Noname manuscript No.
(will be inserted by the editor)

Anisotropy preserving DTI processing

Anne Collard · Silvère Bonnabel · Christophe Phillips · Rodolphe Sepulchre

Received: date / Accepted: date

Abstract Statistical analysis of Diffusion Tensor Imaging (DTI) data requires a computational framework that is both numerically tractable (to account for the high dimensional nature of the data) and geometric (to account for the nonlinear nature of diffusion tensors). Building upon earlier studies exploiting a Riemannian framework to address these challenges, the present paper proposes a novel metric and an accompanying computational framework for DTI data processing. The proposed approach grounds the signal processing operations in interpolating curves. Well-chosen interpolating curves are shown to provide a computational framework that is at the same time tractable and information relevant for DTI processing. In addition, and in contrast to earlier methods, it provides an interpolation method which preserves anisotropy, a central information carried by diffusion tensor data.

This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office. The scientific responsibility rests with its author(s).

A. Collard, C. Phillips, R. Sepulchre
Departement of Electrical Engineering and Computer Science, University of Liège, 4000 Liège, Belgium
Tel.: +32-4366 3749
E-mail: anne.collard@ulg.ac.be

R. Sepulchre
University of Cambridge, Department of Engineering, Trumpington Street, Cambridge CB2 1PZ, United Kingdom

S. Bonnabel
Robotics center, Mathématiques et Systèmes, Mines ParisTech, Boulevard Saint-Michel 60, 75272 Paris, France

C. Phillips
Cyclotron Research Centre, University of Liège, 4000 Liège, Belgium

Keywords diffusion tensor MRI · interpolation · spectral decomposition · anisotropy · quaternions · Riemannian manifold

1 Introduction

Diffusion-weighted imaging (DWI) allows non-invasive quantification of the self diffusion of water in vivo. In biological tissues, characterized by cell membranes and cytostructures, the movement of water is restricted because of these barriers. In tissues such as white matter, which is highly directional, the resulting movement of water is therefore anisotropic. In this way, high diffusion anisotropy reflects the underlying directional arrangement of white matter fibre bundles. Diffusion measurements (which use the same tools as magnetic resonance imaging (MRI)) can characterize this anisotropy. The most common representation of the directional diffusion is through the use of diffusion tensors, a formalism introduced by Basser et al in 1994 (Basser et al, 1994). Since then, other higher level representations have been introduced, such as the Q-Ball Imaging (Tuch, 2004) and the Diffusion Kurtosis Imaging (Jensen and Helpern, 2010). In the context of Diffusion Tensor Imaging (DTI), each voxel of the image contains a diffusion tensor, which is derived from a set of DWI measured in different directions. A diffusion tensor is a symmetric positive definite matrix whose general form is given by

$$\mathbf{D} = \begin{pmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{xy} & D_{yy} & D_{yz} \\ D_{xz} & D_{yz} & D_{zz} \end{pmatrix} \tag{1}$$

where D_{xx}, D_{yy}, D_{zz} relate the diffusion flows to the concentration gradients in the x, y and z directions. The off-diagonal terms reflect the correlation between diffusion flows and concentration gradients in orthogonal directions. This diffusion tensor can be graphically represented as an ellipsoid. This ellipsoid takes the three eigenvectors of the matrix as principal axes (representing the three principal directions of diffusion). The length of the axes, related to the intensities of diffusion along them, is determined by the eigenvalues. Diffusion tensor images can thus be viewed as fields of ellipsoids.

Classical image processing methods have been developed for scalar fields. As a result, early processing of DTI data first converted the tensor information into scalar data, for instance focusing on the scalar measure of fractional anisotropy (FA), see *e.g.* (Alexander and Gee, 2000). However, the tensor nature of DTI data soon motivated a generalization of signal processing methodological frameworks to tensor fields. In particular, methods based on the Riemannian geometry of symmetric positive definite matrices have emerged (Pennec et al, 2006; Fletcher and Joshi, 2007; Castaño-Moraga et al, 2006; Gur and Sochen, 2007; Batchelor et al, 2005; Lenglet et al, 2006, 2009) because the geometric framework provides a nonlinear generalization of calculus in linear spaces.

The present paper also adopts a geometric framework but puts the emphasis on interpolation curves as the central mathematical object required for signal processing in nonlinear spaces. Interpolating curves need *not* be geodesics to be practically relevant. Instead, the choice of interpolation curves is problem dependent and a subtle trade-off between practical considerations (such as numerical tractability) and theoretical considerations (such as invariance properties by certain group actions for the robustness of the processing operations). We advocate the use of interpolating curves that interpolate separately the eigenvalues of the tensor and its orientation. This is similar in spirit to several previous works (Weldeselassie et al, 2009; Tschumperlé and Deriche, 2001; Ingalhalikar et al, 2010) and offers good control on the interpolation of the spectral properties and in particular anisotropy, a key DTI information.

The current work is different from the works grounded in the affine-invariant geometry of the cone (Pennec et al, 2006; Arsigny et al, 2007), which suffer from swelling anisotropy effect but are nevertheless widely used in practice mainly because of their computational advantage. In the present paper, we recover the tractability of the Log-Euclidean framework by handling rotation in the space of quaternions, taking inspiration of their widespread use in robotics. Our approach is also different from the Procrustes analysis proposed in (Dryden et al, 2009; Zhou, 2010; Zhou et al, 2013). The geometry proposed in those papers shares the invariance properties of our framework but does not enjoy similar computational and anisotropy preserving properties.

Our 'spectral-quaternion' interpolating curves are easy to compute, and allow for a number of desirable properties, including geometric interpolation of the eigenvalues (and of the volume), linear interpolation of (Hilbert) anisotropy, and shadowing of rotations when the anisotropy is small, that is, when the orientation is uncertain. The paper focuses on the theoretical foundations of the proposed framework, not restricted to DTI processing, but also illustrates the potential of the proposed approach for DTI processing through basic filtering and interpolation operations needed in this application. The paper is organized as follows: Section 2 introduces four different interpolating curves for symmetric definite-positive matrices. Section 3 studies the mathematical properties of those curves, namely their computational cost, invariances, and the way they deal with diffusion information. Section 4 explores the computation of weighted means using our framework, which is important in all the processing steps. Section 5 uses the interpolating curves to define metrics and similarity measures. Section 6 illustrates the result of our framework in different applications, while Section 7 contains concluding remarks.

