# Segmenting Fiber Bundles in Diffusion Tensor Images

Alvina Goh and René Vidal

Center for Imaging Science, Johns Hopkins University, Baltimore, MD 21218, USA

**Abstract.** We consider the problem of segmenting fiber bundles in diffusion tensor images. We cast this problem as a manifold clustering problem in which different fiber bundles correspond to different submanifolds of the space of diffusion tensors. We first learn a local representation of the diffusion tensor data using a generalization of the locally linear embedding (LLE) algorithm from Euclidean to diffusion tensor data. Such a generalization exploits geometric properties of the space of symmetric positive semi-definite matrices, particularly its Riemannian metric. Then, under the assumption that different fiber bundles are physically distinct, we show that the null space of a matrix built from the local representation gives the segmentation of the fiber bundles. Our method is computationally simple, can handle large deformations of the principal direction along the fiber tracts, and performs automatic segmentation without requiring previous fiber tracking. Results on synthetic and real diffusion tensor images are also presented.

## 1 Introduction

Diffusion Tensor Imaging (DTI) is a 3-D imaging technique that measures the restricted diffusion of water in living tissues. Water diffusion is represented mathematically with a symmetric positive semi-definite (SPSD) tensor field $D : \mathbb{R}^3 \to \mathrm{SPSD}(3) \subset \mathbb{R}^{3 \times 3}$ that measures the diffusion in a direction $\mathbf{v} \in \mathbb{R}^3$ as $\mathbf{v}^\top D \mathbf{v}$. The direction of maximum diffusion is indicative of the orientation of fibers in highly anisotropic tissues. Therefore, DTI can be used to analyze the local orientation and anisotropy of tissue structures, and infer the organization and orientation of tissue components. For example, DTI allows one to distinguish the different anatomical structures of the brain such as the corpus callosum, cingulum, or fornix, noninvasively.

In order to make DTI beneficial for both diagnostic as well as clinical applications, it is necessary to develop image analysis methods for registering DT images, extracting and tracking fibers, segmenting bundles of fibers with different orientation, etc. However, as the space of diffusion tensors is not Euclidean, traditional image analysis techniques need to be revised to handle the new mathematical structure of the data.

**Related work.** It is well-known [1, 2, 3, 4, 5] that the traditional Euclidean distance is not the most appropriate metric for the Riemannian symmetric space $\mathrm{SPSD}(r)$, where $r$ is the dimension of the matrices. This has motivated several frameworks for tensor computing that incorporate different Riemannian properties of SPSD matrices [1, 2, 3]. Applications to interpolation and filtering of tensor fields have shown encouraging results.

Although there exists extensive literature studying the problem of classifying gray matter, white matter and cerebrospinal fluid from MR images, there is relatively lesser

work done on the problem of segmenting different white matter structures from DTI. A first family of segmentation methods [6] reduces the tensor data to a scalar anisotropic measure and then applies standard image segmentation methods to the scalar data. However, reducing the tensor field to a scalar measure eliminates the directional information, thereby reducing the discriminative power. For instance, when two fiber bundles are oriented in different directions, but have the same anisotropy, this method will fail.

A second family of segmentation methods extracts the fiber tracts from the tensor data and then segments these tracts using a measure of similarity between pairs of curves, such as the Euclidean distance between two fibers, or the ratio of the length of corresponding portions of the fibers to the overall length of the pairs [7]. In [8], fibers are reduced to a feature vector extracted from the statistical moments of the fibers, and segmentation is done by applying normalized cuts to these feature vectors. Unfortunately, there are several problems with this approach. Most notably, accurate extraction of fiber tracts in the presence of noise in DT data remains an obstacle. Most tractography methods start at a user specified point and follow the direction of the principal eigenvector of the tensor until a termination criterion is reached. Observe that a slight error in the estimation of the principal tensor direction at one voxel will likely result in tracking a different fiber. This error will propagate as tracking continues, so the extracted fiber could be completely wrong. Also, it is known that the estimation of a tensor is poor in areas where two different fiber bundles cross at an angle. Thus, the likelihood of fiber tracking veering off course is high in regions of crossing fibers. Even if the fiber tracts were correctly estimated, comparing 3-D curves in a mathematical rigorous manner remains an open question. In order to overcome the shortcomings resulting from the local decision-making of the tractography methods, stochastic approaches [9, 10] use a measure of connectivity between brain regions. However, these methods do not give the explicit segmentation of the fiber bundles.

A third family of segmentation methods attempts to sidestep these issues by segmenting the tensor data directly, without first extracting the fiber tracts. These methods make use of a metric on $\mathrm{SPSD}(3)$, such as the Euclidean metric $\mathrm{trace}(\mathbf{D}_1 \mathbf{D}_2^\top)$ [11, 12], or the normalized tensor scalar product $\frac{\mathrm{trace}(\mathbf{D}_1 \mathbf{D}_2^\top)}{\mathrm{trace}(\mathbf{D}_1)\mathrm{trace}(\mathbf{D}_2)}$ [13]. These metrics are then combined with classical segmentation, such as spectral clustering [11, 14] or level set methods [12, 13]. However, as these methods are designed to segment discrete tensors rather than continuous fiber bundles, they fail to segment fiber bundles correctly whenever the tensors in a bundle present high variability, e.g., in a long curved tract. These issues have motivated the usage of more sophisticated metrics such as the log-Euclidean metric [1], the information theoretic metric [4] or the affine-invariant metric [2, 3, 15]. For example, in [4], the diffusion tensor is interpreted as the covariance matrix of a local Gaussian distribution. The distance measure between two matrices is based on the Kullback-Leibler divergence between the two Gaussian probability density functions induced by the two matrices. As the Kullback-Leibler divergence is not symmetric, the J-divergence which is the mean of the two divergences is used. More recently, locally-constrained region based methods that could handle variability in a fiber bundle have been proposed [16, 17, 18]. In these methods, fiber bundles are segmented by minimizing an energy function in a probabilistic framework. These energy minimization techniques, combined with different metrics, work well in general. However,
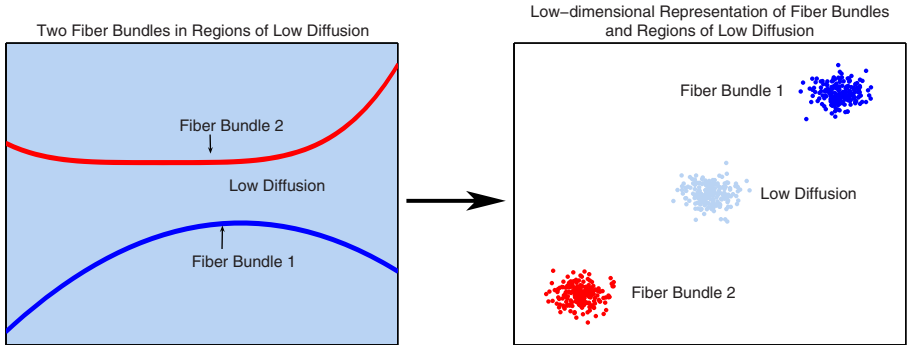
**Fig. 1.** Pictorial view of how our algorithm works. Left: Two fiber bundles surrounded by a region of low diffusion. Right: Low-dimensional representation learned by our algorithm, in which different fiber bundles form different clusters.

