

Robust Tensor Splines for Approximation of Diffusion Tensor MRI Data

Angelos Barmountis, Baba C. Vemuri, John R. Forder
University of Florida
Gainesville, FL 32611, USA

{abarmpou, vemuri}@cise.ufl.edu, jforder@mbi.ufl.edu

Abstract

In this paper, we present a novel and robust spline approximation algorithm given a noisy symmetric positive definite (SPD) tensor field. Such tensor fields commonly arise in the field of Medical Imaging in the form of Diffusion Tensor (DT) MRI data sets. We develop a statistically robust algorithm for constructing a tensor product of B-splines – for approximating and interpolating these data – using the Riemannian metric of the manifold of SPD tensors. Our method involves a two step procedure wherein the first step uses Riemannian distances in order to evaluate a tensor spline by computing a weighted intrinsic average of diffusion tensors and the second step involves minimization of the Riemannian distance between the evaluated spline curve and the given data. These two steps are alternated to achieve the desired tensor spline approximation to the given tensor field. We present comparisons of our algorithm with four existing methods of tensor interpolation applied to DT-MRI data from fixed heart slices of a rabbit, and show significantly improved results in the presence of noise and outliers. We also present validation results for our algorithm using synthetically generated noisy tensor field data with outliers. This interpolation work has many applications e.g., in DT-MRI registration, in DT-MRI Atlas construction etc. This research was in part funded by the NIH ROI NS42075 and the Department of Radiology, University of Florida.

1. Introduction

Data processing and analysis of matrix-valued image data is becoming quite common as imaging sensor technology advances allow for the collection of matrix-valued data sets. In Medical Imaging, in the last decade, it has become possible to collect magnetic resonance image (MRI) data that measures the apparent diffusivity of water in tissue in vivo. A 2-tensor has been commonly used to approximate the diffusivity profile at each lattice point of the image lattice [2]. This approximation yields a diffusion tensor (DT-

MRI) data set which is a matrix-valued image. These tensors are elements of the space of the 3×3 positive-definite matrices denoted by $P(3)$. Mathematically, these positive definite Diffusion Tensors belong to a Riemannian Symmetric space, where a Riemannian metric assigns an inner product to each point of this space. By using this metric, one can compute geodesic distances between the elements of this space (diffusion tensors) and compute various statistics on this space [4, 6, 9, 10].

Processing of DT-MRI data sets has scientific significance for health, information and other sciences. Most of the applications require preprocessing of the DT-MRI data which more often than not involves interpolation of the diffusion tensor fields. Directly performing interpolation of the coefficients of the Diffusion Tensor matrices, does not preserve most of the properties, such as the values of the determinant and eigenvalues etc., of the Diffusion Tensors. Thus, it motivates one to seek alternative methods to achieve interpolation.

In [4], a Riemannian metric was proposed for geodesic computation between two tensors. Having a tensor field, e.g. a volume image of DT-MRI, we can use the geodesic curves between spatially consecutive tensors in order to interpolate the dataset. However, this geodesic curve computation between tensors (in [4]) did not impose higher order smoothness constraints in achieving the interpolation. Thus, although there is continuity of the obtained interpolated dataset, higher order continuity is lacking.

Recently, a Log-Euclidean metric was proposed in [1] for computing with tensors. In their work, the elements from the space of positive definite tensors $P(3)$, are mapped to their tangent space, $Sym(3)$, using the matrix logarithmic mapping. The tangent space forms a vector space of dimension \mathbb{R}^6 for diffusion tensors in \mathbb{R}^3 . Therefore, one can use the Euclidean norm for computations in this tangent space and finally by using the inverse mapping, the data are mapped back to the space of positive definite matrices $P(3)$. More recently, a similar approach for interpolation of tensor fields was proposed in [3]. In their work, the tensors are mapped to the tangent space of a reference tensor, using the

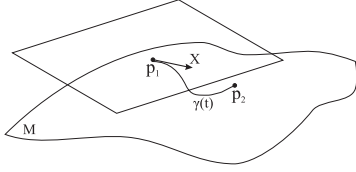


Figure 1. Tangent space of the manifold M of diffusion tensors at point \mathbf{p}_1 . The tangent vector \mathbf{X} points to the direction of geodesic $\gamma(t)$ between the points \mathbf{p}_1 and \mathbf{p}_2 .

Riemannian Logarithmic mapping. The elements of the tangent space are treated as vectors, which are interpolated using radial basis functions and finally are mapped back using the inverse mapping. Both frameworks are quite interesting and have advantages due to their high computational efficiency in comparison to the geodesic distance framework. However, a statistically robust version of these frameworks need to be developed and although the geodesics obtained by such procedures are “similar” to those obtained by using the Riemannian metric, quantitatively they are not the same. This is because by mapping the data to the tangent space, we lose the information about the curvature of the Riemannian manifold of diffusion tensors.

Interpolation of matrix-valued images can be done by filtering such datasets using various regularization methods. Regularization of matrix-valued images can be achieved using a PDE-based approach proposed in [12]. Although this approach can be used for denoising and interpolation of matrix-valued images, the lack of use of an appropriate metric that is defined on the space of positive definite tensors could lead to undesirable limitations in the solution. Another tensor field regularization method was proposed in [13] using Normalized Convolution and Markov Random Fields (MRF) in a Bayesian Framework. The SPD tensors are treated as vectors in 6D and their components are treated independently. This can lead to inaccuracies in the regularized solution.

In this paper, we present a novel and robust spline approximation algorithm given a noisy diffusion tensor field. We develop a robust algorithm for constructing a tensor product of B-splines using the Riemannian metric of the manifold of symmetric positive definite tensors. We present comparisons of our algorithm with four existing methods of interpolation for Diffusion Tensor MRI data from fixed heart slices of a rabbit, and show much improved results in the presence of noise and outliers. We also present validation results for our algorithm using synthetically generated noisy tensor field data with outliers.

