, in [78], it has been argued that a clear and sharp atlas is important for the accuracy of atlas-based multiple image registration. In the clinical setting, a characteristically clear and sharp atlas image as an informative representative of a subpopulation of patients is certainly more meaningful and accessible to medical practitioners than a fuzzy atlas image without clearly identifiable anatomical details.

This loss of image details and structures can be traced back to the atlas construction step (M-step) of the iCluster algorithm in which the atlas is computed as the arithmetic average of group-wise registered images, a popular atlas-construction paradigm first introduced in [32]. It is our observation that the unbiased diffeomorphic atlas construction algorithm often produces blurry and fuzzy atlas images unless the input images are highly homogeneous in their structures. Unfortunately, this is usually not the case and the computed atlases almost always suffer from the loss of image details in various degrees. This undesirable shortcoming can be explained in both practical and theoretical terms. In practice, the existing numerical algorithms used for computing group-wise registration cannot guarantee a globally optimal solution, and unless the images are unusually similar, non-negligible registration errors cannot be avoided in general. Coupled with the suboptimal choices of regularization constants that are often difficult to determine, it is not difficult to imagine a blurry and fuzzy atlas computed as the pixel-wise arithmetic mean of many and inaccurately group-wise registered images. The second and more geometric explanation can be offered as in Figure 2-1. The smooth variation of the input images allows us to assume (e.g., [16]) that these images belong to some smooth submanifold $\mathbf{M}$ in the ambient image space $\mathcal{I}$. Unfortunately, this manifold structure is usually difficult to model directly and perhaps more importantly, it is difficult to utilize in the group-wise registration framework. Therefore, in essence, the unbiased diffeomorphic atlas construction algorithm computes the mean of the images in the ambient image space $\mathcal{I}$, and as the figure shows, this does not guarantee that the

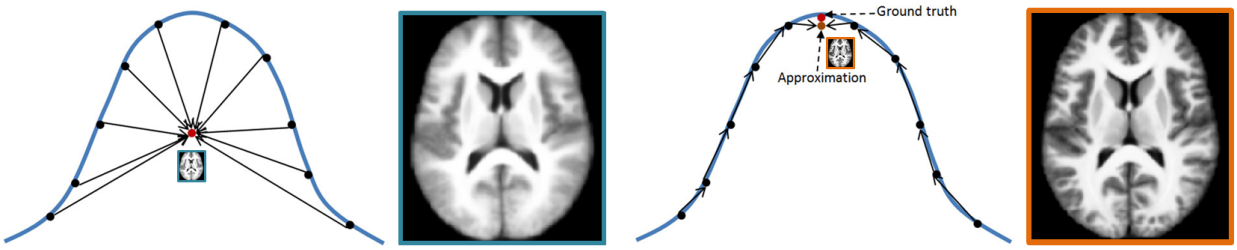


Figure 2-1. Ten points on an one-dimensional manifold. **Left:** Mean for the ten points computed using the metric of the ambient space (iCluster). **Right:** Mean computed using the metric of the submanifold **M** (The proposed method). Notice the greater preservation of image features and structures of the right atlas.

result will be on or near the submanifold **M**. This results in the loss of common image features and renders the atlas images often blurry and fuzzy.

In the context of multiple atlas construction from a large image collection, we propose a novel atlas construction algorithm based on two novel conceptual improvements that aim to correct the shortcomings described above. First, unlike the EM algorithm proposed in [54], we will explicitly decouple the clustering and atlas construction steps. This decoupling provides us with greater flexibility in applying manifold learning methods (e.g., [6, 34]) to model the manifold **M**, an objective that is difficult to accomplish using the traditional EM-based gaussian mixture model as in [54]. In particular, we will require that the atlas images are rotation invariant in the sense that suppose $\mathbf{I}$ is the atlas for the image collection $\{\mathbf{I}_1, \cdots, \mathbf{I}_n\}$ and if n different rotations g_i are applied to these images to form a new collection $\{g_1(\mathbf{I}_1), \cdots, g_n(\mathbf{I}_n)\}$, then the atlas $\tilde{\mathbf{I}}$ for the latter collection should be related to $\mathbf{I}$ by some rotation. This technical requirement means that we can no longer assume that the input images belong to some submanifold **M** in the image space $\mathcal{I}$. Instead, the correct assumption should be that modulo rotations, the images belong to a submanifold **M** in an (abstract) quotient space **Q** of $\mathcal{I}$ [45, 46], and manifold learning provides a suitable context and natural solution for modeling **M** in this abstract setting. Furthermore, its flexibility also allows us to readily incorporate labeled images in a semi-supervised context, again an objective that would be awkward

to formulate using the frameworks proposed earlier (e.g., [32, 54]). Second, we will abandon the usual paradigm of computing the atlas image as the arithmetic mean of all group-wise registered images. Instead, following the manifold assumption, we will propose a new paradigm for atlas image construction by computing the atlas as the weighted arithmetic mean of a few group-wise registered images. Figure 2-1 illustrates the main idea that on the manifold $\mathbf{M}$, the mean (atlas) should be entirely determined only by its neighbors, and its computation could be formulated in a two-step process that first determines the neighbors of the abstract mean on the manifold $\mathbf{M}$ (localization) and then renders the mean as an atlas image using these neighbors (realization). In particular, these two steps utilize the input images differently: in the localization step, all images are used to determine the neighbors of the abstract mean on $\mathbf{M}$ while in the following realization step, only these neighbors are used to render the atlas image through weighted group-wise registration. To the best of our knowledge, the explicit identification and investigation of these two separate and critical processes in atlas construction have not been reported or proposed in the literature.

The proposed multiple atlas construction algorithm incorporates the ideas discussed above: A k-nearest neighbor graph (k-NN) is used to model the manifold structure of $\mathbf{M}$ [6, 34], and the graph partition algorithm is applied to the k-NN graph to compute clusters of the input images. Each atlas is computed as the mean m of the images belong to each cluster on $\mathbf{M}$. In general, the metric on $\mathbf{M}$ is unknown; however, the pairwise geodesic distances $d_{ij} = \mathbf{d}_{\mathrm{M}}(\mathbf{I}_i, \mathbf{I}_j)$ between images can be estimated using the metric in the quotient space $\mathbf{Q}$. An important element in the proposed algorithm is to estimate the geodesic distances $\mathbf{d}_{\mathrm{M}}(m, \mathbf{I}_i)$ between the abstract mean m on $\mathbf{M}$ and images $\mathbf{I}_i$ using the pairwise geodesic distances d_{ij} as the inputs. We formulate this as a convex optimization problem, and the solution to this optimization problem will provide an approximation to $\mathbf{d}_{\mathrm{M}}(m, \mathbf{I}_i)$. This will allow us to locate the mean m on the manifold $\mathbf{M}$ in the localization step by determining its neighbors (given a threshold). In

the realization step, we use the weighted group-wise registration method to form the atlas images in each cluster given the estimated neighbors of the atlas. Several atlases computed by the proposed method are shown in Figure 2-13, and in comparison, they retain considerable amount of important image features and much better preservation of image structures. An extensive evaluation and validation of the proposed algorithm using OASIS data will be reported later, and the experimental results on OASIS have shown that, with greater preservation of important image features and details, the image atlases produced by the proposed algorithm enjoy better image sharpness when compared with atlases obtained using exiting methods.

2.2 Related Work

Image atlas as a standard template has been widely used for the discovery of dense correspondences between MRI images of different subjects. Thus it is natural to construct the image atlas based on group-wise image registration (e.g., [32]). In this type of approach, the atlas formation is cast into an iterative energy minimization framework: input images are non-rigidly registered to a template in the common domain and the template is updated by averaging the aligned images. Unfortunately, because of the large number of variables and the non-convex cost functional, this algorithm, similar to many others, cannot avoid the prickly traps of local minima. Moreover, the rigidity of the framework makes it difficult to incorporate other useful models and information in the computation. As alluded to in the previous section, the atlas generated by this method essentially computes the atlas with respect to the the ambient image space while disregarding the potentially useful assumption of an implied manifold structure by the input images.

In recent years, manifold learning techniques have percolated into the medical image analysis community, and they have found successes in several recent works. [26] performed brain population analysis by projecting input brain images onto a low-dimensional space using Isomap [34]. [29] proposed a framework for registering

images to the atlas on anatomical manifolds. Their empirical manifold is constructed from input images as a k-NN graph. A sample from the input population with the smallest sum of squares of geodesic distances to all input images is selected as the atlas. [31] introduced a hierarchical groupwise registration framework using the k-NN Isomap that provides the intrinsic structure of the input image dataset for their algorithm. However, all the methods discussed above construct a single atlas for the entire input image set, which implicitly assumes a homogeneous population, an assumption that is inappropriate for studying heterogeneous image data sets that are far more common and important in medical imaging analysis. iCluster is an EM-based algorithm presented by [55] for computing multiple template images for an image population. The algorithm fits a Gaussian mixture model to the input images and the input images are not assumed to possess an intrinsic manifold structure. Furthermore, it is well-known that the optimization in the EM-algorithm cannot guarantee the globally optimal solution, and therefore, good initializations of the EM-algorithm, which may not be available or even possible, are critical for the success of the method.

Three pertinent features distinguishing our method from the iCluster [55] and many previously published methods. First, the proposed method explicitly models the intrinsic manifold structure of the input images using manifold learning (k-NN graph) and the manifold modeling takes place in the abstract quotient space instead of the usual image space. Second, our method decouples the clustering step from the atlas construction step that allows the application of more powerful clustering algorithm to produce more homogeneous clusters. Finally, as mentioned in the motivation, we explicitly identify two important and separate steps in the atlas formation process: localization and realization. We formulate a convex cost function in the localization step that bypasses the difficulty of globally minimizing the complicated non-convex functional, and in the realization step, we group-wise register only a few images, a task that can be accomplished efficiently and accurately due to both the number of images as well as their homogeneity.

2.3 Theoretical Framework for Atlas Construction

This section presents the proposed algorithm for computing multiple atlases from a collection of images. The primary aim of our algorithm is to produce sharp atlas images that retain clear important anatomical structures common to the subpopulation of images they represent. Furthermore, we also require that the computed atlases are invariant with respect to image rotations, an important aspect of the proposed method that will be elaborated later. The strategy for achieving these two somewhat disparate goals is to utilize the notion of manifold structure implied by the images. Specifically, we will assume, as in many manifold-based learning methods (e.g.,[7, 88]), that the input images are samples from an unknown manifold $\mathbf{M}$, and the manifold structure is then modeled using a graph G computed from the input images and some metric (e.g., similarity) information among the images. The rotational invariance requirement implies that the manifold of interest $\mathbf{M}$ should be considered as a submanifold of a quotient space (quotient by the rotations), and computationally, this requires rotationally-invariant metric for computing the graph G. The graph G allows us to partition the image collection into subcollections, in both supervised and semi-supervised fashion, such that an atlas can be computed from each subcollection of images using only metrical information between the images.

Before delving into the details, we will fix the notations that will be used throughout the following discussion. Let $\mathcal{I}$ denote the space of images, and we define images in $\mathcal{I}$ formally as L^2-functions on a finite image domain $\Omega \subset \mathbb{R}^d$ ($d = 2$ for 2D images and $d = 3$ for 3D images). Let $\mathcal{C} \equiv \{\mathbf{I}_1, \cdots, \mathbf{I}_n\}$ denote an input collection of n images and $\mathcal{A} \equiv \{\mathbf{I}_{a_1}, \cdots, \mathbf{I}_{a_t}\}$ the collection of t atlases for the image collection $\mathcal{C}$. A rotation $g \in \mathbf{SO}(d)$ in the image domain Ω transforms an image $\mathbf{I}$ into a rotated image $g(\mathbf{I})$ according to the formula $g(\mathbf{I})(x) = \mathbf{I}(g^{-1}(x))$ for $x \in \Omega$ and $\mathbf{I} \in \mathcal{I}$.

2.3.1 Rotational Invariance and the Geometry of the quotient space Q

The rotation invariance property described above can be formulated precisely as follows. Let $\mathcal{C}$ denote the input image collection and $\mathcal{C}' \equiv \{\mathbf{I}'_1, \cdots, \mathbf{I}'_n\}$ are images obtained by applying n rotations to images in $\mathcal{C}$: $\mathbf{I}'_i = g_i(\mathbf{I}_i)$ for $g_i \in \mathbf{SO}(d), 1 \leq i \leq n$, the atlases $\mathcal{A}' \equiv \{\mathbf{I}'_{a_1}, \cdots, \mathbf{I}'_{a_t}\}$ computed for $\mathcal{C}'$ are different from the atlases in $\mathcal{A}$ up to rotations, i.e., $\mathbf{I}'_{a_j} = r_j(\mathbf{I}_{a_j})$ for $r_j \in \mathbf{SO}(d), 1 \leq j \leq t$. The rotational invariance property requires us to work not in the image space $\mathcal{I}$ but in its quotient space $\mathbf{Q}$, the quotient space of $\mathcal{I}$ by the rotation group $\mathbf{SO}(d)$ [45, 46]. The quotient space $\mathbf{Q}$ is a space that parameterizes the $\mathbf{SO}(d)$-orbits in $\mathcal{I}$ and in this work, we will assume that $\mathbf{Q}$ has a manifold structure induced from the manifold structure of $\mathcal{I}$. Let $\pi : \mathcal{I} \to \mathbf{Q}$ denote the canonical projection map that sends each image $\mathbf{I} \in \mathcal{I}$ to the unique $\mathbf{SO}(d)$-orbit $[\mathbf{I}] \in \mathbf{Q}$ containing $\mathbf{I}$: $\pi(\mathbf{I}) = [\mathbf{I}]$.

There is a well-known bijective correspondence [45] between metrics on $\mathbf{Q}$ and $\mathbf{SO}(d)$-invariant metric on $\mathcal{I}$. Recall that a $\mathbf{SO}(d)$-invariant metric on $\mathcal{I}$ satisfies the following condition

$$\mathbf{d}_{\mathcal{I}}(\mathbf{I}_1, \mathbf{I}_2) = \mathbf{d}_{\mathcal{I}}(g(\mathbf{I}_1), g(\mathbf{I}_2)),$$

for any two images $\mathbf{I}_1, \mathbf{I}_2 \in \mathcal{I}$ and $g \in \mathbf{SO}(d)$. The correspondence between metrics $\mathbf{d}_{\mathbf{Q}}$ on $\mathbf{Q}$ and $\mathbf{SO}(d)$-invariant metrics $\mathbf{d}_{\mathcal{I}}(x, y)$ on $\mathcal{I}$ is provided by the formula, [45]

$$\mathbf{d}_{\mathbf{Q}}([\mathbf{I}_1], [\mathbf{I}_2]) = \min_{x \in [\mathbf{I}_1], y \in [\mathbf{I}_2]} \mathbf{d}_{\mathcal{I}}(x, y), \tag{2–1}$$

for any two points $[\mathbf{I}_1], [\mathbf{I}_2] \in \mathbf{Q}$. Note that $[\mathbf{I}_1], [\mathbf{I}_2]$ are realized in $\mathcal{I}$ as $\mathbf{SO}(d)$-orbits, and $\mathbf{d}_{\mathbf{Q}}$ simply computes the distance between the two orbits in $\mathcal{I}$ as measured by $\mathbf{d}_{\mathcal{I}}$ in $\mathcal{I}$.

The two related metrics $\mathbf{d}_{\mathcal{I}}, \mathbf{d}_{\mathbf{Q}}$ allows us to go between $\mathcal{I}$ and $\mathbf{Q}$. In particular, any computation relating to a metric on $\mathbf{Q}$ can be equivalently formulated using its corresponding $\mathbf{SO}(d)$-invariant metric on $\mathcal{I}$. Fortunately, many metrics in $\mathcal{I}$ can be easily shown to be $\mathbf{SO}(d)$-invariant, such as the usual L^2-metric and the following metric

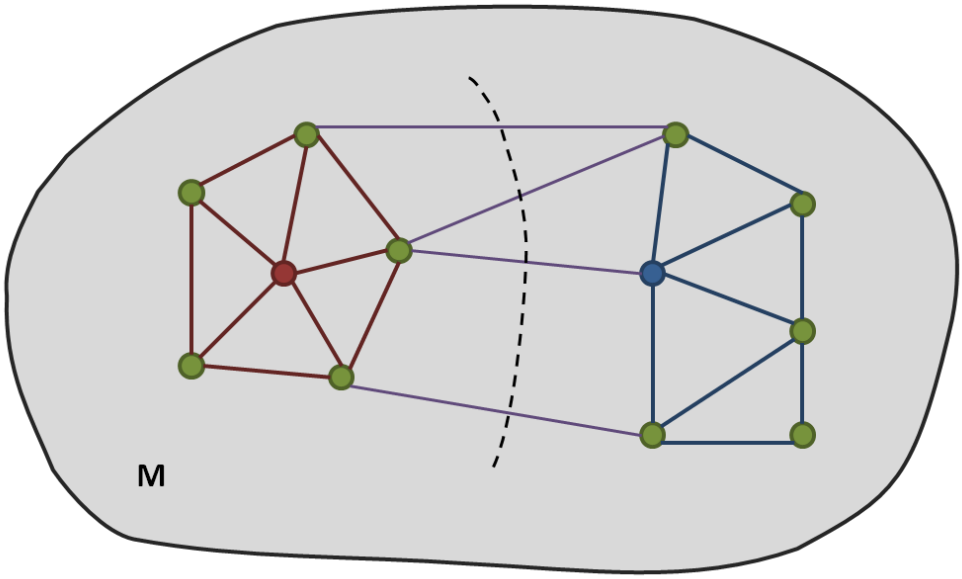


Figure 2-2. Semi-supervised graph partition. **M** is the underlying manifold. Red and blue nodes are labeled vertices for two different classes. Green nodes are unlabeled vertices. The dashed curve is the cut that generates the partition (subgraphs).

proposed in [16]:

$$\mathbf{d}_{\mathcal{I}}(\mathbf{I}_1, \mathbf{I}_2) = \min_{h_i} \left(\int_{\Omega} |\mathbf{I}_1(h_i(x)) - \mathbf{I}_2|^2 dx + \int_0^1 \int_{\Omega} \|Lv_i\|^2 dx ds \right)^{1/2} \tag{2–2}$$

where v_i is the time-dependent vector field that defines the diffeomorphic flow from the identity to the diffeomorphism h_i, and L is a second-order elliptic operator. Given the images $\mathbf{I}_i$ in $\mathcal{C}$, the manifold **M** that will be modeled pertains to the points $\{[\mathbf{I}_1], \cdots, [\mathbf{I}_n]\}$ in the quotient space **Q**. As a submanifold of **Q**, **M** is naturally equipped with the induced Riemannian metric, and in the following, we will denote $\mathbf{d}_{\mathbf{M}}([\mathbf{I}_i], [\mathbf{I}_j])$ the geodesic distance function on **M**.

2.3.2 Graph representation of the manifold M

Because the manifold **M** does not have explicit representation, we follow the common approach that uses a graph $G = (V, E)$ to characterize its manifold structure, where V, E are the node and edge sets, respectively. In this construction, the node set $V = \{[\mathbf{I}_1], \cdots, [\mathbf{I}_n]\}$ is provided by the points $[\mathbf{I}_i]$ in the quotient space **Q**. The edge set

E that defines the connectivity of G is determined by the pair-wise distances among the points $[\mathbf{I}_i]$ via the standard k-nearest neighbor (k-NN) construction. That is, an edge is formed between every node and its k nearest neighbors. A simple example is shown in Figure 2-1, and in this example, each node is connected to its two nearest neighbors ($k = 2$) and the resulting graph gives a good approximation to the underlying curve.

Because of the heterogeneity of some image sets, we partition the graph G into t subgraphs in order to compute multiple atlases to represent the image sets. Spectral clustering and related methods provide a readily available algorithms for partitioning the graph G. Popular and Well-known algorithms include Ratio Cut [28] and Normalized Cut [57], and our unsupervised graph partition algorithm based on Normalized Cut is outlined in Algorithm 1. The algorithm partitions the graph using the graph Laplacian matrix $\mathbf{L}$ that is computed from the similarity matrix $\mathbf{W}$ that encodes all the metrical information provided by the nodes $[\mathbf{I}_i]$:

$$W_{ij} = \begin{cases} \exp(-\frac{\mathrm{d}_{\mathrm{Q}}([\mathrm{I}_i],[\mathrm{I}_j])^2}{\sigma^2}) & \text{if } j \in \mathcal{N}_i \text{ or } i \in \mathcal{N}_j, \\ 0 & \text{otherwise,} \end{cases} \tag{2–3}$$

where $\mathcal{N}_i$ and $\mathcal{N}_j$ are k nearest neighborhoods. The parameter σ is empirically estimated by $\sigma = \frac{1}{n}\sum_{i=1}^{n} \mathbf{d}_{\mathrm{Q}}([\mathbf{I}_i], [\mathbf{I}_{i_k}])$ where $[\mathbf{I}_{i_k}]$ is the k-th nearest neighbor of $[\mathbf{I}_i]$. The graph Laplacian matrix is defined as

$$\mathbf{L} = \mathbf{D} - \mathbf{W} \tag{2–4}$$

where $\mathbf{D}$ is a $n \times n$ diagonal matrix with $D_{ii} = \sum_{j=1}^{n} W_{ij}$.

After the partition, we obtain t subgraphs $G_i = (V_i, E_i)$, where $V_1, \ldots, V_t$ is the partition of the set V and $E_i \subset E$ only includes edges whose two vertices are in V_i.

The above unsupervised framework can be easily extended to a semi-supervised framework with partially labeled data. This extension is relevant because in many medical imaging applications, a small number of data labeled by experts are usually available, and they should be utilized to improve the analysis and classification of the

Algorithm 1 Unsupervised graph partition based on spectral clustering

1: Compute similarity matrix $\mathbf{W}$ and graph Laplacian matrix $\mathbf{L}$.
2: Solve the generalized eigenvector problem,

$$\mathbf{Lu} = \lambda \mathbf{Du}.$$

Let $\mathbf{u}_1, \dots, \mathbf{u}_t$ be t eigenvectors corresponding to the smallest eigenvalues and form the matrix $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_t] \in \mathbb{R}^{n \times t}$.
3: Let $\mathbf{y}_i \in \mathbb{R}^t$ be the i-th row of $\mathbf{U}$, where $i = 1, \dots, n$ and cluster the points $\mathbf{y}_1, \cdots, \mathbf{y}_n$ into t clusters using K-means algorithm.

remaining unlabeled data. Inspired by the ideas from semi-supervised learning [7, 88], we consider the graph partition problem with partially labeled nodes as a classification problem that transfers the known labels to the unlabeled data. A two-class example is shown in Figure 2-2. Red and blue nodes are labeled vertices for two different classes. Green nodes are unlabeled vertices which will be classified using transductive learning algorithms that gives the two subgraphs with red and blue nodes and their edges, respectively.

We will consider the two-class partition problem first with the label set $Y = \{-1, 1\}$. The classifier is defined as a function: $f : \mathbf{M} \to Y$. Because any function $f \in L^2(\mathbf{M})$ can be written in terms of eigenfunctions of the Laplacian, $f(x) = \sum_{i=1}^{\infty} a_i u_i(x)$, where u_i are eigenfunctions: $\Delta u_i = \lambda_i u_i$ and Δ is the Laplace-Beltrami operator. Thus we could approximate the classifier on the basis of labeled data and use it to classify unlabeled data. Let the sample set be $\mathcal{X} = \{x_1, \dots, x_s, x_{s+1}, \dots, x_n\} \subset \mathbf{M}$. Without loss of generality, we assume the first s data are labeled and others are unlabeled. The label set is $\{c_1, \dots, c_s\}$, where $c_i \in Y$. Given the graph Laplacian $\mathbf{L}$, we need to solve the eigenvector problem $\mathbf{Lu} = \lambda \mathbf{u}$. Let $\mathbf{u}_1, \dots, \mathbf{u}_p$ be eigenvectors corresponding to the p smallest eigenvalues. $\mathbf{u}_i = [u_{i1}, \dots, u_{in}]^T$, where $i = 1, \dots, p$. Then we compute the approximated classifier from the labeled data by minimizing the following cost function

$$\mathcal{E}(\mathbf{a}) = \sum_{i=1}^{s} \left(c_i - \sum_{j=1}^{p} a_j u_{ji} \right)^2 = \|\mathbf{Ua} - \mathbf{c}\|^2, \tag{2–5}$$

where the coefficients $\mathbf{a} = [a_1, \dots, a_p]^T$, the labels $\mathbf{c} = [c_1, \dots, c_s]^T$ and the basis matrix $\mathbf{U} = \begin{pmatrix} u_{11} & \cdots & u_{p1} \\ \vdots & \ddots & \vdots \\ u_{1s} & \cdots & u_{ps} \end{pmatrix}$. The closed-form solution of this optimization problem is $\mathbf{a} = (\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T\mathbf{c}$. For the unlabeled data x_i, $i > s$, we set the label to be 1 if $\sum_{j=1}^{p} a_j u_{ji} \geq 0$, otherwise the label is set to -1. For t-classes graph partition, $t > 2$, we follow the *one-versus-the-rest* approach which is commonly used in multiclass support vector machines [72]. We build t separate two-class classifiers f_k, where $k = 1, \dots, t$. The k-th classifier is estimated by using the data from class-k as the positive samples and the data from remaining $t - 1$ classes as the negative samples. Then t coefficients $\mathbf{a}_1, \dots, \mathbf{a}_t$ are computed and will be used to classify unlabeled data, where $\mathbf{a}_k = [a_{k1}, \dots, a_{kp}]^T$. For x_i, $i > s$, the label is set to be $\arg\max_k f_k(x_i) = \sum_{j=1}^{p} a_{kj} u_{ji}$. This transductive graph partition algorithm is shown in Algorithm 2.

Algorithm 2 semi-supervised graph partition

1: Compute similarity matrix **W** and graph Laplacian matrix **L** using both labeled and unlabeled data.
2: Solve the eigenvector problem,

$$\mathbf{L}\mathbf{u} = \lambda\mathbf{u}.$$

Let $\mathbf{u}_1, \dots, \mathbf{u}_p$ be p eigenvectors corresponding to the p smallest eigenvalues.
3: For t-class labeled data, we estimate t separate two-class classifiers by minimizing the following cost functions,

$$\mathcal{E}(\mathbf{a}_k) = \sum_{i=1}^{s} \left(c_i - \sum_{j=1}^{p} a_{kj} u_{ji} \right)^2$$

where $k = 1, \dots, t$.
4: We use the coefficients $\mathbf{a}_1, \dots, \mathbf{a}_t$ computed from step 3 to classify the unlabeled data. For x_i, $i > s$, the label is set to $\arg\max_k \sum_{j=1}^{p} a_{kj} u_{ji}$.