2 Interpolating curves in $\mathrm{S}^+(3)$

Given two positive numbers $s_1 > 0$ and $s_2 > 0$, geometric interpolation between s_1 and s_2 is defined by the parametric curve

$$\begin{aligned} s(t) &= \exp((1-t)\log s_1 + t\log s_2) \\ &= s_1 \exp(t \log\left(\frac{s_2}{s_1}\right)), \quad 0 \le t \le 1 \end{aligned} \tag{2}$$

which can be viewed as a linear interpolation between the logarithms of the scalars, followed by an exponentiation. Viewing positive definite matrices as matrix generalizations of positive numbers, we search for matrix analogues of the interpolating curve (2). Table 1 summarizes four such families. We briefly review their underlying geometry.

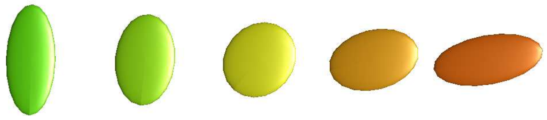
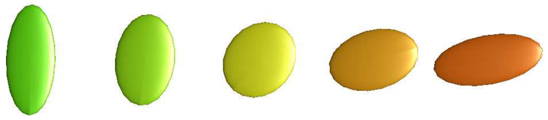
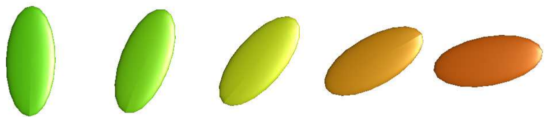
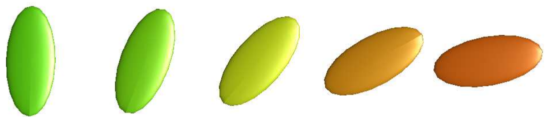
2.1 Affine-invariant Riemannian geometry

The interpolating curve

$$\mathbf{S}(t) = \mathbf{S}_1^{1/2} \exp(t \log(\mathbf{S}_1^{-1/2}\mathbf{S}_2\mathbf{S}_1^{-1/2}))\mathbf{S}_1^{1/2} \tag{3}$$

has the geometric interpretation of a geodesic (*i.e.* minimal length curve) for the affine-invariant metric (Pennec et al, 2006). The use of the affine-invariant Riemannian framework was first proposed for DTI processing

Table 1 Four matrix interpolating curves on the space of positive definite matrices $\mathrm{S}^+(3)$.

 Affine-invariant interpolation $\mathbf{S}(t) = \mathbf{S}_1^{1/2} \exp(t \log(\mathbf{S}_1^{-1/2}\mathbf{S}_2\mathbf{S}_1^{-1/2}))\mathbf{S}_1^{1/2}$	 Log-Euclidean interpolation $\mathbf{S}(t) = \exp((1-t)\log(\mathbf{S}_1) + t\log(\mathbf{S}_2))$
 Spectral interpolation $\mathbf{S}(t) = \mathbf{R}(t)\Lambda(t)\mathbf{R}(t)^T$, with $\mathbf{R}(t) = \mathbf{R}_1 \exp(t\log(\mathbf{R}_1^T\mathbf{R}_2))$ and $\Lambda(t) = \exp((1-t)\log(\Lambda_1) + t\log(\Lambda_2))$	 Spectral Quaternion interpolation $\mathbf{S}(t) = \mathbf{R}_q(t)\Lambda(t)\mathbf{R}_q(t)^T$ $\mathbf{R}_q$: rotation matrix corresponding to $\mathbf{q}(t) = \frac{(1-t)\mathbf{q}_1 + t\mathbf{q}_2}{\|(1-t)\mathbf{q}_1 + t\mathbf{q}_2\|}$ $\Lambda(t) = \exp((1-t)\log(\Lambda_1) + t\log(\Lambda_2))$

in 2006, see (Pennec et al, 2006; Lenglet et al, 2006), see also (Fletcher and Joshi, 2007; Fletcher et al, 2009; Castaño-Moraga et al, 2006; Gur and Sochen, 2007; Batchelor et al, 2005; Lenglet et al, 2009). The affine-invariant geometry is the natural geometry of $\mathrm{S}^+(3)$, the space of 3×3 positive definite matrices, considered as the homogeneous space $\mathrm{Gl}(3)/\mathrm{O}(3)$, where $\mathrm{Gl}(3)$ is the space of general linear matrices (representing all the possible affine transformations) and $\mathrm{O}(3)$ is the space of orthogonal matrices of size 3, see (Smith, 2005). This geometry also plays an important role in statistics (Smith, 2005; Pennec et al, 2006; Ando et al, 2004; Petz and Temesi, 2005; Moakher and Zéraï, 2011; Moakher, 2005; Burbea and Rao, 1982; Skovgaard, 1984) and in convex optimization (Nesterov et al, 1994). A limitation of this geometry for DTI processing is illustrated in Table 1: the midpoint of the interpolation curve between two anisotropic tensors tends to be isotropic.

2.2 Log-Euclidean Riemannian geometry

The interpolating curve

$$\mathbf{S}(t) = \exp((1-t)\log(\mathbf{S}_1) + t\log(\mathbf{S}_2)) \tag{4}$$

has the geometric interpretation of a geodesic for the Log-Euclidean metric (Arsigny et al, 2007). Here one uses the matrix logarithm to define a global embedding of $\mathrm{S}^+(3)$ into the (linear) space of symmetric matrices (*i.e.* we exploit the property that the nonlinear space $\mathrm{S}^+(3)$ is mapped to the linear space $\mathbb{R}^{3\times 3}$ by the (matrix) log mapping). This is similar to the definition of geometric interpolation for scalars (2), which uses the scalar logarithm to map positive scalars on real numbers. The Log-Euclidean metric was first proposed for DTI processing in (Arsigny et al, 2006). It has become a popular framework for DTI processing because of its computational advantage over the affine-invariant geometry (Goodlett et al, 2009; Chiang et al, 2008; Ingalhalikar et al, 2010; Castro et al, 2007; Weldeselassie and Hamarneh, 2007; Arsigny et al, 2006; Fillard et al, 2007; Yeo et al, 2009, 2008; Lepore et al, 2006; Awate et al, 2007). Table 1 suggests that it is a good substitute for the affine-invariant interpolation but that it suffers the same limitation regarding anisotropy. Indeed, the Log-Euclidean geometry is extrinsic, to the contrary of the affine-invariant one which is intrinsic. Extrinsic geometry produces a good approximation of the intrinsic one while close to the identity, but the quality of the approximation decreases while moving away from it.