The rest of the paper is organized into the following sections: In section 2, we present the geometry of the space of Diffusion Tensors and the related mathematical background. In section 3, we make a brief review of B-splines. This is followed by the presentation of a novel algorithm

for computing splines on a given symmetric positive definite tensor field in section 3. Next, we present tensor splines using the Log-Euclidean metric as an improvement over recent work in [1]. Then, we present a robust tensor spline approximation algorithm. Finally, in section 4, comparisons of our algorithm with four existing methods of interpolation for DT MRI data are presented. Validation results using synthetically generated noisy tensor field data with outliers are also presented.

2. The space of Diffusion Tensors

In this section we present the geometry of the space of Diffusion Tensors and the related mathematical background that is required for doing computations in this space. More detailed expositions on this material maybe found in [9, 6, 4]. The space of Diffusion Tensors can be formulated as a Riemannian Symmetric space [5], where a Riemannian metric assigns an inner product to each point of this space. By using this metric we can compute geodesic distances between diffusions tensors and calculate statistics on this space [4, 6, 9].

A symmetric positive definite 2-tensor is a $n \times n$ matrix that belongs to the space of all symmetric positive-definite matrices denoted by $P(n)$. DT-MRI datasets are a field of such tensor usually in two or three-dimensions. $P(n)$ space is a connected Riemannian manifold [5]. The Lie group $GL^+(n)$ of all $n \times n$ real matrices \mathbf{g} with $|\mathbf{g}| > 0$ acts on $P(n)$ and smoothly maps its elements back to $P(n)$ via the group action $\phi(\mathbf{g}, \mathbf{p}) = \mathbf{g}\mathbf{p}\mathbf{g}^T$.

At each point \mathbf{p} of $P(n)$ the Riemannian metric is given by the following inner product $\langle \mathbf{X}, \mathbf{Y} \rangle_{\mathbf{p}} = \text{trace}(\mathbf{g}^{-1}\mathbf{X}\mathbf{p}^{-1}\mathbf{Y}\mathbf{g}^{-T})$, where $\mathbf{g} \in GL^+(n)$ and $\mathbf{p} = \mathbf{g}\mathbf{g}^T$ and $\mathbf{X}, \mathbf{Y} \in \text{Sym}(n)$ are two tangent vectors at the point $\mathbf{p} \in P(n)$. At any point $\mathbf{p} \in P(n)$ the tangent space is defined by the space of the $n \times n$ symmetric matrices.

In figure 1, \mathbf{p}_1 is a point in $P(n)$ and \mathbf{X} is a tangent vector at \mathbf{p}_1 . There is a unique geodesic $\gamma(t)$ starting at \mathbf{p}_1 for $t = 0$ and having $\gamma'(0) = \mathbf{X}$. The point on the geodesic $\gamma(t)$ at $t = 1$ can be computed by using the Riemannian exponential map $\gamma(1) = \text{Exp}_{\mathbf{p}_1}(\mathbf{X})$. The Riemannian exponential map $\text{Exp}_{\mathbf{p}}(\mathbf{X})$ at a point $\mathbf{p} \in P(n)$ maps a tangent vector \mathbf{X} at \mathbf{p} to $\gamma(1)$, where $\gamma(t)$ is the geodesic starting at $\gamma(0) = \mathbf{p}$ having $\gamma'(0) = \mathbf{X}$. $\text{Exp}_{\mathbf{p}}(\mathbf{X})$ can be computed using

$$\text{Exp}_{\mathbf{p}}(\mathbf{X}) = (\mathbf{g}\mathbf{v})\text{exp}(\mathbf{\Sigma})(\mathbf{g}\mathbf{v})^T \quad (1)$$

where $\mathbf{g} \in GL^+(n)$ and $\mathbf{p} = \mathbf{g}\mathbf{g}^T$. Let $\mathbf{Y} = \mathbf{g}^{-1}\mathbf{X}\mathbf{g}^{-T}$, then \mathbf{v} and $\mathbf{\Sigma}$ are the matrix of eigenvectors and the diagonal matrix of eigenvalues of \mathbf{Y} respectively. Therefore in equation (1) $\text{exp}(\mathbf{\Sigma})$ is the matrix exponential map which is given

by,

$$\exp(\mathbf{X}) = \sum_{i=0}^{\infty} \frac{1}{i!} \mathbf{X}^i. \quad (2)$$

The inverse map is the Riemannian Logarithmic map $\text{Log}_{\mathbf{p}_1}(\mathbf{p}_2)$, which maps $\mathbf{p}_2 \in P(n)$ to a tangent vector \mathbf{X} at \mathbf{p}_1 , such that if $\gamma(0) = \mathbf{p}_1$ and $\gamma(1) = \mathbf{p}_2$ then $\mathbf{X} = \gamma'(0)$. $\text{Log}_{\mathbf{p}_1}(\mathbf{p}_2)$ can be computed using,

$$\text{Log}_{\mathbf{p}_1}(\mathbf{p}_2) = (\mathbf{g}\mathbf{u})\log(\mathbf{\Lambda})(\mathbf{g}\mathbf{u})^T \quad (3)$$

where, $\mathbf{g} \in GL^+(n)$ and $\mathbf{p}_1 = \mathbf{g}\mathbf{g}^T$. Let $\mathbf{y} = \mathbf{g}^{-1}\mathbf{p}_2\mathbf{g}^{-T}$, then \mathbf{u} and $\mathbf{\Lambda}$ are the matrix of eigenvectors and the diagonal matrix of eigenvalues of \mathbf{y} respectively. The distance between the two tensors \mathbf{p}_1 and \mathbf{p}_2 equals to $\|\text{Log}_{\mathbf{p}_1}(\mathbf{p}_2)\|_{\mathbf{p}_1}$ where $\|\cdot\|_{\mathbf{p}_1}$ is the Riemannian norm at the point \mathbf{p}_1 .