they suffer from convergence to a local minimum, and often require a user-specified initial segmentation.

**Paper contributions.** In this paper, we present an algorithm for segmenting fiber bundles in DT images. Our algorithm is algebraic, thus it requires no initialization; operates directly on tensor data, thus it requires no fiber extraction or tracking; and is designed to deal with long curved fiber bundles. More specifically, we assume that tensors at adjacent voxels within a fiber bundle are similar (similar eigenvectors and eigenvalues), while tensors at distant voxels could be very different, even if they lie in the same bundle. Under this assumption, we show that one can map each diffusion tensor to a point in a low-dimensional linear space in such a way that tensors tracing out a fiber bundle are mapped to nearby points, while different fiber bundles are mapped to distinct clusters, as shown in Fig. 1. This is achieved by using a new manifold clustering technique called Locally Linear Diffusion Tensor Clustering (LLDTC), which is a natural generalization of locally linear embedding (LLE) [19] from Euclidean to diffusion tensor data. The generalization is based on the Riemannian framework with the affine-invariant and log-Euclidean metrics [1, 3]. In particular, we will adopt the generic framework first proposed in [20] and extended to clustering of probability density functions in [21].

The combination of techniques from Riemannian geometry and manifold learning has already been used to perform statistical analysis and segmentation of diffusion MRI data [22] by considering the tensors as points lying on a single manifold. However, as noted in [23], diffusion MRI data belongs to a union of manifolds with different dimensions and densities. [23] shows that it is possible to characterize neuro-anatomical areas by considering the data as points clouds, and clustering these points into different groups by estimating the dimension and density around each data point based on its $k$ nearest neighbors. Our algorithm, while modeling the data as a union of manifolds, does not require different fiber bundles to have different dimensions to achieve clustering.

**Paper outline.** §2 reviews the classical LLE algorithm for a nonlinear manifold using the Euclidean metric. §3 presents the extension of LLE to the space of diffusion tensors

using the Riemannian framework. §4 shows how to segment DTI fiber bundles that correspond to different submanifolds using LLDTC framework. §5 presents experimental results on synthetic and real data, and §6 gives the conclusions.

## 2  Review of Locally Linear Embedding in Euclidean Spaces

Let $X = \{\mathbf{x}_i \in \mathbb{R}^D\}_{i=1}^n$ be a set of $n$ data points sampled from a low-dimensional manifold embedded in $\mathbb{R}^D$. The goal of nonlinear dimensionality reduction (NLDR) is to find a set of $n$ vectors $\{\mathbf{y}_i \in \mathbb{R}^d\}_{i=1}^n$, where $d \ll D$, such that nearby points remain close and distant points remain far. Existing NLDR techniques can be divided in two main groups. Global NLDR techniques, such as Isomap [26], try to preserve global properties of the data. Local NLDR techniques, such as LLE [19], Laplacian eigenmaps [24] and Hessian LLE [25], try to preserve local properties obtained from small neighborhoods around the datapoints. In particular, LLE exploits the fact that the local neighborhood of a point on the manifold can be well approximated by the affine subspace spanned by the $k$ nearest neighbors of the point. The key idea of LLE is to find a low-dimensional embedding of the data that preserves the coefficients of such affine approximations. More specifically, the LLE algorithm can be summarized as follows.

1. *Nearest neighbor search:* For each data point $\mathbf{x}_i \in X$, find its $k$ nearest neighbors ($k$NN) $\{\mathbf{x}_{i_j}\}_{j=1}^k$ according to the Euclidean distance.
2. *Least squares fit*: Find a matrix of weights $W \in \mathbb{R}^{n \times n}$ whose entries $W_{ij}$ minimize the reconstruction error

$$\varepsilon(W) = \sum_{i=1}^n \|\mathbf{x}_i - \sum_{j=1}^n W_{ij}\mathbf{x}_j\|^2 = \sum_{i=1}^n \|\sum_{j=1}^n W_{ij}(\mathbf{x}_i - \mathbf{x}_j)\|^2 = \sum_{i=1}^n \mathrm{dist}^2(\mathbf{x}_i, \widehat{\mathbf{x}}_i) \tag{1}$$

subject to the constraints (i) $W_{ij} = 0$ if $\mathbf{x}_j$ is not a $k$-nearest neighbor of $\mathbf{x}_i$ and (ii) $\sum_{j=1}^n W_{ij} = 1$. In (1), $\widehat{\mathbf{x}}_i = \mathbf{x}_i + \sum_{j=1}^n W_{ij}\overrightarrow{\mathbf{x}_i \mathbf{x}_j}$ is the linear interpolation of $\mathbf{x}_i$ and its $k$NN. The solution to this problem can be computed as

$$\begin{bmatrix} W_{i\,i_1} & W_{i\,i_2} & \cdots & W_{i\,i_k} \end{bmatrix} = \frac{\mathbf{1}^\top C_i^{-1}}{\mathbf{1}^\top C_i^{-1} \mathbf{1}} \in \mathbb{R}^{1 \times k}, \tag{2}$$