2.3 Spectral geometry

The interpolation curve given by the equations

$$\mathbf{S}(t) = \mathbf{U}(t)\Lambda(t)\mathbf{U}(t)^T, \tag{5}$$

$$\mathbf{U}(t) = \mathbf{U}_1 \exp(t\log(\mathbf{U}_1^T\mathbf{U}_2)) \tag{6}$$

$$\Lambda(t) = \exp((1-t)\log(\Lambda_1) + t\log(\Lambda_2)) \tag{7}$$

has the geometric interpretation of a geodesic in the product space $\mathcal{M} = \mathrm{SO}(3) \times \mathrm{D}^+(3)$, with the bi-invariant metric of $\mathcal{M}$ viewed as a Lie group. $\mathrm{SO}(3)$ is the special orthogonal group of rotation matrices with determinant equal to 1, and $\mathrm{D}^+(3)$ is the group of diagonal matrices with positive elements. The Lie group $\mathcal{M}$ is closely related to $\mathrm{S}^+(3)$ because of the spectral decomposition of a positive definite matrix

$$\mathbf{S} = \mathbf{U}\Lambda\mathbf{U}^T\,, \tag{8}$$

where $\mathbf{U}$ is an orthogonal matrix containing the eigenvectors (principal axes) of the ellipsoid and

$$\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \lambda_3)$$

is the diagonal matrix of eigenvalues (axes lengths). We will assume that $\lambda_1 \geq \lambda_2 \geq \lambda_3 > 0$ and impose $\det(\mathbf{U}) = 1$, so that U belongs to the space of special orthogonal matrices, SO(3).

Consider $\mathrm{S}^+_*(3)$ the subset of $\mathrm{S}^+(3)$ with distinct eigenvalues $\lambda_1 > \lambda_2 > \lambda_3$. Every tensor of $\mathrm{S}^+_*(3)$ can be represented by four distinct elements of the Lie group $\mathcal{M}$, see Figure 1. The four rotation matrices of three rotations of angle π around the principal axes plus the identity form the discrete group G. The formal identification

$$\mathrm{S}^+_*(3) \simeq (\mathrm{SO}(3)/\mathrm{G}) \times \mathrm{D}^+(3)\,,$$

where $\simeq$ is the standard notation to indicate a diffeomorphism between spaces, thus justifies the spectral interpolation (5) provided that $\mathbf{U}_1$ and $\mathbf{U}_2$ are univocally selected among the possible discrete spectral decompositions of $\mathbf{S}_1$ and $\mathbf{S}_2$. Denoting GU_2 the discrete group of four rotation matrices associated to the spectral decomposition of $\mathbf{S}_2$, a reasonable choice is to choose $\mathbf{U}_1$ arbitrarily and choose $\mathbf{U}_2$ according to

$$\mathbf{U}_2 = \arg \min_{\mathbf{U} \in \mathrm{GU}_2} d_{\mathrm{SO}(3)}(\mathbf{U}_1, \mathbf{U}) \tag{9}$$

$$= \arg \min_{\mathbf{U} \in \mathrm{GU}_2} || \log(\mathbf{U}_1^T \mathbf{U}) ||_2 \tag{10}$$

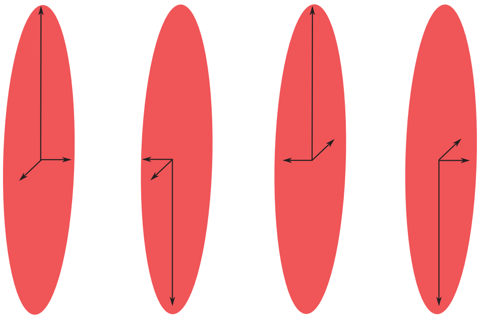


Fig. 1 Non-uniqueness of the spectral decomposition. Four different rotation matrices parametrize the same tensor.

The choice of the spectral geometric framework for DTI processing has been first proposed in (Tschumperlé and Deriche, 2001). Table 1 suggests that the decoupling of rotations and scaling in interpolation is favorable to the conservation of anisotropy along the interpolation curve.

The main reason why spectral interpolation is not commonly used in DTI processing is computational. The 'realignment step' involved in (10) is a local operation that has to be performed for each tensor of any image, leading to prohibitive computations for large-scale DTI processing.

2.4 Spectral quaternion interpolation

The interpolating curve given by the equations

$$\mathbf{S}(t) = \mathbf{U}_q(t) \Lambda(t) \mathbf{U}_q(t)^T \tag{11}$$

where

$$\Lambda(t) = \exp((1-t)\log(\Lambda_1) + t\log(\Lambda_2))\,,$$

$\mathbf{U}_q(t)$ is the rotation matrix associated to the unit quaternion $\mathbf{q}(t) \in \mathbb{H}_1$ and $\mathbf{q}(t)$ satisfies

$$\mathbf{q}(t) = \frac{(1-t)\mathbf{q}_1 + t\mathbf{q}_2}{||(1-t)\mathbf{q}_1 + t\mathbf{q}_2||}\,. \tag{12}$$

has the geometric interpretation of a geodesic in the product space $\mathbb{H}_1 \times \mathrm{D}^+(3)$ where $\mathbb{H}_1$ is the group of quaternions of norm 1 equipped with the Euclidean metric of its natural embedding in $\mathbb{R}^4$. Using the embedding of SO(3) into the linear space of quaternion is very common in robotics in order to save on computational time. In this sense, the interpolating curve (11) can be considered as a computationally convenient substitute of (5), in the same way as (4) simplifies the computation of (3).

A unit quaternion is generally denoted by $\mathbf{q} = (a, \mathbf{V})$ where a is associated to the angle of rotation by $\theta = 2\arccos(a)$ and $\mathbf{V}$ is associated to the axis $\mathbf{w}$ of rotation through $\mathbf{w} = \mathbf{V}/\sin(\theta/2)$. From $\mathbf{q}$, the associated rotation matrix $\mathbf{R}$ is given by

$$\mathbf{R} = \exp \begin{pmatrix} 0 & -w_3\theta & w_2\theta \\ w_3\theta & 0 & -w_1\theta \\ -w_2\theta & w_1\theta & 0 \end{pmatrix}\,. \tag{13}$$

The construction of $\mathbf{q}$ from $\mathbf{R}$ is given by

$$\theta = \arccos((\mathrm{trace}(\mathbf{R}) - 1)/2) \tag{14}$$

$$\mathbf{w} = \frac{1}{2\sin\theta} \begin{pmatrix} R_{3,2} - R_{2,3} \\ R_{1,3} - R_{3,1} \\ R_{2,1} - R_{1,2} \end{pmatrix}\,. \tag{15}$$

Finally, we have $a = \cos(\theta/2)$, $\mathbf{V} = \sin(\theta/2)\mathbf{w}$. Note that the opposite quaternion given by $(-a, -\mathbf{V})$ represents the same rotation matrix. Using this representation, rotations can be manipulated as Euclidean vectors, which decreases the computational cost.