For any real number t the point $\gamma(t)$ in the geodesic between \mathbf{p}_1 and \mathbf{p}_2 can be computed as

$$\gamma(t) = (\mathbf{g}\mathbf{u})\text{Diag}(\mathbf{\Lambda}^t)(\mathbf{g}\mathbf{u})^T \quad (4)$$

where \mathbf{g} , \mathbf{u} and $\mathbf{\Lambda}$ are the same as in the equation of Riemannian Logarithmic map, and $\text{Diag}(\mathbf{X})$ are the diagonal elements of symmetric matrix \mathbf{X} . Equation 4 can be easily derived from $\gamma(t) = \text{Exp}_{\mathbf{p}_1}(\mathbf{X}, t) = (\mathbf{g}\mathbf{v})\exp(\mathbf{\Sigma}t)(\mathbf{g}\mathbf{v})^T$ [4] and 3.

Using the Riemannian metric one can also compute statistics of Diffusion Tensors [4, 6, 9]. The average tensor of a set of Diffusion Tensors, can now be computed as that tensor which minimizes the sum of squared Riemannian distances between itself and the given set of tensors.

In the following section the Riemannian exponential and logarithmic maps and the equation of a geodesic between two diffusion tensors will be used in order to define and compute tensor splines.

3. Tensor Spline Approximation

In this section we present a novel and robust spline approximation algorithm given a noisy symmetric positive definite tensor field. Our algorithm involves the use of the Riemannian distance between SPD tensors in order to evaluate a tensor spline by computing a weighted intrinsic average of tensors. By intrinsic average, we mean, an average that minimizes the Riemannian distance between the unknown average and the members of the population whose average is being sought. This module is then used in a robust tensor product B-spline fitting method involving the minimization of the Riemannian distance between the spline function and the data.

This section has three subsections. First we make a brief review of B-splines. Next, we present a novel algorithm for computing splines on a given symmetric positive definite tensor field. After that we present tensor splines using the Log-Euclidean metric (described earlier). Finally a robust tensor spline approximation technique is presented.

3.1. B-splines

The equation for $k-1$ degree B-spline with $n+1$ control points (c_0, c_1, \dots, c_n) and $n+k+1$ knots $(t_{-k+1}, t_{-k+2}, \dots, t_{n+1})$, is

$$S(t) = \sum_{i=0}^n N_{i,k}(t)c_i \quad (5)$$

where $t_0 \leq t \leq t_{n+1-(k-1)}$. Each control point is associated with a basis function $N_{i,k}$ where

$$N_{i,1} = \begin{cases} 1 & \text{if } t_i \leq t < t_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

and

$$N_{i,k}(t) = N_{i,k-1}(t) \frac{t - t_i}{t_{i+k-1} - t_i} + N_{i+1,k-1}(t) \frac{t_{i+k} - t}{t_{i+k} - t_{i+1}} \quad (7)$$

$N_{i,k}(t)$ functions are polynomials of degree $k-1$. Cubic basis functions $N_{i,4}$ can be used for a 3rd degree B-spline. Knots must be series of monotonic increasing numbers.

One useful property of the functions $N_{i,k}(t)$ is that $N_{i,k}(t) \geq 0$, for all i and $\sum_{i=0}^n N_{i,k}(t) = 1$. Considering

the above properties, functions $N_{i,k}(t)$ behave as blending functions and eq. 5 is a weighted average of the control points c_i .

3.2. Tensor Splines

In the symmetric positive definite tensor space $P(n)$, a tensor $\gamma(t)$, where $t \in R$ is a scalar, that lies on the geodesic between two tensors \mathbf{p}_1 and \mathbf{p}_2 is given by equation (4). By definition, a geodesic curve is the shortest path between two points. Having a dataset of tensors, e.g., a volume of DT-MRI, we can use the geodesic curves between spatially consecutive tensors in order to interpolate the dataset. However, a geodesic curve does not contain information about the neighborhood of the two interpolating points. Thus, although there is continuity of the interpolated dataset, there is lack of higher degree of continuity. It is more natural to have a higher degree of continuity in the interpolant within smoothly varying regions. Recent work in [7], on continuous tensor field approximation achieves smoothness, however, a Riemannian framework is not employed for tensor calculations. In this section we define tensor splines which are curves in the input space $P(n) \times \mathfrak{R}^m$, $(P(3) \times \mathfrak{R}^2$ in the case of a 2D diffusion tensor field) constructed using the geometry of the space of symmetric positive definite tensors. Note that doing spline interpolation on $P(n)$ is not meaningful in itself, in this context, as there is no ordering of points (matrices) on $P(n)$, which reflects the ordering imposed by the lattice in 2D/3D. Hence, interpolation on $P(3) \times \mathfrak{R}^2$ or $P(3) \times \mathfrak{R}^3$ is the right thing to do because the

