Interpolating 3D Diffusion Tensors in 2D Planar Domain by Locating Degenerate Lines

Chongke Bi[1], Shigeo Takahashi[1], and Issei Fujishiro[2]

[1] Graduate School of Frontier Sciences, The University of Tokyo, Japan
[2] Department of Information and Computer Science, Keio University, Japan

Abstract. Interpolating diffusion tensor fields is a key technique to visualize the continuous behaviors of biological tissues such as nerves and muscle fibers. However, this has been still a challenging task due to the difficulty to handle possible degeneracy, which means the rotational inconsistency caused by degenerate points. This paper presents an approach to interpolating 3D diffusion tensors in 2D planar domains by aggressively locating the possible degeneracy while fully respecting the underlying transition of tensor anisotropy. The primary idea behind this approach is to identify the degeneracy using minimum spanning tree-based clustering algorithm, and resolve the degeneracy by optimizing the associated rotational transformations. Degenerate lines are generated in this process to retain the smooth transitions of anisotropic features. Comparisons with existing interpolation schemes will be also provided to demonstrate the technical advantages of the proposed approach.

1 Introduction

Recent development of visualization techniques for tensor fields has provided an effective means of understanding biological tissues especially in medical applications. Diffusion tensor magnetic resonance imaging (DT-MRI) is such an example where the associated tensor fields are obtained by measuring the motion of water molecules.

In general, a tensor field is obtained as a grid of tensor samples, and thus requires appropriate interpolation of such discrete samples to explore the structures of underlying features. For interpolating diffusion tensor fields, it is important to retain the smooth transition of anisotropic features inherent in the given tensor fields, especially around degenerate points, where at least two of three eigenvalues are equivalent [1]. Zheng et al. [2] proved that degenerate line, which connects degenerate points (the detail will be introduced in Section 4), is the most stable topological structure for 3D tensors, while degenerate points are unstable. However, existing interpolating schemes cannot obtain these degenerate lines to retain the smooth transition of anisotropic features. Figure 1 describes such limitations. Figure 1(a) presents a diffusion tensor field containing two degenerate points, and a degenerate line is obtained using our scheme, as shown in Figure 1(e). However, in Figures 1(b) and (c), we cannot observe such degenerate line while discontinuities appear in Figure 1(d). Note that the color of each ellipsoid indicates the anisotropy of the corresponding tensor value, which is represented by the FA value (Eq. (4)) of the tensor.
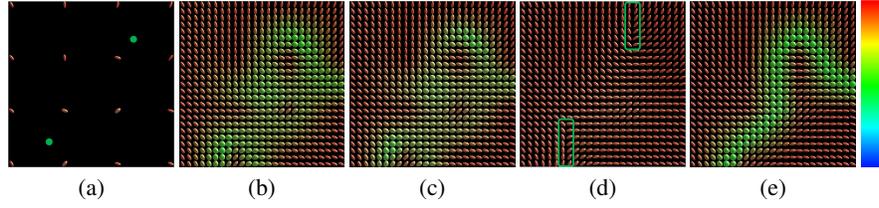
Fig. 1. Interpolating a diffusion tensor field containing two degenerate points. (a) Original tensor samples. Results with the (b) component-wise, (c) Log-Euclidean, (d) geodesic-loxodrome, and (e) proposed interpolation scheme.

This paper presents an approach to interpolating diffusion tensor fields by locating degenerate points and generating degenerate lines. The main idea is to cluster discrete tensor samples with similar anisotropy and orientation using a minimum spanning tree strategy, in order to locate degenerate points, which are connected by degenerate lines. Figure 1(e) presents the result that a degenerate line is obtained between the two degenerate points, which is the primary advantage of the proposed method over the existing interpolation schemes. In this paper, we introduce our method to interpolate 3D tensors in 2D planar domain, which is an initial step of our research in 3D tensor fields.

The remainder of this paper is organized as follows: Section 2 introduces several mathematical prerequisites for diffusion tensors, and then provides a brief survey on related work. Our approach for interpolating tensor fields is detailed first for 1D domain in Section 3, and then for 2D cases even with tensor degeneracy in Section 4. The effectiveness of the proposed approach is presented through the comparison with existing interpolation schemes in Section 5, followed by the conclusion of this paper in Section 6.