where $\mathbf{1}$ is the vector of all ones, and $C_i \in \mathbb{R}^{k \times k}$ is the local Gram matrix at $\mathbf{x}_i$, i.e. $C_i(j,l) = (\mathbf{x}_j - \mathbf{x}_i) \cdot (\mathbf{x}_l - \mathbf{x}_i)$. The matrix of weights $W$ is invariant to rotations, scalings and translations of each data point and its neighbors.
3. *Sparse eigenvalue problem*: Find vectors $\{\mathbf{y}_i \in \mathbb{R}^d\}_{i=1}^n$ that minimize the error

$$\phi(Y) = \sum_{i=1}^n \|\mathbf{y}_i - \sum_{j=1}^n W_{ij}\mathbf{y}_j\|^2 = \mathrm{trace}(Y^\top M Y), \tag{3}$$

where $Y = [\mathbf{y}_1, \ldots, \mathbf{y}_n]^\top \in \mathbb{R}^{n \times d}$, subject to the constraints (i) $\sum_{i=1}^n \mathbf{y}_i = \mathbf{0}$ (centered at the origin) and (ii) $\frac{1}{n}\sum_{i=1}^n \mathbf{y}_i \mathbf{y}_i^\top = I$ (unit covariance). The optimal

solution is the matrix $Y$ whose columns are the $d$ eigenvectors of the matrix $M = (I - W)^\top (I - W)$ associated with its second to $(d+1)$-st smallest eigenvalues. The first eigenvector of $M$ is discarded, because it is the vector of all ones, $\mathbf{1} \in \mathbb{R}^n$, with 0 as its eigenvalue. This is because $\sum_{j=1}^{n} W_{ij} = 1$, hence $W\mathbf{1} = \mathbf{1}$, and $M\mathbf{1} = \mathbf{0}$.

In principle, the LLE algorithm is designed for data lying in a single connected submanifold of Euclidean space. In the next sections we will show how LLE can be extended to data lying in multiple submanifolds of SPSD(3).

## 3   Locally Linear Embedding in the Space of SPSD Matrices

The LLE framework presented in §2 is applicable in the presence of one connected manifold with unknown structure. Therefore, every operation has to be approximated by the corresponding Euclidean operation, e.g., finding the $k$NN using the Euclidean distance, minimizing the Euclidean reconstruction error and doing Euclidean interpolation of a point and its neighbors. For diffusion tensors, however, the manifold structure is known. Moreover, previous work has shown that the Euclidean distance is not the most appropriate metric for SPSD matrices [1, 2, 3, 4, 5]. For example, in [3] it is clearly illustrated that Euclidean averaging of tensors leads to a *tensor swelling effect* in which the resulting determinant of the mean is larger than the original determinants.

In this section, we will show how to extend LLE to diffusion tensors using the affine-invariant and log-Euclidean metrics [1, 3] instead of the Euclidean metric. Under these metrics, closed-form formulae for Riemannian operations, such as the geodesic distance, geodesic interpolation, etc., are available. We will make use of the generic framework proposed in [20] for such closed-form Riemannian structures. Since the information about the local geometry of the manifold is essential only in the first two steps of the LLE algorithm, modifications are made only to these two stages, i.e. how to select the $k$NN and how to compute the matrix $W$ representing the local geometry of fiber bundles using the new metrics. Given $W$, the calculation of the low-dimensional representation remains the same as in the Euclidean case.

**Selection of Riemannian $k$NN.**  The first step of the LLE algorithm is the computation of the $k$NN associated with each data point. To that end, consider any two tensors $\mathbf{D}(x_1)$ and $\mathbf{D}(x_2)$ at coordinates $x_1$ and $x_2$. Notice that there are two ways of measuring similarity: the Riemannian metric between the tensors $\mu(\mathbf{D}(x_1), \mathbf{D}(x_2))$ and the Euclidean distance between the coordinates $\|x_1 - x_2\|$. A weighing factor has been used in [11] to control the trade-off between these two distances. However, since our objective here is to cluster fiber bundles, we must choose a distance accordingly. Clearly $\mu$ alone does not suffice, because two tensors in a bundle may be very different from each other. Since nearby tensors within a bundle are similar, we select the $k$NN of $\mathbf{D}(x)$ as follows.

**Definition 1.** *The $k$NN of a tensor $\mathbf{D}(x)$ at $x$ are the $k$ tensors $\mathbf{D}(x_1), \ldots, \mathbf{D}(x_k)$ that minimize $\mu(\mathbf{D}(x), \mathbf{D}(x_i))$, subject to $\|x - x_i\| \leq R$, for a given radius $R > 0$.*

Notice that our definition is essentially a combination of the $k$NN and $\epsilon$-neighborhood used to build a graph from similarity matrices in spectral clustering [27].

**Riemannian Calculation of $W$.** The second step of LLE is to compute the matrix of weights $W \in \mathbb{R}^{n \times n}$. For this purpose, we need to define a reconstruction error similar to (1), and an interpolation method that allows us to express a tensor $\mathbf{D}_i \doteq \mathbf{D}(x_i)$ as an "affine combination" of its $k$NN $\{\mathbf{D}_j \doteq \mathbf{D}(x_j)\}$. Both the reconstruction error and the interpolation method depend on the Riemannian metric chosen. We will illustrate our algorithm using the affine-invariant [3] and log-Euclidean metrics [1].