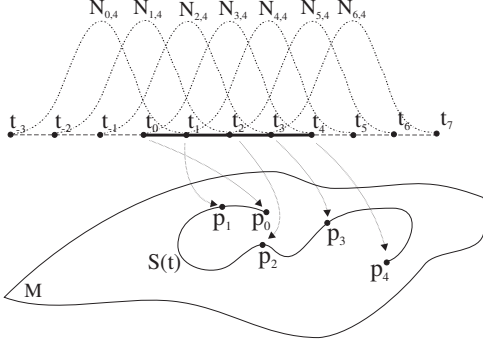


Figure 2. A 3rd degree tensor spline $S(t)$, that passes through 5 given tensors \mathbf{p}_i of a 1-D tensor field. The given points \mathbf{p}_i and the points of the tensor spline $S(t)$ are SPD matrices, elements of the Riemannian manifold M . However the tensor spline is in $P(n) \times \mathfrak{R}$ since the tensors lie on a 1-D lattice. 7 control points \mathbf{c}_i are required and 11 knots t_i . The association between basis functions $N_{i,4}(t)$, knots t_i and given data points \mathbf{p}_i are displayed in this figure.

tensors in DTI are defined on a lattice in 2D or 3D. Another issue to keep in mind is that we are defining tensor-splines by doing weighted intrinsic averages on $P(n)$ and choosing the weight functions to be B-splines. As an illustration of interpolation on a 1D grid of tensors, figure (2) depicts the idea of using weight functions (B-splines here) to perform weighted average of tensors using the Riemannian metric. This weighted averaging leads to the desired degree spline interpolant of the diffusion tensor data. Applications of tensor splines are numerous e.g., higher order interpolation between diffusion tensors, smoothing and filtering diffusion tensor data sets, and compression of such data sets.

Let us assume that we have a set of N diffusion tensors ($\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_{N-1}$) on a one dimensional grid, and we want to interpolate between them. Linear (1st degree) interpolation on the tensor space can be achieved by computing points on the geodesics connecting two consecutive diffusion tensors. Higher degree interpolation can be done by using a set of control points and a knots vector. A $k-1$ degree tensor spline that fits to our data requires $N-1+k-1$ control points ($\mathbf{c}_0, \mathbf{c}_1, \dots, \mathbf{c}_{N-1+k-2}$) which are also tensors and $N+2(k-1)$ monotonically increasing knots ($t_{-k+1}, t_{-k+2}, \dots, t_{N-1+k-1}$). A tensor $S(t)$, where $t \in [t_j, t_{j+1})$, which is a point on a tensor spline, can be now computed by generalizing the equation (5) to the space of tensors. We can compute the value $S(t)$ of the $k-1$ degree B-spline of tensors for a particular t value, by calculating a weighted intrinsic average, $\widetilde{\sum}$, of the control tensors \mathbf{c}_i , where the weights equal to the basis functions $w_i = N_{i,k}(t)$, discussed earlier.

$$S(t) = \widetilde{\sum}_{i=0}^n w_i \mathbf{c}_i \quad (8)$$

The intrinsic weighted average of tensors is defined using

Riemannian distance instead of Euclidean, and it is the minimizer of the function

$$\min_{\mu \in P(n)} \rho(\mu) = \min_{\mu \in P(n)} \frac{1}{2} \sum_{i=0}^n w_i \text{dist}(\mu, \mathbf{c}_i)^2 \quad (9)$$

where $\text{dist}(\cdot, \cdot)$ is the Riemannian geodesic distance. The weighted average can be computed using a gradient descent algorithm which is an extension of the algorithm described by Pennec [8] for computing the mean of tensors. The gradient of $\rho(\mu)$ is given by

$$\nabla_{\mu} \rho = - \sum_{i=0}^n w_i \text{Log}_{\mu}(\mathbf{c}_i) \quad (10)$$

Thus the intrinsic weighted average of a set of diffusion tensors can be computed by the following procedure:

```

input :  $\mathbf{c}_1, \dots, \mathbf{c}_N \in P(n)$ 
           $w_1, \dots, w_N$  weights
output:  $\mu \in P(n)$ , the weighted mean
 $\mu_0 \leftarrow \mathbf{I}$ ;
 $i \leftarrow 0$ ;
while  $\|\mathbf{X}_i\| > e$  do
  |  $\mathbf{X}_i \leftarrow -\nabla_{\mu_i} \rho$ ;
  |  $\mu_{i+1} \leftarrow \text{Exp}_{\mu_i}(\mathbf{X}_i)$ ;
end

```

Algorithm 1: Intrinsic Weighted Mean of Tensors

In order to fit a tensor spline to the diffusion tensor data, we have to approximate the control tensors of such a spline. A tensor spline that fits to our data, minimizes the Riemannian distance of the given tensors from the tensor spline curve.

$$E = \frac{1}{2N} \sum_{i=0}^{N-1} \text{dist}(S(t_i), \mathbf{p}_i)^2 \quad (11)$$

In this energy expression, the Riemannian metric should be used for the distance calculation, since the tensor space, where the data and control points belong, is a Riemannian manifold. We need to find a set of control points ($\mathbf{c}_0, \mathbf{c}_1, \dots, \mathbf{c}_{N-1+k-2}$) that form the spline $S(t)$ which minimizes the energy E . The gradient of E with respect to \mathbf{c}_j is then given by,

$$\nabla_{\mathbf{c}_j} E = \frac{1}{2N} \sum_{i=0}^{N-1} \nabla_{S(t_i)} \text{dist}(S(t_i), \mathbf{p}_i)^2 \nabla_{\mathbf{c}_j} S(t_i) \quad (12)$$

The gradient of the square distance between $S(t_i)$ and \mathbf{p}_i with respect to $S(t_i)$ equals,

$$\nabla_{S(t_i)} \text{dist}(S(t_i), \mathbf{p}_i)^2 = -2 \text{Log}_{S(t_i)}(\mathbf{p}_i) \quad (13)$$

where $\text{Log}_{S(t_i)}(\mathbf{p}_i)$ is the Riemannian logarithmic map (eq. 1). However, $\text{Log}_{S(t_i)}(\mathbf{p}_i)$ is a tangent vector at $S(t_i)$.