2   Related Work

A 3D diffusion tensor can be represented by three real eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3 > 0$, together with the corresponding eigenvectors e_1, e_2, and e_3 that form an orthonormal basis, which can be visualized as an ellipsoid as shown in Figure 2(a). The shape of the ellipsoid depends on the eigenvalues, which are defined as tensor anisotropy. Several metrics for evaluating such anisotropy have been proposed [3], which include linearity (C_l), planarity (C_p), sphericity (C_S), and Fractional Anisotropy (FA) as follows:

$$C_l = (\lambda_1 - \lambda_2)/(\lambda_1 + \lambda_2 + \lambda_3), \tag{1}$$

$$C_p = 2(\lambda_2 - \lambda_3)/(\lambda_1 + \lambda_2 + \lambda_3), \tag{2}$$

$$C_s = 3\lambda_3/(\lambda_1 + \lambda_2 + \lambda_3), \tag{3}$$

$$FA = \frac{\sqrt{3\left((\lambda_1 - \bar{\lambda})^2 + (\lambda_2 - \bar{\lambda})^2 + (\lambda_3 - \bar{\lambda})^2\right)}}{\sqrt{2\left(\lambda_1^2 + \lambda_2^2 + \lambda_3^2\right)}},$$

$$\text{where} \quad \bar{\lambda} = (\lambda_1 + \lambda_2 + \lambda_3)/3. \tag{4}$$
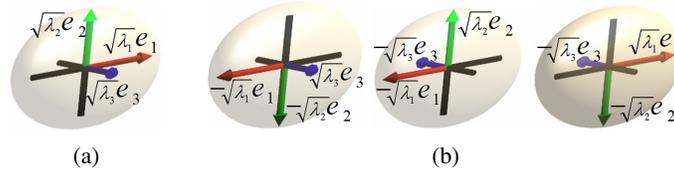
Fig. 2. Sign ambiguity of eigenvectors. (a) A tensor can be represented by an ellipsoid. The directions of the coordinate axes are represented by the arrows. Different colors are assigned to the axes for representing the magnitudes of eigenvalues. (b) Three other possible definitions of the tensor in (a).

Note that $C_l + C_p + C_s = 1$ and $0 \leq C_l, C_p, C_s, FA \leq 1$.

As described in the literature [4], the history of tensor interpolation methods started with naïve schemes as usual, including component-wise interpolation of tensor matrices. These methods, however, incur undesirable change in the tensor anisotropy, and cannot generate degenerate lines between degenerate points, as seen in Figure 1(b). In addition, positive-definiteness of the tensor matrix may not be preserved since the linear interpolation has been applied to each component of the tensor matrix in this scheme.

To alleviate this problem, Batchelor et al. [5] defined a distance function so that we can interpolate the tensors by tracking the corresponding geodesic path on a nonlinearly curved space. Their approach still incurs undesirable transition of the anisotropic features along the interpolated tensors when the associated rotational angle is relatively large. Furthermore, Fletcher et al. [6] modeled the space of diffusion tensors as a Riemannian symmetric manifold and introduced a framework for the statistical analysis of that space. However, their methods suffer from high computational costs because the geodesic path invokes long iterative numerical computations.

Recently, Arsigny et al. [7] developed a Riemannian metric called *Log-Euclidean* to provide a faster computational algorithm. This has been accomplished by transforming tensor samples into their matrix logarithms so that we can perform the tensor interpolation using Euclidean operations. However, it still incurs unnecessary change in the anisotropy of the interpolated tensors, as shown in Figure 1(c). Kindlmann et al. [8] presented a novel tensor interpolation method called the *geodesic-loxodrome*, which discriminates between the isotropic and anisotropic components of the tensors first and then interpolates each component individually. This accomplishes high quality interpolating results, however, at the cost of longer computation times again. The method may also incur undesirable discontinuity over the domain when its boundary has redundant rotation of the tensor orientation, as shown in Figure 1(d).