*Affine-invariant metric.* From [3], we know that the affine-invariant metric is given by

$$\mu_{AI}(\mathbf{D}_i, \mathbf{D}_j) = \| \log(\mathbf{D}_i^{-\frac{1}{2}} \mathbf{D}_j \mathbf{D}_i^{-\frac{1}{2}}) \|_F = \sqrt{\operatorname{trace}\left(\log(\mathbf{D}_i^{-\frac{1}{2}} \mathbf{D}_j \mathbf{D}_i^{-\frac{1}{2}})^2\right)}, \quad (4)$$

where $\|\cdot\|_F$ is the Frobenius norm and $\log(\cdot)$ is the matrix logarithm. We also know that the geodesic linear interpolation of $\mathbf{D}_i$ by tensors $\{\mathbf{D}_j\}_{j=1}^n$ with weights $\{W_{ij}\}_{j=1}^n$ is

$$\widehat{\mathbf{D}}_{AI,i} = \mathbf{D}_i^{\frac{1}{2}} \exp\left(\sum_{j=1}^n W_{ij} \log(\mathbf{D}_i^{-\frac{1}{2}} \mathbf{D}_j \mathbf{D}_i^{-\frac{1}{2}})\right) \mathbf{D}_i^{\frac{1}{2}}, \quad (5)$$

where $\exp(\cdot)$ is the matrix exponential. Therefore, instead of minimizing the Euclidean reconstruction error (1), we minimize the affine-invariant reconstruction error

$$\varepsilon_{AI}(W) = \sum_{i=1}^n \mu_{AI}^2(\mathbf{D}_i, \widehat{\mathbf{D}}_{AI,i}) = \sum_{i=1}^n \| \sum_{j=1}^n W_{ij} \log(\mathbf{D}_i^{-\frac{1}{2}} \mathbf{D}_j \mathbf{D}_i^{-\frac{1}{2}}) \|_F^2, \quad (6)$$

subject to $W_{ij} = 0$ if $\mathbf{D}_j$ is not a $k$NN of $\mathbf{D}_i$ and $\sum_j W_{ij} = 1$. Therefore, the optimal weights are obtained as in (2), with the local Gram matrix $C_i \in \mathbb{R}^{k \times k}$ defined as

$$C_i(j, l) = \operatorname{trace}(\log(\mathbf{D}_i^{-\frac{1}{2}} \mathbf{D}_j \mathbf{D}_i^{-\frac{1}{2}}) \log(\mathbf{D}_i^{-\frac{1}{2}} \mathbf{D}_l \mathbf{D}_i^{-\frac{1}{2}})). \quad (7)$$

*Log-Euclidean metric.* From [1], we know that the log-Euclidean metric is given by

$$\mu_{LE}(\mathbf{D}_i, \mathbf{D}_j) = \| \log \mathbf{D}_i - \log \mathbf{D}_j \|_F, \quad (8)$$

and the geodesic linear interpolation of $\mathbf{D}_i$ by tensors $\{\mathbf{D}_j\}_{j=1}^n$ with weights $\{W_{ij}\}_{j=1}^n$ is

$$\widehat{\mathbf{D}}_{LE,i} = \exp(\sum_{j=1}^n W_{ij} \log \mathbf{D}_j). \quad (9)$$

Thus, $W$ is obtained by minimizing the log-Euclidean reconstruction error

$$\varepsilon_{LE}(W) = \sum_i \mu_{LE}^2(\mathbf{D}_i, \widehat{\mathbf{D}}_{LE,i}) = \sum_{i=1}^n \| \sum_{j=1}^n W_{ij}(\log \mathbf{D}_i - \log \mathbf{D}_j) \|_F^2, \quad (10)$$

subject to $W_{ij} = 0$ if $\mathbf{D}_j$ is not a $k$NN of $\mathbf{D}_i$ and $\sum_j W_{ij} = 1$. The optimal weights are obtained as in (2), with the local Gram matrix $C_i \in \mathbb{R}^{k \times k}$ defined as

$$C_i(j, l) = \operatorname{trace}((\log \mathbf{D}_i - \log \mathbf{D}_j)(\log \mathbf{D}_i - \log \mathbf{D}_l)). \quad (11)$$

Thanks to (7) and (11), we can calculate $W$ exactly and the matrix $M$ is computed as before, i.e. , $M = (I - W)^\top (I - W)$.

**Calculation of the Embedding Coordinates.** The last step of LLE is to find a Euclidean low-dimensional representation. Given $W$, this step is independent of the Riemannian structure. Hence, one can find the embedding coordinates as described in §2. That is, the embedding coordinates are the $d$ eigenvectors of the matrix $M$ associated with its second to $(d + 1)$-st smallest eigenvalues.

## 4   Locally Linear Diffusion Tensor Clustering (LLDTC)

In this section, we present our algorithm for segmenting fiber bundles in the brain that are separated, e.g., segmenting the cingulum and the corpus callosum into different groups. As each fiber bundle defines a different submanifold, the segmentation problem is equivalent to the problem of clustering $m$ submanifolds in the Riemannian space SPSD(3). In particular, we will make use of the Riemannian manifold clustering algorithm in [20].

The LLE algorithm provides a low-dimensional representation of a set of $n$ points under the assumption that the $n$ points are $k$-connected, i.e. for any two points $\mathbf{z}_1, \mathbf{z}_2 \in X$ there exists an ordered sequence of points in $X$ having $\mathbf{z}_1$ and $\mathbf{z}_2$ as endpoints, such that any two consecutive points in the sequence have at least one $k$-nearest neighbor in common. We extend the results of §2 and §3 in order to cluster data lying in a union of $m$ $k$-connected submanifolds. The important assumption we make is that *no kNN of a data point in one submanifold lies in a different submanifold*. At first, this may seem as a very strong assumption. However, Def. 1 ensures that this assumption is approximately true. For instance, consider two spatially close fiber bundles such as the corpus callosum and the cingulum. We know that the corpus callosum is mostly oriented in a left-right direction whereas the cingulum is oriented in the anterior-posterior direction. Even though these two bundles are close to each other spatially, the distance between tensors on different bundles in terms of the Riemannian SPSD metric $\mu$ is significantly large. Therefore by Def. 1, tensors on different bundles are not connected. Consider now two tensors $\mathbf{D}_1$ and $\mathbf{D}_2$ on the same bundle, but spatially separated and having very different orientations. It follows from Def. 1 that these two tensors are not connected. However, as the fiber connecting the two tensors is smooth, there is a sequence of tensors connecting $\mathbf{D}_1$ and $\mathbf{D}_2$. In short, by making use of the locality property in both the coordinate and tensor space to separate two fiber bundles, the aforementioned assumption is fulfilled.

Proposition 1 states the main result of [20] adapted to our scenario. This proposition shows that in the case of a disconnected union of $m$ $k$-connected submanifolds, the matrix $M$ has at least $m$ zero eigenvalues, whose eigenvectors give the clustering of the data. This is a general result that is applicable to both Euclidean and Riemannian LLE. The interested reader is referred to [20] for the proof of Proposition 1.