Since, the gradient of the energy (eq. 12) is with respect to \mathbf{c}_j , we need to express the gradient of eq. 13 by using tangent vectors at point \mathbf{c}_j . Taking this into consideration, equation 13 can be approximated by the formula $\Lambda_{\mathbf{c}_j}(\mathbf{p}_i, S(t_i)) = \text{Log}_{\mathbf{c}_j}(\mathbf{p}_i) - \text{Log}_{\mathbf{c}_j}(S(t_i))$, so we obtain

$$\nabla_{S(t_i)} \text{dist}(S(t_i), \mathbf{p}_i)^2 \approx -2\Lambda_{\mathbf{c}_j}(\mathbf{p}_i, S(t_i)) \quad (14)$$

Furthermore, the gradient of $S(t_i)$ with respect to \mathbf{c}_j in the equation 12 is

$$\nabla_{\mathbf{c}_j} S(t_i) = N_{j,k}(t_i) \quad (15)$$

Using equations 15 and 14 in 12 we obtain

$$\nabla_{\mathbf{c}_j} E = -\frac{1}{N} \sum_{i=0}^{N-1} \Lambda_{\mathbf{c}_j}(\mathbf{p}_i, S(t_i)) N_{j,k}(t_i) \quad (16)$$

Starting with an initial guess of the control tensors, we can update them by using the gradient descent technique. The new values \mathbf{c}'_j of control tensors will be

$$\mathbf{c}'_j = \text{Exp}_{\mathbf{c}_j} \left(\frac{1}{N} \sum_{i=0}^{N-1} \Lambda_{\mathbf{c}_j}(\mathbf{p}_i, S(t_i)) N_{j,k}(t_i) \right) \quad (17)$$

where $\text{Exp}(\cdot)$ is the Riemannian exponential map (eq. 1). The initial guess of the control tensors can be either the given data or the average tensor of the given tensors. The gradient descent algorithm is summarized below:

```

input :  $N$  tensors ( $\mathbf{p}_0, \dots, \mathbf{p}_{N-1}$ ),
           $N+2(k-1)$  monotonically increasing
          knots ( $t_{-k+1}, \dots, t_{N-1+k-1}$ )
           $k$ , and a small value  $e$ 
output:  $N-1+k-1$  control tensors
          ( $\mathbf{c}_0, \dots, \mathbf{c}_{N-1+k-2}$ )

 $\|\mathbf{X}_0\| \leftarrow e+1$ ;
while  $\sum_j \|\mathbf{X}_j\| > e$  do
  for  $j=0$  to  $N-1+k-2$  do
     $\mathbf{X}_j \leftarrow$  zero matrix ;
    for  $i=0$  to  $N-1$  do
       $\mathbf{X}_j \leftarrow \mathbf{X}_j +$ 
       $\Lambda_{\mathbf{c}_j}(\mathbf{p}_i, S(t_i)) N_{j,k}(t_i)$  ;
    end
     $\mathbf{c}'_j \leftarrow \text{Exp}_{\mathbf{c}_j} \mathbf{X}_j$  ;
  end
   $\mathbf{c} \leftarrow \mathbf{c}'$  ;
end

```

Algorithm 2: Control Tensors estimation

The error introduced by the approximation of equation 14 can be large, if the tensor spline approximation $S(t_i)$ is far from the target \mathbf{p}_i . While $S(t_i)$ moves closer to \mathbf{p}_i during the spline fitting procedure, the error introduced by the

approximation of equation 14 moves to zero. By setting a small number e , the outer loop of algorithm 2 will be iterated enough times in order the error of equation 14 to be as small as we need. Hence the control tensors \mathbf{c}_j , which are obtained as the output of algorithm 2, are estimated without the assumption that there is no curvature in the manifold of SPD matrices. On the contrary, this assumption was used in [1] and [3].

The knot sequence can be parameterized by different ways that preserve the monotonically increasing property of the knots series. In the experiments we used the series of integer numbers 1,2,3... as the knot sequence.

Tensor Splines can be fit to higher dimensional tensor fields, by simply extending the presented algorithms to the new dimensionality. For example consider the case of a 2D $N \times M$ tensor field. In this case we are doing interpolation on $P(n) \times \mathbb{R}^2$. A $k-1$ degree tensor spline that fits on our data requires $(N-1+k-1) \times (M-1+k-1)$ control tensors and $(N+2(k-1)) \times (M+2(k-1))$ monotonically increasing in both the dimensions, the knots $(t_{-k+1, -k+1}, \dots, t_{N-1+k-1, M-1+k-1})$. Note that in this case the knots are vectors with 2 elements, one for each parametric dimension. Finally the new basis functions are formed by the tensor product of 1-dimensional basis functions $N_{i,j,k}([t_1 t_2]) = N_{i,k}(t_1) N_{j,k}(t_2)$.

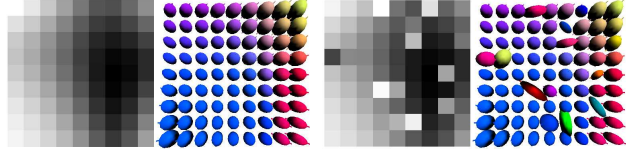
3.3. Log-Euclidean Splines

Recently, in [1], Arsigny et. al., proposed a new Log-Euclidean metric for tensor calculations. By using this metric, the diffusion tensors are mapped, with the matrix logarithmic map to the space of the symmetric matrices $\text{Sym}(n)$. Therefore, we can make Euclidean calculations in this space and finally by using the matrix exponential mapping, the data are mapped back to the space of positive definite matrices $P(n)$. Although the tensors obtained by this procedure are "similar" to those obtained by using the Riemannian metric, quantitatively they are not the same. This is because by mapping the data to the tangent space, we lose the information about the curvature of the Riemannian manifold of diffusion tensors.