Any of the aforementioned approaches tried to transform tensor matrices to some specific nonlinear space and perform the interpolation by finding an optimal transition path between the tensors in that space. However, less attention has been paid to the eigenstructures of the tensor matrices. Merino-Caviedes et al. [9] developed a method for interpolating 2D diffusion tensors defined over the 2D planar domain, where they constitute a 3D Euclidean space spanned by the two eigenvalues and the angle between the primary eigenvector and the *x*-axis of the 2D domain. Hotz et al. [10] presented a so-
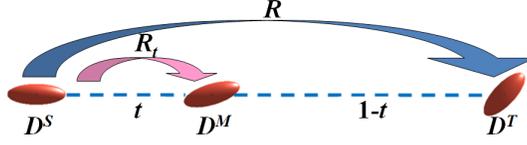
Fig. 3. Interpolating a 1D diffusion tensor field.

phisticated model based on the eigenstructures of the 2D diffusion tensors. In this work, they linearly interpolated between each pair of the eigenvalues and the corresponding pair of eigenvectors component-wise separately. They also located degenerate points over 2D triangulated domain. Readers can refer to a more complete survey in [11].

3 Interpolating 1D Tensor Fields

3.1 Eigenstructure-based Tensor Representation

In this paper, an approach to interpolating diffusion tensors by employing an eigenstructure-based representation is proposed. Nonetheless, such interpolation scheme has not been fully tackled so far, because it cannot provide a unique description of a tensor.

This comes from the fact that each tensor has sign ambiguity in its eigenvector directions since both $Ae_i = \lambda_i e_i$ and $A(-e_i) = \lambda_i(-e_i)(i = 1,2,3)$ hold simultaneously, where A represents the matrix representation of the tensor. Thus, even when we suppose that the three eigenvalues suffice the condition $\lambda_1 \geq \lambda_2 \geq \lambda_3 > 0$, and the associated eigenvectors are all normalized to unit vectors to form a right-handed coordinate system, we still have four different representations for a single tensor, as shown in Figure 2.

For interpolating 1D tensor fields, we first establish the correspondence between each adjacent tensor samples, and then individually interpolate between each pair for seeking the smooth transition of tensor anisotropy values and its associated orientations.

3.2 Optimizing correspondence between tensors

Suppose that we have two tensor samples D^S and D^T, while their normalized eigenvectors are represented as $\{e_1^S, e_2^S, e_3^S\}$, and $\{e_1^T, e_2^T, e_3^T\}$, respectively. The rotation matrix R that transforms between D^S and D^T can be formulated as:

$$R = (p_1 e_1^T, p_2 e_2^T, p_3 e_3^T)(e_1^S, e_2^S, e_3^S)^{-1}, \tag{5}$$

where $p_i (i = 1,2,3)$ is defined to be the sign of each eigenvector e_i, in such a way that $p_i = \pm 1 (i = 1,2,3)$ and $\prod_{i=1}^3 p_i = 1$. The rotation angle θ between D^S and D^T is given by:

$$\theta = \arccos|(\mathrm{tr}R - 1)/2|, \tag{6}$$

where $\mathrm{tr}R$ is the trace of R. We assume $\theta \in [0, \pi/2]$, to remove redundant rotation.

3.3 Interpolation using eigenvalues and eigenvectors

Having fixed the eigenvector directions of two tensor samples, we interpolate their corresponding eigenvalues and eigenvectors individually. Suppose that we calculate the interpolated tensor D^M at the ratio of $t : (1-t)$ in the range [0, 1] between D^S and D^T, as shown in Figure 3. We calculate the three eigenvalues λ_i^M ($i = 1,2,3$) of D^M by linearly interpolating between the eigenvalues of D^S and D^T, and three eigenvectors e_i^M ($i = 1,2,3$) by linearly interpolating the associated rotation angle between them as:

$$\lambda_i^M = (1-t)\lambda_i^S + t\lambda_i^T, \quad \text{and} \tag{7}$$
$$(e_1^M, e_2^M, e_3^M) = R^t(e_1^S, e_2^S, e_3^S). \tag{8}$$

4 Interpolating 2D Tensor Fields

In order to extend the previous formulation to 2D planar domains, we need to handle the following two important technical issues:

1. In 2D cases, the rotational transformation depends on two parameters that define the parameterization of the 2D planar domain. We have to take care of the order of applying the rotation matrices since the rotations do not commute with each other.
2. We need to remove the rotational inconsistency around degenerate points.