The realignment step in the quaternion space uses the Euclidean (chordal) distance

$$d(\mathbf{q}_1, \mathbf{q}_2) = ||\mathbf{q}_1 - \mathbf{q}_2|| \tag{16}$$

Because $\mathbb{H}_1$ provides a double covering of SO(3), eight different quaternions represent the same orientation.

We denote by $\mathcal{Q}$ this set of quaternions. Realignment in $\mathbb{H}_1$ thus involves solving

$$d(\mathcal{Q}_1, \mathcal{Q}_2) = \min_{\mathbf{q}_2 \in \mathcal{Q}_2} ||\mathbf{q}_1 - \mathbf{q}_2|| \tag{17}$$

with $\mathbf{q}_1^r$ arbitrarily chosen in $\mathcal{Q}_1$. This quaternion will be called the *reference* quaternion.

Because $\mathbf{q}_1^r$ and $\mathbf{q}_2$ have unit norm, we have the simplification

$$||\mathbf{q}_1^r - \mathbf{q}_2||^2 = 2 - 2\,\mathbf{q}_1^r.\mathbf{q}_2 \tag{18}$$

which replaces (17) with

$$\mathbf{q}_2^a = \arg\max_{\mathbf{q}_2 \in \mathcal{Q}_2} \mathbf{q}_1^r.\mathbf{q}_2 \tag{19}$$

$$d(\mathcal{Q}_1, \mathcal{Q}_2) = ||\mathbf{q}_1^r - \mathbf{q}_2^a|| \tag{20}$$

where $\mathbf{q}_2^a$ is called the *realigned* quaternion. The computation of (19) and (20) is very fast compared to (10) : the eight scalar products $\mathbf{q}_1^r.\mathbf{q}_2$ can be computed through a single matrix product between the 1×4 vector representing $(\mathbf{q}_1^r)^T$ and the 4×8 matrix formed by the eight quaternions $\mathbf{q}_2$. In contrast, computing the distance (10) requires four logarithms of product of 3×3 matrices, which is expensive. The selection of the parametrization of rotations as quaternions thus enables the framework to be computationally tractable.

3 Computational and invariance properties

The four interpolation methods discussed in the previous section are valid matrix generalizations of the scalar formula (2). Here we compare their relative merits for DTI processing.

3.1 Computational cost

The two interpolation methods shown in the left column of Table 1 (equations (3) and (5)) are intrinsic whereas the methods shown in the right column of Table 1 (equations (4) and (11)) are extrinsic, that is, the interpolating curve is a projection on the manifold of an interpolating curve in the embedding space. Extrinsic geometries lead to significant computational savings because they reduce the use of matrix exponential and matrix logarithm. Table 2 illustrates how the difference translates into computational savings on a (non optimized) MATLAB code. We expect that the qualitative difference between intrinsic curves and extrinsic interpolations will scale up dramatically in real DTI signal processing.

Table 2 Computational time of computing 1000 distances between a reference and random samples from a Wishart distribution. The computations are performed on a Intel Core 2 Duo 2,66 GHz with 4Go of RAM machine using a (non optimized) MATLAB code.

Affine-invariant	Log-Euclidean	Spectral	Spectral-quaternions
0.47 s	0.17 s	0.65 s	0.11 s

3.2 Invariances

The scalar interpolation formula (2) is invariant by scaling, that is

$$\forall \lambda > 0 : s(t; \lambda s_1, \lambda s_2) = \lambda\, s(t; s_1, s_2)\,. \tag{21}$$

This invariance property is desirable when processing positive measurements that provide physical intensities, because it makes the process robust to units (or calibration). Likewise, the four matrix interpolation formulas are invariant both by congruence and by scaling, that is

$$\forall \lambda \in \mathbb{R}^+, \forall \mathbf{U} \in \mathrm{SO}(3) :$$

$$\mathbf{S}(t; \lambda \mathbf{U}\mathbf{S}_1\mathbf{U}^T, \lambda \mathbf{U}\mathbf{S}_2\mathbf{U}^T) = \lambda \mathbf{U}\mathbf{S}(t; \mathbf{S}_1, \mathbf{S}_2)\mathbf{U}^T \tag{22}$$

This invariance property makes the process framework robust to the choice of physical units and calibration for DTI data, a desirable property in application. It should be noted that formula (3) possesses a larger invariance group :

$$\forall \mathbf{A} \in \mathrm{Gl}(3) :$$

$$\mathbf{S}(t; \mathbf{A}\mathbf{S}_1\mathbf{A}^T, \mathbf{A}\mathbf{S}_2\mathbf{A}^T) = \mathbf{A}\mathbf{S}(t; \mathbf{S}_1, \mathbf{S}_2)\mathbf{A}^T \tag{23}$$

In fact, the interpolating curve (3) is the unique interpolation formula being invariant to an arbitrary congruence transformation, hence its name. This property can be highly desirable in other applications (see e.g. (Smith, 2005)), but it is the intrinsic cause of the anisotropy degradation observed in Table 1.

3.3 Volume and spectral interpolation

The volume of the ellipsoid represented by a positive definite matrix $\mathbf{S}$ is the product of its eigenvalues, *i.e.* its determinant. The four interpolation formulas discussed in this paper enjoy the remarkable property

$$\det(\mathbf{S}(t, \mathbf{S}_1, \mathbf{S}_2)) =$$

$$\exp((1-t)\log(\det(\mathbf{S}_1)) + t\log(\det(\mathbf{S}_2))) \tag{24}$$

In other words, the determinant of the tensor along the interpolating curve satisfies the scalar interpolation formula (2). In particular, the volume is preserved along the interpolation curve between two tensors of same volume. This property is in contrast with the 'swelling' effect observed with simpler interpolation methods and has been a main motivation to promote a Riemannian framework for DTI processing (Arsigny et al, 2006; Fillard et al, 2007).