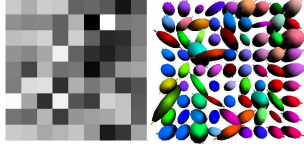
Using the Log-Euclidean metric we can also fit a spline to the logarithmically mapped data into the symmetric space, and after that we can map the interpolated tensors back to the tensor space using the exponential mapping. In the section 4, we provide a quantitative comparison between tensor splines and Log-Euclidean Splines.

3.4. Robust Tensor Splines

The presence of outliers is common in DT-MRI data due to noise in the original data obtained from the DT-MRI sensors [11]. A robust interpolation algorithm should detect these outlier tensors and reject them from further consideration in any processing algorithms applied to the dataset.



(a) The FA map and Gaussian el- (b) The tensor field of fig. 3(a) with lipsoids of the subsampled synthetic 10% outliers tensor field



(c) The tensor field of fig. 3(a) with noise

Figure 3. Synthetic tensor fields used in the experiments.

A robust function can be used in the energy function, in order to weight the given data \mathbf{p}_i appropriately. We need to use a robust function that assigns weights in the interval $[0, 1]$, where weights which are almost zero imply rejection of the corresponding data point. Furthermore, high weights should be assigned to the data points whose distance from the corresponding points on the unknown spline curve is small and on the other hand lower weights should be assigned to the data points whose distance from the corresponding points on the unknown spline is larger. Let us consider the following function $\phi(x) = -se^{-\frac{x}{s}}$. The derivative of this function is $\psi(x) = e^{-\frac{x}{s}}$ and it has the aforementioned properties. By using the above function ϕ , the energy function that we are going to minimize can be written as

$$E = \frac{1}{2N} \sum_{i=0}^{N-1} \phi(\text{dist}(S(t_i), \mathbf{p}_i)^2) \quad (18)$$

The gradient of the energy with respect to the control tensors now becomes

$$\nabla_{\mathbf{c}_j} E = -\frac{1}{N} \sum_{i=0}^{N-1} \psi(\text{dist}(S(t_i), \mathbf{p}_i)^2) \times \Lambda_{\mathbf{c}_j}(\mathbf{p}_i, S(t_i)) N_{j,k}(t_i)$$

In the above equation the quantity $\psi(\text{dist}(S(t_i), \mathbf{p}_i)^2)$, weights the given data points \mathbf{p}_i , leading to a spline approximation that is robust to outliers. The distance function $\text{dist}(\cdot, \cdot)$, as it was previously mentioned, measures the Riemannian distance between the tensors.

4. Experimental results

In this section, we present several experiments with noisy synthetic as well as real diffusion tensor data. We also present comparisons with four other existing methods to demonstrate the performance of our proposed algorithm.

We synthesized a tensor field on a 2D lattice of size 33×33 (see fig. 4(a)). For the generation of this field, we chose four diffusion tensors which were then positioned at the corners of the given rectangular lattice. These diffusion tensors were arbitrarily selected from a real DT-MRI data set of a fixed rabbit heart. The *tensors at the rest of the lattice points were obtained via geodesic interpolation* using the methods described earlier. The left plate of figure 4(a) depicts the Fractional Anisotropy (FA) map (see [2] for definition of FA) and the right plate depicts the dominant eigenvector field. The x, y, z components of the dominant eigenvectors of the tensors were assigned Red-Green-Blue (RGB) colors respectively for visualization purposes.

We then subsampled the tensor field by a factor of four and figure 3(a) is a depiction of the same. The Gaussian ellipsoids that represents the diffusion tensors of the synthetic tensor field are shown in the right plate of this figure, and the FA map is shown in the left plate. Then, we generated two noisy tensor fields by adding noise to the subsampled tensor field. The first noisy tensor field (fig. 3(b)) contains 10% outliers which were also arbitrarily selected from a set of outlier diffusion tensors that were contaminating the previously mentioned data set.

For the generation of the second noisy tensor field (fig. 3(c)) a 3×3 symmetric matrix $\mathbf{X}_{i,j}$ was randomly generated from a Gaussian distribution, for every lattice point of the subsampled tensor field $\mathbf{p}_{i,j}$. These matrices were used as vectors in the tangent space of SPD matrices. Then the noisy tensor field of fig. 3(c) was obtained by applying the Riemannian Exponential map $\text{Exp}_{\mathbf{p}_{i,j}}(\mathbf{X}_{i,j})$.

Finally we interpolated the noisy tensor fields by a factor of 4, using the four existing methods a) Riemannian geodesic interpolation [4], b) Log-Euclidean linear interpolation [1], c) Tangent space interpolation [3], d) PDE-based anisotropic non-linear diffusion/interpolation [12], e) our robust tensor spline method with degree 3 (cubic) and also using our own modification of the Log-Euclidean metric, namely, f) a Log-Euclidean spline. The results are presented in figure 4. We use two methods to measure the distance of the estimated tensor fields from the ground truth tensor field of figure 4(a); a) the Riemannian metric and b) the length of the error vector defined as the difference between the ground truth and the estimated dominant eigenvector of the diffusion tensors respectively. These errors are computed at each voxel and the mean and standard deviation of these errors are reported in table 1 for the two noisy data sets. As evident and expected, the error is much smaller for our algorithm in comparison to the others. Moreover, since our method is robust to the presence of outliers in the data, its performance is least affected among all the methods as the corrupting mechanism of the original tensor field was changed. This conclusively demonstrates superior interpolation performance of our algorithm over other exist-