2.3.3 Estimating mean from pairwise geodesic distances

After the graph partition, we need to construct an atlas for each subgraph. Without loss of generality, we suppose $\{x_1, \cdots, x_k\}$ are nodes of one subgraph G_a on a manifold $\mathbf{M}$. Let $m \in \mathbf{M}$ denote the mean of points $\{x_1, \cdots, x_k\}$ on $\mathbf{M}$ and $\mathbf{d}_{\mathrm{M}}$ denotes

the Riemannian geodesic distance on $\mathbf{M}$. Although we do not have the analytic representation for $\mathbf{d}_{\mathrm{M}}$, we could approximate $\mathbf{d}_{\mathrm{M}}(x_i, x_j)$, $1 \leq i, j \leq k$ with $\mathbf{d}_{\mathrm{Q}}$ and the graph structure of G_a. For each edge $(x_p, x_q) \in E_a$, let the distance $\mathbf{d}_{\mathrm{Q}}(x_p, x_q)$ be its weight. Since a geodesic is the shortest path between two points on the manifold, we could use Dijkstra's algorithm or Floyd-Warshall algorithm [15] to find the shortest path between two nodes x_i, x_j on the graph and use it to approximate the geodesic on the manifold $\mathbf{M}$. One possible way to determine the mean m is the following. Let $a_i = \mathbf{d}_{\mathrm{M}}(x_i, m), 1 \leq i \leq k$. Determining a_i is of course equivalent to locating m on $\mathbf{M}$, and a_i can be determined as the solution to an optimization problem given by:

$$
\begin{aligned}
\min \quad & a_1^2 + a_2^2 + \cdots + a_k^2 \\
\mathtt{s.t.} \quad & a_i \geq 0, \ i = 1, \cdots k \\
& a_i + a_j \geq m_{ij}, \\
& a_i + m_{ij} \geq a_j, \\
& a_j + m_{ij} \geq a_i, i, j = 1, \cdots k \\
& \mathtt{HDC}_{\mathrm{M}}(a_i, m_{ij}),
\end{aligned} \tag{2–6}
$$

where $m_{ij} = \mathbf{d}_{\mathrm{M}}(x_i, x_j)$ and $\mathtt{HDC}_{\mathrm{M}}(a_i, m_{ij})$ denote the remaining higher-degree constraints among a_i, m_{ij}. The collection of constraints in $\mathtt{HDC}_{\mathrm{M}}$ depends on $\mathbf{M}$ as a metric space and in general, for two different manifolds, their corresponding $\mathtt{HDC}$ will be different. The linear inequality constraints between a_i and m_{ij} are imposed because $\mathbf{d}_{\mathrm{M}}$ satisfies the triangle inequality.

For our problem, the manifold $\mathbf{M}$ and its metric $\mathbf{d}_{\mathrm{M}}$ are unknown, and consequently, the constraints in $\mathtt{HDC}_{\mathrm{M}}$ cannot be known. However, as an approximation, we solve the following optimization problem in $\mathbb{R}^k$ and use the solution $a_1, \cdots, a_k$ as an approximation

to the solution of the above problem:

$$\begin{aligned} \min \quad & a_1^2 + a_2^2 + \cdots + a_k^2 \\ \text{s.t.} \quad & a_i \geq 0, \ i = 1, \cdots k \\ & a_i + a_j \geq m_{ij}, \\ & a_i + m_{ij} \geq a_j, \\ & a_j + m_{ij} \geq a_i, i, j = 1, \cdots k \end{aligned} \tag{2–7}$$

The optimization problem is clearly convex since the objective function is strictly convex (the Hessian is positive-definite everywhere) and the domain, which is the intersection of half-spaces, is also convex. This particular type of optimization problem (with quadratic cost function and linear inequality constraints) can be solved efficiently even with a large number of variables and constraints. And because the objective function is strictly convex, the solution is unique [8]. Furthermore, the solution is stable with respect to the input parameters m_{ij} in the sense that small perturbations of m_{ij} will not significantly alter the solution $a_1, \cdots, a_k$ [8]. For computing robust $\mathbf{L}^1$-median [22] instead of mean, we simply need to change the cost function to $a_1 + a_2 + \cdots + a_k$.

2.3.4 Computing the image atlases

The result of the previous localization step provides us with the pairwise distances $\mathbf{d}_{\mathrm{M}}(m, [\mathbf{I}_i])$, and at this point, the atlas has been located only in the abstract manifold $\mathbf{M}$ in terms of these distances. In the realization step, we will render the atlas image using the distance data $\mathbf{d}_{\mathrm{M}}(m, [\mathbf{I}_i])$ and the input images. Specifically, given the vertex set $\{[\mathbf{I}_1], \cdots, [\mathbf{I}_k]\}$ of subgraph G_a and the corresponding non-negative numbers a_i as estimates on the geodesic distances $\mathbf{d}_{\mathrm{M}}(m, [\mathbf{I}_i])$, where $i = 1, \ldots, k$, we realize the image atlas using a weighted group-wise registration approach. Because the closer a sample image to the atlas on the manifold $\mathbf{M}$, the more contribution it will give in atlas

realization, the weight is in the form of an exponential function $\exp(-a_i^2/\sigma^2)$. Since the weight will be close to 0 if a_i is large, we determine a small number of points in $\{[\mathbf{I}_1], \cdots, [\mathbf{I}_k]\}$ that are close to m as measured by $\mathbf{d}_{\mathrm{M}}$, and m is then approximated from these points with respect to the metric $\mathbf{d}_{\mathrm{Q}}$ in $\mathbf{Q}$. In practice, a positive integer K is specified and K points $[\mathbf{I}_{i_1}], \cdots, [\mathbf{I}_{i_K}]$ with shortest distances to m are chosen. We require that $K \geq 1+$ dimension of the manifold.

We could approximate m with a weighted mean of the K points $[\mathbf{I}_{i_j}]$, with weights $w_j, 1 \leq j \leq K$:

$$[\mathbf{I}] = \underset{[\mathrm{I}']\in \mathrm{Q}}{\operatorname{argmin}} \sum_{j=1}^{K} w_j \mathbf{d}_{\mathrm{Q}}^2([\mathbf{I}_{i_j}], [\mathbf{I}'])$$

The weights w_j can be constructed from the estimated geodesic distances a_{i_j}:

$$w_j = b_j / \left(\sum_{k=1}^{K} b_k \right)$$

where $b_j = \exp(-a_{i_j}^2/\sigma^2), 1 \leq j \leq K$, for some $\sigma > 0$. Using the metric defined by Equation 2–2, the corresponding atlas $\mathbf{I}$ is computed by solving the variational problem

$$\min_{\tilde{\mathrm{I}}, h_j, s_j} \sum_{j=1}^{K} w_j \int_{\Omega} |\mathbf{I}_{i_j}(h_j s_j(x)) - \tilde{\mathbf{I}}|^2 dx + \int_0^1 \int_{\Omega} \|Lv_j\|^2 dx ds. \tag{2–8}$$

Overall, our multiple atlases construction approach is outlined in Algorithm 3.

Algorithm 3 Multiple atlases construction on image manifolds

1: Construct k-NN graph for image data on the manifold $\mathbf{M}$ and use Algorithm 1 to partition the graph into t subgraphs. If there are labeled data, apply the semi-supervised graph partition discussed in Algorithm 2.
2: For each of t subgraphs,

- locate the mean by solving the optimization problem in equation 2–7.
- realize the image atlas by the weighted group-wise registration.

2.4 Experiments

In this section, we present the experimental results of our method against traditional atlas construction techniques on synthetic and real image data. The traditional methods

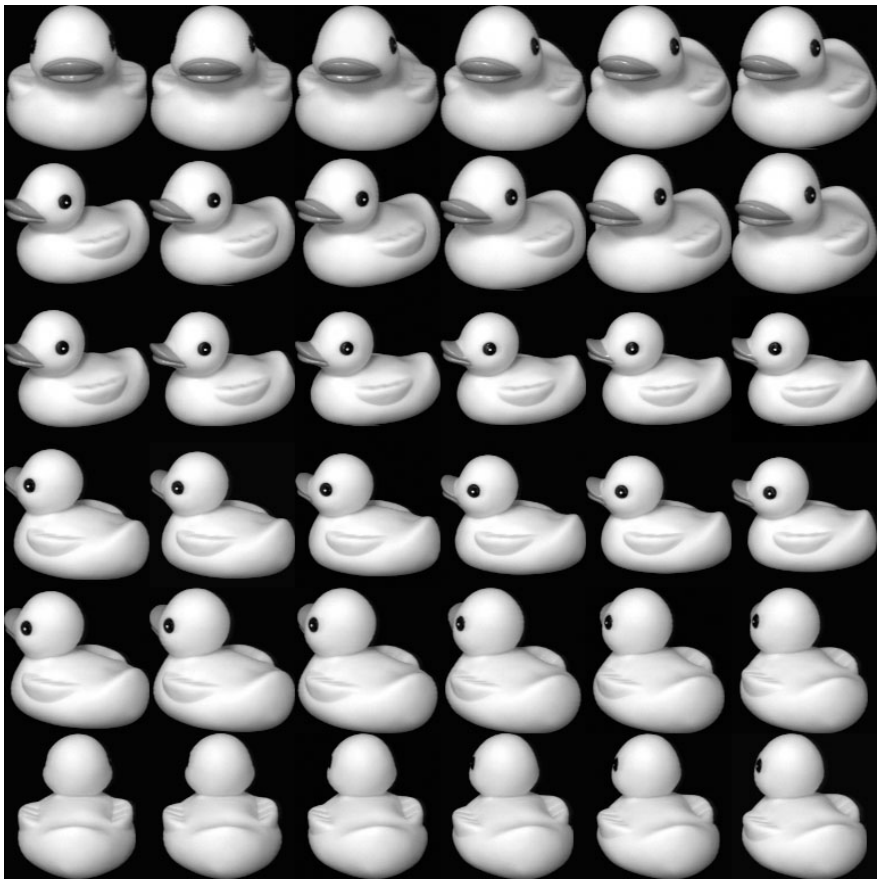


Figure 2-3. Samples of input duck images.

used as comparisons in following experiments are non-rigid group-wise registrations [22, 32, 54].

2.4.1 Experimental Results on COIL-20

The first experiment is performed on Columbia Object Image Library (COIL-20)[47] dataset. These images were taken by rotating each object through 360 degrees with respect to a fixed camera. Since in COIL-20 the underlying manifold of 72 images for each object is one-dimensional, this database is commonly used to test manifold learning algorithms and is appropriate to show the advantages of our atlas (mean) image construction method. In this experiment, the nearest neighbor parameter K is set to 4.

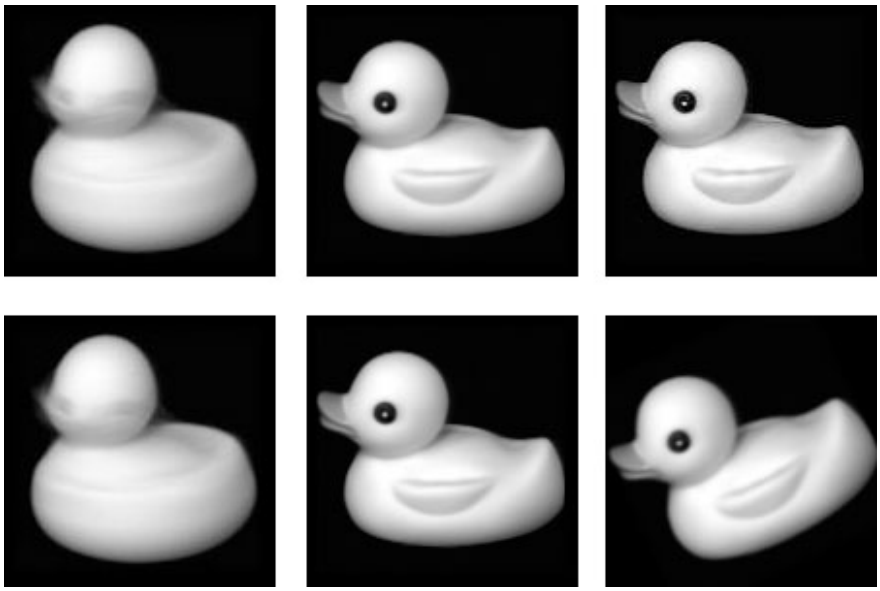


Figure 2-4. **First Row**: Karcher mean obtained by the competing method (left) and our algorithm (middle). The right image represents the ground truth. **Second Row**: Median image obtained by the competing method (left) and our algorithm (middle). The right image represents our result for pre-processed data by rotating images with different angles.

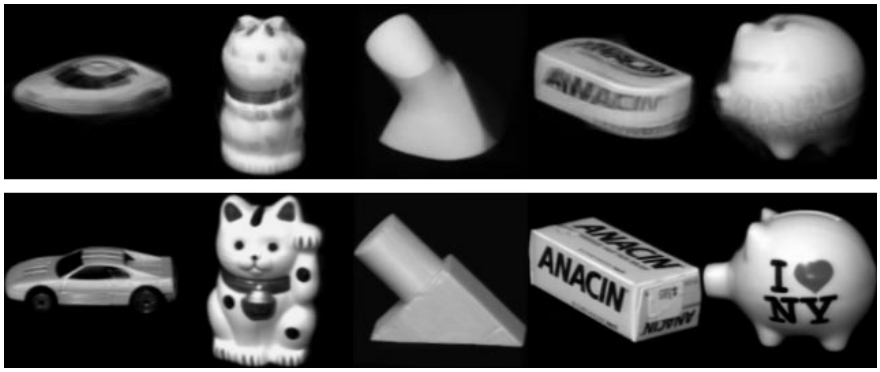


Figure 2-5. Experimental results for five objects from COIL-20. The images in the **top row** are the means generated by traditional method. Our results are shown in the **bottom row**.

We choose the 'duck' image as an example to start our discussion. 36 images at poses from 5 degrees to 90 degrees and from -5 degrees to -90 degrees are chosen to be the input of our experiment. Some smaples of input images are shown in Figure 2-3. The Karcher means obtained by the competing non-rigid group-wise registration method and our algorithm are shown in the first row of Figure 2-4. Since input images are generated by rotating the object with respect to one axis, it is reasonable to use the duck image at 0 degree pose as ground truth atlas that is shown in Figure 2-4.
It is obvious that our result is closer to the ground truth than the atlas (mean) image generated by the competing method. Because the competing method does not respect the manifold on which the data is resident, it is not surprising that its results are blurry and lose substantial details of the input duck. In comparison to the robust median images (second row of Figure 2-4), our method also yields a better result. Five more results for COIL-20 objects are presented in Figure 2-5.

Because our method is based on orbit space which quotients out the rotation, if the input images are rotated by some degrees, their projections on orbit space are invariant and the new mean would be on the same orbit of the old one, which means the two means would be different only by a rotation. We construct a new image set by rotating each input image a different angle from 20 degrees to 30 degrees. As shown in Figure 2-4, the mean for this new image set computed by our method satisfies the property mentioned above.

2.4.2 Experimental Results on Synthetic Images

In the second experiment, we synthetically generate 56 images from bunny mesh model. The model is centered at the origin and we sample 56 locations on a sphere surrounding the model. These 56 locations are (approximately) uniformly sampled with respect to a cental (apex) location ([0 0 1]), with maximum angular deviation from the central location of 30-degree. In all the renderings, we assume simple Lambertian reflectance model with the distant lighting direction parallel to the viewpoint (camera

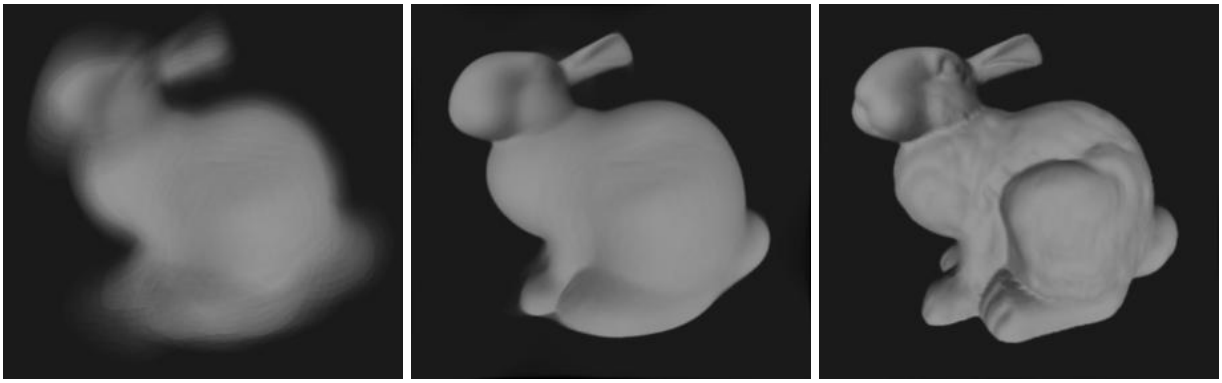


Figure 2-6. Left to right: Straightforward Averaging, Unbiased Diffeomorphic Atlas and Our Atlas.

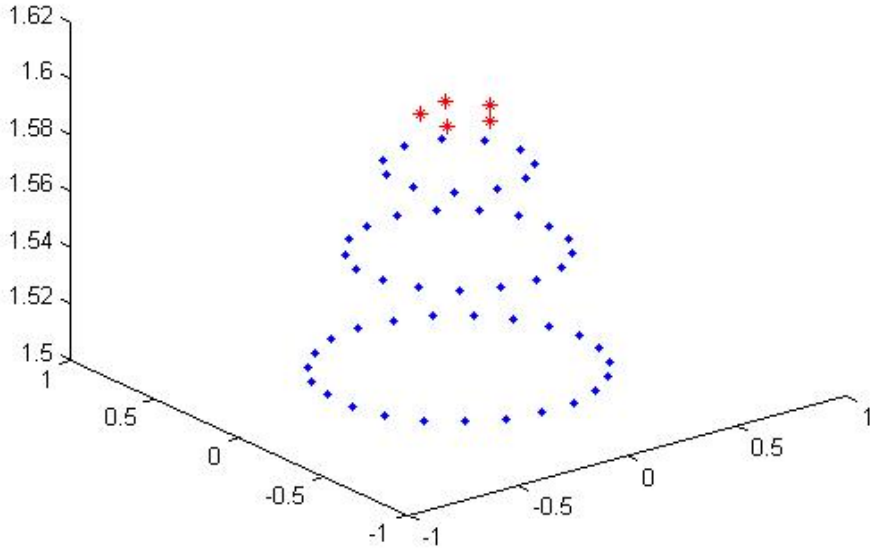


Figure 2-7. Camera positions of input images. Red points represent mean image's neighbors computed by our method

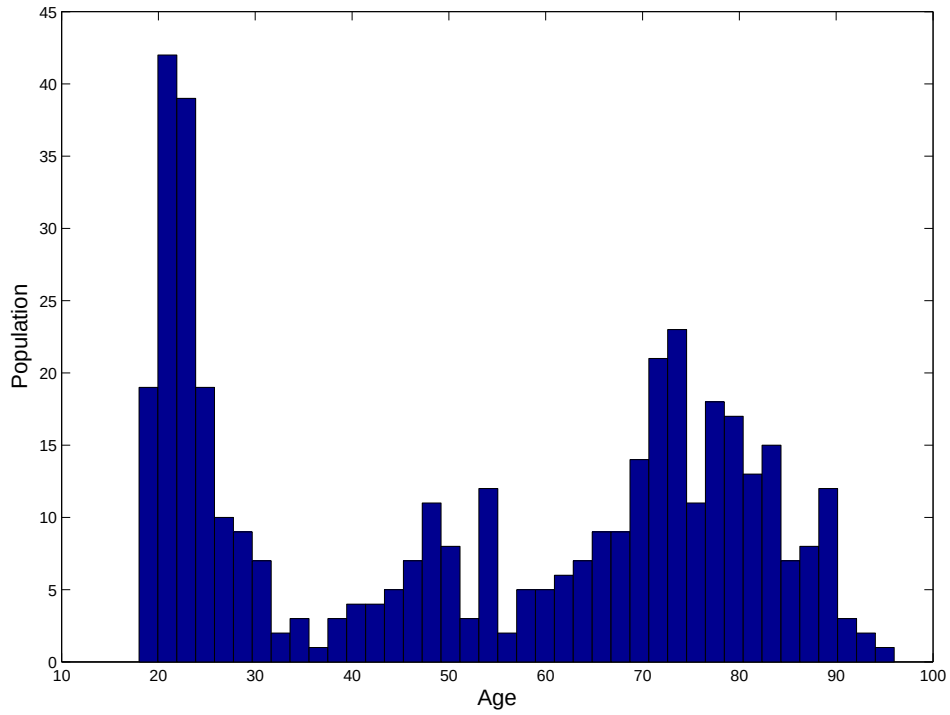


Figure 2-8. The histogram of age for all subjects in OASIS data set. 40 bins are used.

location). We choose a value of 5 for the parameter K in this experiment. Figure 2-6 shows the atlas constructed by our method and the two comparisons (straightforward averaging and the unbiased diffeomorphic atlas). Note that outputs from the two competing methods are not satisfactory as many details and features are lost or smoothed out.

The 56 camera locations on the sphere are shown as blue points in the Figure 2-7. The red points represent the five neighbors of mean computed by our algorithm. From the results depicted here, it is evident that the our atlas is close to the expected mean for this collection of points on the sphere.

2.4.3 Multiple atlases for OASIS data

In this section, we apply the proposed algorithms to construct atlases from collections of MR images. Specifically, the image data used in our experiment are the MR images from the freely available Open Access Series of Imaging Studies (OASIS)

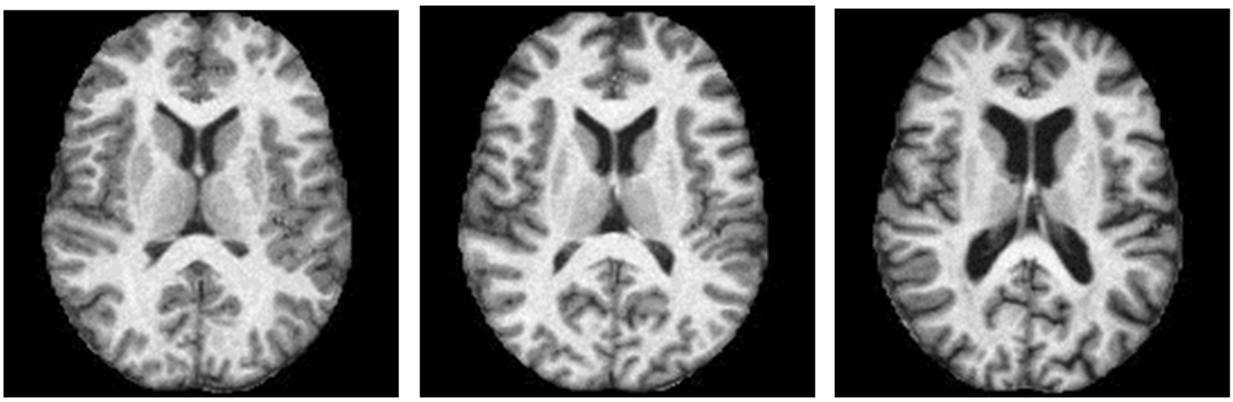
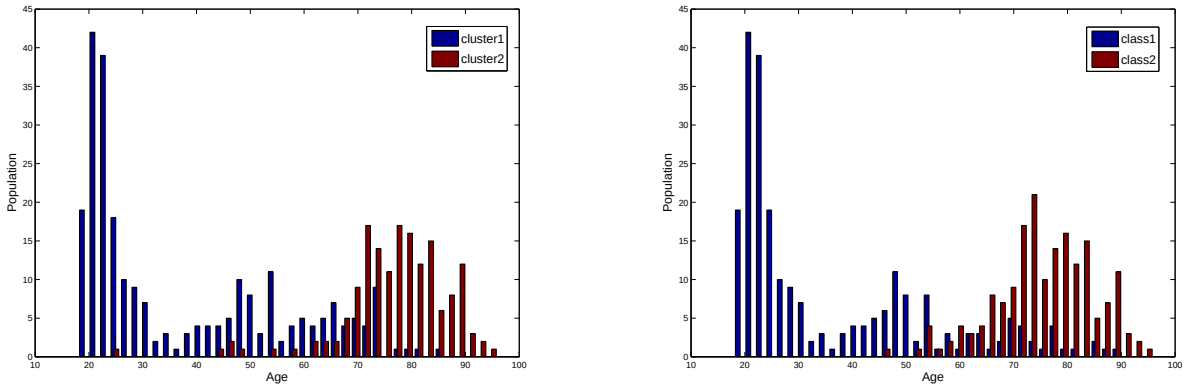


Figure 2-9. Sample images from youth, middle-aged and elderly subjects.

data set [43]. OASIS contains T1 weighted MR brain images from a cross-sectional population of 416 subjects. Each MRI scan has a resolution of $176 \times 208 \times 176$ voxels. The ages of the subjects range from 18 to 96. The histogram of age for all OASIS subjects is shown in Figure 2-8.

Because the structural difference in the brain across different age groups can be significant [44], the OASIS data set is heterogeneous. The axial views of three sample subjects are shown in Figure 2-9. Therefore, it's better to construct multiple atlases instead of a single mean template image to represent the OASIS data.