**Proposition 1.** *Let $\{\mathbf{D}_i\}_{i=1}^n$ be a set of tensors drawn from a disconnected union of $m$ $k$-connected $d$-dimensional submanifolds of SPSD(3). Then, there exist $m$ vectors*

$\{\mathbf{v}_j\}_{j=1}^m$ *in the null space of* $M$ *such that* $\mathbf{v}_j$ *corresponds to the* $j$*th group of points, i.e.* $\mathbf{v}_{ij} = 1$ *if the* $i$*-th data point is in the* $j$*th group, and* $\mathbf{v}_{ij} = 0$ *otherwise.*

With real data, we still have distinct clusters, but the between-cluster weights are not exactly 0. Therefore, the matrix $M$ is a perturbed version of the ideal case. Nevertheless, it is well-known from perturbation theory [28] that if the perturbation is small or the eigengap is big, the eigenvector $\mathbf{v}_j$ is equal to the ideal indicator vectors $(\mathbf{0}, ., \mathbf{1}, ., \mathbf{0})^\top$ of the $j$-th cluster up to a small error term. Hence, it is reasonable to expect that, instead of mapping the data points on $m$ submanifolds to $m$ points, Riemannian LLE will generate a collection of $n$ points distributed around $m$ cluster centers. Therefore, the k-means algorithm will still be able to separate the groups from each other.

Notice that when computing a basis for $\ker(M)$, we do not necessarily obtain the set of membership vectors, but rather linear combinations of them, including the vector $\mathbf{1}$. In general, linear combinations of segmentation eigenvectors still contain the segmentation of the data. Hence, we can cluster the data into $m$ groups by applying k-means to the columns of a matrix whose rows are the $m$ eigenvectors in the null space of $M$. From §3 and Proposition 1, we have the following linear algebraic tensor clustering algorithm.

### Locally Linear Diffusion Tensor Clustering Algorithm (LLDTC)

1. *Nearest neighbors search:* For each tensor $\mathbf{D}_i \in \text{SPSD}(3)$ at coordinate $x_i \in \mathbb{R}^3$, find the $k$ tensors $\{\mathbf{D}_{i_j}\}$ at coordinates $x_{i_j}$ located within a fixed spatial radius $R$ from $x_i$, i.e. $\|x_i - x_{i_j}\| \le R$, that have the smallest tensor distance $\mu$, where $\mu$ is

$$\mu(\mathbf{D}_i, \mathbf{D}_j) = \begin{cases} \|\mathbf{D}_i - \mathbf{D}_j\|_F, & \text{Euclidean.} \\ \|\log \mathbf{D}_i - \log \mathbf{D}_j\|_F, & \text{Log-Euclidean.} \\ \|\log(\mathbf{D}_i^{-\frac{1}{2}} \mathbf{D}_j \mathbf{D}_i^{-\frac{1}{2}})\|_F, & \text{Affine-invariant.} \end{cases}$$

2. *Least squares fit:* Compute the $k$ nonzero entries of the $i$-th row of the weight matrix as $\begin{bmatrix} W_{i\,i_1} & \cdots & W_{i\,i_k} \end{bmatrix} = \frac{\mathbf{1}^\top C_i^{-1}}{\mathbf{1}^\top C_i^{-1}\mathbf{1}} \in \mathbb{R}^{1 \times k}$, where $C_i$ is the local Gram matrix for $\mathbf{D}_i$

$$C_i(j,l) = \begin{cases} \text{trace}((\mathbf{D}_i - \mathbf{D}_j)(\mathbf{D}_i - \mathbf{D}_l)), & \text{Euclidean.} \\ \text{trace}((\log \mathbf{D}_i - \log \mathbf{D}_j)(\log \mathbf{D}_i - \log \mathbf{D}_l)), & \text{Log-Euclidean.} \\ \text{trace}(\log(\mathbf{D}_i^{-\frac{1}{2}}\mathbf{D}_j\mathbf{D}_i^{-\frac{1}{2}})\log(\mathbf{D}_i^{-\frac{1}{2}}\mathbf{D}_l\mathbf{D}_i^{-\frac{1}{2}})), & \text{Affine-invariant.} \end{cases}$$

3. *Clustering:* Compute the $m$ eigenvectors $\{\mathbf{v}_j\}_{j=1}^m$ of $M = (I - W)^\top(I - W)$ associated with its $m$ smallest eigenvalues and apply k-means to the rows of $[\mathbf{v}_1 \cdots \mathbf{v}_m]$ to cluster the tensors into $m$ different groups.

## 5   Experiments

**Synthetic data.** We first test our algorithm on synthetic data in order to validate the segmentation performance of the different metrics. For this purpose, we generate a 3D synthetic tensor field containing two distinct fiber bundles (straight and curved) generated by taking the tensors to be oriented according to the tangential direction of two
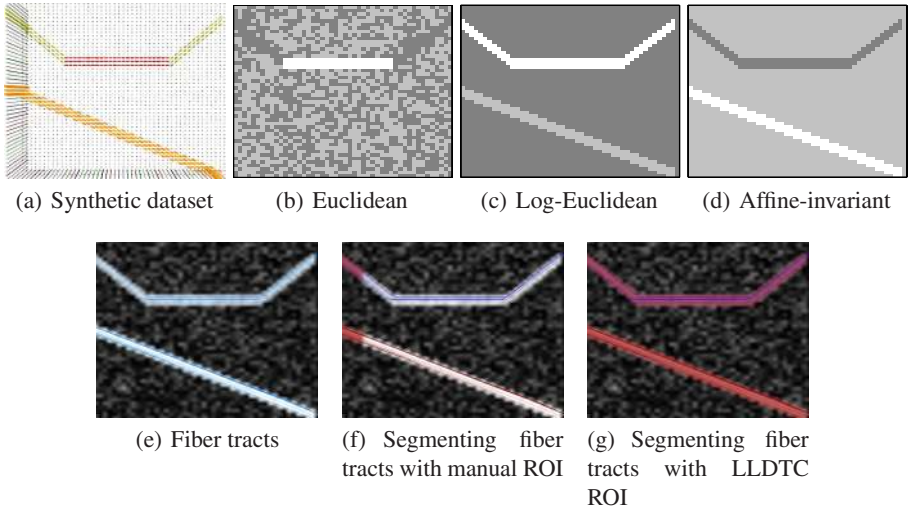
(a) Synthetic dataset        (b) Euclidean        (c) Log-Euclidean        (d) Affine-invariant