4.1 Combination of rotations for 2D cases

For the noncommutative multiplication of rotation matrices, we alleviate the problem by employing Alexa's formulation on linear combination of transformations [12], which enables us to handle the multiplication of rotation matrices as their linear sum. For example, in the square region confined by the discrete tensor samples D_{00}, D_{01}, D_{11}, and D_{10}, where the region is defined as a 2D parametric domain $(s,t) \in [0,1] \times [0,1]$ (Figure 4). If we denote Alexa's commutative multiplication operator by $\oplus$, we can define the tensor D at parametric coordinates (s,t), using the bilinear interpolation as:

$$R = R_x^{s(1-t)} \oplus R_y^{(1-s)t} \oplus R_{xy}^{st}, \tag{9}$$

where, R_x, R_y, and R_{xy} represent the rotation matrices between D_{00} and D_{01}, D_{00} and D_{10}, and D_{00} and D_{11}, respectively. Now the eigenvectors of D can be obtained by applying R to those of D_{00}.

4.2 Locating tensor degeneracy

For 3D diffusion tensors, a tensor is defined as *a degenerate point* if at least two of three eigenvalues are equivalent [1]. Zheng et al. [2] proved that the most stable topological structure for 3D diffusion tensors is *a degenerate line* which connects degenerate points in tensor fields. We define the square containing a degenerate point as *a degenerate cell*.

We will introduce how to locate the position of degenerate points. Since each degenerate point is contained in some degenerate cell, we can locate degenerate cells
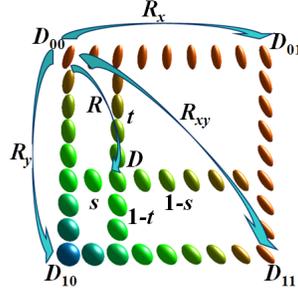
Fig. 4. Interpolating a 2D diffusion tensor field.

instead of degenerate points. A minimum spanning tree (MST)-based clustering algorithm is employed so that we can group tensor samples (or clusters) that share similar anisotropic values and their associated orientations. This is accomplished by introducing the following dissimilarity metric that evaluates the proximity between the neighbor tensor samples:

$$d(D^S, D^T) = \alpha|C_l^S - C_l^T| + \beta|C_p^S - C_p^T| + \gamma(|\theta_{S,T}|/(\pi/2)), \qquad (10)$$

where, C_l^S and C_l^T represent the C_l values of the two tensor samples D_S and D_T, and C_p^S and C_p^T are the corresponding C_p values. In addition, $\theta_{S,T}$ is the minimal rotation angle between the right-handed coordinate systems defined by the two sets of eigenvector directions. This is calculated by selecting one representation for each tensor (Figure 2). This metric satisfies the fundamental axioms for metric spaces, and tries to evaluate both the differences in the anisotropy and the rotational angle between two tensors. Our experiments suggest that the parameter setting $\alpha = 4$, $\beta = 2$ and $\gamma = 1$ is reasonable for this purpose because we are more likely to group high anisotropic tensor samples in earlier stages of this clustering process. Figure 5(a) shows such an example, where we find the most similar pair of samples among the candidate adjacent pairs and connect the pair with MST-based clustering. We continue this process until all the tensor samples fall into a single cluster as shown in Figure 5(b).

After finishing this process, we can identify the pair of adjacent tensor samples as *a degenerate pair* if their rotation angle is more than $\pi/2$. By counting the number of degenerate pairs, we can locate degenerate cells. This is because a degenerate cell contains an odd number of degenerate pairs (Figure 6). Figure 5(c) shows such an example, where two degenerate points are located. However, from the interpolated result in Figure 5(d), we find that the rotational inconsistency exists just on the degenerate pairs between the two degenerate points. Therefore, we should introduce degenerate lines to remove the rotational inconsistency and connect the degenerate points.