We at first construct two atlases for the collection of 416 subjects in OASIS set. Both unsupervised and semi-supervised methods are used. In all the experiments, we set $k = 10$ when constructing k-NN graphs. For the unsupervised graph partition, we use two clusters. For the semi-supervised graph partition, we randomly choose 20 subjects younger than 50 years old and 20 subjects older than or equal to 50 as the labeled samples. Other subjects are considered as unlabeled samples. The histograms of the partition results using both methods are displayed in Figure 2-10. Table 2-1 compares the means and standard deviations of ages of subjects in both classes computed by EM-based iCluster algorithm [55], our unsupervised and semi-supervised methods. The clustering result of our unsupervised method is similar to iCluster, while our semi-supervised method gives slightly smaller mean ages of both young and old



(A) (B)

Figure 2-10. Two classes partition. **A)** unsupervised method. **B)** semi-supervised method.

groups. The partial labels in semi-supervised learning provide useful information to classify more subjects aged between 50 and 70 into the older group. As shown in Figure 2-11, the atlases generated by iCluster are blurry, while the atlases computed by our methods are substantially sharper with more structural details. This is not surprising because our atlases are computed as means on the manifolds, instead of the means in the ambient space. Thus, our atlases retain important brain structures and they genuinely look like real brains when compared with the real samples in Figure 2-9.

Table 2-1. Means and standard deviations of subject ages for two classes partition

	iCluster	unsupervised	semi-supervised
Class 1	39.1±19.9	37.3±18.7	33.7±17.0
Class 2	77.8±9.3	77.0±9.9	75.3±9.6

Figure 2-8 also tells us that there is a cluster for middle aged subjects in OASIS cross-sectional set. Thus in our second experiment, we choose three clusters for our unsupervised atlases construction algorithm. For the semi-supervised algorithm, we divide the OASIS population into three groups based on the histogram of ages shown in Figure 2-8: young (subjects younger than 40), middle-aged (between 40 and 60) and older adults (older than 60) at first. Then we randomly sample 20 subjects from each group as the labeled data and consider the remainder unlabeled. The

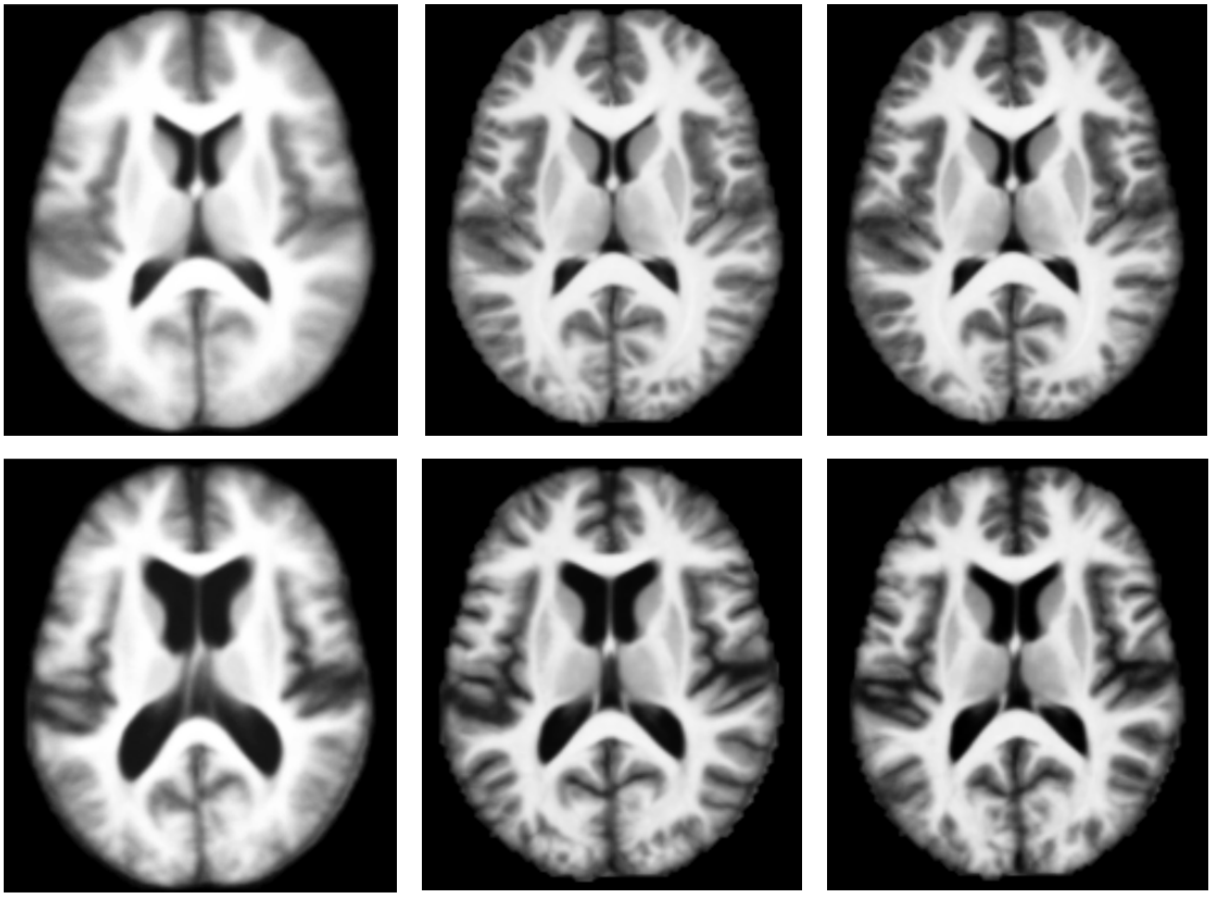
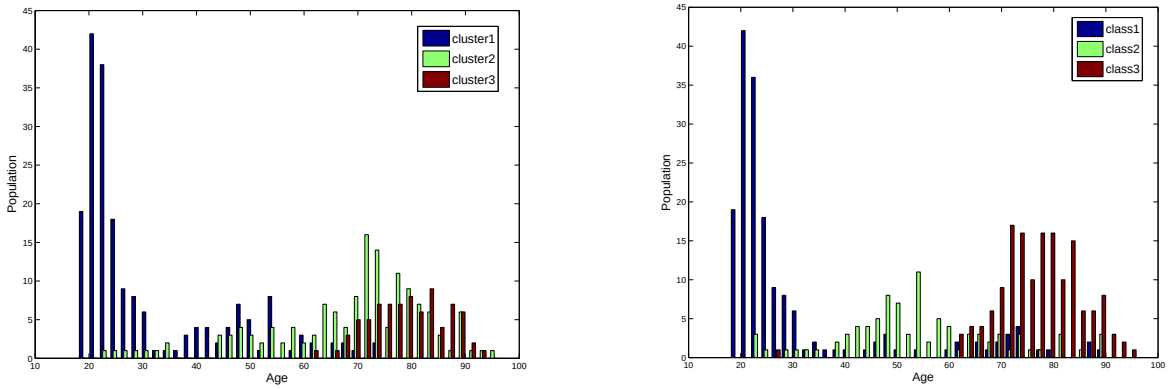


Figure 2-11. Two atlases for OASIS data. **First row**: atlas for young subjects. **Second row**: atlas for older subjects. **A)** atlases using iCluster. **B)** atlases using our unsupervised algorithm. **C)** atlases using our semi-supervised algorithm.

histograms of both unsupervised and semi-supervised partition results are shown in Figure 2-12. For our unsupervised method, the middle aged cluster has a significant overlap in age with the older cluster. A similar result was reported in [55]. These clustering methods cannot simply find the middle age mode in the age distribution, because they can only discover the dominant structural modes and there is an obvious structural difference between old (around 60) and elderly (around 75) subjects. Since the semi-supervised method uses more information from partially labeled data, it gives a better classification for different modes in age distribution. Table 2-2 compares the means and standard deviations of ages of subjects in all three classes computed by



(A) (B)

Figure 2-12. Three classes partition. **A)** unsupervised method. **B)** semi-supervised method.

iCluster, our unsupervised and semi-supervised methods, and Figure 2-13 shows the corresponding atlases constructed by these three algorithms. The clustering result of our unsupervised method is similar to iCluster, but our method builds much sharper atlases, which are very useful in applications such as atlas based segmentation and tensor based morphometry. The mean age of class 2 returned by our semi-supervised partition method is 54.6, which is more appropriate for the middle aged mode in the age distribution, compared with corresponding mean ages returned by iCluster and our unsupervised methods.

Table 2-2. Means and standard deviations of subject ages for three classes partition

	iCluster	unsupervised	semi-supervised
Class 1	31.2±14.5	30.3±13.9	30.0±16.9
Class 2	68.9±13.6	67.1±15.2	54.6±14.9
Class 3	79.6±7.5	79.6±7.3	77.1±8.5

With the age information for some subjects, our semi-supervised method has ability to construct more than three atlases across ages, which is akin to the population shape regression [16]. The regression method requires ages of all subjects in the data set, while our method just needs class labels for small amount of images. The OASIS data can be partitioned into seven classes with 10 years interval. We randomly choose 10

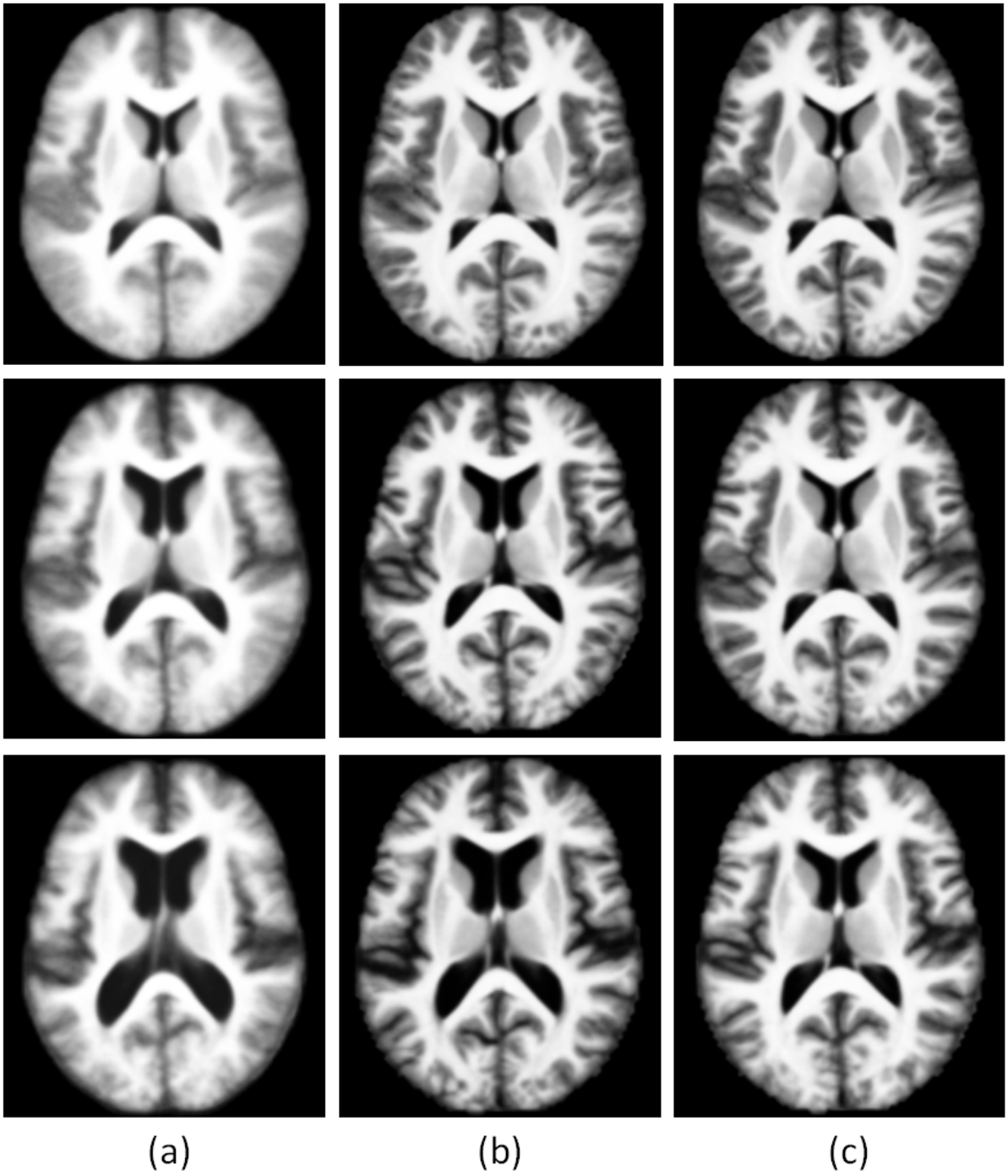


Figure 2-13. Three atlases for OASIS data. **First row**: atlas for young subjects. **Second row**: atlas for middle aged subjects. **Third row**: atlas for elderly subjects. **A)** atlases by iCluster. **B)** atlases by our unsupervised algorithm. **C)** atlases by our semi-supervised algorithm.

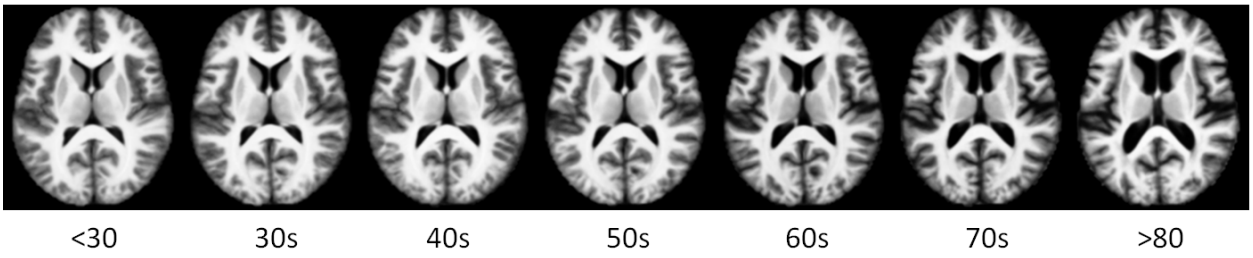


Figure 2-14. Atlases across different ages for subjects in OASIS data set.

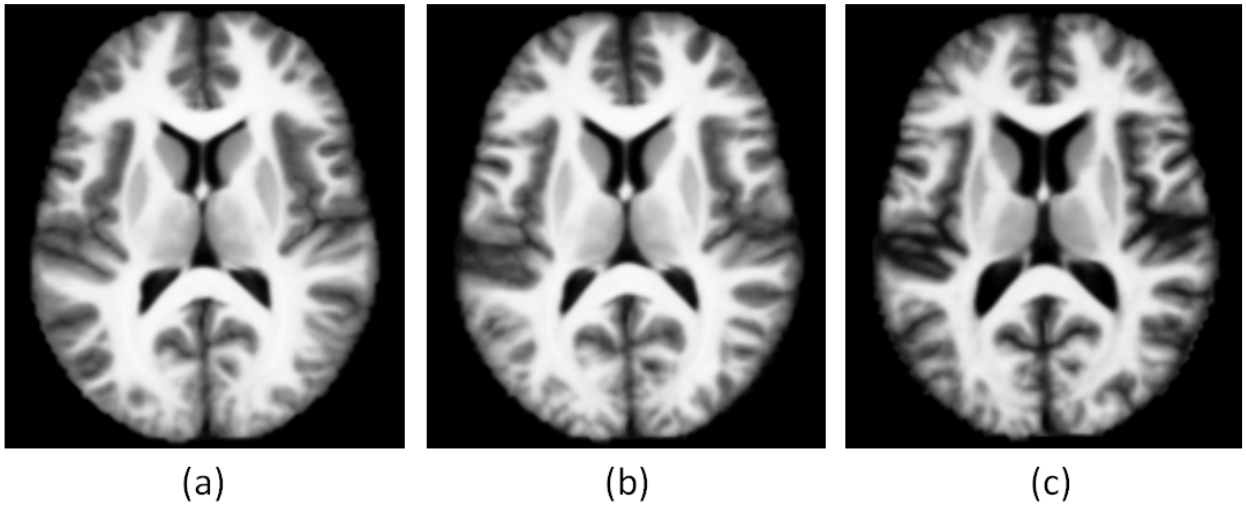


Figure 2-15. Atlases generated by our semi-supervised algorithm for female subjects in OASIS data set. **A)** atlas for young subjects. **B)** atlas for middle aged subjects. **C)** atlas for elderly subjects.

subjects from each class as the labeled data and consider the remainder unlabeled. Note that only 16.8% of images are labeled. Our atlases across different ages for OASIS data are shown in Figure 2-14. It's obvious that the ventricles enlarge with aging and the expansion is accelerated after 60. This agrees with the findings reported in [44] and [16].

2.4.4 Atlases for male and female subjects

The OASIS data set includes 160 male subjects and 256 female subjects. In this experiment, we build three atlases for both sex groups using our semi-supervised algorithm. 15 subjects from each age class are used as labeled data. Our brain atlases for female are shown in Figure 2-15 and brain atlases for male are shown in Figure 2-16. Since OASIS cross-sectional data set provides the segmentations of cerebrospinal

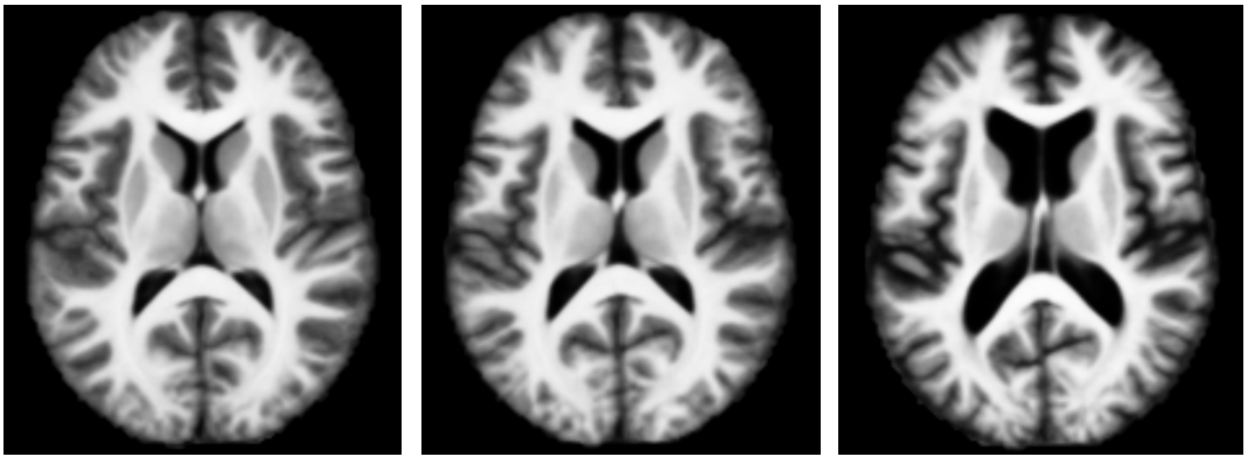


Figure 2-16. Atlases generated by our semi-supervised algorithm for male subjects in OASIS data set. **A)** atlas for young subjects. **B)** atlas for middle aged subjects. **C)** atlas for elderly subjects.

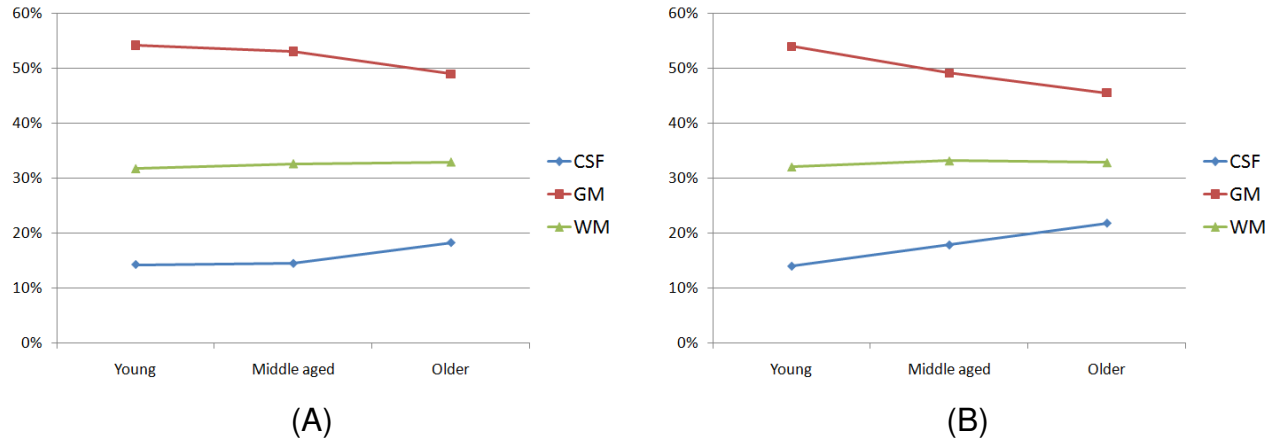


Figure 2-17. Volume ratios of CSF, GM and WM in atlases for **A)** female subjects and **B)** male subjects.

fluid (CSF), gray matter (GM) and white matter (WM) for each image, a brain tissue classification map could be generated for each atlas during the atlas construction process. The volume percentage of CSF, GM and WM for female and male atlases are plotted in Figure 2-17. Males and females appear to have the similar volume ratio changes. The percentages of CSF increase with age and the percentages of GM decrease with age. There are no significant changes for the ratio of WM.

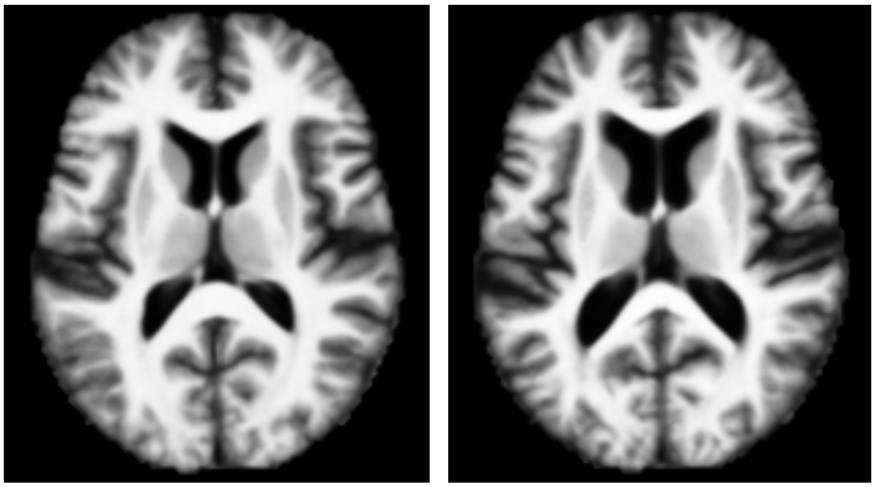


Figure 2-18. Axial view of atlases for **A)** old subjects and **B)** dementia patients.

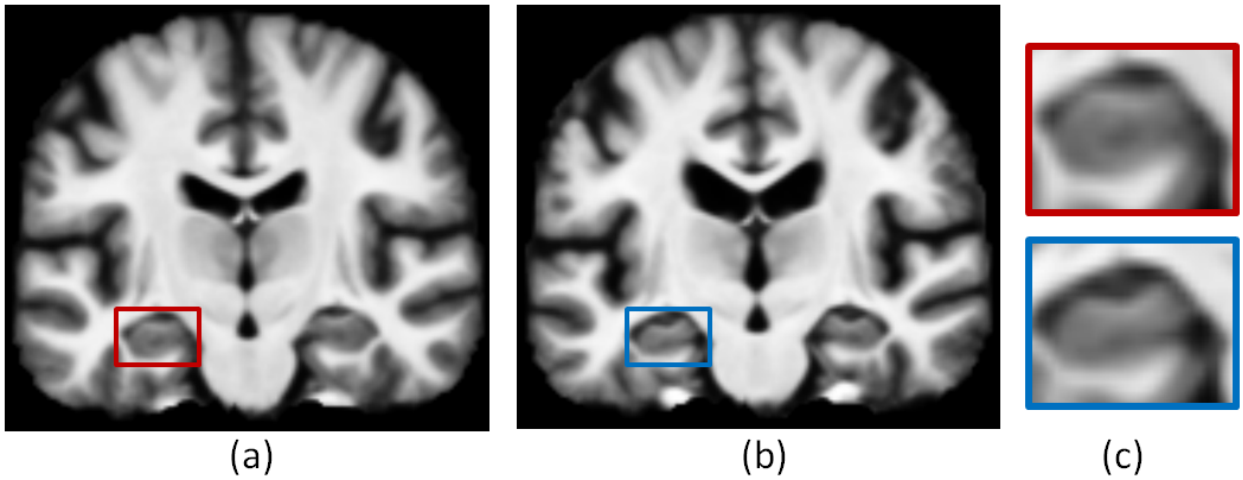


Figure 2-19. Coronal view of atlases for **A)** old subjects and **B)** dementia patients. **C)** the zoom in of hippocampus regions.

2.4.5 Atlases for dementia patients and healthy elderly subjects

The OASIS data set also includes one hundred subjects who have been clinically diagnosed with very mild to moderate Alzheimer's disease (AD) using the Clinical Dementia Rating (CDR) scale. Since all the AD subjects in OASIS data set are older than 60, to make a comparison of the normal aged brain and the AD patient's brain, we choose 98 healthy subjects (CDR score is zero) aged 60 or older as the normal group. Then we build atlases separately for AD subjects and healthy elder subjects. The axial

views of our atlases are shown in Figure 2-18 and the corresponding coronal views are shown in Figure 2-19. We can find that the AD atlas has severely enlarged ventricles and shrinkage of hippocampus, compared with the atlas for the non-demented old subjects.

CHAPTER 3
A GEOMETRIC FRAMEWORK FOR TENSOR FIELDS ANALYSIS

3.1 Motivation and Related Work

In Chapter 2, we construct a few mean templates to represent a collection of images. Methods considering higher order statistics such as variance are usually used to improve the group-wise image analysis. Diffusion Tensor Imaging (DTI) introduced by Basser et al. [4] allows one to measure the diffusion of water molecules in tissue, which is modeled using a second order tensor within each voxel. The local patterns of diffusion are useful to reveal microscopic structural properties of the measured underlying tissue, which may not be visible in other imaging modalities such as anatomical MRI and CT. Because DT images, commonly represented as fields of symmetric positive-definite matrices (SPDs), have special geometric structures, many traditional methods in medical image analysis have been extended to this new tensor valued data form. Examples include diffusion tensor field denoising [51], DTI segmentation [37, 75] and DTI registration [85, 87]. Tensor field data sets are also commonly encountered in Tensor-Based Morphometry (TBM) [38, 39]. Deformation field is obtained by warping each subject to the template using non-rigid image registration techniques. Then the Jacobian is used to construct a deformation tensor in each voxel. Voxel-based statistical methods are commonly used for longitudinal study and comparison between healthy and diseased subjects.