(e) Fiber tracts        (f) Segmenting fiber tracts with manual ROI        (g) Segmenting fiber tracts with LLDTC ROI

**Fig. 2.** Segmentation of a synthetic dataset. Fig. 2(a) shows the visualization of the data with each tensor at each voxel represented by an ellipsoid. Figs. 2(b)–2(d) show LLDTC clustering results using the Euclidean, log-Euclidean and affine-invariant metrics. Fig. 2(e) shows the fiber tracking results from MedINRIA. Fig. 2(f) shows two ROI masks marked manually at the beginning of the two bundles, and the fiber tracts extracted by TD. Fig. 2(f) show two ROI masks extracted automatically from the segmentation given by LLDTC, and the fiber tracts extracted by TD.

curves. The background contains tensors without any orientation (isotropic). The eigenvalues of the tensors are independently corrupted by Gaussian noise. Fig. 2(a) shows the dataset with each tensor represented by an ellipsoid whose major axis indicates the dominant diffusion direction. Figs. 2(b)-2(d) show the clustering results using the Euclidean, log-Euclidean, and affine-invariant metrics, respectively. Observe that the Riemannian metrics give the correct segmentation, while the Euclidean metric fails.

In order to extract and cluster all the fibers in each one of the two bundles, we can manually select a region of interest (ROI), and then track fibers that pass through voxels in that ROI. Alternatively, the ROI can be defined automatically from the segmentation given by LLDTC. To evaluate which method is able to find most of the fibers in a bundle, we first extracted all the fibers in the two bundles using the freely available software MedINRIA [29]. The fiber tracking algorithm used here is tensor deflection (TD) [30]. Fig. 2(e) shows the extracted fibers. Notice that TD gives good results when the bundles are straight, but fails in regions of high curvature. We then manually marked two ROI masks located at the beginning of each fiber bundle. These masks are shown in red and pink in Fig. 2(f). We also used the segmentation generated by LLDTC with the affine-invariant metric to define two other ROI masks, shown in red and pink in Fig. 2(g). Since we have ground truth for the two synthetic fiber bundles, we can compare the extracted volume with the true volume of the fiber bundle. For the straight fiber bundle, the manual method gave $97.8\%$ of the bundle, whereas LLDTC achieved $98.6\%$. For the curved fiber bundle, the manual method gave $90.8\%$ of the bundle, while LLDTC achieved $98.6\%$. Hence, by using LLDTC to automatically generate a ROI, we can obtain a good estimate of the fiber bundles, even when tracking is not completely accurate.

**Real data.** We also test the LLDTC algorithm in the segmentation of the corpus callosum and the cingulum from real DTI data using the affine-invariant metric. The corpus callosum is the major communications conduit linking the two hemispheres of the human brain. The two cerebral hemispheres are responsible for distinct and dissimilar cognitive processes, as well as control of contralateral motion and proprioception. Consisting of over 200 million individual nerve fibers, it provides not only a physical, but also a functional connection essential for the coordination of our motor, language, and cognitive abilities. The cingulum bundle, measuring 5-7 mm in diameter, runs dorsal to the corpus callosum, and is the most prominent fiber bundle of the limbic lobe. Many studies have suggested that some functions of the cingulate gyrus depend on the integrity of its connections with other parts of the neuronal network. Therefore, the cingulum bundle, which serves to connect the cingulate cortex with other regions, would be important in the maintenance of the processing of cognitive functions. The corpus callosum and cingulum bundles have been studied extensively using DTI in many clinical populations, including Alzheimer's disease [31], schizophrenia [32] and
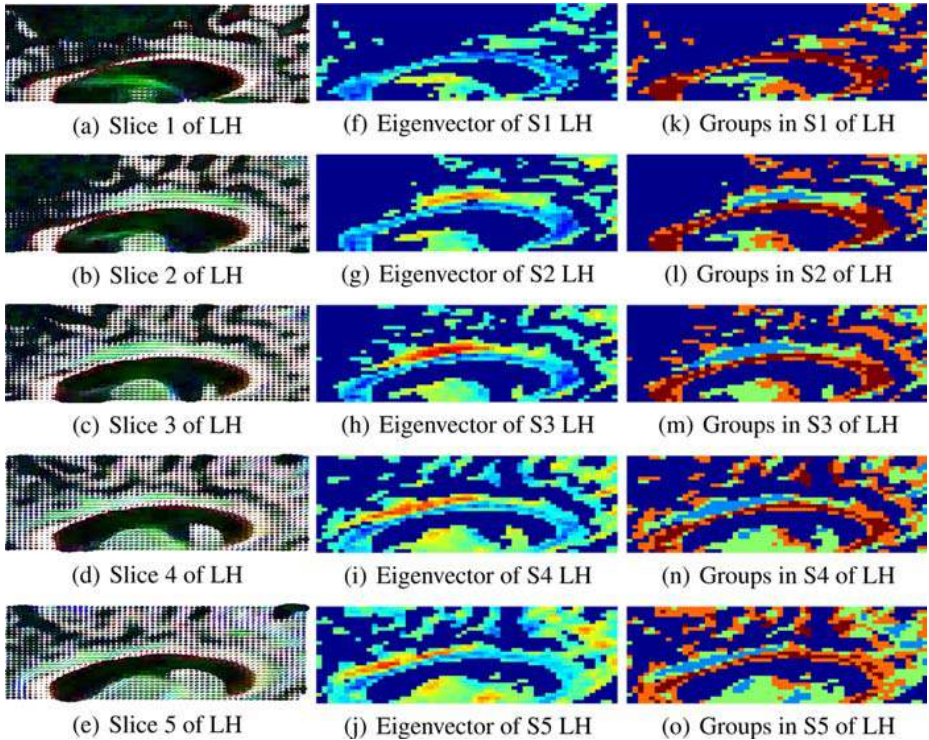