4.3 Rotational inconsistency around degenerate points

The main idea of resolving the rotational inconsistency is to transform aforementioned degenerate pairs into non-degenerate ones. For this purpose, we optimize the rotational
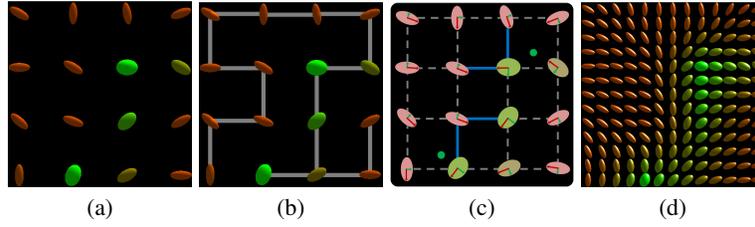
Fig. 5. MST-based clustering in tensor field: (a) Original tensor samples (b) The final MST that covers all the tensor samples. (c) Degenerate points (green points) and degenerate pairs (blue lines) obtained through the MST-based clustering algorithm. (d) Interpolated result obtained through the MST-based clustering algorithm only.

Fig. 6. Configurations of degenerate pairs in a unit square. A pair of tensor samples is drawn in blue if it is degenerate, and the square is shaded in red if it contains a degenerate point.

transformation between the two end tensor samples. This is achieved by selecting one of the two tensors, and changing the order of its eigenvectors to minimize the rotational angle in between.

Figure 7 illustrates this process. We focus on a degenerate pair indicated by the red segment in Figure 7(a), and try to transform it into non-degenerate one. We basically select one of the two tensors (circled by a broken circle in red in Figure 7(a)) as the one that has not been visited yet, while the tensor with low anisotropy is more likely to be selected if both are unvisited. The order of its eigenvectors is then rearranged in order to minimize the rotational angle between these two tensors where the first and second eigenvectors are exchanged in this case. Finally, we label the adjusted tensor as visited, and check the incident pairs as represented by the yellow segments in Figure 7(b), because the change in the tensor representation may transform the neighbor pairs into degenerate ones. Now we select another degenerate pair and repeat this process, as shown in Figure 7(c), until all the degenerate pairs are resolved into non-degenerate ones.

Actually, our method tries to introduce isotropic tensors in the region between the neighbor degenerate pairs, and these isotropic tensors constitute of degenerate lines. This has been finished by optimizing the rotational angle between the two end tensors on degenerate pairs. Therefore, our method is able to remove the rotational inconsistency by generating degenerate lines, as shown in Figure 7(d).

Furthermore, degenerate lines do not affect the anisotropy of the region without degenerate pairs. The reason is that the orders of the eigenvalues of all the tensors in such region change at the same time. For example, there is no degenerate pairs in the
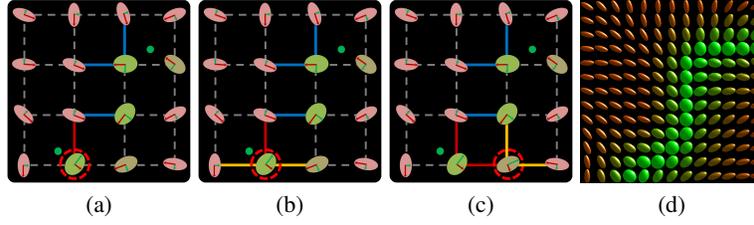
Fig. 7. Generating degenerate lines by resolving degenerate pairs of tensor samples: (a) Find one of the remaining degenerate pairs in red, and transform it into non-degenerate one. (b) Check if the pairs around the selected tensor sample have redundant rotations. (c) If they exist, select them and transform them into non-degenerate ones. We continue this process until all the remaining degenerate pairs are resolved. (d) Result after rotational inconsistency has been resolved, where a degenerate line is generated between the two degenerate points.

rightmost and bottommost cell in Figure 7(a). Before we transform the degenerate pairs into non-degenerate ones, all the orders of their eigenvalues are $\{\lambda_1, \lambda_2, \lambda_3\}$. After we finish transforming the degenerate pairs into non-degenerate ones, all the orders of their eigenvalues are changed into $\{\lambda_2, \lambda_1, \lambda_3\}$. Therefore, the anisotropy of interpolated tensors in such region will not be affected, as shown in Figure 7(d).