Because of the cone structure of the symmetric positive-definite matrix space, the Riemannian geometry of the SPD tensors and its applications to medical image analysis problems that require statistical analysis of ensembles of SPD matrices, have been the focus of intensive study in the past several years, e.g. [23, 51, 56]. In [51], Pennec et al. developed a Riemannian framework for computing statistics on SPD tensors. A Riemannian metric with superior properties over Euclidean metric was introduced. In [56], Schwartzman discussed the geometry of positive-definite matrices and studied

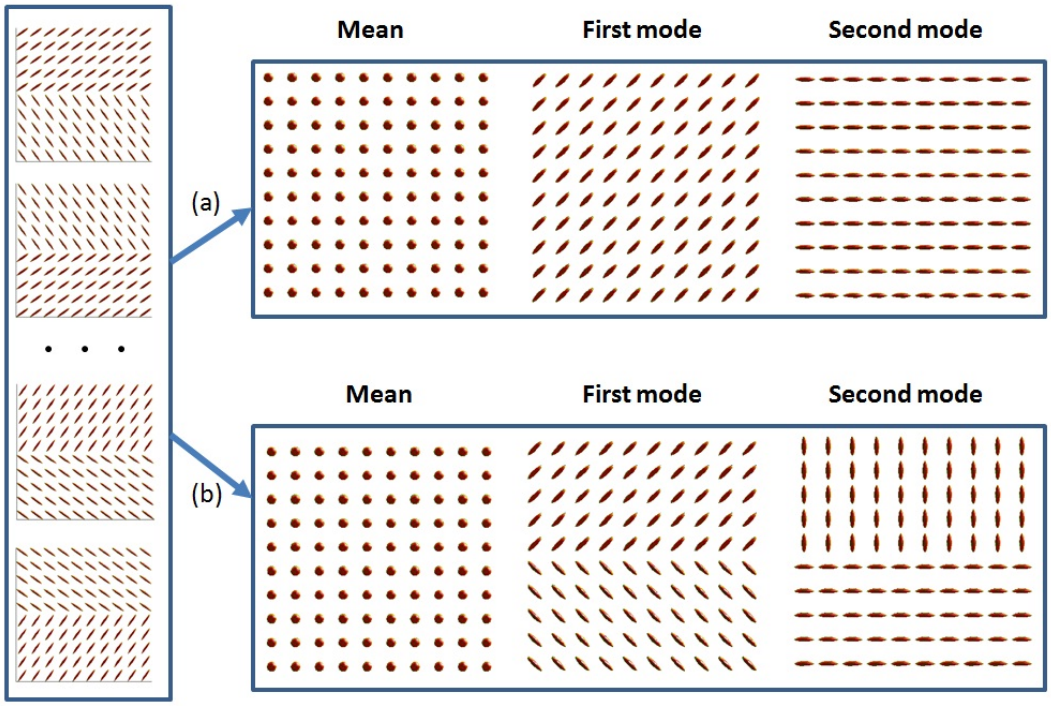


Figure 3-1. **Left column**: Eight input tensor fields. **A)** Mean, first mode and second mode of tensor fields computed using a pixel-based approach: means and modes are determined at each pixel independently using eight tensors. Note that the two modes are constant fields. **B)** Mean, first mode and second mode of tensor fields computed using the proposed method.

probability distributions defined on SPD matrices. Principal Geodesic Analysis (PGA), as a generalization of Principal Component Analysis (PCA) for data on a Riemannian manifold, was introduced in [24]. In [23], Fletcher and Joshi described PGA-based methods for statistical analysis of diffusion tensors, in the context of computing intrinsic mean of a collection of SPD matrices and characterizing their variance.

We remark that all these earlier works have focused on the statistical analysis of symmetric positive-definite matrices, while the data obtained in medical imaging is tensor field in its entirety. There is a need to develop statistical analysis methods on a collection of tensor fields where each field is treated as a single entity, similar to the group analysis of intensity images. Although the methods for SPD matrices can be applied to study the statistics of tensor fields using a voxel-based approach where tensor fields are aligned and the statistics are gathered independently for every voxel,

it is clearly insufficient and inadequate as it fails to capture global interaction patterns in the tensor fields. This point can be best illustrated using a simple example shown in Figure 3-1. Here, we generate four pairs of 10×10 tensor fields. Each tensor field contains two constant subfields occupying the top and bottom regions of the domain. Tensor fields in each pair differ by a reflection that swaps the top and bottom regions. For this collection of eight tensor fields, pixel-wise statistics fails to capture the global patterns in the tensor fields as the first and second modes are all constant fields. This result is not surprising as a pixel-based approach only considers tensors at each location independently, and it completely ignores the possible correlation between tensors at different locations.

We propose a Riemannian framework for statistical analysis of a set of tensor fields that is capable of capturing the global correlation within the fields. Specifically, each tensor field can be represented by a point in the Riemannian product space. The Riemannian exponential map of the tensor field and the corresponding logarithmic map are defined. We also introduce an invariant Riemannian metric in the space of tensor fields. We extend intrinsic mean computation algorithm and Principal Geodesics Analysis (PGA) to Riemannian symmetric products. This provides a principled method for linearizing the problem by mapping tensor fields to a Euclidean space that is the tangent space at their intrinsic mean tensor field. The global correlations of the tensor field are then captured using Principle Component Analysis (PCA) in the tangent space, and the modes of variation for the tensor fields can be determined first in the tangent space and followed by the exponential map. Moreover, dimensionality reduction of tensor valued images is particularly important as one major difficulty of working with the space of tensor fields is its dimension. For example, the space of $100 \times 100 \times 100$ tensor-valued images has 6×10^6 dimension, and dimensionality reduction is necessary for most applications and analysis. In our PGA framework, each tensor field is recorded as just a list of real coefficients instead of a matrix valued image. Thus much less

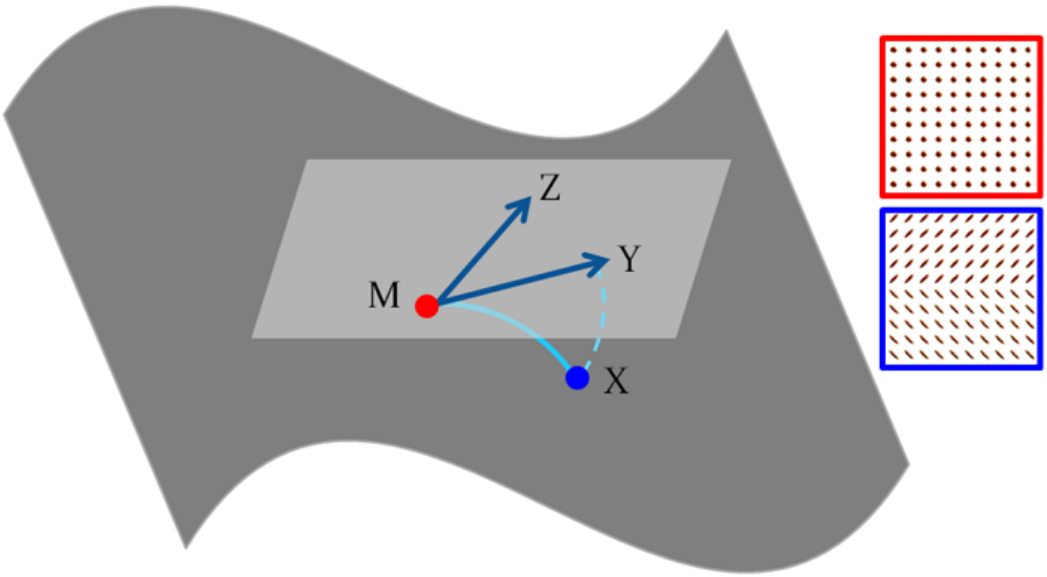


Figure 3-2. The space of tensor fields. **M** and **X** are two tensor fields. The gray plane represents the tangent space at **M**. **Y**, **Z** are two tangent vectors. The light blue curve is the geodesic between **M** and **X**.

space is taken for each tensor image. We also present a novel tensor field classification approach using the geodesic distances to submanifolds as the feature for classification. The proposed method is evaluated using the Open Access Series of Imaging Studies (OASIS) dataset and a collection of rat brain diffusion tensor images. Experimental results have demonstrated the superiority of our method compared to the voxel-based approach.

3.2 Statistical Analysis in the Space of Tensor Fields

In this section, we present the details of statistical analysis in the space of tensor fields, and we will consider only fields of symmetric positive-definite matrices. Let $P(n)$ denote the space of $n \times n$ symmetric positive-definite matrices and $Sym(n)$ denote the space of $n \times n$ symmetric matrices. A tensor field defined on a domain Ω in $\mathbb{R}^K$ is treated as a function $f : \Omega \to P(n)$. Since both diffusion tensor fields and deformation tensor fields used in medical imaging are almost always defined over a grid (pixels or voxels), Ω will be a collection of m points in $\mathbb{R}^K$, and we identify the space of tensor fields on Ω

with the product $P(n)^m = \underbrace{P(n) \times P(n) \times \cdots \times P(n)}_{m}$, which is a differentiable manifold of dimension $n(n+1)m/2$. Thus a tensor field $\mathbf{X}$ in $P(n)^m$ is represented as an m-tuple $(X_1, X_2, \ldots, X_m)$, where each X_i is a symmetric positive-definite matrix, the value of $\mathbf{X}$ at one point in Ω.

3.2.1 Geometry of Tensor Fields

The space $P(n)$ is a symmetric Riemannian manifold [23] with $GL(n)$ as the symmetry group. This can be generalized directly to product spaces $P(n)^m$ using product Riemannian structure, and in particular, the Riemannian geodesic distances, log and exponential maps have closed-form expressions. Figure 3-2 illustrates the space of SPD matrix fields. Specifically, the group $GL(n)^m$ acts transitively on $P(n)^m$ with the action $\phi : GL(n)^m \times P(n)^m \to P(n)^m$ given by $\phi_{\mathrm{G}}(\mathbf{X}) = (G_1 X_1 G_1^T, \ldots, G_m X_m G_m^T)$, where each $G_i \in GL(n)$ is a $n \times n$ invertible matrix and X_i is a $n \times n$ positive-definite matrix. Intuitively, the group action of $GL(n)^m$ can be interpreted as a change of coordinates in the product space of tensor fields $P(n)^m$. The tangent space of $P(n)^m$ at any point can be identified with $Sym(n)^m$ since the tangent space of a product manifold is the product of tangent spaces. Let $\mathbf{Y}, \mathbf{Z} \in T_{\mathrm{M}} P(n)^m$ be two tangent vectors at $\mathbf{M} \in P(n)^m$. The product Riemannian metric gives the inner product between the two vectors as

$$\langle \mathbf{Y}, \mathbf{Z} \rangle_{\mathrm{M}} = \sum_{i=1}^{m} \mathrm{tr}(Y_i M_i^{-1} Z_i M_i^{-1}). \tag{3–1}$$

This scaled inner product is invariant under the group action of $GL(n)^m$. Specifically, given the group action ϕ on $P(n)^m$, the differential map of ϕ is

$$d\phi_{\mathrm{G}}(\mathbf{Y}) = (G_1 Y_1 G_1^T, \ldots, G_m Y_m G_m^T).$$

For any $\mathbf{G} \in GL(n)^m$,

$$\langle d\phi_{\mathrm{G}}(\mathbf{Y}), d\phi_{\mathrm{G}}(\mathbf{Z}) \rangle_{\phi_{\mathrm{G}}(\mathrm{M})} = \langle \mathbf{Y}, \mathbf{Z} \rangle_{\mathrm{M}} . \tag{3–2}$$

Using this metric, the geodesic passing through $\mathbf{M}$ in the direction of $\mathbf{Y}$ is uniquely given by

$$\gamma_{\mathbf{M}}(t;\mathbf{Y}) = \left(G_1 \exp(G_1^{-1} Y_1 G_1^{-T} t) G_1^T, \ldots, G_m \exp(G_m^{-1} Y_m G_m^{-T} t) G_m^T\right) \tag{3–3}$$

where $G_i \in GL(n)$ such that $\mathbf{M} = \left(G_1 G_1^T, \ldots, G_m G_m^T\right)$. The Riemannian exponential map at $\mathbf{M}$ which maps the tangent vector $\mathbf{Y}$ to a point in $P(n)^m$ is defined as the value of the geodesic $\gamma_{\mathbf{M}}(t;\mathbf{Y})$ at $t = 1$. In our case, the closed-form representation of the Riemannian exponential map is

$$\mathrm{Exp}_{\mathbf{M}}(\mathbf{Y}) = \left(G_1 \exp(G_1^{-1} Y_1 G_1^{-T}) G_1^T, \ldots, G_m \exp(G_m^{-1} Y_m G_m^{-T}) G_m^T\right) \tag{3–4}$$

where $G_i \in GL(n)$ such that $\mathbf{M} = \left(G_1 G_1^T, \ldots, G_m G_m^T\right)$.

Its inverse map, Riemmannian logarithmic map can be easily obtained. Given $\mathbf{X} \in P(n)^m$, the log map at $\mathbf{M}$ is

$$\mathrm{Log}_{\mathbf{M}}(\mathbf{X}) = \left(G_1 \log(G_1^{-1} X_1 G_1^{-T}) G_1^T, \ldots, G_m \log(G_m^{-1} X_m G_m^{-T}) G_m^T\right). \tag{3–5}$$

Using this definition of log map in $P(n)^m$, the geodesic distance between two tensor fields $\mathbf{M}$ and $\mathbf{X}$ is computed by

$$d(\mathbf{M},\mathbf{X}) = \|\mathrm{Log}_{\mathbf{M}}(\mathbf{X})\| = \sqrt{\sum_{i=1}^{m} \mathrm{tr}\left(\log^2(G_i^{-1} X_i G_i^{-T})\right)}. \tag{3–6}$$

3.2.2 Statistics on the Space of Tensor Fields

Using the formula above for the geodesic distance, we define the intrinsic mean of a collection of N tensor fields $\mathbf{X}_1, \ldots, \mathbf{X}_N$ as the tensor field that minimizes the sum of squared geodesic distances:

$$\mathbf{M} = \arg\min_{M \in P(n)^m} \frac{1}{N} \sum_{i=1}^{N} d(M, \mathbf{X}_i)^2. \tag{3–7}$$

The sum of squares on the RHS above can be re-written as a sum over all points in Ω. This implies that the value of $\mathbf{M}(p)$ of $\mathbf{M}$ at one point $p \in \Omega$ is the usual Karcher mean

in $P(n)$ of $\mathbf{X}_1(p), \cdots, \mathbf{X}_N(p)$. In particular, because the space of SPD matrices $P(n)$ is complete and has non-positive sectional curvature, the Karcher mean is unique on $P(n)$ [23]. This shows that $\mathbf{M}$ will be unique as well, and it can be computed using the following gradient descent algorithm.

Algorithm 4 Intrinsic mean tensor field

1: Set the initial mean tensor field $\mathbf{M}_0$ as one of N input tensor fields $\mathbf{X}_i$. Set the initial step set $\tau = 1$ and iteration $k = 0$.
2: Compute the average on the tangent space at $\mathbf{M}_k$:

$$\mathbf{Y}_k = \frac{1}{N} \sum_{i=1}^{N} \mathrm{Log}_{\mathbf{M}_k} \mathbf{X}_i.$$

3: If $k > 0$ and $\|\mathbf{Y}_k\| > \|\mathbf{Y}_{k-1}\|$, adjust step size $\tau = \tau/2$, set $\mathbf{Y}_k = \mathbf{Y}_{k-1}$, $\mathbf{M}_k = \mathbf{M}_{k-1}$.
4: Project back to the space of tensor fields:

$$\mathbf{M}_{k+1} = \mathrm{Exp}_{\mathbf{M}_k}(\tau \mathbf{Y}_k).$$

5: Stopping condition: if $\|\mathbf{Y}_k\| < \epsilon$, output $\mathbf{M}_{k+1}$ as the intrinsic mean tensor field, otherwise set $k = k + 1$ and go to step 2.

After obtaining the intrinsic mean $\mathbf{M}$ of the input tensor fields $\mathbf{X}_1, \ldots, \mathbf{X}_N$, we will determine the modes of variation using PGA. Specifically, we use the log map to map all the tensor fields to the tangent space at $\mathbf{M}$, $x_i = \log_{\mathbf{M}}(X_i)$. This is a Euclidean space in which we can analyze the data points $x_1, \cdots, x_N$ using principal component analysis. We define the principal vectors $\mathbf{V}_1, \cdots, \mathbf{V}_k$ in $T_{\mathbf{M}} P(n)^m$ according to the following equations:

$$\mathbf{V}_1 = \arg \max_{\|\mathbf{V}\|=1} \sum_{i=1}^{N} \langle \mathbf{V}, \mathrm{Log}_{\mathbf{M}}(\mathbf{X}_i) \rangle_{\mathbf{M}}^2,$$

$$\mathbf{V}_k = \arg \max_{\|\mathbf{V}\|=1} \sum_{i=1}^{N} \sum_{j=1}^{k-1} \langle \mathbf{V}_j, \mathrm{Log}_{\mathbf{M}}(\mathbf{X}_i) \rangle_{\mathbf{M}}^2 + \langle \mathbf{V}, \mathrm{Log}_{\mathbf{M}}(\mathbf{X}_i) \rangle_{\mathbf{M}}^2.$$

The orthonormal vectors $\mathbf{V}_i$ spanned a K-dimensional subspace S_K that best approximates $x_1, \cdots, x_N$ in the least-squares sense, and they can be computed using PCA. By exponentiating vectors in $\mathbf{S}_K$, $\mathrm{Exp}_{\mathrm{M}}\left(\sum_{k=1}^{d} \alpha_k \mathbf{V}_k\right)$, where α_k tells the variation of k*th* mode, we obtain the geodesic submanifold $\mathbf{S}_K \subset P(n)^m$ that can serve as a good approximation of the input tensor fields.

There are two important details that differ from the usual application of PGA [23]. First, except at the identity, the inner product defined in Equation 3–1 does not correspond to the standard Euclidean inner product, which is required for the familiar PCA algorithm. Therefore, we first transform the data to the tangent space at the identity, which is accomplished via the following transform, $\mathbf{X} = (X_1, \cdots X_m) \in T_{\mathrm{M}} P(n)^m$

$$d\phi_{\mathrm{G}^{-1}}(\mathbf{X}) = \left(G_1^{-1} X_1 G_1^{-T}, \ldots, G_m^{-1} X_m G_m^{-T}\right), \tag{3–8}$$

where $\mathbf{G} = (G_1, \ldots, G_m)$ is such that $\mathbf{M} = \left(G_1 G_1^T, \ldots, G_m G_m^T\right)$. Once the data have been mapped to $T_{\mathrm{I}} P(n)^m$, we can apply the usual PCA algorithm to obtain principal vectors $\mathbf{U}_i, i = 1, \cdots, K$. They are then transformed back to $T_{\mathrm{M}} P(n)^m$ using $\mathbf{V}_i = d\phi_{\mathrm{G}}(\mathbf{U}_i)$. Due to the high-dimensionality of $P(n)^m$, we use the Gram matrix instead of the usual covariance matrix when computing the principal vectors in the tangent space, which is the approach used in many computer vision applications such as the Eigenfaces [68]. The complete algorithm is summarized in Algorithm 5.

Algorithm 5 PGA for Tensor Fields

1: **Input** N tensor fields $\mathbf{X}_1, \ldots, \mathbf{X}_N \in P(n)^m$.
2: Compute intrinsic mean $\mathbf{M}$ of input tensor fields using the method listed Algorithm 4.
3: Compute $\mathbf{Y}_i = \mathrm{Log}_{\mathrm{M}}(\mathbf{X}_i)$ for $i = 1, \ldots, N$.
4: Translate $\mathbf{Y}_i$ to the tangent space of identity $\mathbf{I}$.
5: Perform PCA in $T_{\mathrm{I}} P(n)^m$ and get eigenvectors $\mathbf{U}_i$.
6: Translate $\mathbf{U}_i$ back to get $\mathbf{V}_i$ in the tangent space of $\mathbf{M}$.

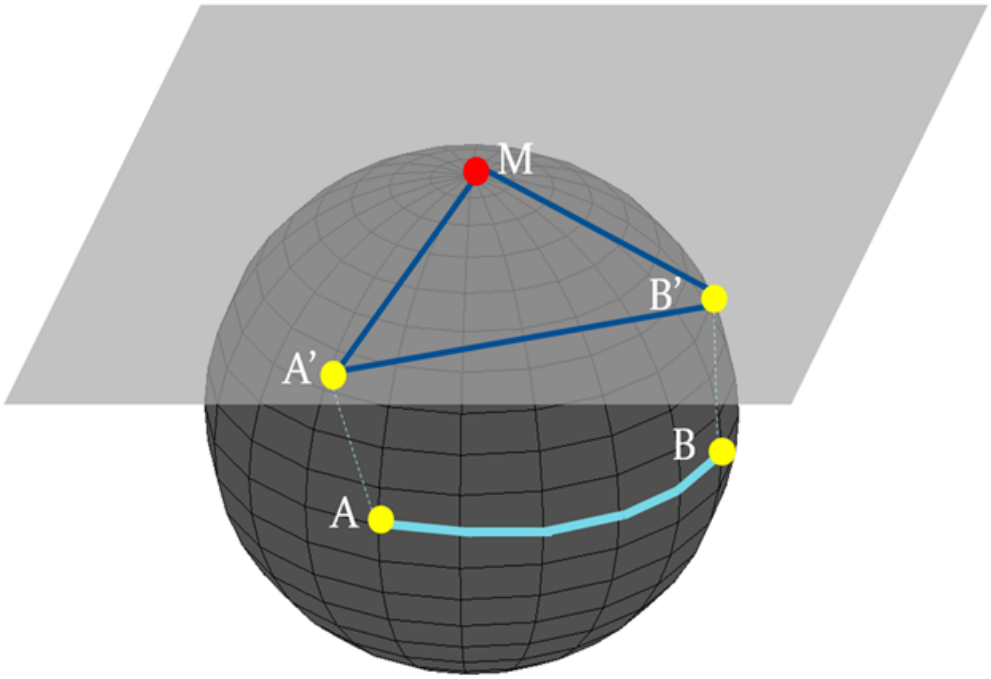


Figure 3-3. The log map on the unit sphere. **A** and **B** are two points on the unit sphere. **A**′ and **B**′ are their log projections to the tangent plane at **M**.

3.3 Tensor Fields Classification

We can formulate a tensor field classification algorithm using the principal directions and geodesic submanifolds. One common method for solving classification problems on Riemannian manifold is to map input data to the tangent space and do the classification in the tangent space [79]. However, this approach does not respect the geometry of the manifold as the geodesic distance between two points on the manifold are usually not the same or even commensurate with the distance between their images in the tangent space. One example is shown in Figure 3-3. Let $\mathbf{A} = (1, 0, 0)$, $\mathbf{B} = (0, 1, 0)$, $\mathbf{M} = (0, 0, 1)$ be three points on the unit sphere. $\mathbf{A}'$ and $\mathbf{B}'$ are the log projections of **A** and **B** to the tangent plane at **M**. The spatial relation of three points on the sphere is not preserved on the tangent plane at **M**, since the distance between $\mathbf{A}'$ and $\mathbf{B}'$ ($\frac{\sqrt{2}}{2}\pi$) is not the same as the geodesic distance between **A** and **B** on the unit sphere ($\frac{\pi}{2}$). A more

principled approach is to use the distances to geodesic submanifolds as the feature for classification.

Assume a binary classification problem, and the training tensor fields are labelled as one of the two classes. For each label $k(k = 1, 2)$, we compute a low-dimensional geodesic submanifold $\mathbf{S}_k$ using training tensor fields with label k. For a test tensor field $\mathbf{X}$, we can determine its class by comparing the geodesic distances $d_k = \min_{Y \in S_k} d(\mathbf{X}, \mathbf{Y})$. A tensor field is classified as belonging to class k if d_k is smaller than the other geodesic distance. The key step in this algorithm is to find the minimizer in $\mathbf{S}_k$ that gives the geodesic distance d_k. Since any point in the geodesic submanifold $\mathbf{S}_k$ can be written as $\mathrm{Exp}_{\mathrm{M}} \left(\sum_{i=1}^{d} \alpha_i \mathbf{V}_i \right)$, where $\mathbf{M}$ is the mean and $\alpha_1, \ldots, \alpha_d$ are real coefficients, d_k can be solved via the following optimization problem in $\mathbb{R}^d$,

$$\min_{\alpha_1, \ldots, \alpha_d} d \left(\mathbf{X}, \mathrm{Exp}_{\mathrm{M}} \left(\sum_{i=1}^{d} \alpha_i \mathbf{V}_i \right) \right)^2 . \tag{3–9}$$

Unfortunately, minimizing Equation 3–9 can be time-consuming for large tensor fields. Therefore, we approximate d_k by the geodesic distance $d(\mathbf{X}, \mathbf{Z})$ between $\mathbf{X}$ and $\mathbf{Z}$ defined by

$$\mathbf{Z} = \mathrm{Exp}_{\mathrm{M}} \left(\sum_{i=1}^{d} \mathbf{V}_i \langle \mathbf{V}_i, \mathrm{Log}_{\mathrm{M}}(\mathbf{X}) \rangle_{\mathrm{M}} \right) . \tag{3–10}$$

That is, we obtain $\mathbf{Z}$ by first map $\mathbf{X}$ to the tangent space at $\mathbf{M}$ using Log map and project it onto the principal subspace S_d. The projection is then mapped down to the manifold using the exponential map to get Z. In our experiments discussed in the next section, the differences between the approximated distance and the one computed by the optimization are less than 1% and have no major influence on the classification results. We summarize the tensor fields classification algorithm in Algorithm 6. Our algorithm can be easily extended to multi-class cases by computing geodesic submanifolds for each class in the training step and assigning the test tensor field to the class with the smallest geodesic distance.

Algorithm 6 Tensor Fields Classification

1: **Training** compute geodesic submanifolds $\mathbf{S}_1$ and $\mathbf{S}_2$ by using PGA on training tensor fields of different classes separately.
2: **Testing** for each test tensor field, compute their geodesic distances d_1 and d_2 to submanifolds $\mathbf{S}_1$ and $\mathbf{S}_2$ respectively. If $d_1 < d_2$, classify the test data to class1, otherwise, set it to class2.