Because the interpolation of rotations and eigenvalues is decoupled in the two spectral interpolation methods (5) and (11), they enjoy a further spectral interpolation property :

$$\lambda_i\left(\mathrm{S}(t;\mathbf{S}_1,\mathbf{S}_2)\right) = \exp((1-t)\log(\lambda_i(\mathbf{S}_1)) + t\log(\lambda_i(\mathbf{S}_2))) \quad (25)$$

This spectral interpolation is a key property for the anisotropy interpolation.

3.4 Anisotropy interpolation

Geometrically, any anisotropy scalar measure should be a scale invariant distance to identity. The Hilbert metric (Birkhoff, 1957) is precisely a projective distance that can be defined in arbitrary cones. It leads to the following definition that we refer to as Hilbert anisotropy (HA) in the sequel:

$$\mathrm{HA} = d_{\mathrm{H}}(\mathbf{S},\mathbf{I}) = \log\left(\frac{\lambda_{\max}}{\lambda_{\min}}\right) \quad (26)$$

where $\lambda_{\max}$ and $\lambda_{\min}$ are respectively the maximum and minimum eigenvalues of $\mathbf{S}$. The HA index possesses all the required properties for an anisotropy index, *i.e.*

- HA ≥ 0 and HA $= 0$ only for isotropic tensors.
- HA is invariant to rotations: $\mathrm{HA}(\mathbf{S}) = \mathrm{HA}(\mathbf{USU}^T)$ for all $\mathbf{U} \in \mathrm{O}(3)$.
- HA is invariant by scaling, $\mathrm{HA}(\mathbf{S}) = \mathrm{HA}(\alpha\mathbf{S})$, $\forall\alpha \in \mathbb{R}_+$ (it means that anisotropy only depends on the shape of the tensor and not on its size).
- HA is a dimensionless number. This property is desirable and natural, as the anisotropy of the tensor physically reflects the microscopic anisotropy of the tissues, which is independent from the diffusivity.

The spectral interpolation formula (25) leads to a particularly attractive formula for Hilbert anisotropy interpolation

$$\mathrm{HA}(\mathbf{S}(t;\mathbf{S}_1,\mathbf{S}_2)) = (1-t)\,\mathrm{HA}(\mathbf{S}_1) + t\,\mathrm{HA}(\mathbf{S}_2)\,. \quad (27)$$

In the following, we let HA_t denote $\mathrm{HA}(\mathbf{S}(t;\mathbf{S}_1,\mathbf{S}_2))$. In other words, Hilbert anisotropy is linearly interpolated along the interpolation curves (5) and (11). Other measures of anisotropy have been used in the literature : fractional anisotropy (FA), relative anisotropy (RA) (Basser and Pierpaoli, 1996), geodesic anisotropy (GA)(Fletcher and Joshi, 2007). With $\lambda_m = (\lambda_1+\lambda_2+\lambda_3)/3$ and $\lambda_{gm} = \sqrt[3]{\lambda_1\lambda_2\lambda_3}$, these indices are given respectively by

$$\mathrm{FA} = \sqrt{\frac{3}{2}}\frac{\sqrt{\sum_{i=1}^{3}(\lambda_i-\lambda_m)^2}}{\sqrt{\sum_{i=1}^{3}\lambda_i^2}}\,, \quad (28)$$

$$\mathrm{RA} = \frac{1}{\sqrt{3}}\frac{\sqrt{\sum_{i=1}^{3}(\lambda_i-\lambda_m)^2}}{\lambda_m} \quad (29)$$

$$\mathrm{GA} = \sqrt{\sum_{i=1}^{3}(\log(\lambda_i)-\log(\lambda_{gm}))^2}\,. \quad (30)$$

They do not lead to linear interpolation as the Hilbert anisotropy but they are qualitatively equivalent in that they evolve monotonically along the interpolating curve. This is in contrast with the evolution of anisotropy along the interpolating curve (3) and (4). Figure 2 illustrates how the anisotropy may degrade along such an interpolating curve. This 'anisotropy swelling effect' is undesirable for a processing framework in which anisotropy carries important information.

3.5 Singularities and uncertainty

A potential shortcoming of the proposed spectral interpolation framework is that it is only defined on the subset $\mathrm{S}_*^+(3)$ where eigenvalues are distinct. In situations of two or three similar eigenvalues, small measurements errors may lead to significant differences in the interpolating curves.
One should bear in mind that an isotropic measurement is also much less certain about orientation than an anisotropic measurement (Parker et al, 2003). In other words, eigenvalue separation of positive definite matrix correlates with the orientation information contained in the physical measurement.

In order to account for this uncertainty, we introduce the real function

$$f(x) = \frac{(\beta x)^4}{(1+(\beta x)^4)} \quad (31)$$

where β is a parameter to be defined in the sequel. The function f defines a smooth sigmoidal transition between the limits $f(0)=0$ and $f(\infty)=1$. The function of Eq. (31) will be used to rescale the interpolation between orientations of tensors in a way that mitigates the importance of the orientation information in case of low anisotropy.

Consider two tensors $\mathbf{S}_1, \mathbf{S}_2$, with $\mathbf{S}_1$ isotropic. The orientation of $\mathbf{S}_1$ (encoded by the associated quaternion

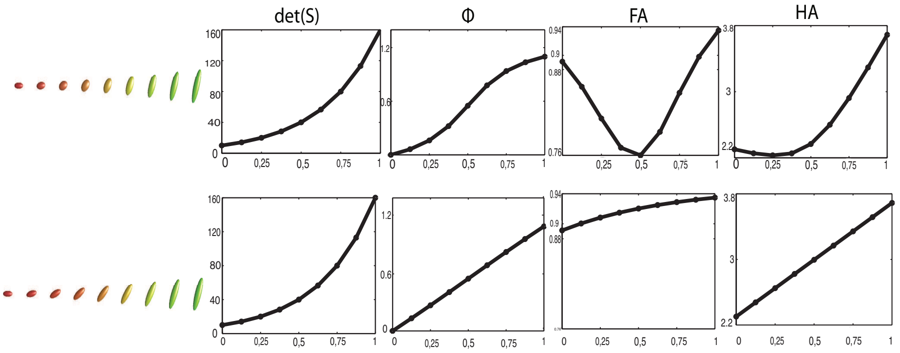


Fig. 2 Interpolating curves between two tensors. *Top*: Log-Euclidean interpolation. *Bottom*: Spectral-quaternion interpolation. The represented indices are (from left to right) the determinant, the angle between the principal eigenvectors of the first tensor and the considered one, the fractional anisotropy and the Hilbert anisotropy. Anisotropy evolves monotonically along spectral-quaternions interpolating curves, in contrast to Log-Euclidean interpolation.