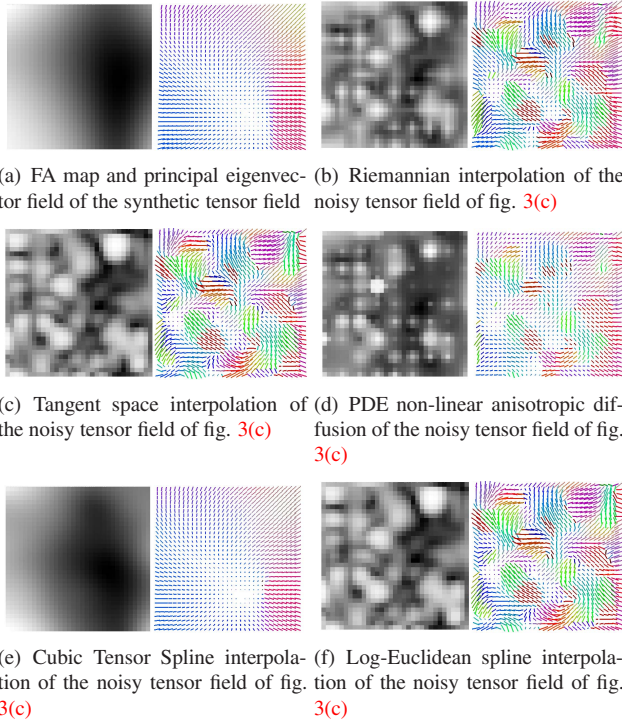


Figure 4. Comparison of interpolation methods.

ing methods.

In figure 5, the first row depicts three images. (a) Shows a slice of a T2-weighted MR image obtained using a zero gradient field. A white box has been used to identify the region of interest (ROI) containing a reasonably noisy section in the DT-MRI slice. The rest of the plates in this figure depict the processing limited to this ROI. Figure (b) depicts the FA map of the data. Higher brightness indicates higher FA values in this depiction. Note the random changes in brightness values in this image caused due to the presence of noise and outliers in the tensor field. Figure (c) depicts the FA computed after applying tensor spline approximation. Note the relatively smooth appearance in brightness corresponding to the FA values, which is biologically more consistent in the shown ROI. The part (d) of the same figure depicts an ellipsoidal visualization (in color) of the original tensor field in the ROI. Note the presence of the outliers in the bottom right part of this image. Image (e) of the same figure shows the result after applying our tensor spline approximation. Notice that all of the outliers have been rejected and the field has been interpolated smoothly.

Finally figure 6 shows another real data example. In this figure the proposed Tensor Spline approximation algorithm is compared with the recently proposed Log-Euclidean geodesic interpolation [1], discussed earlier. The vector field of the dominant eigenvector of the original data is depicted in 6(b). Note that the data contain noise and outliers. Figures 6(c) and 6(d) show the interpolation results

Interpolation errors of the tensor field of fig. 3(b)

	Riem. μ	Riem. σ	SSD μ	SSD σ
Riem. Geodesic	0.3097	0.4765	0.5107	0.4443
Log-Euc Geodesic	0.3126	0.4809	0.6848	0.3556
Tangent space int.	0.4624	0.5719	0.6414	0.3957
PDE interpolation	0.2374	0.4285	0.2022	0.4545
Log-Euc Spline	0.4628	0.5714	0.6553	0.3839
Tensor Spline	0.0008	0.0003	0.0018	0.0494

Interpolation errors of the tensor field of fig. 3(c)

	Riem. μ	Riem. σ	SSD μ	SSD σ
Riem. Geodesic	0.9744	0.4881	0.6462	0.4235
Log-Euc Geodesic	0.9855	0.4976	0.6917	0.3657
Tangent space int.	1.2321	0.5998	0.6680	0.3917
PDE interpolation	0.8394	0.5129	0.5265	0.5084
Log-Euc Spline	1.2315	0.5996	0.5894	0.3764
Tensor Spline	0.2487	0.0917	0.3003	0.5453

Table 1. Mean error and standard deviation for the two noisy synthetic tensor fields

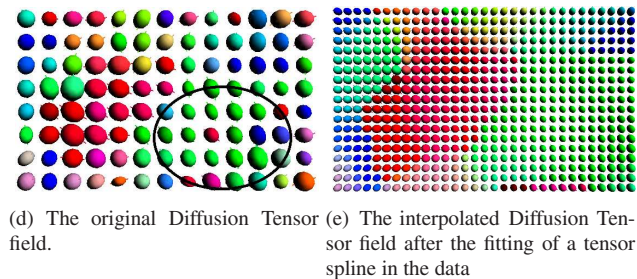
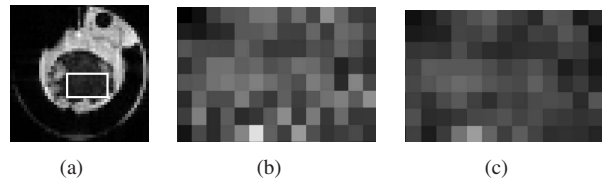


Figure 5. a) A DT-MRI slice of a rabbit heart. The next figures come from the marked region of this image. b) The Fractional Anisotropy (FA) map of the original data. c) The FA map after the fitting of a cubic tensor spline in the data. d) and e) show the ellipsoid visualization of the corresponding tensor fields.

of Log-Euclidean geodesic and Tensor Spline algorithm respectively. Since the Log-Euclidean method is not robust the interpolation result still contains a large amount of the outliers seen in the original data. Notice that in the Tensor Spline approximation results these outliers have been rejected. Figure 6(a) depicts the FA map of the original and the interpolated data. In the case of Tensor Spline approximation (6(a) bottom) the FA map of the approximated data, has also been interpolated smoothly.