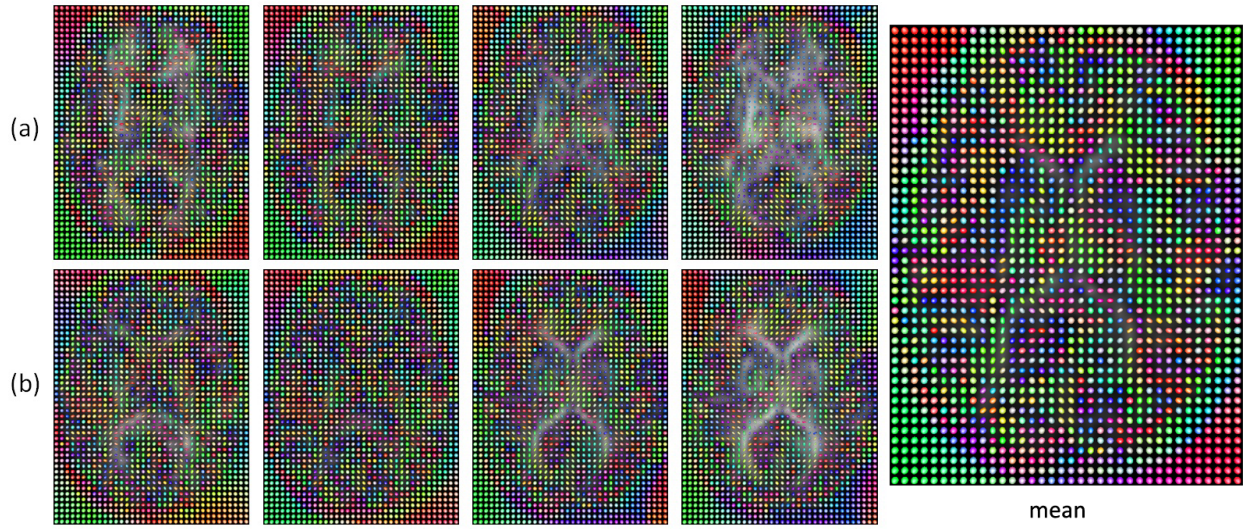


Figure 3-4. Statistical Analysis for deformation tensor fields from old age group. For better visualization, we downsample the images, only axial view is shown and set FA as the background in the display. **A)** Tensor field variation along the first principal direction. From left to right, the coefficient α_1 is $-2\sigma_1, -\sigma_1, \sigma_1, 2\sigma_1$. **B)** Tensor field variation along the second principal direction. From left to right, the coefficient α_2 is $-2\sigma_2, -\sigma_2, \sigma_2, 2\sigma_2$. **Right column**: Mean tensor field.

3.4 Experimental Results

In this section, we apply the proposed algorithms to perform statistical analysis for tensor fields from real medical imaging data. Both deformation tensor fields generated from MR brain images in OASIS database and a set of rat brain diffusion tensor images are used in our experiments.

3.4.1 Deformation tensor fields

Freely available Open Access Series of Imaging Studies (OASIS) MRI data set contains T1 weighted MR brain images from a cross-sectional population of 416 subjects aged between 18-96 [43]. Each brain image has a resolution of $176 \times 208 \times 176$

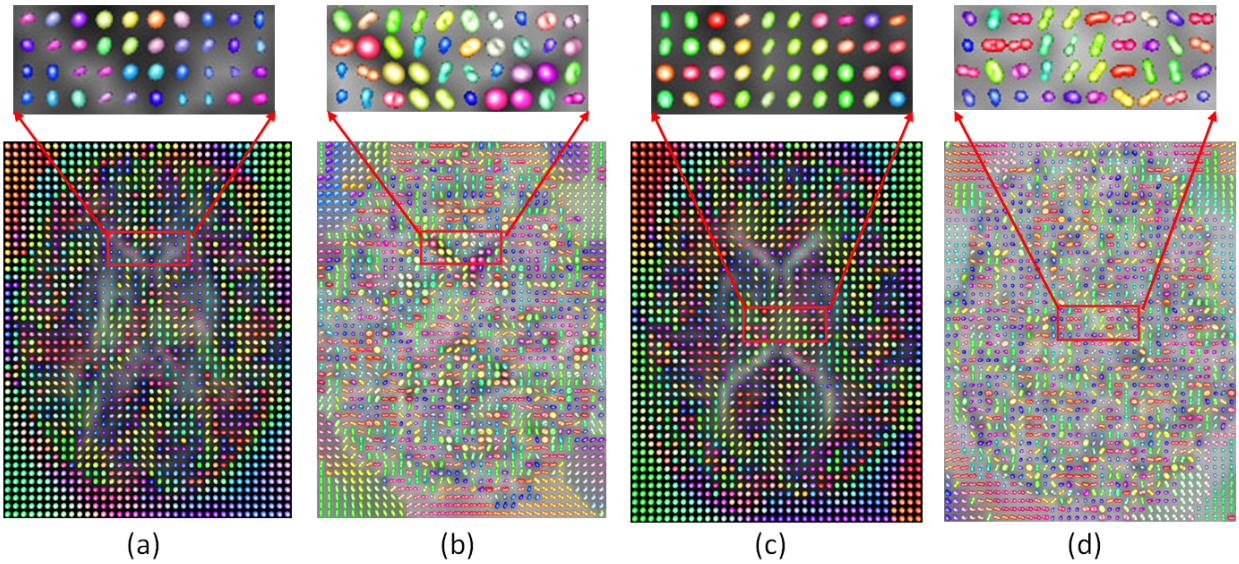


Figure 3-5. Comparisons with voxel-based method. (a) and (c) are the first mode (σ_1) and second mode (σ_2) computed using Algorithm Five. (b) and (d) are the first mode and the second mode computed using voxel-wise PGA. FA is used as the background in the display.

voxels. Because the structural difference in the brain across different age groups can be significant [44], the OASIS data set is heterogeneous. We divided 416 subjects three groups: young subjects (40 or younger), middle-aged subjects (between 40 and 60 years of age) and old subjects (60 or older). There is a subset of old subjects who have been clinically diagnosed with very mild to moderate Alzheimers disease (AD) using the Clinical Dementia Rating (CDR) scale. We compute the atlas for all the MR images in the OASIS data set using a group-wise nonrigid registration [32], and this also gives the the deformation field from each image to the atlas. For each voxel, we compute the Jacobian matrix J of the deformation field and build the deformation strain tensor $S = (J^T J)^{1/2}$. This gives a deformation tensor field for every subject.

In the first experiment, we characterize the mean and variation in the tensor fields within an age group by computing the modes of variation using Algorithm 4 and Algorithm 5. The dimension of the geodesic submanifold is set at 20 after examining the eigenvalue distribution. Figure 3-4 displays the intrinsic mean tensor field and the variations along the first two principal directions for the old group. The comparison

with the mean and modes computed using the voxel-based approach in shown in Figure 3-5. We can clearly see that the modes computed using voxel-based method are fragmentary and they do not reflect the global structures of the tensor fields. This is not surprising because voxel-based method does not consider correlations between different voxels.

For the second experiment, we test our tensor-field classification algorithm. We randomly divide the brain images for each age group into four subsets. Images from one of the subsets are the test images, while other three subsets give the training images. The training images are used to compute the geodesic submanifolds $\mathbf{S}_d$, and we classify the test images for every pair of age groups using Algorithm 6. We use a four-fold cross-validation in the experiments to fully evaluate the algorithm on OASIS data set. We compared the performance of our algorithm with the Log-Euclidean nearest neighbor method that maps each tensor field to the tangent space of the mean. Low-dimensional feature vectors are generated using PCA projection, and the classification is done using nearest neighbor of these feature vectors. We have also tested the proposed algorithm on classification for healthy and diseased (Alzheimer) brain images. All results are tabulated in Table 3-1. The experimental results indicate

Table 3-1. Tensor Fields Classification on OASIS

	O vs. Y	**O vs. M**	**M vs. Y**	**AD vs. Control**
Nearest Neighbor	92.43%	87.74%	78.42%	84.29%
Submanifold Projection	96.43%	90.23%	84.32%	88.57%

that the deformation strain tensor fields do capture the subtle structural changes in the brain images across different age groups (and also for diseased samples), and the good classification results show the importance and value of doing statistical analysis on the space of tensor fields.

3.4.2 Diffusion tensor fields

In the third experiment, we use a set of rat brain diffusion tensor images. In the preprocessing step, all images are aligned using similarity transforms, and in each

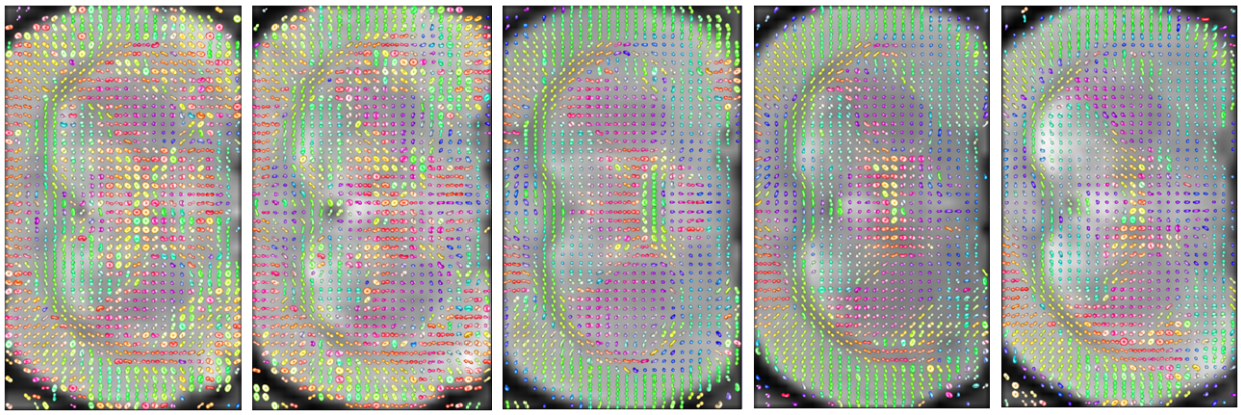


Figure 3-6. Five samples from input rat brain diffusion tensor images. S_0 is set as the background in the display. We have enhanced the anisotropy for better visualization.

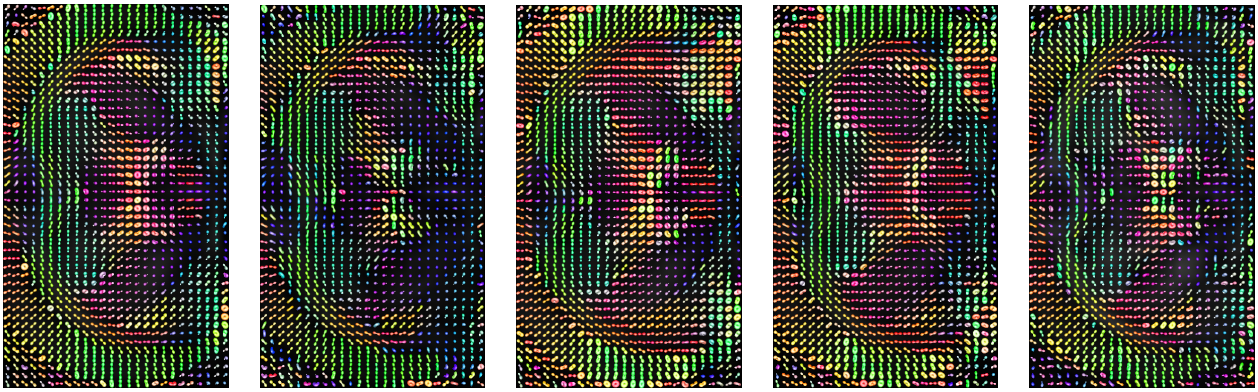


Figure 3-7. Statistical analysis for diffusion tensor images. **A)** mean tensor field. **B-E)** tensor field variation along the first four principal directions.

image, a region of interest of size $46 \times 27 \times 7$ is manually selected. Figure 4.4.2 displays five sample slices from the input 3D diffusion tensor images. We enhance the tensor anisotropy in the visualization to better display the rat brain structures. The diffusion tensor fields from 50 subjects are chosen to construct the principal geodesic submanifold approximation of the image set. The intrinsic mean tensor field and the variations along the first four principal directions are shown in Figure 3-7. Our principal geodesic submanifold representation clearly captures important anatomical structures such as white matter, putamen and nucleus accumbens. We also test the reconstruction

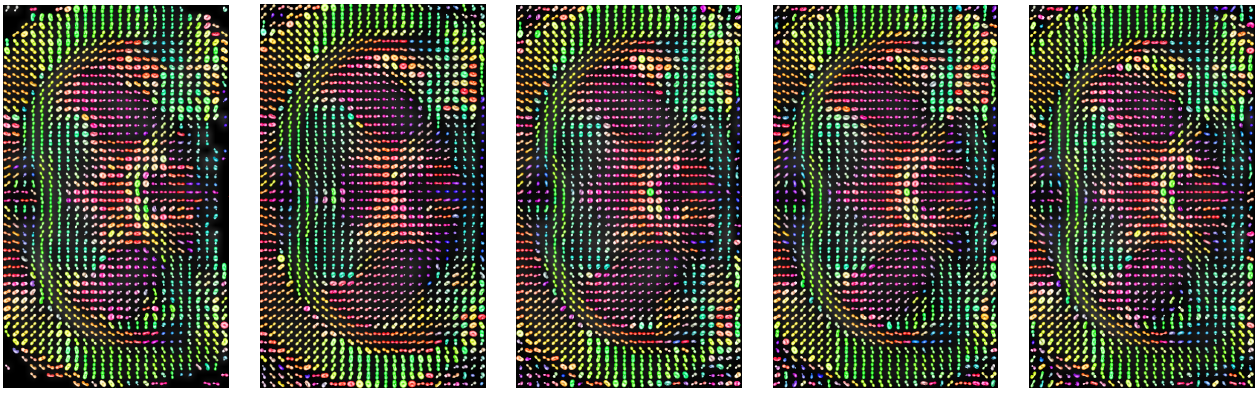


Figure 3-8. Diffusion tensor image reconstruction. **A)** one slice of original diffusion tensor image. **B-E)** reconstructed tensor images in principal subspaces with dimensions 5, 15, 25, 35 respectively.

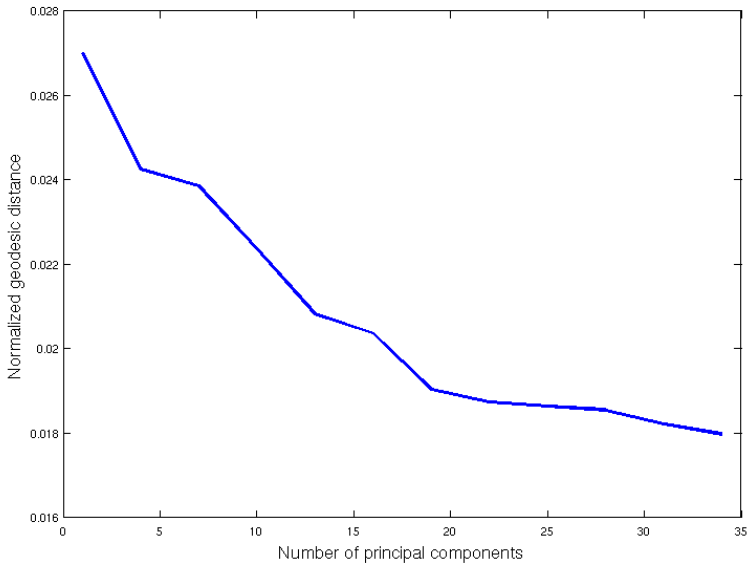


Figure 3-9. Reconstruction errors between the test tensor field and its projections on the geodesic submanifolds of different dimensions.

performance of our model by computing the projections of a new tensor field to the geodesic submanifolds generated by different numbers of principal components. Figure 3-8 displays the original input diffusion tensor image $\mathbf{I}_0$ and its projections $\mathbf{I}$ to the low dimensional geodesic submanifolds computed using Algorithm 5. Normalized geodesic distance $\frac{1}{m}d(\mathbf{I}_0, \mathbf{I})$ is used to quantitatively measure the reconstruction error, which is plotted in Figure 3-9. As the number of the principal components increases, the projections in the geodesic submanifold more accurately approximated the original diffusion tensor image.

CHAPTER 4
NONNEGATIVE FACTORIZATION OF DIFFUSION TENSOR IMAGES

4.1 Motivation

In this chapter, we introduce the novel notion of nonnegative factorization of tensor fields and its application to the segmentation of diffusion tensor images (DTI). Tensor images (or tensor fields) abound in medical imaging applications, with well-known examples such as DTIs, conductivity tensor images (CTI) and elasticity tensors in Elastography. There is an increasing demand for a principled and versatile method that can automatically extract important features from single as well as multiple tensor images, and our work takes a step forward in this direction by investigating the general factorization problem for tensor fields. Specifically, we formulate the factorization as an optimization problem, and as the main technical contribution, we present an iterative algorithm that can efficiently and reliably optimize the objective function and compute good-quality factorizations.

Matrix factorization in the form of factoring a data matrix $\mathbf{V}$ into a product of the basis matrix $\mathbf{W}$ and the coefficient matrix $\mathbf{H}$ appears frequently in computer vision and image processing. Examples include the principal component analysis (PCA) and the Tomasi-Kanade factorization from multi-view 3D structure recovery. Algebraically, the factorization attempts to "discover" the linear subspace spanned by the columns of $\mathbf{W}$ that can be used to approximate the data given as the columns of $\mathbf{V}$. Instead of the full subspace, the linear model can be further constrained by requiring that the data vectors belong to (or can be approximated by) the cone generated by the columns of $\mathbf{W}$. Since a point in a cone can be written as a nonnegative linear combination of the cone's generators, this gives rise to the nonnegative matrix factorization (NMF) first introduced in [36]. Specifically, using images as data, the matrices appeared in an NMF are all required to have nonnegative components. While the unconstrained matrix factorization admits a comparatively straightforward solution using linear algebra,

nonnegative matrix factorization is generally more difficult to solve as its objective function is non-convex and there exists no known linear algebraic methods that can compute its solution in closed form. Nevertheless, the optimization is not particularly involved as the nonnegative constraint can be easily enforced, and NMF is almost always solved using an alternating sequence of nonnegative linear least squares with variables in $\mathbf{W}$ and $\mathbf{H}$ separately. More importantly, the nonnegativity constraint has been argued in the original paper [36] to be more capable, compared with the unconstrained factorization, of capturing features that are common among the data. In particular, the columns of the basis matrix $\mathbf{W}$ can be considered as the discovered common parts among the input data, and the coefficient matrix $\mathbf{H}$ provides the weights for reconstructing each input data using these parts. The power and versatility of NMF have been demonstrated to various degrees by a wide range of successful applications that include data clustering [17], parts discovery [41] and MR image analysis [33].

The proposed nonnegative factorization of tensor-valued images is an extension of the nonnegative matrix factorization method above. In our context, components in the matrices $\mathbf{V}$ and $\mathbf{W}$ are represented as symmetric positive semi-definite (PSD) tensors of rank two and $\mathbf{H}$ is the coefficient matrix with nonnegative real components. In particular, the nonnegativity constraint in our more general setting has now acquired two different realizations: the nonnegative constraint on the components of $\mathbf{H}$ as in the original NMF and the generalized nonnegative constraint on the components of $\mathbf{W}$ that they belong to the space of $\mathbf{PSD}(n)$ of $n \times n$ symmetric positive semi-definite matrices. Geometrically, this corresponds to replacing the nonnegative real line (intensity values) with the space $\mathbf{PSD}(n)$ (diffusion tensors), and it is their respective cone structures that permit us to formulate nonnegative factorization using these spaces. While our extension is easy to understand conceptually, the resulting optimization problem in our case is considerably more difficult to solve because of the generalized nonnegative constraint on $\mathbf{W}$ and the dimension of $\mathbf{PSD}(3)$. The former requires generalized matrix

inequalities to enforce the constraint [8] and the latter introduces a large number of variables in the objective function. Therefore, a major portion of this chapter is devoted to an optimization algorithm that can reliably and efficiently compute the factorization.

Having overcome this computational hurdle, we will next show that the proposed tensor images factorization method can be successfully applied to segmentation problems for single and multiple DTI images. For a single image, the data matrix $\mathbf{V}$ is an $1 \times n$ array of PSD tensors, and a direct clustering on the coefficient matrix $\mathbf{H}$ gives the segmentation. For multiple tensor images, the basis matrix $\mathbf{W}$ gives as before a "part-decomposition" of the collection, and the common parts given as the columns of $\mathbf{W}$ can usually be realized as tensor fields with local support. In the current medical imaging literature, algorithms that segment DTI images can be broadly classified into two categories, the level-set based methods (e.g., [37, 75]) and the methods based on combinatorial optimization such as graph cuts [76]. In terms of its underlying motivation and numerics, our factorization-based method offers a completely different approach to the segmentation problem. We have validated the proposed method using both synthetic and real tensor images. Preliminary segmentation results from single and multiple DTI images have indicated that the proposed factorization-based approach is a viable and competitive alternative method for segmenting tensor images.

4.2 Preliminaries

Given an $n \times m$ non-negative matrix $\mathbf{V}$, each of whose column $\mathbf{v}_i$ represents an input image, non-negative matrix factorization (NMF) [36] attempts to factor it into two matrices with non-negative components, $\mathbf{V} \approx \mathbf{WH}$, where $\mathbf{W}$ denotes the $n \times r$ basis matrix and $\mathbf{H}$ denotes the $r \times m$ coefficient matrix. According to the geometric interpretation elucidated in [18], NMF determines a cone $\Sigma_W = \{\mathbf{x} | \mathbf{x} = \sum_{j=1}^{r} h_j \mathbf{w}_j,\ h_j \geq 0\}$ that approximates the input images with the basis elements (or generators of Σ_W) $\mathbf{w}_j$ given by the j-th column of $\mathbf{W}$.

For tensor-valued images, there is a $d \times d$ symmetric, positive semi-definite matrix associated to each pixel (or voxel). $d = 3$ for diffusion tensor images. The mathematics for nonnegative factorization of tensor images is straightforward as it only requires a cone structure that is provided by $\mathbf{PSD}(d)$, the space of $d \times d$ symmetric positive semi-definite matrices. A collection of m tensor images of size n can be arranged into a block matrix $\mathbf{V} = \begin{pmatrix} V_{11} & \cdots & V_{1m} \\ \vdots & \ddots & \vdots \\ V_{n1} & \cdots & V_{nm} \end{pmatrix}$, where $V_{ki} \in \mathbf{PSD}(d)$, $k = 1, \ldots, n$ and $i = 1, \ldots, m$. Each of the m columns represents one tensor image. As before, nonnegative factorization for tensor images attempts to factor $\mathbf{V}$ into a product of the basis matrix $\mathbf{W}$ the coefficient matrix $\mathbf{H}$ whose elements are non-negative real numbers:

$$\mathbf{V} \approx \mathbf{W} * \mathbf{H} = \begin{pmatrix} W_{11} & \cdots & W_{1r} \\ \vdots & \ddots & \vdots \\ W_{n1} & \cdots & W_{nr} \end{pmatrix} * \begin{pmatrix} h_{11} & \cdots & h_{1m} \\ \vdots & \ddots & \vdots \\ h_{r1} & \cdots & h_{rm} \end{pmatrix}, \tag{4–1}$$

where the blocks W_{ij} in $\mathbf{W}$ are matrices in $\mathbf{PSD}(d)$, and the blockwise product $*$ is defined as

$$\mathbf{W} * \mathbf{H} = \begin{pmatrix} \sum_{j=1}^{r} W_{1j} h_{j1} & \cdots & \sum_{j=1}^{r} W_{1j} h_{jm} \\ \vdots & \ddots & \vdots \\ \sum_{j=1}^{r} W_{nj} h_{j1} & \cdots & \sum_{j=1}^{r} W_{nj} h_{jm} \end{pmatrix}. \tag{4–2}$$

r in the above equation is the number of basis elements (generators) used in the factorization, and it is clear that our nonnegative factorization reduces to the usual NMF when $d = 1$. We remark that the nonnegative factorization can be considered as a generative model for the collection of input tensor images as each input tensor image is approximated by a non-negative linear combination of r columns of $\mathbf{W}$, each of which can be considered as a tensor image. To determine $\mathbf{W}, \mathbf{H}$ from the data matrix $\mathbf{V}$, we

formulate a constrained optimization problem that minimizes the cost function

$$E(\mathbf{W}, \mathbf{H}) = \frac{1}{2} \sum_{i=1}^{m} \sum_{k=1}^{n} \| V_{ki} - \sum_{j=1}^{r} W_{kj} h_{ji} \|_F^2 \tag{4–3}$$

with the constraints $W_{kj} \succeq 0$ and $h_{ji} \geq 0$, $i = 1, \ldots, m$, $j = 1, \ldots, r$ and $k = 1, \ldots, n$. $\succeq$ denotes matrix inequality and $\| \cdot \|_F$ denotes the Frobenius norm.

4.3 Algorithm and implementation details

In this section, we present an efficient algorithm that solves the constrained optimization defined above. While the objective function $E(\mathbf{W}, \mathbf{H})$ is not convex, it is convex with respect to the two block variables **W** and **H**. A common approach to solve this type of constrained optimization problem is the block coordinate descent method [48], and in particular, we can alternatively fix one block variable and improve the other as shown in Algorithm 7. Grippo et al [27] have shown that every limit point of the

Algorithm 7 Alternating Non-negative Factorization

Initialize $\mathbf{H}^1 \geq 0$.
For $t = 1, 2, \ldots$

- $\mathbf{W}^{t+1} = \underset{\mathbf{W}}{\operatorname{argmin}} \; E(\mathbf{W}, \mathbf{H}^t)$, s.t. $W_{kj} \succeq 0, \forall k, j$.
- $\mathbf{H}^{t+1} = \underset{\mathbf{H}}{\operatorname{argmin}} \; E(\mathbf{W}^{t+1}, \mathbf{H})$, s.t. $\mathbf{H} \geq 0$.

sequence $\{\mathbf{W}^t, \mathbf{H}^t\}$ generated by Algorithm 7 is a stationary point of the optimization problem defined in (3). Since both sub-problems are convex, our algorithm can easily be shown to be provably convergent. The details for solving the two sub-problems efficiently will be discussed below.