$\mathbf{q}_1$) is totally arbitrary. As a result, it seems logical *not* to take into account this orientation in the interpolation between $\mathbf{S}_1$ and $\mathbf{S}_2$. This suggest to fix $\mathbf{q}_2$ along the entire interpolation curve, which means weighting $\mathbf{q}_1$ by a vanishing weight in the formula (12).

On the other hand, if we consider two very anisotropic tensors (sharing the same eigenvalues), then their orientations $\mathbf{q}_1, \mathbf{q}_2$ should be fully taken into account in the interpolation process by opting for the linear interpolation (12).

The function f can be used as a smooth transition between those two limit cases. Noticing that the case where one tensor is isotropic corresponds to

$$\min(\mathrm{HA}_1, \mathrm{HA}_2) = 0\,,$$

while the case where both are very anisotropic corresponds to $\min(\mathrm{HA}_1, \mathrm{HA}_2)$ being large, we thus let

$$\alpha(\mathrm{HA}_1, \mathrm{HA}_2) = f(\min(\mathrm{HA}_1, \mathrm{HA}_2))\,. \tag{32}$$

The linear quaternion interpolation in (12) can be replaced with

$$\mathbf{q}(t) = w_1^*(t)\mathbf{q}_1 + w_2^*(t)\mathbf{q}_2 \tag{33}$$

where

$$w_1^*(t) = (1-t)\frac{\alpha(\mathrm{HA}_1, \mathrm{HA}_t)}{\bar{\alpha}}\,,$$

$$w_2^*(t) = t\frac{\alpha(\mathrm{HA}_t, \mathrm{HA}_2)}{\bar{\alpha}}$$

and where $\bar{\alpha} = (1-t)\alpha(\mathrm{HA}_1, \mathrm{HA}_t) + t\alpha(\mathrm{HA}_t, \mathrm{HA}_2)$ ensures that $w_1^* + w_2^* = 1$. HA_t is defined in Eq.(27).

Defined this way, the interpolation is consistent with the orientation uncertainty. Indeed when one tensor, say $\mathbf{S}_1$, is isotropic we have $w_1^*(t) \equiv 0$. More generally as soon as $\mathrm{HA}_1 < \mathrm{HA}_2$ we have $w_1^* < 1-t$ as $f(\mathrm{HA}_1) < f(\mathrm{HA}_t)$, meaning that the basis $\mathbf{q}_1$ is given less weight than in the linear interpolation case (12). Finally, when $\min(\mathrm{HA}_1, \mathrm{HA}_2) \to \infty$, we recover the linear interpolation (12). Figure 3 illustrates the effect of this rescaling in the case of unequal anisotropies. For the simulations of this paper, we choose $\beta = 0.6$ so that $\alpha(\mathrm{HA}_1, \mathrm{HA}_2)$ becomes very close to 1 (more than 0.9) when both anisotropies are at least equal to 3, which means the Hilbert ratio between the largest and smallest eigenvalue is at least 20 (for example, $\lambda_1 = 10$ and $\lambda_3 = 0.5$). In this case both tensors can be considered as very anisotropic and the linear interpolation (12) is recovered. The parameter β thus appears as a tuning parameter such that $f^{-1}(0.9)/\beta$ (where f^{-1} is the inverse of f) is considered as a high Hilbert anisotropy. The parameter β can be considered as a user parameter.

3.6 Choice of two frameworks

In the rest of the paper, we focus our comparisons only between the Log-Euclidean framework and the spectral-quaternion framework. This is because Table 1 suggests that no significative differences can be observed between the Spectral interpolation and the Spectral Quaternion one. This is further illustrated in Figure 4, which shows the angular difference between the dominant eigenvectors of the tensors interpolated with either the spectral or the spectral quaternion methods. This difference is very small, as its maximum is lower than $8\,10^{-3}$. Moreover, the Spectral-Quaternion method shares the interesting invariances properties of the spectral method, as well as its 'anisotropy preserving' feature. For similar reasons, the Log-Euclidean interpolation is a good approximation of the affine-invariant one. Since

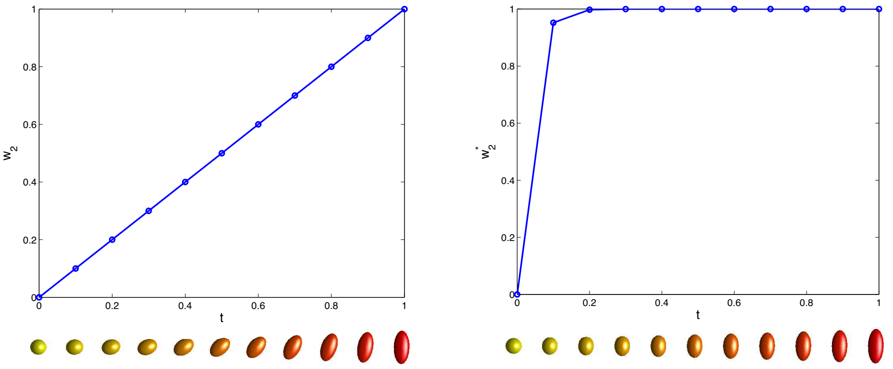


Fig. 3 Impact of the rescaling on interpolating curve. For each interpolation, the quaternion interpolation follows $\mathbf{q}(t) = w_1^*(t)\mathbf{q}_1 + w_2^*(t)\mathbf{q}_2$. *Left*: $w_1^* = 1 - t, \quad w_2^* = t$, which corresponds to the linear interpolation of Eq. (12). *right*: anisotropy is used to rescale the interpolation, and $w_2^*(t) = t^{\frac{\alpha(\mathrm{HA}_t,\mathrm{HA}_2)}{\bar{\alpha}}}$. Since the anisotropy of the first tensor is very low, the interpolated orientation is mostly based on the one of the second tensor (which is equivalent to $w_2 \approx 1$).

the Log-Euclidean and Spectral Quaternion frameworks are computationally cheaper than the affine-invariant and spectral ones, the following of the paper will focus on those two methods only.