5   Results and Discussions

In this section, we demonstrate the effectiveness of our approach in the 2D tensor fields without degenerate points and with degenerate points, respectively. We also compare the results of our approach with those obtained by other existing schemes.

Figure 8(a) presents a 2D case where 6×6 discrete tensor samples guide the underlying "X"-like shape. No degenerate points are included in this dataset. Figures 8(b), (c), and (d) show interpolated tensor samples obtained using the component-wise, Log-Euclidean and our schemes, respectively. The two conventional schemes unexpectedly incur low anisotropic features around the crossing of the two anisotropic line features, while our method can still maximally preserve the underlying anisotropic structures.

Figure 9 shows results of interpolating a real human brain DT–MRI dataset ($256 \times 256 \times 30$). Figure 9(a) is the 17th axial slice of the original dataset, which is downsampled into 128×128, and Figure 9(b) is the zoom-up view of the region boxed in a square in Figure 9(a) where two fibers intersect with each other. To interpolate the tensor samples in this region, a degenerate line should be obtained to separate these two fibers. Figures 9(c), (d), (e), and (f) show the interpolation results with the component-wise, Log-Euclidean, geodesic-loxodrome, and our schemes, respectively. The results show that our scheme can produce a degenerate line composed by lower anisotropic tensors, which separates these two fibers, and our scheme can also respect the anisotropic features of the two fibers. However, neither of the component-wise or Log-Euclidean schemes can respect the underlying anisotropic features of the left fiber appropriately. The geodesic-loxodrome scheme can fully respect the anisotropy of the two fibers. Un-

(a) original samples   (b) component-wise   (c) Log-Euclidean   (d) our scheme
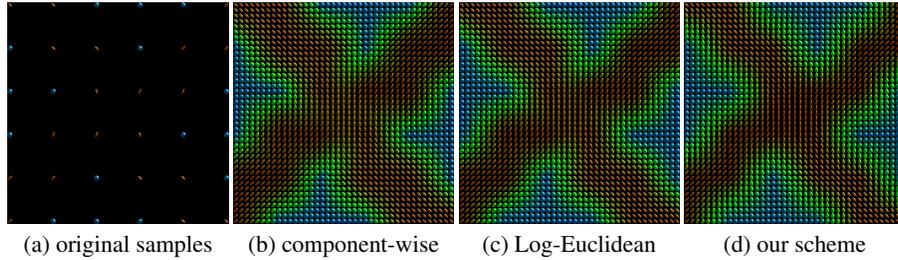
Fig. 8. Interpolating a 2D diffusion tensor field. (a) Original tensor samples. Results with the (b) component-wise, (c) Log-Euclidean, and (d) our interpolation schemes.

fortunately, all these three schemes cannot generate the degenerate line to separate the two fibers.

6   Conclusion

An approach to interpolating diffusion tensor fields through the analysis of the associated eigenvalues and eigenvectors has been presented in this paper. Compared with other existing interpolation schemes, the present approach can maximally respect the underlying anisotropy of the given dataset, especially in the tensor fields containing degenerate points. In our method, degenerate points can be connected by degenerate lines, which are the most stable topological structure for 3D tensors, by employing MST-based algorithm. We also solve the non-commutative property of matrix composition by taking advantage of Alexa's linear combination of transformations [12].

However, the present approach may not be able to effectively handle noisy datasets, where such anisotropic features are rather scattered over the data domain. We are currently working on extending our 2D scheme to 3D so as to enable ones to perform detailed analysis of complex fiber structures.