4.3.1 Optimization with respect to the basis matrix W

When **H** is fixed, the optimization problem (3) reduces to a quadratic semi-definite programming problem with a large number of positive semi-definite matrices as the constrained variables. This is a challenging optimization problem without readily available solvers as most available semi-definite programming solvers such as SeDuMi

[62] and SDPT3 [69] require the linear objective functions. Toh et al. [67] proposed an inexact primal-dual algorithm to solve a special class of convex quadratic semi-definite programming problem. However, their algorithm only deals with a single positive semi-definite matrix and cannot be directly applied to our optimization problem. Instead, we will exploit the special feature in our problem that $d = 3$ is a relatively small number and design a specific algorithm based on primal-dual path-following interior-point method to solve this subproblem.

The primal problem is given by

$$\begin{aligned} &\min_{W} \frac{1}{2}\sum_{i=1}^{m}\sum_{k=1}^{n}\| V_{ki} - \sum_{j=1}^{r} W_{kj}h_{ji}\|_F^2 \\ &s.t.\ W_{kj} \succeq 0, k = 1, \dots, n,\ j = 1, \dots, r. \end{aligned} \tag{4–4}$$

We introduce the $d \times d$ symmetric matrices Z_{kj} associated with the matrix inequalities $W_{kj} \succeq 0$, where $k = 1, \dots, n,\ j = 1, \dots, r$. The Lagrangian of problem (4) is then

$$L(\mathbf{W}, \mathbf{Z}) = \frac{1}{2}\sum_{i=1}^{m}\sum_{k=1}^{n}\| V_{ki} - \sum_{j=1}^{r} W_{kj}h_{ji}\|_F^2 - \sum_{k=1}^{n}\sum_{j=1}^{r}\mathrm{Tr}(Z_{kj}W_{kj}). \tag{4–5}$$

If we take the derivative of $L(\mathbf{W}, \mathbf{Z})$ with respect to W_{kj} and set it to zero, we have

$$Z_{kj} = -\sum_{i=1}^{m}(V_{ki} - \sum_{l=1}^{r} W_{kl}h_{li})h_{ji}. \tag{4–6}$$

Substituting this expression back into the Lagrangian gives the dual problem

$$\begin{aligned} &\max_{Z} \sum_{i=1}^{m}\sum_{k=1}^{n}\left(-\frac{1}{2}\| V_{ki} - \sum_{j=1}^{r} W_{kj}h_{ji}\|_F^2 + < V_{ki} - \sum_{j=1}^{r} W_{kj}h_{ji}, V_{ki} >\right) \\ &s.t. \sum_{i=1}^{m}\left(V_{ki}h_{ji} - \sum_{l=1}^{r} W_{kl}h_{ji}h_{li}\right) + Z_{kj} = 0 \\ &\quad Z_{kj} \succeq 0, k = 1, \dots, n,\ j = 1, \dots, r. \end{aligned} \tag{4–7}$$

For primal-dual interior point method, we use the perturbed Karush-Kuhn-Tucker (KKT) conditions:

$$\begin{aligned} Z_{kj} + \sum_{i=1}^{m}(V_{ki} - \sum_{l=1}^{r} W_{kl}h_{li})h_{ji} = 0 \\ W_{kj}Z_{kj} = \nu_{kj}I \\ W_{kj} \succeq 0,\ Z_{kj} \succeq 0 \\ k = 1, \dots, n,\ j = 1, \dots, r \end{aligned} \tag{4–8}$$

where ν_{kj} are positive parameters. Given the current iterate $(\mathbf{W}, \mathbf{Z})$, the search direction $(\Delta\mathbf{W}, \Delta\mathbf{Z})$ at each interior-point iteration is the solution of the following Newton system

$$\sum_{l=1}^{r} \Delta W_{kl}(\sum_{i=1}^{m} h_{li}h_{ji}) - \Delta Z_{kj} = R_{d_{kj}} := Z_{kj} + \sum_{i=1}^{m}(V_{ki} - \sum_{l=1}^{r} W_{kl}h_{li})h_{ji} \tag{4–9}$$

$$H_{P_{kj}}(\Delta W_{kj}Z_{kj} + W_{kj}\Delta Z_{kj}) = \sigma\mu_{kj}I - H_{P_{kj}}(W_{kj}Z_{kj} + \Delta W_{kj}\Delta Z_{kj}) \tag{4–10}$$

where $\mu_{kj} =< W_{kj}, Z_{kj} > /d$ and $\sigma \in (0, 1)$ is the centering parameter. The symmetrization scheme H_P defined as $H_P(M) = \frac{1}{2}[PMP^{-1} + P^{-T}M^TP^T]$ is required here to generate symmetric ΔW_{kj} and ΔZ_{kj}. Inspired by [66], we choose $P_{kj} = T_{kj}^{-1/2}$, where T_{kj} is the Nesterov-Todd scaling matrix satisfying $T_{kj}Z_{kj}T_{kj} = W_{kj}$. The linearization of equation (10) gives

$$\mathcal{E}_{kj}(\Delta W_{kj}) + \mathcal{F}_{kj}(\Delta Z_{kj}) = R_{c_{kj}} := \sigma\mu_{kj}I - H_{P_{kj}}(W_{kj}Z_{kj}) \tag{4–11}$$

where $\mathcal{E}_{kj}$ and $\mathcal{F}_{kj}$ are linear operators. By eliminating ΔZ_{kj} in equations (9) and (11), we obtain

$$\sum_{l=1}^{r}(\sum_{i=1}^{m} h_{li}h_{ji})\Delta W_{kl} + \mathcal{F}_{kj}^{-1}\mathcal{E}_{kj}(\Delta W_{kj}) = R_{d_{kj}} + \mathcal{F}_{kj}^{-1}R_{c_{kj}}. \tag{4–12}$$

Therefore, we just need to solve n linear systems to get $\Delta\mathbf{W}$. Each linear system includes $\frac{d(d+1)r}{2}$ equations. Because both d and r are small for our problem, these linear systems can be solved efficiently, and $\Delta\mathbf{Z}$ can be computed easily using equation (9). The detail steps of the algorithm is shown in Algorithm 8.

4.3.2 Optimization with respect to coefficient matrix H

With a fixed **W**, (3) becomes a convex optimization problem. Since

$$\frac{\partial E}{\partial h_{ji}} = -\sum_{k=1}^{n} \mathrm{Tr}(V_{ki} W_{kj}) + \sum_{k=1}^{n} \sum_{l=1}^{r} \mathrm{Tr}(W_{kj} W_{kl}) h_{li}, \tag{4–13}$$

setting $\frac{\partial E}{h_{ji}} = 0$ gives

$$\sum_{l=1}^{r} (\sum_{k=1}^{n} < W_{kj}, W_{kl} >) h_{li} = \sum_{k=1}^{n} < V_{ki}, W_{kj} > . \tag{4–14}$$

Thus we just need to solve a non-negative least squares problem $\mathbf{A}_{r\times r}\mathbf{H}_{r\times m} = \mathbf{B}_{r\times m}$ to get h_{ji}, where $A_{jl} = \sum_{k=1}^{n} < W_{kj}, W_{kl} >$ and $B_{ji} = \sum_{k=1}^{n} < V_{ki}, W_{kj} >$, for all i, j, l. In our implementation, we use the fast active set method proposed by Van Benthem and Keenan [71] to solve this large-scale nonnegative least squares problem.

Once we have obtained the basis matrix **W**, we can easily compute the "projection" of a new diffusion tensor image $\mathbf{X} = (X_1, \ldots, X_n)^T$ by solving the following optimization problem

$$\min_{\mathrm{y}\geq 0} \sum_{k=1}^{n} \|X_k - \sum_{j=1}^{r} W_{kj} y_j\|_F^2 \tag{4–15}$$

where each X_k is a diffusion tensor and $\mathbf{y} = (y_1, \ldots, y_r)^T$ is the nonnegative coefficient vector. This problem, similar to the subproblem with respect to **H** above, can also be efficiently solved using the same fast active set method.

4.3.3 Diffusion tensor image segmentation

Because of its relations to K-means [17] and probabilistic latent semantic analysis [25], nonnegative matrix factorization has been widely used in data clustering, e.g., [83]. In this section, we formulate the diffusion tensor image segmentation problem as a special case of our more general nonnegative factorization problem with spatial constraints.

Specifically, given a diffusion tensor image of size m, we arrange the m tensors in a row to form the data matrix $\mathbf{V} = (V_1, \ldots, V_m)$. The original nonnegative factorization

Algorithm 8 Quadratic semi-definite programming for nonnegative factorization

1: **Initialization** $W_{kj} \succeq 0$, $Z_{kj} \succeq 0$, $\forall k, j$ and $\tau = 0.9$.
2: **Convergence test** Stop the iteration if the accuracy measure ϕ is sufficiently small.

$$\phi = \frac{\sum_{k=1}^{n} \sum_{j=1}^{r} < W_{kj}, Z_{kj} >}{1 + |\mathbf{pobj}| + |\mathbf{dobj}|}$$

where **pobj** and **dobj** are values of primal and dual objective functions.
3: **Predictor step** Compute the predictor search direction $(\delta\mathbf{W}, \delta\mathbf{Z})$ by choosing $\sigma = 0$.
4: **Predictor step-length** Compute $\alpha_p = \min(1, \tau\alpha)$. α is the maximum step length that can be taken so that for $k = 1, \ldots, n$ and $j = 1, \ldots, r$, $W_{kj} + \alpha\delta W_{kj}$ and $Z_{kj} + \alpha\delta Z_{kj}$ remain positive semidefinite.
5: **Centering rule** Set

$$\sigma = \frac{\sum_{k,j} < W_{kj} + \alpha_p \delta W_{kj}, Z_{kj} + \alpha_p \delta Z_{kj} >}{\sum_{k,j} < W_{kj}, Z_{kj} >}.$$

6: **Corrector step** Compute the search direction $(\Delta\mathbf{W}, \Delta\mathbf{Z})$ using

$$R_{c_{kj}} = \sigma\mu_{kj} I - H_{P_{kj}}(W_{kj} Z_{kj} + \delta W_{kj} \delta Z_{kj}).$$

7: **Corrector step-length** Compute α_c similar to step 4 with $(\delta\mathbf{W}, \delta\mathbf{Z})$ replaced by $(\Delta\mathbf{W}, \Delta\mathbf{Z})$.
8: Update $(\mathbf{W}, \mathbf{Z})$ to the next iterate $(\mathbf{W}^+, \mathbf{Z}^+)$.

$$W_{kj}^+ = W_{kj} + \alpha_c \Delta W_{kj}, \; Z_{kj}^+ = Z_{kj} + \alpha_c \Delta Z_{kj}.$$

9: Update the step-length parameter by $\tau^+ = 0.9 + 0.08\alpha_c$.

problem is modified as

$$\min_{\mathrm{W,H}} \frac{1}{2} \sum_{i=1}^{m} \| V_i - \sum_{j=1}^{r} W_j h_{ji} \|_F^2 + \frac{\lambda}{2} \sum_{(k,l)\in\Omega} \sum_{j=1}^{r} (h_{jk} - h_{jl})^2 \qquad (4\text{–}16)$$

with nonnegative constraints $W_j \succeq 0$ and $h_{ji} \geq 0$, for all i, j. The first term is simply the objective function in (3) given $\mathbf{V}$ as a row vector. The second term is the spatial smoothness (soft) constraint that requires neighboring pixels to have similar coefficients, and in the equation above, Ω denotes the edge set of the discrete image graph, and λ the coupling parameter. The optimization problem (16) can be solved using the same

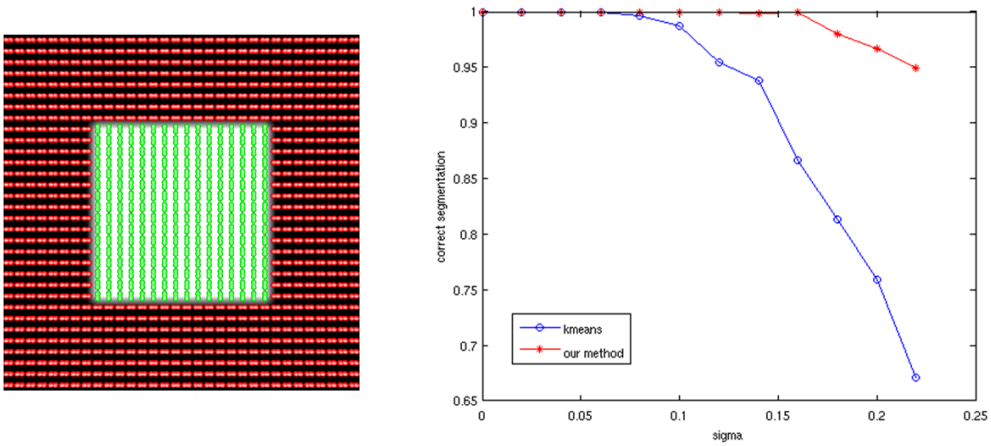


Figure 4-1. Segmentation experiment on synthetic tensor field. **Left**: Input tensor field without noise. **Right**: Segmentation accuracy vs. different levels of added Gaussian noise $N(0, \sigma)$ with covariance σ.

alternating method discussed above as the second term is also quadratic in **H**. Once the coefficient matrix **H** has been determined, we cluster the columns of **H** to produce the diffusion tensor image segmentation. In our implementation, we use K-means for this last clustering step.

4.4 Experiments

In this section, we present two sets of experimental results. The first experiment is on diffusion tensor image segmentation using the segmentation algorithm outlined in the previous section. For the second set of experiments, we work with multiple images, and the results demonstrated that, as for scalar-valued images, meaningful parts can be discovered or detected using nonnegative factorization.

4.4.1 Diffusion tensor image segmentation

Synthetic Tensor Images In this experiment, we test the accuracy of the segmentation algorithm using synthetic tensor images with various levels of added noise. We first generate the 32×32 tensor image **T** shown in Figure 4-1 that contains only two tensors: diagonal matrices $diag(0.5, 0.25, 0.25)$ and $diag(0.25, 0.5, 0.25)$, and this defines the ground truth of the segmentation. Different levels of Gaussian noise $N(0, \sigma)$ are added to the tensor image **T** to generate noisy tensor images, and we

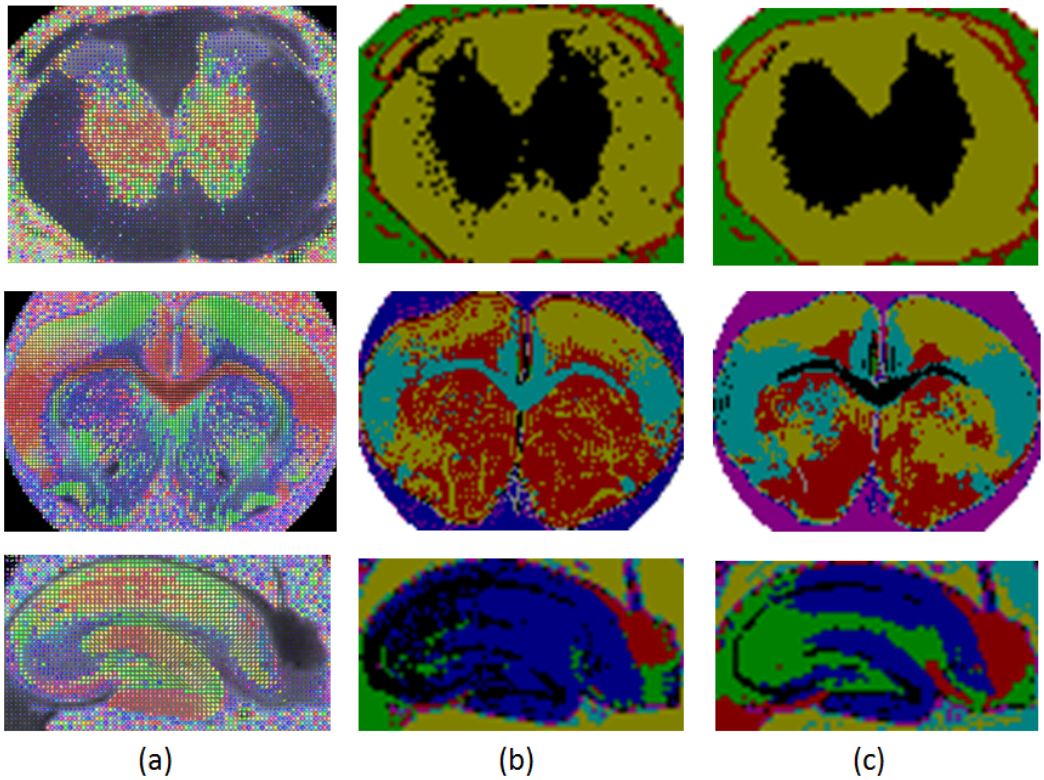
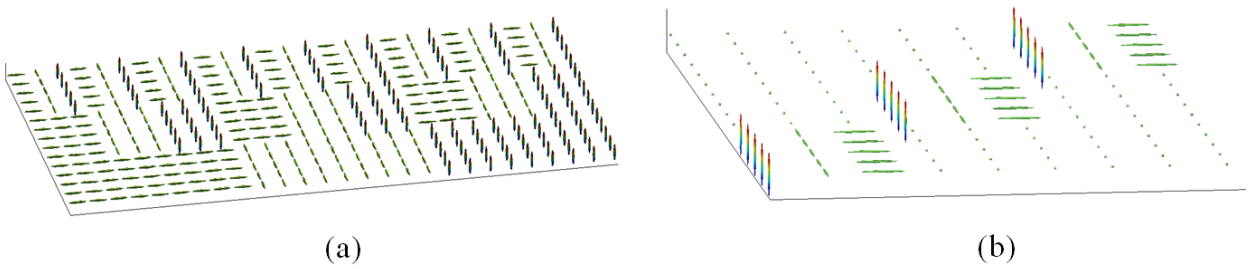


Figure 4-2. Diffusion tensor image segmentation. **First Row**: Spinal cord of a rat. **Second Row**: Corpus callosum of a rat. **Third Row**: Hippocampus of a rat. Columns **A)** Diffusion tensor images. **B)** Segmentation results using K-means. **C)** Segmentation results using our method. Segments are color-coded.

compare the segmentation accuracy of our method with the segmentation algorithm based on clustering the pixels using K-means (on the tensors). In this experiment, the segmentation accuracy is defined by the percentage of correctly segmented pixels, and the comparison across different noise levels is plotted in Figure 4-1. The result clearly shows the robustness of our method when compared with K-means, especially in the presence of substantial amount of noise.

Diffusion Tensor Images We next present segmentation results on three real diffusion tensor images shown in Figure 4-2. These are DTI images of the spinal cord, corpus callosum and an isolated hippocampus of a rat. The data were acquired using a PGSE with TR=1.5s, TE=28.3ms, bandwidth= 35Khz. Twenty-one diffusion weighted



(a) (b)

Figure 4-3. Nonnegative factorization on multiple synthetic tensor fields. **A)** Visualization of the data matrix **V**, each of whose columns represents a 5×3 tensor image. **B)** Visualization of the basis matrix **W**, each of whose columns represents a basis tensor image.

images with a b-value of 1250s/mm^2 were collected. The sizes of the regions of interest for rat spinal cord, corpus callosum and hippocampus are 71×61, 101×74 and 71×39, respectively. In this experiment, the number of clusters for these images are four, eight and seven, respectively, and the number of basis elements (columns in **W**) is set to five for all three images. Again, we compared our method with the K-means based segmentation and the results consistent demonstrate that our method can produce anatomically more accurate and meaningful segmentation than the straightforward K-means clustering. Finally, we note that both algorithms use K-means for clustering pixels. However, in our method, K-means is applied only to the coefficient vectors in **H**, while in the comparison method, K-means is applied directly to the tensors. While the clustering power of the nonnegative factorization for matrices are well-known for scalar-valued data (e.g., [17]), our experimental results provide the first convincing evidence of its clustering power for tensor-valued data as the two sets of experiments have shown that there is no reasons to expect that a direct clustering on tensors would produce desired segmentation results. However, direct clustering on coefficient vectors does yield satisfactory results.

4.4.2 Nonnegative factorization with multiple tensor fields

For a suitable collection of images, it is well-known that NMF has the ability to automatically determine similar decompositions of the images into their common parts

and provide part-based representations for the images in the collection [18]. In this experiment using synthetic data, we show that nonnegative factorization also has similar capability for tensor images. Twenty-seven tensor fields with size 5×3 (15 pixels) are generated and form the columns of the data matrix $\mathbf{V}$ as shown in Figure 4-3(a). We factor $\mathbf{V}$ into the coefficient matrix $\mathbf{H}$ and the basis matrix $\mathbf{W}$ shown in Figure 4-3(b). Our factorization algorithm correctly determines the nine basis elements (columns of $\mathbf{W}$) required to form the data matrix $\mathbf{V}$, and the coefficient matrix returned by our algorithm is a sparse matrix. Furthermore, the L^2 factorization error is less than 8.57×10^{-10}.

Finally, we present a preliminary result on applying our nonnegative factorization method to automatically discover and segment anatomically important regions from a collection of 53 rat brain diffusion tensor images. In the preprocessing step, all images are aligned using similarity transforms, and in each image, a region of interest of size $46 \times 27 \times 7$ is manually selected. The left column of Figure 4 displays five sample slices from the input 3D diffusion tensor images. We apply the factorization algorithm with $r = 5$ (five basis elements) to this collection of DTIs, and one sample slice from each of the five basis images found by the algorithm is shown on the right column in Figure 4-4. Important anatomical regions such as white matter, putamen and nucleus accumbens are clearly represented in the five basis images.

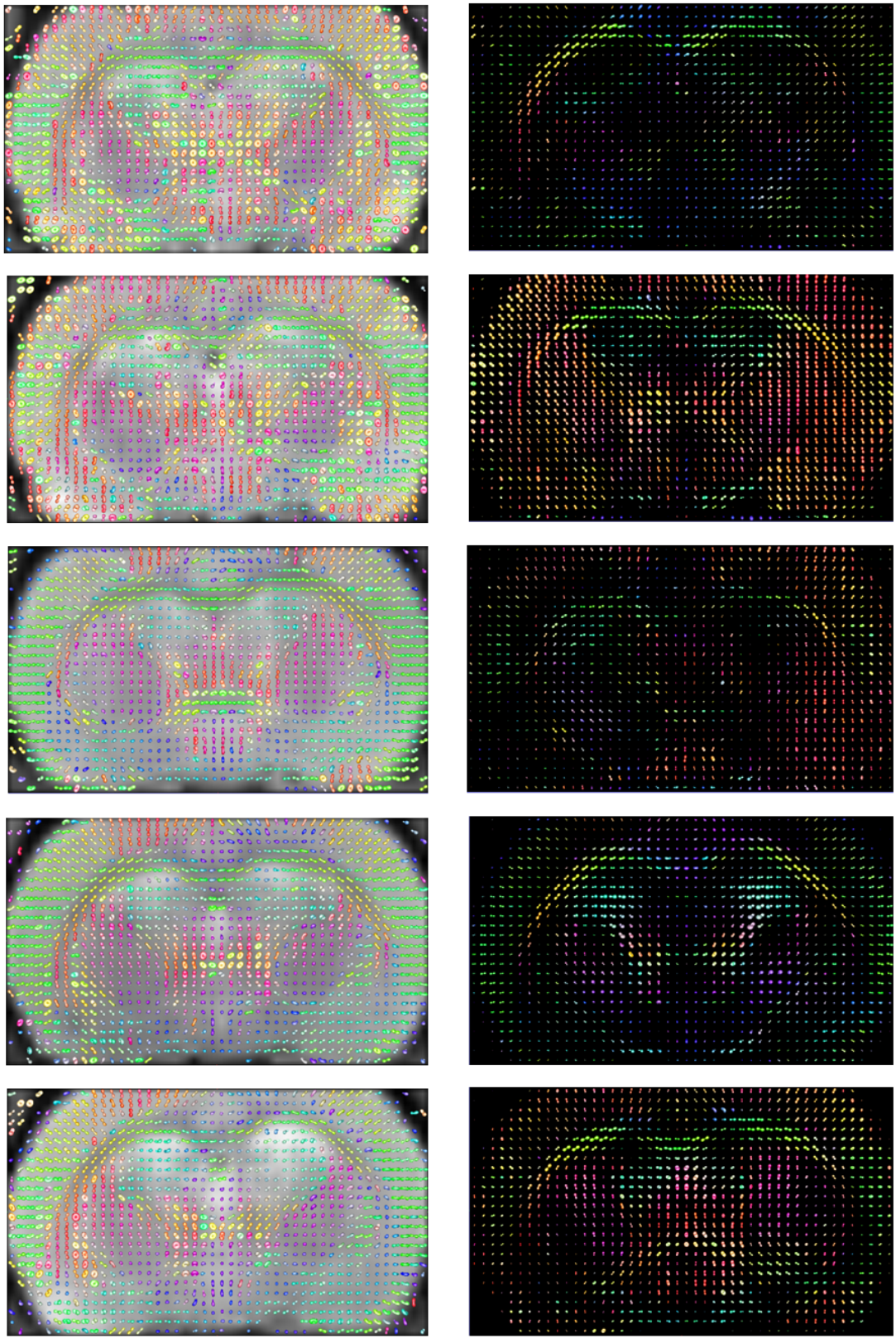


Figure 4-4. **Left column**: Five sample slices from 53 input rat brain diffusion tensor images. **Right column**: Sample slices from the five basis tensor images produced by the proposed nonnegative factorization algorithm.

CHAPTER 5
DICTIONARY LEARNING ON RIEMANNIAN MANIFOLDS

5.1 Motivation

Dictionary learning, which seeks to find a collection of atoms for sparse representation of the input data, has been widely used in computer vision applications such as image recognition, classification and restoration (e.g., [2]). Under this model, each data point is assumed to be generated linearly using only a small number of atoms, and this linear sparsity assumption is responsible for much of its generalization power and success. However, the underlying linear process requires that the data points as well as the atoms are treated as vectors in some vector space $\mathbb{R}^d$, and the dictionary is learned from the input data using only the vector space structure (and its associated inner product). For many applications in compute vision and medical image analysis that involve data points belonging to some known Riemannian manifolds such as the space of symmetric positive-definite matrices [23], hyper-spheres for parameterizing square-root densities [61], Stiefel and Grassmann manifolds [10], etc., the existing extrinsic approaches that completely ignore the potentially important intrinsic structure implied by the data is clearly inadequate and unsound (if they can be applied at all). To remedy this deficiency and inadequacy, we take a small step in this direction by investigating the problem of extending dictionary learning framework to incorporate intrinsic geometry implied by the input data.