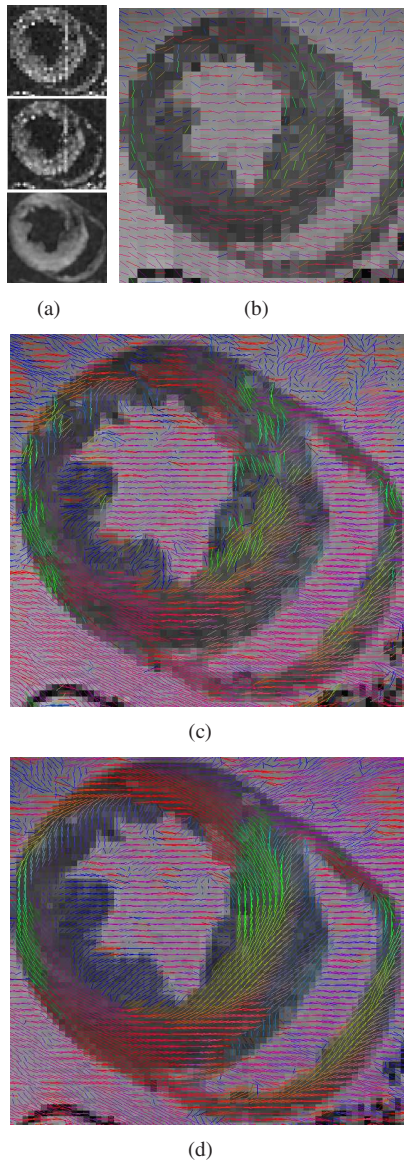


Figure 6. Real DTI from a rabbit heart: a) FA map of the original data (top), after Log-Euclidean interpolation (middle), after cubic Tensor Spline approximation (bottom). The rest of the plates in this figure depict the dominant eigenvector field of (b) the original data, (c) Log-Euclidean interpolation, (d) Tensor Spline approximation.

5. Conclusions

In this paper, we presented a novel and robust spline approximation algorithm that we dub, tensor-splines, given a noisy symmetric positive definite tensor field. We presented comparisons of our algorithm with four existing methods of interpolation for Diffusion Tensor MRI data from fixed heart slices of a rabbit, and presented significantly improved results in the presence of noise and outliers. We also presented validation results for our algorithm using syntheti-

cally generated noisy tensor field data with outliers. Our future work will be focused on applying this algorithm as a module in applications such as registration of DT-MRI data sets, tractography etc. [15, 14].

References

- [1] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. Fast and Simple Calculus on Tensors in the Log-Euclidean Framework. In *Proceedings of MICCAI*, LNCS, pages 259–267, 2005. 1, 2, 5, 6, 7
- [2] P. Basser, J. Mattiello, and D. Lebihan. Estimation of the Effective Self-Diffusion Tensor from the NMR Spin Echo. *J. Magn. Reson. B*, 103:247–254, 1994. 1, 6
- [3] P. Fillard, V. Arsigny, X. Pennec, P. Thompson, and N. Ayache. Extrapolation of sparse tensor fields: Application to the modeling of brain variability. In *IPMI*, pages 27–38, 2005. 1, 5, 6
- [4] P. Fletcher and S. Joshi. Principal geodesic analysis on symmetric spaces: Statistics of diffusion tensors. *Proc. of CVAMIA*, pages 87–98, 2004. 1, 2, 3, 6
- [5] S. Helgason. *Differential geometry, Lie groups, and symmetric spaces*. American Mathematical Society, 2001. 2
- [6] C. Lenglet, M. Rousson, R. Deriche, and O. Faugeras. Statistics on Multivariate Normal Distributions: A Geometric Approach and its Application to Diffusion Tensor MRI. *INRIA Research Report 5242*, 2004. 1, 2, 3
- [7] S. Pajevic, A. Aldroubi, and P. Basser. Continuous Tensor Field Approximation of Diffusion Tensor MRI Data. In *Visualization and Image Processing of Tensor Fields*. Springer, 2005. 3
- [8] X. Pennec. Probabilities and statistics on Riemannian manifolds: basic tools for geometric measurements. *IEEE Workshop on Nonlinear Signal and Image Processing*, 1999. 4
- [9] X. Pennec, P. Fillard, and N. Ayache. A Riemannian framework for tensor computing. *International Journal of Computer Vision*, 65, 2005. 1, 2, 3
- [10] R. San-Jose Estepar, S. Haker, and C.-F. Westin. Riemannian Mean Curvature Flow. In *ISVC*, volume 3804 of LNCS, pages 613–620, December 2005. 1
- [11] Z. Wang, B. Vemuri, Y. Chen, and T. Mareci. A Constrained Variational Principle for Direct Estimation and Smoothing of the Diffusion Tensor Field from DWI. *IPMI*, pages 660–671, 2003. 5
- [12] J. Weickert and M. Welk. Tensor field interpolation with PDEs. *Preprint 142, Saarland University*, 2005. 2, 6
- [13] C.-F. Westin, M. Martin-Fernandez, C. Alberola-Lopez, J. Ruiz-Alzola, and H. Knutsson. Tensor Field Regularization Using Normalized Convolution and markov random fields in a bayesian framework. In J. Weickert and H. Hagen, editors, *Visualization and Image Processing of Tensor Fields*. Springer, 2005. In Press. 2
- [14] D. Xu, S. Mori, M. Solaiyappan, P. C. Zijl, and C. Davatzikos. A framework for callosal fiber distribution analysis. *Neuroimage*, 17(3):1131–1143, November 2002. 8
- [15] H. Zhang, P. A. Yushkevich, and J. C. Gee. Registration of Diffusion Tensor Images. *CVPR*, 1:842–847, 2004. 8