References

1. Hesselink, L., Levy, Y., Lavin, Y.: The topology of symmetric, second-order 3D tensor fields. IEEE Transactions on Visualization and Computer Graphics **3** (1997) 1–11
2. Zheng, X., Pang, A.: Topological lines in 3D tensor fields. In: Proceedings of the Conference on Visualization '04, IEEE Computer Society (2004) 313–320
3. Westin, C., Peled, S., Gudbjartsson, H., Kikinis, R., Jolesz, F.: Geometrical diffusion measures for MRI from tensor basis analysis. In: Proceedings of ISMRM '97. (1997) 1742

(a) original samples   (c) component-wise   (e) geodesic-loxodrome

(b) zoom-up view of the target region   (d) Log-Euclidean   (f) our scheme
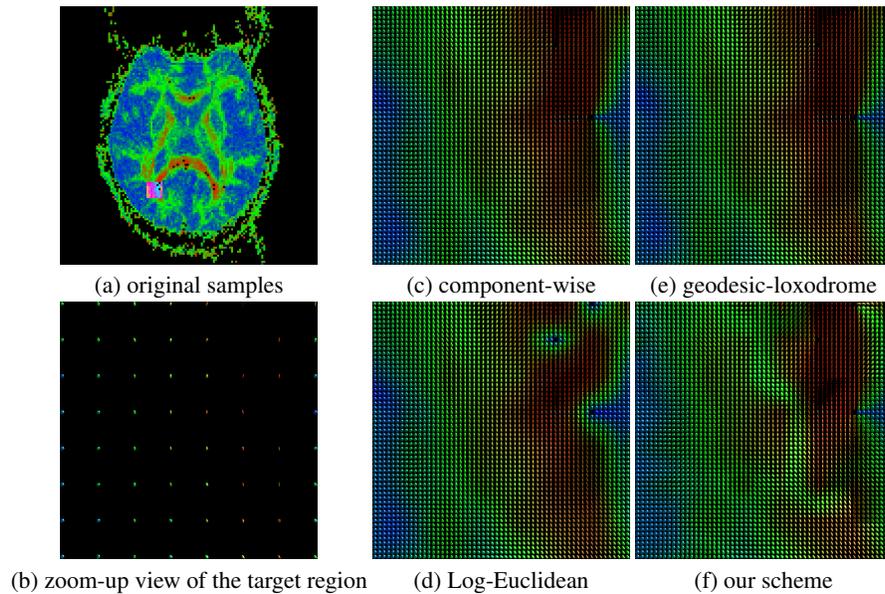
Fig. 9. Human brain DT–MRI dataset: (a) Original data samples. (b) The zoom-up view of the region where two anisotropic features intersect. Interpolation results obtained using (c) component-wise, (d) Log-Euclidean, (e) geodesic-loxodrome, and (f) our interpolation schemes.

4. Westin, C.F., Maier, S.E., Mamata, H., Nabavi, A., Jolesz, F.A., Kikinis, R.: Processing and visualization for diffusion tensor MRI. Medical Image Analysis **6** (2002) 93–108
5. Batchelor, P.G., Moakher, M., Atkinson, D., Calamante, F., Connelly, A.: A rigorous framework for diffusion tensor calculus. Magnetic Resonance in Medicine **53** (2005) 221–225
6. Fletcher, P.T., Joshi, S.: Riemannian geometry for the statistical analysis of diffusion tensor data. Signal Processing **87** (2007) 250–262
7. Arsigny, V., Fillard, P., Pennec, X., Ayache, N.: Log-Euclidean metrics for fast and simple calculus on diffusion tensors. Magnetic Resonance in Medicine **56** (2006) 411–421
8. Kindlmann, G., Estepar, R., Niethammer, M., Haker, S., Westin, C.: Geodesic-Loxodromes for diffusion tensor interpolation and difference measurement. In: Proceedings Medical Image Computing and Computer-Assisted Intervention. Volume 4791 of Springer LNCS. (2007) 1–9
9. Merino-Caviedes, S., Martin-Fernandez, M.: A general interpolation method for symmetric second-rank tensors in two dimensions. In: Proceeding of IEEE International Symposium on Biomedical Imaging. (2008) 931–934
10. Hotz, I., Sreevalsan-Nair, J., Hamann, B.: Tensor field reconstruction based on eigenvector and eigenvalue interpolation. In: Scientific Visualization: Advanced Concepts. Volume 1. (2010) 110–123
11. Laidlaw, D., Weickert, J., eds.: Visualization and Processing of Tensor Fields. Springer (2009)
12. Alexa, M.: Linear combination of transformations. ACM Transactions on Graphics **21** (2002) 380–387