The applicability and suitability of applying existing dictionary learning methods to solve vision problems that have to deal with manifold-valued data can be two thorny issues to consider. First, as a prerequisite, the data manifold must admit an embedding into some $\mathbb{R}^d$ in order to be able to apply the existing dictionary learning methods. However, for most manifolds, such as Grassmann and Stiefel manifolds, there simply does not exist known canonical embedding into $\mathbb{R}^d$ (or such embedding is difficult to compute). Second, even in the case when the existing method can be applied,

due to their extrinsic viewpoint, important intrinsic properties of the data may not be represented in the dictionary. This can be illustrated by a simple example that it is possible that two points x, y on the manifold $\mathcal{M}$ have a large geodesic distance separating them but under the embedding $i : \mathcal{M} \to \mathbb{R}^d$, $i(x), i(y)$ has a small distance in $\mathbb{R}^d$. Therefore, sparse coding using dictionary learned in $\mathbb{R}^d$ is likely to code $i(x), i(y)$ (and hence x, y) using the same set of atoms with similar coefficients. Clearly, this will be undesirable and unsatisfactory if the applications require tasks such as classification and clustering, for which one would prefer the sparse coding to reflect some degree of actual similarity (i.e., geodesic distance) between the two samples x, y.

While the above example provides the motivation for seeking an extension to the existing dictionary learning framework to the more general Riemannain setting, it is by no means obvious how the extension should be correctly formulated. Let $\mathcal{M}$ denote the Riemannian manifold on which a collection of data points $x_1, \cdots, x_n$ are given. At the minimum, the goal of dictionary learning on $\mathcal{M}$ is to compute a collection of atoms $\{a_1, \cdots, a_m\} \subset \mathcal{M}$, also points on $\mathcal{M}$, such that each data point x_i can be generated using only a small number of atoms (sparsity). In the Euclidean setting, this is usually formulated as

$$\min_{D, w_1, \cdots, w_n} \sum_{i=1}^{n} \|x_i - Dw_i\|^2 + \mathbf{Sp}(w_i), \tag{5–1}$$

where D is the matrix with columns composed of the atoms a_i, w_i the sparse coding coefficients and $\mathbf{Sp}(w_i)$ the sparsity promoting term. One immediate technical hurdle that any satisfactory generalization needs to overcome is the lack of a global linear structure that will allow the data to be generated from the atoms. Instead, the Riemannian geometry provides only local linear structures through the Riemannian exponential and logarithm maps [59], and by moving to the more general Riemannian setting, we essentially trade the unique global linear structure with infinitely many local linear structures, which is the main source of the various technical difficulties present in

our generalization. However, this diversity of linear structures also provides us with an opportunity to formulate the dictionary learning using a data specific approach.

Specifically, we will formally modify each summand in Equation 5–1 so that the sparse coding of a data x_i with respect to the atoms $\{a_1, \cdots, a_m\} \subset \mathcal{M}$ is now obtained by minimizing

$$\min_{w_i} \| \sum_{j=1}^{m} w_{ij} \log_{x_i} a_j \|_{x_i}^2 + \mathbf{Sp}(w_i), \tag{5–2}$$

with the important affine constraint that $\sum_{j=1}^{m} w_{ij} = 1$, where $w_i = (w_{i1}, \ldots, w_{im})^T$. That is, we are using the Riemannian exponential and logarithm maps at each data point x to define the generative process, and the sparse approximation of a given data point is first computed in its tangent space $T_x\mathcal{M}$ and then realized on $\mathcal{M}$ by applying the exponential map. We remark that this formulation is entirely coordinate-independent since each $\log_{x_i} a_j$ is coordinate-independent, and Equation 5–2 can be minimized using any local chart and its associated basis for $T_{x_i}\mathcal{M}$ (with a result that will be independent of these choices). Furthermore, although this affine constraint is not present in most existing dictionary learning algorithms, it is the consequence of our data-specific approach: a subspace S in a given coordinates system is represented as an affine subspace in a different coordinates system with a different origin. In short, Equation 5–2 is the direct generalization of linear sparsity condition with the exception now the origin has been moved to the data point x_i. Computationally, the resulting optimization problem using Equation 5–2 can be effectively minimized. In particular, the gradient of the cost function in some cases admits closed-form formulas and in general, they can be evaluated numerically to provide the required inputs for the gradient-based optimization algorithm on manifolds [19].

To validate the proposed method, we have applied the dictionary learning framework to several classification problems in computer vision and medical imaging analysis. The feature vectors specific for these problems belong to several well-known Riemannian manifolds, and the experimental results show that the dictionary learned

using the proposed framework can and does indeed provide real improvements when compared with other direct approaches. Finally, before embarking on presenting the details of the proposed method, we mention that for technical reasons, we will assume that the Riemannian manifold $\mathcal{M}$ under consideration is complete and for each point $x \in \mathcal{M}$, the Riemannian log map $\log_x$ at x can be defined on $\mathcal{M}$ outside of a subset of codimension at least one. This technical condition is to ensure that for almost every arbitrary point $y \in \mathcal{M}$, $\log_x y$ can be uniquely defined, and for the Riemannian manifolds considered in this chapter, this condition will always be satisfied.

5.2 Related Work

The sparse coding framework which approximates the observed signals with the linear combination of a small number of atoms from a fixed dictionary has been successfully used in many computer vision and pattern recognition applications [77]. Dictionary learning proposed in the seminal work by Olshausen and Field [50] seeks to learn a codebook leading to the sparse decomposition of the observations. In recent research, it has been shown that the learned dictionary gives superior performance in many computer vision tasks such as image denoising [2] and image classification [84]. In [42], Mairal et al. introduced an online dictionary learning algorithm which fits very large datasets better than the batch procedures. The combination of group-structured sparsity regularization and online dictionary learning has been discussed in [63]. Sra et al. proposed a generalized dictionary learning method for symmetric positive definite (SPD) matrices [60]. They did not consider the Riemannian structure of the SPD matrix space and the SPD matrix was approximated by nonnegative weighted sum of rank-1 matrices.

5.3 A Brief on Riemannian Manifold

We now present a very brief introduction to some well known material from Riemmanian manifolds and refer the reader to [59] for details. A manifold $\mathcal{M}$ of dimension d is a topological space that is locally homeomorphic to open subsets of

the Euclidean space $\mathbb{R}^d$. With a globally defined differential structure, manifold $\mathcal{M}$ becomes a differentiable manifold. The tangent space at $x \in \mathcal{M}$ indicated by $T_x\mathcal{M}$ is a vector space that contains all the tangent vectors to $\mathcal{M}$ at x. A Riemannian metric on $\mathcal{M}$ associates to each point $x \in \mathcal{M}$ a differentiable varying inner product $\langle \cdot, \cdot \rangle_x$ in the tangent space $T_x\mathcal{M}$. Let x_i, x_j be two points on the manifold $\mathcal{M}$. The geodesic curve $\gamma : [0, 1] \to \mathcal{M}$ is a smooth curve with minimum length connecting x_i and x_j. Let $v \in T_x\mathcal{M}$ be a tangent vector to the manifold at x. There exists a unique geodesic γ_v satisfying $\gamma_v(0) = x$ with initial tangent vector v. Then the exponential map is defined as $\exp_x(v) = \gamma_v(1)$. The inverse of the exponential map $\exp_x$ is called the log map and denoted by $\log_x : \mathcal{M} \to T_x\mathcal{M}$. Given two points $x_i, x_j \in \mathcal{M}$, $\log_{x_i}$ maps the point x_j to the unique tangent vector at x_i that is the initial velocity of the geodesic γ with $\gamma(0) = x_i$ and $\gamma(1) = x_j$. The geodesic distance between x_i and x_j is computed by $\mathrm{dist}(x_i, x_j) = \| \log_{x_i}(x_j) \|_{x_i}$.

Given a set of points $x_1, \ldots, x_n$ on a Riemannian manifold $\mathcal{M}$, we can calculate the intrinsic mean on $\mathcal{M}$ by solving the optimization problem

$$\bar{x} = \arg\min_{x \in \mathcal{M}} \frac{1}{n} \sum_{i=1}^{n} \mathrm{dist}(x, x_i)^2. \tag{5–3}$$

The properties of the Riemannian intrinsic mean have been studied by Karcher [35]. In general, the intrinsic mean $\bar{x}$ may not have a closed form. Thus optimization algorithms on manifolds are usually used to compute $\bar{x}$ [1, 51].

5.4 Dictionary Learning on Riemannian Manifolds

Given a collection of signals $x_1, \ldots, x_n \in \mathbb{R}^d$, classical dictionary learning methods [2, 40, 49] try to find a dictionary $D \in \mathbb{R}^{d \times m}$ which includes m atoms such that each signal x_i can be represented as a sparse linear combination of these atoms $x_i \approx Dw_i$, where $w_i \in \mathbb{R}^m$. As others [42, 84], we formulate the dictionary learning problem using l_1 regularization on w_i

$$\min_{D, w_i} \sum_{i=1}^{n} \left(\|x_i - Dw_i\|_2^2 + \lambda \|w_i\|_1 \right) \tag{5–4}$$

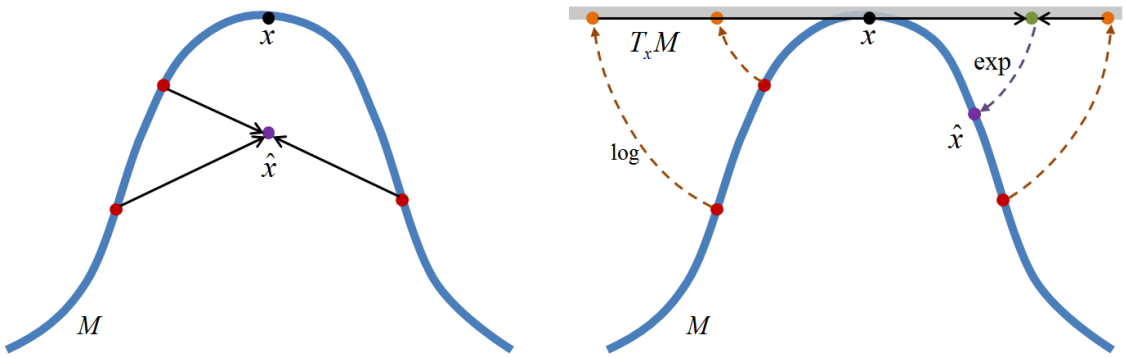


Figure 5-1. The blue curve denotes a one-dimensional manifold $\mathcal{M}$. x is a data point on $\mathcal{M}$ and the red points represent the dictionary atoms. **Left:** the linear approximation $\hat{x}$ with the dictionary atoms may not be on the manifold $\mathcal{M}$. **Right:** our approach projects the atoms to the tangent space at x and performs the linear combination on the vector space $T_x\mathcal{M}$. The exponential map guarantees the approximation $\hat{x}$ is always on $\mathcal{M}$.

where λ is a regularization parameter. Recent extensions of the classical dictionary learning include online dictionary learning [42] and using different regularization terms such as group-structured sparsity [63] and local coordinate constraint [74].

We generalize the classical dictionary learning techniques in Euclidean space to Riemannian manifolds. Suppose $\mathcal{M}$ represents a Riemannian manifold. Let $x_1, \dots, x_n \in \mathcal{M}$ be a collection of n data points on the manifold $\mathcal{M}$. Let $a_1, \dots, a_m \in \mathcal{M}$ be atoms of the learned dictionary $\mathcal{D} = \{a_1, \dots, a_m\}$. Because of the geometric structure of $\mathcal{M}$, we cannot use the linear combination of atoms $\hat{x}_i = \sum_{j=1}^{m} w_{ij} a_j$ to represent the data x_i, since $\hat{x}_i$ may not even be on the manifold $\mathcal{M}$. Thus we use the geodesic linear interpolation on Riemannian manifold, and x_i can be represented by

$$\hat{x}_i = \exp_{x_i}\left(\sum_{j=1}^{m} w_{ij} \log_{x_i}(a_j)\right) \tag{5–5}$$

where $\exp_{x_i}$ and $\log_{x_i}$ are exponential and logarithmic maps at x_i, and $w_{ij} \in \mathbb{R}$ are weights. Note that in order to approximate data point x_i, we project all the atoms in the dictionary to the tangent space at x_i. Because $T_{x_i}\mathcal{M}$ is a vector space, we can perform linear combination $v_i = \sum_{j=1}^{m} w_{ij} \log_{x_i}(a_j)$ on $T_{x_i}\mathcal{M}$. Then the approximation $\hat{x}_i$ can be

represented as the exponential map of v_i at x_i. A 1D case is illustrated in Figure 5-1. We hope to build a dictionary that minimizes the sum of reconstruction error for each data point. Here we define

$$\begin{aligned} E_{\text{data}} &= \sum_{i=1}^{n} \text{dist}(x_i, \hat{x}_i)^2 = \sum_{i=1}^{n} \| \log_{x_i}(\hat{x}_i) \|_{x_i}^2 \\ &= \sum_{i=1}^{n} \| \sum_{j=1}^{m} w_{ij} \log_{x_i}(a_j) \|_{x_i}^2. \end{aligned} \tag{5–6}$$

By using the l_1 sparsity regularization, the dictionary learning on the manifold $\mathcal{M}$ can be written as the following optimization problem

$$\begin{aligned} &\min_{\mathbf{W}, \mathcal{D}} \sum_{i=1}^{n} \| \sum_{j=1}^{m} w_{ij} \log_{x_i}(a_j) \|_{x_i}^2 + \lambda \|\mathbf{W}\|_1 \\ &s.t. \sum_{j=1}^{m} w_{ij} = 1, i = 1, \dots, n \end{aligned} \tag{5–7}$$

where $\mathbf{W} \in \mathbb{R}^{n \times m}$ and the (i, j) entry of $\mathbf{W}$ is written as w_{ij}. The affine constraint $\sum_{j=1}^{m} w_{ij} = 1$ used in our formulation has also appeared in several recent papers on dictionary learning such as sparse subspace clustering [20] and local coordinate coding [86]. The affine constraint means that we are using affine subspaces to approximate the data instead of the usual subspaces, which are simply affine subspaces based at the origin. Moving away from vector spaces to general Riemannian manifolds, there is no corresponding notion of the origin that can be used to define subspaces, and this simple geometric fact requires the abandonment of the usual subspaces in favor of general affine subspaces. We can also introduce other regularizations to our dictionary learning framework instead of the l_1 norm. For example, the localization regularization used in local coordinate coding [86] can be generalized to $E_{\text{reg}} = \sum_{i=1}^{n} \sum_{j=1}^{m} |w_{ij}| \| \log_{x_i}(a_j) \|_{x_i}^2$. Similar to classical dictionary learning methods, we could use the iterative method to solve this optimization problem:

1. **Sparse coding step:** fix the dictionary $\mathcal{D}$ and optimize with respect to the coefficients $\mathbf{W}$.

2. **Codebook optimization step:** fix **W** and optimize with respect to the dictionary $\mathcal{D}$.

The first step becomes a regular sparse coding problem and many fast algorithms can be used to solve it. The second subproblem is much more challenging, because the dictionary atoms are on the manifold $\mathcal{M}$ and the optimization methods in Euclidean space are not appropriate any more.

Algorithm 9 K-means on manifold $\mathcal{M}$

Input: A set of data $\mathcal{X} = \{x_1, \dots, x_n\}$ on the manifold $\mathcal{M}$ and the number of clusters K.
Output: Clusters $c_1, \dots, c_K$ and a partition of the input data $\mathcal{L} = \{l_1, \dots, l_n\}$, where each label $l_i \in \{1, \dots, K\}$

1. Randomly select K elements from $\mathcal{X}$ as the initialization of clusters.

2. For each data x_i, set the label as

$$l_i = \arg\min_k \operatorname{dist}(x_i, c_k).$$

3. For each cluster k, let $\mathcal{X}_k = \{x_i | l_i = k\}$, the mean can be computed by

$$c_k = \arg\min_{c \in \mathcal{M}} \sum_{x \in \mathcal{X}_k} \operatorname{dist}(c, x)^2.$$

4. Stop if $\mathcal{L}$ does not change, otherwise go to step 2.

We present a line search based algorithm on Riemannian manifold to update the dictionary $\mathcal{D}$. Let $f(a_1, \dots, a_m)$ be the cost function to be minimized. At first, we need to initialize the atoms in dictionary. One possible initialization is using m clusters of the data $x_1, \dots, x_n$ generated by K-means algorithm. Because of the manifold structure, K-means on $\mathcal{M}$ shown in Algorithm 9 is a little different from the traditional K-means algorithm in Euclidean space. Then we use the line search on manifold to optimize $f(a_1, \dots, a_m)$. The basic idea is to find a descent direction v on the tangent space, and then walk a step along the geodesic γ whose initial velocity is v. The details are listed in Algorithm 10. The convergence analysis of the line search method on manifold is discussed in [1].

In the next two sections, we give two concrete examples and use our dictionary learning framework on symmetric positive definite (SPD) matrix space and square-root density function space.

Algorithm 10 Line search on Riemannian manifold

Input: A set of data $\mathcal{X} = \{x_1, \dots, x_n\}$ on the manifold $\mathcal{M}$, coefficients $\mathbf{W} \in \mathbb{R}^{n\times m}$ and initial dictionary atoms $a_1^0, \dots, a_m^0$.
Output: $(a_1^*, \dots, a_m^*)$ that minimize the cost function $f(a_1, \dots, a_m)$.

1. Set scalars $\alpha > 0$, $\beta, \sigma \in (0, 1)$ and initialize $k = 0$.
2. Compute $\mathrm{grad} f(a_1^k, \dots, a_m^k) = (\frac{\partial f(a_1^k)}{\partial a_1}, \dots, \frac{\partial f(a_m^k)}{\partial a_m})$
3. Pick $\eta^k = (\eta_1^k, \dots, \eta_m^k) = -\mathrm{grad} f$, where $\eta_i^k \in T_{a_i^k}\mathcal{M}$.
4. Find the smallest s such that

$$f(\exp_{a_1^k}(\alpha\beta^s\eta_1^k), \dots, \exp_{a_m^k}(\alpha\beta^s\eta_m^k))$$

$$\leq f(a_1^k, \dots, a_m^k) - \sum_{i=1}^{m} \sigma\alpha\beta^s \|\eta_i^k\|_{a_i^k}.$$

5. Set $a_i^{k+1} = \exp_{a_i^k}(\alpha\beta^s\eta_i^k)$, $i = 1, \dots, m$.
6. Stop if f does not change much, otherwise set $k = k + 1$ and go back to step 2.

5.5 Symmetric Positive Definite Matrix Space

Let $P(d)$ represent the $d \times d$ symmetric positive definite (SPD) matrix space which forms a connected Riemannian manifold [23, 51]. The Riemannian symmetric space $P(d)$ arises naturally from the group action of the general linear group $GL(d)$. The group action of $GL(d)$ on $P(d)$ is the transformation $\phi : GL(d) \times P(d) \to P(d)$ given by $\phi_G(X) = GXG^T$, where $G \in GL(d)$ is a $d \times d$ invertible matrix and $X \in P(d)$ is a $d \times d$ SPD matrix. The tangent space $T_X P(d)$ at every point $X \in P(d)$ can be identified with $\mathrm{Sym}(d)$, which represents the $d \times d$ symmetric matrix space.

Let $Y, Z \in T_M P(d)$ be two tangent vectors at $M \in P(d)$. The Riemannian metric gives the inner product as

$$\langle Y, Z\rangle_M = \mathrm{tr}(YM^{-1}ZM^{-1}) \tag{5–8}$$

where tr represents the matrix trace. The geodesic passing through $M \in P(d)$ in the direction of $Y \in T_M P(d)$ with respect to the inner product defined above is uniquely given by

$$\gamma(t) = G \exp\left(G^{-1} Y G^{-T} t\right) G^T \tag{5–9}$$

where $\exp$ represents the matrix exponential and $G \in GL(d)$ is any square root of M such that $M = GG^T$. Therefore the Riemannian exponential map at M which maps Y to a point in $P(d)$ can be represented as

$$\exp_M(Y) = G \exp\left(G^{-1} Y G^{-T}\right) G^T. \tag{5–10}$$

Given two positive definite matrices $X, M \in P(d)$, the Riemannian logarithmic map $\log_M : P(d) \to T_M P(d)$ is defined as

$$\log_M(X) = G \log(G^{-1} X G^{-T}) G^T \tag{5–11}$$

where $\log$ represents the matrix logarithm. Using the definition of log map in $P(d)$, the geodesic distance between M and X can be computed by

$$\mathrm{dist}(M, X) = \| \log_M(X) \| = \sqrt{\mathrm{tr}(\log^2(G^{-1} X G^{-T}))}. \tag{5–12}$$

In recent years, there is a growing interest on sparse coding, dictionary learning and nonnegative factorization of SPD matrix data [58, 80]. However, in these works an SPD matrix is represented as a nonnegative linear combination of the dictionary atoms, which does not respect the intrinsic Riemannian structure of the SPD matrix space discussed above. In this section, we present a dictionary learning method for symmetric positive definite matrices, which considers the intrinsic geometry of $P(d)$.

Let $X_1, \ldots, X_n \in P(d)$ be a collection of $d \times d$ SPD matrices. $A_1, \ldots, A_m \in P(d)$ are m atoms in the learned dictionary $\mathcal{D}$. $\mathbf{W}$ is a $n \times m$ matrix. Different from the linear combination approximation of a point X_i in $P(d)$, our representation with intrinsic

Riemannian structure can be written as

$$\hat{X}_i = G_i \exp\left(\sum_{j=1}^{m} w_{ij} \log(G_i^{-1} A_j G_i^{-T})\right) G_i^T \tag{5–13}$$

where $G_i \in GL(d)$ such that $G_i G_i^T = X_i$.

If we just use l_1 regularization, the dictionary learning for SPD matrices becomes

$$\begin{aligned} &\min_{\mathbf{W},\mathcal{D}} \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{m} w_{ij} w_{ik} \mathrm{tr}(L_{ij} L_{ik}) + \lambda \|\mathbf{W}\|_1 \\ &s.t. \sum_{j=1}^{m} w_{ij} = 1, i = 1, \dots, n \end{aligned} \tag{5–14}$$

where $L_{ij} = \log(G_i^{-1} A_j G_i^{-T})$, $i = 1, \dots, n$ and $j = 1, \dots, m$. This optimization problem can be solved using the algorithm discussed in section 4.

5.6 Square-root Density Function Space

Probability density functions (pdfs) as a class of constrained non-negative functions have been widely used in many computer vision applications such as texture analysis and shape analysis. Without loss of generality, we restrict to the pdfs defined on the interval $[0, T]$ in this section:

$$\mathcal{P} = \{p : [0, T] \to \mathbb{R} | \forall s, p(s) \geq 0, \int_0^T p(s) ds = 1\}. \tag{5–15}$$

The Fisher-Rao metric has been introduced to study the Riemannian structure formed by the statistical manifold in [52]. For a pdf $p_i \in \mathcal{P}$, the Fisher-Rao metric is defined as

$$\langle v_j, v_k \rangle = \int_0^T v_j(s) v_k(s) \frac{1}{p_i(s)} ds \tag{5–16}$$

where $v_j, v_k \in T_{p_i}(\mathcal{P})$. The Fisher-Rao metric is invariant to reparameterizations of the functions. In order to make the resulting manifold easy to compute on with Riemannian operations, the square root density representation $\psi = \sqrt{p}$ was used [61]. The space of

square root density functions is defined as

$$\Psi = \{\psi : [0, T] \to \mathbb{R} | \forall s, \psi(s) \geq 0, \int_0^T \psi^2(s) ds = 1\} \tag{5–17}$$

As we can see, Ψ forms a convex subset of the unit sphere in a Hilbert space. Then the Fisher-Rao metric can be obtained as

$$\langle v_j, v_k \rangle = \int_0^T v_j(s) v_k(s) ds \tag{5–18}$$

where $v_j, v_k \in T_{\psi_i}\Psi$ are tangent vectors. Given any two functions $\psi_i, \psi_j \in \Psi$, the geodesic distance between these two points is

$$\text{dist}(\psi_i, \psi_j) = \cos^{-1}(\langle \psi_i, \psi_j \rangle) \tag{5–19}$$

which is just the angle between ψ_i and ψ_j on the unit hypersphere. The geodesic at ψ_i with a direction $v \in T_{\psi_i}(\Psi)$ is defined as

$$\gamma(t) = \cos(t)\psi_i + \sin(t)\frac{v}{|v|}. \tag{5–20}$$

Then the exponential map can be represented as

$$\exp_{\psi_i}(v) = \cos(|v|)\psi_i + \sin(|v|)\frac{v}{|v|}. \tag{5–21}$$

To ensure the exponential map is a bijection, we restrict $|v| \in [0, \pi)$. The log map is then given by

$$\log_{\psi_i}(\psi_j) = u \cos^{-1}(\langle \psi_i, \psi_j \rangle) / \sqrt{\langle u, u \rangle} \tag{5–22}$$

where $u = \psi_j - \langle \psi_i, \psi_j \rangle \psi_i$.

Using the geometric expressions of Ψ discussed above, we can perform the dictionary learning on square-root density functions. Let $x_1, \ldots, x_n \in \Psi$ be a collection of square-root density functions, and $a_i, \ldots, a_m \in \Psi$ be atoms in the dictionary $\mathcal{D}$. $\mathbf{W}$ is a

$n \times m$ matrix. If we use l_1 regularization, our dictionary learning framework becomes

$$\begin{aligned} &\min_{\mathbf{W},\mathcal{D}} \sum_{i=1}^{n} \| \sum_{j=1}^{m} w_{ij} \cos^{-1}(\langle x_i, a_j \rangle) \frac{u_{ij}}{|u_{ij}|} \|_{x_i}^2 + \lambda \|\mathbf{W}\|_1 \\ &s.t. \sum_{j=1}^{m} w_{ij} = 1, i = 1, \dots, n. \end{aligned} \tag{5–23}$$

where $u_{ij} = a_j - \langle x_i, a_j \rangle x_i$. This optimization problem can be efficiently solved using the algorithm presented in section 5.4.