4 Weighted means and multidimensional interpolation

4.1 Weighted means

An interpolating curve $\mathbf{S}(t; \mathbf{S}_1, \mathbf{S}_2)$ provides a straightforward definition of weighted mean from

$$\mathrm{mean}((1-t)\mathbf{S}_1, t\mathbf{S}_2) = \mathbf{S}(t; \mathbf{S}_1, \mathbf{S}_2)\,, 0 \le t \le 1\,. \qquad (34)$$

In particular, the unweighted mean obtained for $t = 1/2$ corresponds to the midpoint of the interpolation curve, a definition which agrees with the usual notion of Riemannian mean when the interpolating curve is a geodesic (see (Karcher, 1977; Fletcher and Joshi, 2007)). For the interpolating curve (3), this notion also coincides with the mean concept defined by Ando on abstract cones (Ando et al, 2004).
Extending the concept of mean to more than two positive definite matrices is less straightforward and the topic of current research (Pennec et al, 2006; Ando et al, 2004; Petz and Temesi, 2005; Moakher and Zéraï, 2011; Moakher, 2005). Mean on manifolds is usually defined by the Karcher mean, which minimizes the sum of squared distances. Usually, this definition does not enable to derive a closed-form formula for the computation of the mean of N elements, and this mean is computed through an optimization algorithm, as for example in (Pennec et al, 2006; Fletcher and Joshi, 2007). This is another attractive feature of the Log-Euclidean framework, which provides a closed-form formula for the mean of several matrices (Arsigny et al, 2006).
Here we propose a natural definition of weighted mean using the spectral interpolation curves of Section 2 : given N weights $w_1, \ldots, w_N$ that satisfy $\sum_i w_i = 1$, the weighted w-mean of N tensors $\mathbf{S}_1, \ldots, \mathbf{S}_N$ is defined by $\mathbf{S}_\mu = \mathbf{U}_\mu \Lambda_\mu \mathbf{U}_\mu^T$, where the different components are defined as follows.

- The eigenvalues of the mean tensor are defined from the scalar formula (2):
$$\lambda_{\mu,k} = \exp(\sum_{i=1}^{N} w_i \log(\lambda_{i,k})), \quad k = 1, 2, 3. \qquad (35)$$
$\Lambda_\mu = \mathrm{diag}(\lambda_{\mu,1}, \lambda_{\mu,2}, \lambda_{\mu,3})$.
- To compute the mean orientation, we first select the (weighted) most informative tensor as the reference quaternion $\mathbf{q}^r$. Namely, we choose r as
$$r = \arg\max_i w_i \mathrm{HA}_i \qquad (36)$$
The realigned quaternions are denoted by $\mathbf{q}_{i,r}$.
The mean quaternion is defined as the (chordal) mean in the quaternion space
$$\mathbf{q}_m = \sum_i w_i \mathbf{q}_{i,r} \qquad (37)$$
$$\overline{\mathbf{q}} = \frac{\mathbf{q}_m}{||\mathbf{q}_m||} \qquad (38)$$
$\mathbf{U}_\mu$ is the rotation matrix corresponding to $\overline{\mathbf{q}}$.

The chordal mean (38) of quaternions is the Riemannian mean over the set of rotations using the chordal distance on the set of quaternions(Dai et al, 2010; Sarlette et al, 2007).
The proposed weighted mean is invariant by permutation of the matrices thanks to the selection of a proper

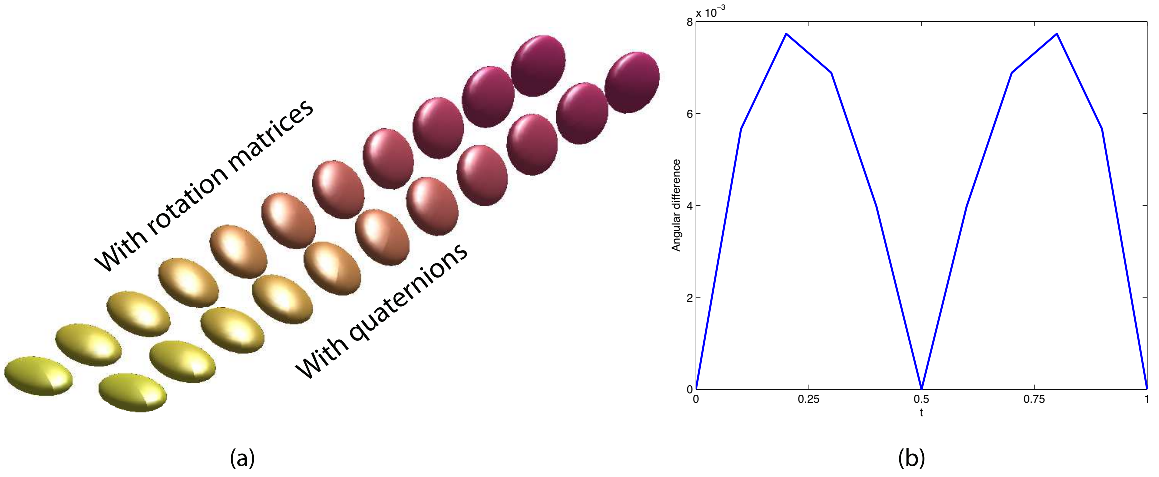


Fig. 4 The Spectral Quaternion interpolating curve is a good approximation of the Spectral one. (a) : Interpolating curves between a prolate and an oblate tensor, with the Spectral framework (top) and the Spectral Quaternion one (bottom). (b) : Angular difference between the first eigenvector of the tensors computed with the spectral method and the one of the tensor computed with the Spectral-Quaternion method. The maximal 'angular error' is lower than $8\,10^{-3}$.

reference orientation, chosen consistently with the information carried by the tensor measurements.
We also emphasize that the only singularity of the proposed mean occurs in the situation $\mathbf{q}_m = 0$, which reflects a situation in which there is so much uncertainty about the average orientation that any choice for $\overline{\mathbf{q}}$ is equally good (or bad).
It follows from the definition and from the properties of spectral-quaternion interpolation curves that the proposed mean is 'information preserving': the determinant of the mean of N tensors is the (geometric) mean of the determinant, the length of each principal axis is the geometric mean of the principal axis lengths, and the Hilbert anisotropy of the mean tensor is the arithmetic mean of the anisotropies

$$\overline{\mathrm{HA}} = \sum_{i=1}^{N} w_i \mathrm{HA}_i \,. \tag{39}$$

Finally, as explained in Section 3.5, it is possible to weight the orientations differently than the eigenvalues, in order to take into account the uncertainty of orientation for isotropic tensors. Following the developments of Section 3.5, a novel weighting of the quaternions, similar to the curve (33), is given by replacing the weights in Eq. (37) by

$$w_i^* = w_i \frac{\alpha(\mathrm{HA}_i, \overline{\mathrm{HA}})}{\bar{\alpha}} \tag{40}$$

where $\alpha(\mathrm{HA}_i, \mathrm{HA}_j)$ is given by Eq. (32) and $\bar{\alpha}$ is the normalizing factor given by $\bar{\alpha} = \sum_i w_i \alpha(\mathrm{HA}_i, \overline{\mathrm{HA}})$.