(a) Slice 1 of LH    (f) Eigenvector of S1 LH    (k) Groups in S1 of LH

(b) Slice 2 of LH    (g) Eigenvector of S2 LH    (l) Groups in S2 of LH

(c) Slice 3 of LH    (h) Eigenvector of S3 LH    (m) Groups in S3 of LH

(d) Slice 4 of LH    (i) Eigenvector of S4 LH    (n) Groups in S4 of LH

(e) Slice 5 of LH    (j) Eigenvector of S5 LH    (o) Groups in S5 of LH

**Fig. 3.** Segmenting the corpus callosum and the cingulum in the left hemisphere (LH) using the affine-invariant metric. The first column shows the visualization of the data in five sagittal slices and the tensor at each voxel is represented by an ellipsoid. The second column shows an eigenvector of the LLE matrix $M$ for each of the sagittal slices. The third column shows the clustering result given by LLDTC. The corpus callosum is segmented into the red cluster and the cingulum into the light blue cluster.

autism [33]. For example, patients who have Alzheimer's disease have reduced fractional anisotropy in the cingulum bundle compared to normal aging patients, suggesting that lower anisotropy is associated with cognitive dysfunction and atrophy of the limbic system [31].

The size of the entire DTI volume of the brain is $128 \times 128 \times 58$ voxels and the voxel size is $2 \times 2 \times 2$ mm. From the visualization of the tensor data, we know the approximate location of each cingulum bundle in the left and right hemispheres. Hence, we reduce the input volume to the algorithm by focusing in this location. In addition, we also mask out voxels with fractional anisotropy that is below a threshold of 0.2 in order to separate white matter from the rest of the brain. We set the value of the spatial radius $R$ to be 5 and the number of nearest neighbors to be 25.

Fig. 3 and 4 show the results of the left and right hemispheres respectively. Figs. 3(a)–3(e) and 4(a)–4(e) show the sagittal slices used and the ellipsoid visualization of the tensors. The corpus callosum is the bundle with the red tensors pointing out of the plane and resembles the letter 'C'. The cingulum, which is significantly smaller,



(a) Slice 1 of RH          (f) Eigenvector of S1 RH          (k) Groups in S1 of RH

(b) Slice 2 of RH          (g) Eigenvector of S2 RH          (l) Groups in S2 of RH

(c) Slice 3 of RH          (h) Eigenvector of S3 RH          (m) Groups in S3 of RH

(d) Slice 4 of RH          (i) Eigenvector of S4 RH          (n) Groups in S4 of RH

(e) Slice 5 of RH          (j) Eigenvector of S5 RH          (o) Groups in S5 of RH
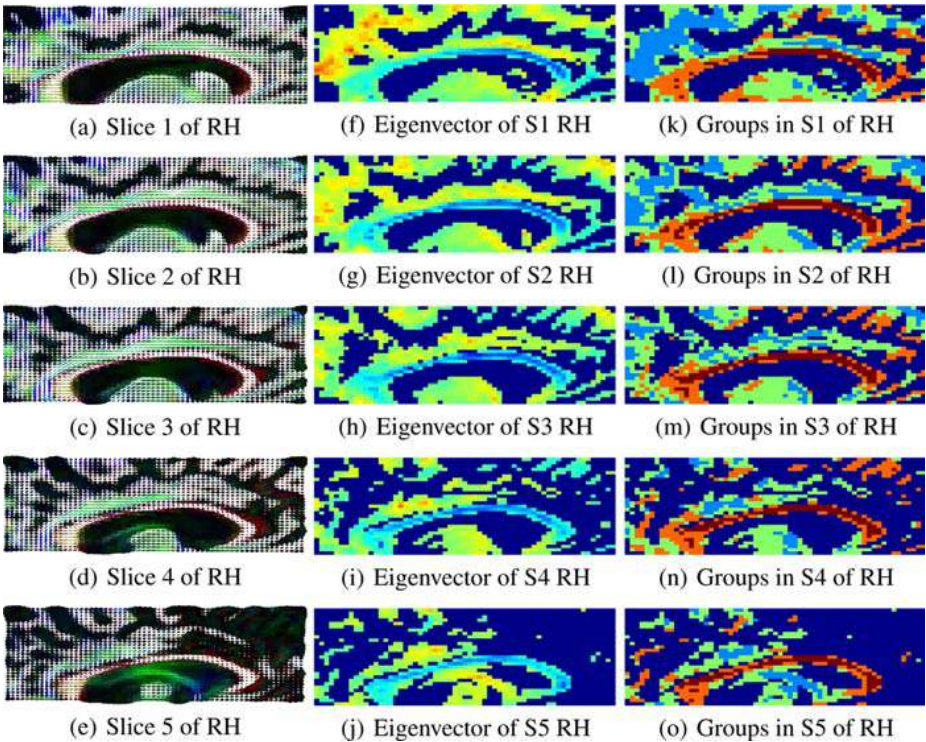
**Fig. 4.** Segmenting the corpus callosum and the cingulum in the right hemisphere (RH) using the affine-invariant metric. The first column shows the visualization of the data in five sagittal slices and the tensor at each voxel is represented by an ellipsoid. The second column shows an eigenvector of the LLE matrix $M$ for each of the sagittal slices. The third column shows the clustering result given by LLDTC. The corpus callosum is segmented into the red cluster and the cingulum into the light blue cluster.

is the bundle left to the corpus callosum with the green tensors oriented vertically. Figs. 3(f)–3(j) and 4(f)–4(j) show an eigenvector of the matrix $M$ for each of the sagittal slice. We see that that the corpus callosum and the cingulum are clustered around different centers. Figs. 3(k)–3(o) and 4(k)–4(o) show the results of LLDTC. In both cases, the corpus callosum forms a distinct cluster (in red). The cingulum is better segmented in the left hemisphere as it consists of the light blue cluster. In the right hemisphere, however, the segmentation of the cingulum is not as distinct. In addition, as our algorithm does not incorporate any smoothness constraint, our segmentation is noisier compared to energy minimization methods such as [16, 17, 18]. However, for the segmentation of the cingulum bundle in [16, 17, 18], a significant effort was required to manually remove voxels in the corpus callosum before running their respective algorithms. Our algorithm, on the other hand, is automatic. Hence, an immediate use for our method is that the output could be used as an automatic initialization for such algorithms.