5.7 Experiments

We evaluated our dictionary learning algorithms on classification experiments. Let $x_1, \dots, x_n \in \mathcal{M}$ be a collection of n training data. At first a codebook $\mathcal{D} = \{a_1, \dots, a_m\}$ is learned from the training data by using the dictionary learning technique proposed in this chapter, where each a_i denotes an atom in the dictionary $\mathcal{D}$. Then for each x_i, we generate the sparse code $\mathbf{w}_i \in \mathbb{R}^m$ based on the learned dictionary $\mathcal{D}$. Note that the sparse coding on Riemannian manifold is just one step in the dictionary learning algorithm with the codebook $\mathcal{D}$ fixed. Similar to [84], a linear SVM is trained for classification. In the testing stage, let $y_1, \dots, y_k \in \mathcal{M}$ be a set of k testing data. The sparse coding can be performed on the testing data in a similar manner. Given each sparse code as a feature vector, the trained SVM model can be used to classify the testing data.

In our experiments, three methods are used for comparison purposes.

- Geodesic K-nearest neighbor (GKNN)
- Support Vector Machine (SVM) on vectorized data
- SVM on sparse codes with the codebook trained by KSVD

GKNN is the K-nearest neighbor classification using the Riemannian geodesic distance instead of the traditional Euclidean distance. It's commonly used in literature for classification on manifolds. K is set to 5 in our experiments. Since the feature vector for SVM should be in Euclidean space, we vectorize the data on manifolds and feed

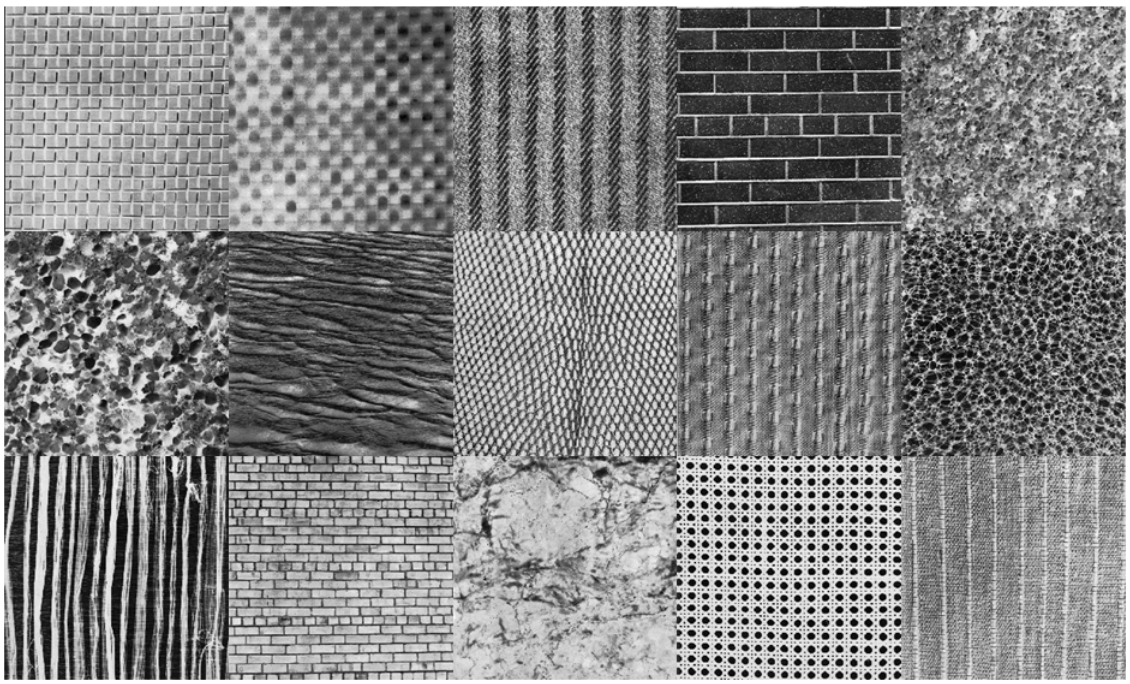


Figure 5-2. Samples of Brodatz textures used in our experiments.

them to the SVM classifier. The LIBSVM [11] package is used in our experiments. We also apply the popular KSVD method [2] to train a dictionary for sparse coding, and then a linear SVM on sparse codes is employed in classification. By comparing the proposed method with the dictionary learning method in vector space, we try to discover whether keeping the Riemannian structure is important in the classification. The experimental comparisons are performed on two public available datasets: Brodatz texture dataset and OASIS brain MRI dataset.

5.7.1 Brodatz dataset

Table 5-1. The texture classification accuracy for different methods on the Brodatz dataset.

	SVM	KSVD+SVM	GKNN	Proposed
16 classes	93.36	94.92	95.70	**99.02**
32 classes	88.67	90.82	91.11	**95.70**

In this experiment, we evaluate the dictionary learning algorithm for SPD matrices. We use region covariance descriptor introduced in [70] for texture classification on

Brodatz dataset [9]. With the similar experimental setting in [58], we construct 16-texture and 32-texture sets using the images from Brodatz dataset. Some sample textures are shown in Figure 5-2. Each 256×256 texture image is separated into 64 blocks of size 32×32. Inside each block, we can construct a 5×5 region covariance matrix using the feature

$$F = \left(I, \left| \frac{\partial I}{\partial x} \right|, \left| \frac{\partial I}{\partial y} \right|, \left| \frac{\partial^2 I}{\partial x^2} \right|, \left| \frac{\partial^2 I}{\partial y^2} \right| \right)^T . \tag{5–24}$$

For the existence of the derivative at each pixel, the texture images are pre-processed by gaussian filters. To make sure each covariance matrix strictly positive definite, we add σE, where σ is a very small positive constant and E denotes the identity matrix. For k-class problem, where k=16 or 32, we obtain $64k$ SPD matrices, which are equally partitioned to the training set and the testing set. The size of the dictionary $\mathcal{D}$ is set to $5k$. The texture classification results are reported in Table 5-1. Our dictionary learning based method performs better than other algorithms in comparison. As we can see, although the proposed method also uses a linear SVM to classify the feature vector in Euclidean space, the sparse code from learned dictionary preserves the geometric information better than the vector representation of SPD matrix.

5.7.2 OASIS dataset

Table 5-2. The classification accuracy for different methods on the OASIS dataset.

	SVM	KSVD+SVM	GKNN	Proposed
Y vs. M	90.04	91.29	91.84	**98.34**
M vs. O	97.32	98.08	100	**100**
O vs. Y	98.12	99.46	100	**100**
Y vs. M vs. O	91.97	94.04	93.18	**98.62**

In this experiment, we evaluate the dictionary learning algorithm for square root densities on the freely available Open Access Series of Imaging Studies (OASIS) database [43]. OASIS contains T1 weighted MR brain images from a cross-sectional population of 416 subjects. Each MRI scan has a resolution of $176 \times 208 \times 176$ voxels. The ages of the subjects range from 18 to 96. We divide the OASIS population into

three groups. 175 subjects younger than 40 years old are designated as young (Y), and 66 subjects between 40 and 60 are designated as middle-aged (M). We consider other 195 subjects older than 60 as old adults (O). Because of the structural difference in the brain across different age groups, we can use the MR images to perform age group classification. At first, we align all the MR images in OASIS dataset using the nonrigid group-wise registration method described in [32]. For each image, we obtain a displacement field. Then the histogram of the displacement vectors is constructed in each image as the feature for classification [12]. In our experiment, the number of bins in each direction is set to $4 \times 4 \times 4$. Thus the 64 dimensional histogram is used as the feature vector for the KSVD and SVM classification, while the square root of the histogram is used in GKNN and our dictionary learning based method. The learned dictionary includes 100 atoms in our experiment. We use 5-fold cross validation and report the classification results in Table 5-2. Note that all four methods give pretty good classification accuracy for Young vs. Old and Middle-aged vs. Old, but our method outperforms the competing methods in the Young vs. Middle-aged experiment. Because the regional brain volume and cortical thickness of adults are relatively stable prior to reaching 60 years [44], the classification of young and middle-aged subjects is more challenging.

CHAPTER 6
CONCLUSIONS

Effective group-wise image analysis using the geometric structure of the data is a big topic. In this dissertation, we have presented several techniques along this direction for both Euclidean image space and non-Euclidean high-dimensional ambient space. Below, we will summarize the contributions.

One of the simple but effective ways to represent a collection of data is to construct a few templates. For instance, popular k-means clustering constructs k templates and uses this low dimensional representation to model the whole data set. With the similar idea, we have proposed a novel approach to multiple image atlas construction problem. A k-NN graph is employed to capture the geometric structure of the image data set. Unsupervised and semi-supervised graph partition methods enable us to build multiple atlases for heterogeneous image data sets. The atlases computed using the proposed algorithm preserve important image features and generally enjoy better image sharpness.

Some data such as diffusion tensor image and deformation tensor field lie in a non-Euclidean ambient space. Thus traditional group-wise image analysis methods in vector space are unable to be directly used to process these tensor valued data. We have presented a Rimannian geometric framework for tensor valued image analysis, which extends Principal Component Analysis (PCA) to the space of tensor fields. The geodesic submanifold generated by our method provides an effective dimensionality reduction of the tensor valued image data in the high-dimensional product space.

Similar to PCA, Nonnegative Matrix Factorization (NMF) also uses the linear combination of a set of bases to approximate each image data. The difference is that PCA requires the bases to be orthonormal, while NMF uses nonnegative constraints on both bases and coefficients. We have proposed a novel method as a generalization of NMF to approximate a collection of diffusion tensor images using nonnegative linear

combinations of basis tensor images. An efficient iterative optimization algorithm is proposed to solve this factorization problem.

In recent years, learned dictionary combined with sparse coding becomes a major tool for group-wise data analysis. Using the sparse regularization, it can automatically select linear subspaces spanned by a small number of bases from the dictionary to approximate the data. Existing dictionary learning algorithms rely heavily on the assumption that the data points are vectors in some Euclidean space. However, in many vision applications, features and data points often belong to some Riemannian manifold with its intrinsic metric structure that is potentially important and critical to the application. We developed a very general dictionary learning framework for data lying on some known Riemannnian manifolds. Using the local linear structures furnished by the Riemannian geometry, we have proposed a novel dictionary learning algorithm that can be considered as data-specific, a feature that is not present in the existing methods.

For the future work, we plan to explore more, innovative representations for group-wise image analysis with solid computational models and robust optimization methods.

REFERENCES

[1] Absil, P.A., Mahony, R., and Sepulchre, R. *Optimization algorithms on matrix manifolds*. Princeton Univ Pr, 2008.

[2] Aharon, M., Elad, M., and Bruckstein, A. "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation." *Signal Processing, IEEE Transactions on* 54 (2006).11: 4311–4322.

[3] Ashburner, J., Hutton, C., Frackowiak, R., Johnsrude, I., Price, C., and Friston, K. "Identifying global anatomical differences: deformation-based morphometry." *Human Brain Mapping* 6 (1998).5-6: 348–357.

[4] Basser, P.J., Mattiello, J., and LeBihan, D. "MR diffusion tensor spectroscopy and imaging." *Biophysical journal* 66 (1994).1: 259–267.

[5] Bazin, P.L. and Pham, D.L. "Statistical and topological atlas based brain image segmentation." *Proceedings of the 10th international conference on Medical image computing and computer-assisted intervention*. 2007, 94–101.

[6] Belkin, M. and Niyogi, P. "Laplacian eigenmaps and spectral techniques for embedding and clustering." *Advances in neural information processing systems* 1 (2002): 585–592.

[7] ———. "Semi-supervised learning on Riemannian manifolds." *Machine Learning* 56 (2004).1: 209–239.

[8] Boyd, A. and Vandenberghe, L. *Convex Optimization*. Cambridge University Press, 2004.

[9] Brodatz, P. *Textures: a photographic album for artists and designers*, vol. 66. Dover New York, 1966.

[10] Cetingul, H.E. and Vidal, R. "Intrinsic mean shift for clustering on Stiefel and Grassmann manifolds." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009, 1896–1902.

[11] Chang, Chih-Chung and Lin, Chih-Jen. "LIBSVM: A library for support vector machines." *ACM Transactions on Intelligent Systems and Technology* 2 (2011): 27:1–27:27.

[12] Chen, T., Rangarajan, A., and Vemuri, B.C. "Caviar: Classification via aggregated regression and its application in classifying oasis brain database." *IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI)*. 2010, 1337–1340.

[13] Cootes, T.F., Edwards, G.J., and Taylor, C.J. "Active appearance models." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23 (2001).6: 681–685.

[14] Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J., et al. "Active shape models-their training and application." *Computer vision and image understanding* 61 (1995).1: 38–59.

[15] Cormen, T.H., Leiserson, C.E., Rivest, R.L., and Stein, C. *Introduction to algorithms*. The MIT press, 2001.

[16] Davis, B., Fletcher, P., Bullitt, E., and Joshi, S. "Population Shape Regression From Random Design Data." *IEEE International Conference on Computer Vision (ICCV)*. 2007.

[17] Ding, C., He, X., and Simon, H.D. "On the equivalence of nonnegative matrix factorization and spectral clustering." *Proc. SIAM Data Mining Conf*. 2005.

[18] Donoho, D.L. and Stodden, V. "When does non-negative matrix factorization give a correct decomposition into parts?" *NIPS*. 2004.

[19] Edelman, A., Arias, T.A., and Smith, S.T. "The geometry of algorithms with orthogonality constraints." *Arxiv preprint physics/9806030* (1998).

[20] Elhamifar, E. and Vidal, R. "Sparse subspace clustering." *CVPR*. 2009, 2790–2797.

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

[22] Fletcher, P., Venkatasubramanian, S., and Joshi, S. "Robust statistics on Riemannian manifold via the geometric median." *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*. 2008.

[23] Fletcher, P.T. and Joshi, S. "Riemannian geometry for the statistical analysis of diffusion tensor data." *Signal Processing* 87 (2007).2: 250–262.

[24] Fletcher, P.T., Lu, C., Pizer, S.M., and Joshi, S. "Principal geodesic analysis for the study of nonlinear statistics of shape." *IEEE Transactions on Medical Imaging* 23 (2004).8: 995–1005.

[25] Gaussier, E. and Goutte, C. "Relation between PLSA and NMF and implications." *ACM SIGIR*. 2005.

[26] Gerber, S., Tasdizen, T., Thomas Fletcher, P., Joshi, S., and Whitaker, R. "Manifold modeling for brain population analysis." *Medical image analysis* 14 (2010).5: 643–653.

[27] Grippo, L. and Sciandrone, M. "On the convergence of the block nonlinear gauss-seidel method under convex constraints." *Operations Research Letters* 26 (2000).3: 127–136.

[28] Hagen, L. and Kahng, A.B. “New spectral methods for ratio cut partitioning and clustering.” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 11 (1992).9: 1074–1085.

[29] Hamm, J., Ye, D.H., Verma, R., and Davatzikos, C. “GRAM: A framework for geodesic registration on anatomical manifolds.” *Medical Image Analysis* 14 (2010).5: 633–642.

[30] Hua, X., Leow, A.D., Parikshak, N., Lee, S., Chiang, M.C., Toga, A.W., Jack Jr, C.R., Weiner, M.W., and Thompson, P.M. “Tensor-based morphometry as a neuroimaging biomarker for Alzheimer’s disease: an MRI study of 676 AD, MCI, and normal subjects.” *Neuroimage* 43 (2008).3: 458–469.

[31] Jia, H., Wu, G., Wang, Q., and Shen, D. “ABSORB: Atlas building by self-organized registration and bundling.” *NeuroImage* 51 (2010).3: 1057–1070.

[32] Joshi, S., Davis, B., Jomier, M., and Gerig, G. “Unbiased Diffeomorphic Atlas Construction for Computational Anatomy.” *NeuroImage* 23 (2004): S151–S160.

[33] Joshi, S., Karthikeyan, S., Manjunath, BS, Grafton, S., and Kiehl, K.A. “Anatomical parts-based regression using non-negative matrix factorization.” *CVPR*. 2010.

[34] J.Tenenbaum, de Silva, V., and Langford, J. “A global geometric framework for nonlinear dimensionality reduction.” *Science* 290(5500) (2000): 2319–2323.

[35] Karcher, H. “Riemannian center of mass and mollifier smoothing.” *Communications on pure and applied mathematics* 30 (1977).5: 509–541.

[36] Lee, D.D. and Seung, H.S. “Learning the parts of objects by non-negative matrix factorization.” *Nature* 401 (1999): 788–791.

[37] Lenglet, C., Rousson, M., and Deriche, R. “DTI segmentation by statistical surface evolution.” *IEEE Trans. on Medical Imaging* 25 (2006).6: 685–700.

[38] Leow, A.D., Klunder, A.D., Jack Jr, C.R., Toga, A.W., Dale, A.M., Bernstein, M.A., Britson, P.J., Gunter, J.L., Ward, C.P., Whitwell, J.L., et al. “Longitudinal stability of MRI for mapping brain change using tensor-based morphometry.” *Neuroimage* 31 (2006).2: 627–640.

[39] Lepore, N., Brun, C., Chou, Y.Y., Chiang, M.C., Dutton, R.A., Hayashi, K.M., Luders, E., Lopez, O.L., Aizenstein, H.J., Toga, A.W., et al. “Generalized tensor-based morphometry of HIV/AIDS using multivariate statistics on deformation tensors.” *IEEE Transactions on Medical Imaging* 27 (2008).1: 129–141.

[40] Lewicki, M.S. and Sejnowski, T.J. “Learning overcomplete representations.” *Neural computation* 12 (2000).2: 337–365.

[41] Li, S.Z., Hou, X.W., Zhang, H.J., and Cheng, Q.S. “Learning spatially localized, parts-based representation.” *CVPR*. 2001.

[42] Mairal, J., Bach, F., Ponce, J., and Sapiro, G. "Online learning for matrix factorization and sparse coding." *The Journal of Machine Learning Research* 11 (2010): 19–60.

[43] Marcus, D., Wang, T., Parker, J., Gsernansky, J.G., Morris, J., and Buckner, R. "Open Access Series of Imaging Studies (OASIS): Cross-Section MRI Data in Young, Middle Aged, Nondemented, and Demented Older Adults." *Journal of Cognitive Neuroscience* 19 (2007): 1498–1507.

[44] Mortamet, B., Zeng, D., Gerig, G., Prastawa, M., and Bullitt, E. "Effects of healthy aging measured by intracranial compartment volumes using a designed MR brain database." *MICCAI*. 2005.

[45] Mumford, D., Fogarty, J., and Kirwan, F. *Geometric Invariant Theory*. Springer, 1994.

[46] Munkres, J. *Topology*. Prentice Hall, 2000.

[47] Nene, S.A., Nayar, S.K., and Murase, H. "Columbia object image library (coil-20)." *Technical Report 005-96, CUCS* (1996).

[48] Nocedal, J. and Wright, S.J. *Numerical optimization*. Springer, 2000.

[49] Olshausen, B.A. and Field, D.J. "Sparse coding with an overcomplete basis set: A strategy employed by V1?" *Vision research* 37 (1997).23: 3311–3325.

[50] Olshausen, B.A. et al. "Emergence of simple-cell receptive field properties by learning a sparse code for natural images." *Nature* 381 (1996).6583: 607–609.

[51] Pennec, X., Fillard, P., and N.Ayache. "A Riemannian Framework for Tensor Computing." *Int. J. Computer Vision* 66 (2006).1: 41–46.

[52] Rao, C.R. "Information and accuracy attainable in the estimation of statistical parameters." *Bull. Calcutta Math. Soc* 37 (1945).3: 81–91.

[53] Roweis, S.T. and Saul, L.K. "Nonlinear dimensionality reduction by locally linear embedding." *Science* 290 (2000).5500: 2323.

[54] Sabuncu, M., Balci, S., and Golland, P. "Discovering Modes of an Image Population through Mixture Modeling." *MICCAI*. 2008.

[55] Sabuncu, M.R., Balci, S.K., Shenton, M.E., and Golland, P. "Image-driven population analysis through mixture modeling." *IEEE Transactions on Medical Imaging* 28 (2009).9: 1473–1487.

[56] Schwartzman, A. *Random ellipsoids and false discovery rates: Statistics for diffusion tensor imaging data*. Ph.D. thesis, Stanford University, 2006.

[57] Shi, J. and Malik, J. "Normalized cuts and image segmentation." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 22 (2000).8: 888–905.

[58] Sivalingam, R., Boley, D., Morellas, V., and Papanikolopoulos, N. "Tensor sparse coding for region covariances." *ECCV*. 2010, 722–735.

[59] Spivak, M. *A comprehensive introduction to differential geometry*. Publish or perish Berkeley, 1979.

[60] Sra, S. and Cherian, A. "Generalized dictionary learning for symmetric positive definite matrices with application to nearest neighbor retrieval." *Machine Learning and Knowledge Discovery in Databases*. 2011, 318–332.

[61] Srivastava, A., Jermyn, I., and Joshi, S. "Riemannian analysis of probability density functions with applications in vision." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2007.

[62] Sturm, J.F. "Using SeDuMi 1. 02, a MATLAB toolbox for optimization over symmetric cones." *Optimization methods and software* 11 (1999).1: 625–653.

[63] Szabo, Z., Poczos, B., and Lorincz, A. "Online group-structured dictionary learning." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2011, 2865–2872.

[64] Talairach, J. and Tournoux, P. *Co-planar stereotaxic atlas of the human brain*, vol. 147. Thieme New York, 1988.

[65] Thomas Yeo, BT, Sabuncu, M.R., Desikan, R., Fischl, B., and Golland, P. "Effects of registration regularization and atlas sharpness on segmentation accuracy." *Medical Image Analysis* 12 (2008).5: 603–615.

[66] Todd, MJ, Toh, KC, and Tutuncu, RH. "On the Nesterov-Todd direction in semidefinite programming." *SIAM Journal on Optimization* 8 (1998).3: 769–796.

[67] Toh, KC, Tutuncu, RH, and Todd, MJ. "Inexact primal-dual path-following algorithms for a special class of convex quadratic SDP and related problems." *Pac. J. Optim* 3 (2007): 135–164.

[68] Turk, M.A. and Pentland, A.P. "Face recognition using eigenfaces." *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. 1991, 586–591.

[69] Tutuncu, R.H., Toh, K.C., and Todd, M.J. "Solving semidefinite-quadratic-linear programs using SDPT3." *Mathematical Programming* 95 (2003).2: 189–217.

[70] Tuzel, O., Porikli, F., and Meer, P. "Region covariance: A fast descriptor for detection and classification." *European Conference on Computer Vision (ECCV)*. 2006, 589–600.

[71] Van Benthem, M.H. and Keenan, M.R. "Fast algorithm for the solution of large-scale non-negativity-constrained least squares problems." *Journal of chemometrics* 18 (2004).10: 441–450.

[72] Vapnik, V.N. *Statistical learning theory*. Wiley, New York, 1998.

[73] Vemuri, BC, Ye, J., Chen, Y., and Leonard, CM. "Image registration via level-set motion: Applications to atlas-based segmentation." *Medical image analysis* 7 (2003).1: 1–20.

[74] Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., and Gong, Y. "Locality-constrained linear coding for image classification." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2010, 3360–3367.

[75] Wang, Z. and Vemuri, B.C. "DTI segmentation using an information theoretic tensor dissimilarity measure." *IEEE Trans. on Medical Imaging* 24 (2005).10: 1267–1277.

[76] Weldeselassie, Y.T. and Hamarneh, G. "DT-MRI segmentation using graph cuts." *Proc. of SPIE Medical Imaging: Image Processing*. 2007.

[77] Wright, J., Ma, Y., Mairal, J., Sapiro, G., Huang, T.S., and Yan, S. "Sparse representation for computer vision and pattern recognition." *Proceedings of the IEEE* 98 (2010).6: 1031–1044.

[78] Wu, G., Jia, H., Wang, Q., and Shen, D. "SharpMean: Groupwise registration guided by sharp mean image and tree-based registration." *NeuroImage* (2011).

[79] Wu, J., Smith, WAP, and Hancock, ER. "Gender classification using shape from shading." *British machine vision conference*. 2007.

[80] Xie, Y., Ho, J., and Vemuri, B. "Nonnegative Factorization of Diffusion Tensor Images and Its Applications." *Information Processing in Medical Imaging (IPMI)*. Springer, 2011, 550–561.

[81] Xie, Y., Ho, J., and Vemuri, B.C. "Image atlas construction via intrinsic averaging on the manifold of images." *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*. 2010.

[82] Xie, Y., Vemuri, B., and Ho, J. "Statistical analysis of tensor fields." *Medical Image Computing and Computer-Assisted Intervention (MICCAI)* (2010): 682–689.

[83] Xu, W., Liu, X., and Gong, Y. "Document clustering based on non-negative matrix factorization." *ACM SIGIR*. 2003.

[84] Yang, J., Yu, K., Gong, Y., and Huang, T. "Linear spatial pyramid matching using sparse coding for image classification." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009, 1794–1801.

[85] Yeo, B.T.T., Vercauteren, T., Fillard, P., Peyrat, J.M., Pennec, X., Golland, P., Ayache, N., and Clatz, O. "DT-REFinD: Diffusion tensor registration with exact finite-strain differential." *IEEE Transactions on Medical Imaging* 28 (2009).12: 1914–1928.

[86] Yu, K., Zhang, T., and Gong, Y. "Nonlinear learning using local coordinate coding." *Advances in Neural Information Processing Systems (NIPS)* 22 (2009): 2223–2231.

[87] Zhang, H., Yushkevich, P.A., and Gee, J.C. "Registration of diffusion tensor images." *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. vol. 1. 2004.

[88] Zhu, X. "Semi-supervised learning literature survey." *Technical Report 1530 Department of Computer Science, University of Wisconsin, Madison.* (2005).

BIOGRAPHICAL SKETCH

Yuchen Xie was born in Datong, Shanxi, P.R. China. He received his Bachelor of Science degree from School of Mathematical Sciences at Peking University in 2006. He then joined the Ph.D. program of the Department of Computer and Information Science and Engineering at the University of Florida. His research interests include computer vision, machine learning, medical imaging and numerical optimization.