## 6   Conclusion

We have presented an algorithm for the automatic segmentation of fiber bundles in DT images. Our method requires no initialization or fiber tracking. Instead, it makes the reasonable assumption that tensors at adjacent voxels within a fiber bundle have similar eigenvectors and eigenvalues. Results on synthetic and real data were encouraging. However, an open problem is to incorporate spatial coherence into the algorithm.

## References

1. Arsigny, V., et al.: Log-Euclidean metrics for fast and simple calculus on diffusion tensors. Magnetic Resonance in Medicine 56, 411–421 (2006)
2. Fletcher, P.T., Joshi, S.: Riemannian geometry for the statistical analysis of diffusion tensor data. Signal Processing 87 (2007)
3. Pennec, X., Fillard, P., Ayache, N.: A Riemannian framework for tensor computing. International Journal of Computer Vision 66, 41–46 (2006)
4. Wang, Z., Vemuri, B.C.: DTI segmentation using an information theoretic tensor dissimilarity measure. IEEE Trans. on Med. Imag. 24, 1267–1277 (2005)
5. Kindlmann, G., et al.: Geodesic-loxodromes for diffusion tensor interpolation and difference measurement. In: Ayache, N., Ourselin, S., Maeder, A. (eds.) MICCAI 2007, Part I. LNCS, vol. 4791, pp. 1–9. Springer, Heidelberg (2007)
6. Zhukov, L., et al.: Level set segmentation and modeling of DT-MRI human brain data. Journal of Electronic Imaging, 125–133 (2003)
7. Ding, Z., et al.: Classification and quantification of neuronal fiber pathways using diffusion tensor MRI. Magnetic Resonance in Medicine 49, 716–721 (2003)
8. Brun, A., et al.: Clustering fiber tracts using normalized cuts. In: Barillot, C., Haynor, D.R., Hellier, P. (eds.) MICCAI 2004. LNCS, vol. 3216, pp. 368–375. Springer, Heidelberg (2004)

9. Friman, O., Farneback, G., Westin, C.F.: A Bayesian approach for stochastic white matter tractography. IEEE Trans. on Med. Imag. 25, 965–978 (2006)
10. Perrin, M., et al.: Fiber tracking in q-ball fields using regularized particle trajectories. In: Information Processing in Medical Imaging (2005)
11. Wiegell, M., et al.: Automatic segmentation of thalamic nuclei from diffusion tensor magnetic resonance imaging. NeuroImage, 391–401 (2003)
12. Wang, Z., Vemuri, B.: Tensor field segmentation using region based active contour model. In: European Conference on Computer Vision, pp. 304–315 (2004)
13. Jonasson, L., et al.: A level set method for segmentation of the thalamus and its nuclei in DT-MRI. Signal Processing 87, 309–321 (2007)
14. Ziyan, U., Tuch, D., Westin, C.F.: Segmentation of thalamic nuclei from DTI using spectral clustering. In: Larsen, R., Nielsen, M., Sporring, J. (eds.) MICCAI 2006. LNCS, vol. 4191, pp. 807–814. Springer, Heidelberg (2006)
15. Lenglet, C., et al.: A Riemannian approach to diffusion tensor images segmentation. In: Information Processing in Medical Imaging (2005)
16. Melonakos, J., et al.: Locally-constrained region-based methods for DW-MRI segmentation. In: MMBIA (2007)
17. Melonakos, J., et al.: Finsler tractography for white matter connectivity analysis of the cingulum bundle. In: Ayache, N., Ourselin, S., Maeder, A. (eds.) MICCAI 2007, Part I. LNCS, vol. 4791, pp. 36–43. Springer, Heidelberg (2007)
18. Awate, S., et al.: A fuzzy, nonparametric segmentation framework for DTI and MRI analysis. IEEE Trans. on Med. Imag. 26, 1525–1536 (2007)
19. Roweis, S., Saul, L.: Think globally, fit locally: Unsupervised learning of low dimensional manifolds. Journal of Machine Learning Research 4, 119–155 (2003)
20. Goh, A., Vidal, R.: Clustering and dimensionality reduction on Riemannian manifolds. In: IEEE CVPR (2008)
21. Goh, A., Vidal, R.: Unsupervised Riemannian clustering of probability density functions. In: ECML PKDD (2008)
22. Wassermann, D., et al.: Diffusion maps clustering for magnetic resonance Q-Ball imaging segmentation. International Journal in Biomedical Imaging (2008)
23. Haro, G., Lenglet, C., Sapiro, G., Thompson, P.M.: On the non-uniform complexity of brain connectivity. In: ISBI, pp. 887–890 (2008)
24. Belkin, M., Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. Neural Information Processing Systems, 585–591 (2002)
25. Donoho, D., Grimes, C.: Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. PNAS 100, 5591–5596 (2003)
26. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science 290, 2319–2323 (2000)
27. von Luxburg, U.: A tutorial on spectral clustering. Stat. and Computing 17 (2007)
28. Horn, R., Johnson, C.: Matrix Analysis. Cambridge University Press, Cambridge (1985)
29. Toussaint, N., et al.: MedINRIA: Medical image navigation and research tool by INRIA. In: Proc. of MICCAI 2007 Workshop on Interaction in med. image analysis and vis. (2007), http://www-sop.inria.fr/asclepios/software/MedINRIA
30. Weinstein, D., et al.: Tensorlines: Advection-diffusion based propagation through diffusion tensor fields. In: IEEE Visualization, San Francisco, pp. 249–254 (1999)
31. Xie, S., et al.: Evaluation of bilateral cingulum with tractography in patients with Alzheimer's disease. Neuroreport 16, 1275–1278 (2005)
32. Foong, J., et al.: Investigating regional white matter in schizophrenia using diffusion tensor imaging. Neuroreport 13, 333–336 (2002)
33. Alexander, A., et al.: Diffusion tensor imaging of the corpus callosum in autism. Neuroimage 34, 61–73 (2007)