4.2 Multidimensional interpolation

As previously advocated in (Zhang et al, 2006; Kindlmann et al, 2007; Arsigny et al, 2007), an adequate interpolation method is important for the processing of diffusion tensor images and particularly for the extension of usual registration techniques (for scalar images) to the case of tensor images. This interpolation scheme is necessary to resample images. Here, we provide a direct generalization of classical interpolation method, where the interpolated value is computed as the weighted mean of the original tensors.

A common definition for multidimensional interpolation is through a weighted average of diffusion tensors (Pennec et al, 2006; Fletcher and Joshi, 2007; Arsigny et al, 2006). The weight associated to each tensor is a function of the grid distance between this tensor and the location of the interpolated tensor. In this work, if $(x_1, x_2, x_3) \in [0,1] \times [0,1] \times [0,1]$ are the coordinates of the interpolated tensor and $(\alpha_1, \alpha_2, \alpha_3) \in \{0,1\} \times \{0,1\} \times \{0,1\}$ the coordinates of the point α of the grid, the following function will be used

$$w_\alpha(x_1, x_2, x_3) = \prod_{i=1}^{3} (1 - \alpha_i + (-1)^{1-\alpha_i} x_i). \tag{41}$$

Figure 2 shows the curve interpolation between two tensors using both the Log-Euclidean and the spectral quaternions frameworks. As in (Zhou, 2010), the variation of the main information conveyed by the tensors is also shown. As previously shown, the Hilbert anisotropy is linearly interpolated by the novel framework, while this information is significantly degraded in the Log-Euclidean framework. A similar behavior is found for the evolution of the fractional anisotropy. Both methods geometrically interpolate the determinant. It is also interesting to analyse the difference in ϕ, the angle between the first eigenvector of the first tensor and the first eigenvector of the weighted mean. The spectral measure produces a quasi linear interpolation of this angle, in contrast to the Log-Euclidean framework.

Using the method described above for computing the weighted means of many tensors, the interpolation of four tensors at the corners of a grid can be computed, as illustrated in Figures 5 and 6. For those figures, the alternative weights proposed in (40) are used. In the left columns of those figures, colors of the tensors are determined by HA. In their middle columns, the evolution of the anisotropy in the grid is represented as a contour map. Each curve in these subfigures represent a constant level of anisotropy. In Figure 5, which performs the weighted mean of 3 very anisotropic (prolate) tensors and an isotropic one, the difference is clearly visible between the two interpolation schemes. The monotone evolution of anisotropy achieved with the spectral interpolation is strongly distorted with the Log-Euclidean interpolation. Figure 6 uses an isotropic tensor (bottom left), a prolate tensor (bottom right) and two oblate tensors (for which $\lambda_1 \approx \lambda_2 > \lambda_3$). In this case, anisotropy varies in a same way using any of the two methods. Finally, the direction of the principal eigenvector of tensors is also shown (right columns). Some differences can be observed, for example on the first row of the subfigures in Figure 5, which corresponds to an interpolation between two orthogonal tensors with the same shape.With the Log-Euclidean method, the principal eigenvector does not vary until the midpoint of the interpolation, where it is rotated by $\pi/2$ rad. On the contrary, the spectral quaternion interpolation results in a monotone rotation of the tensor. In Figure 6, it is interesting to note the difference for the orientation interpolation between the prolate tensor and the oblate one (last columns of the subfigures). Since the principal direction of diffusion (first eigenvector) of the prolate tensor is in the diffusion plan of the oblate tensor, it is not necessary to vary the orientation of the prolate tensor. This is indeed the case when using the Log-Euclidean method, but not with the Spectral-Quaternion framework, which performs a smooth variation of this orientation. This disadvantage of our method could be avoided by using the whole rotation matrix to represent eigenvectors. This could help to weight differently the orientation of a prolate tensor if its principal direction is included in the diffusion plane of an oblate tensor. However, this would imply to weight the orientation interpolation by a factor depending upon the full orientation, which is not permitted by the quaternion representation.

In order to further compare the two frameworks, we implemented some synthetic processing examples which illustrate how approximation errors could propagate in the processing pipeline. The simulations are chosen as in (Thévenaz et al, 2000). The protocol is the following: a succession of $r = 15$ rotations of $2\pi/15 = 24°$ each was applied to some image, such that the output of any given step r_i is the input for the next step r_{i+1}. The final output is then compared to the initial image, see Figure 7. To enable a better interpretation of the results, the maps of squared differences between the properties of the original image and the ones of the output are also shown. The errors in FA are localized, and are bigger in the case of the Log-Euclidean interpolation. There are more errors for the direction of the principal eigenvector, but it should be noted that the largest errors are localized in low anisotropy areas. The uncertainty of orientation is high in these areas, which accounts for large orientation errors.

5 Distance, Riemannian metric and similarity measure

Interpolation curves provide a natural measure of similarity (or closeness) as the length of the interpolating curve in a suitable metric space.

For the scalar interpolation formula (2), the definition

$$d(s_1, s_2) = \int_0^1 \sqrt{g_{s(t)}(\dot{s}(t), \dot{s}(t))} dt = |\log \left(\frac{s_1}{s_2}\right)| \qquad (42)$$

where $g_{s(t)}$ is the chosen scalar product at $s(t)$, coincides with the (scale-invariant) geometric distance between two positive numbers.

Likewise, for the four interpolating formulas considered in this paper, the definition

$$d(\mathbf{S}_1, \mathbf{S}_2) = \int_0^1 \sqrt{g_{\mathbf{S}(t)}(\dot{\mathbf{S}}(t), \dot{\mathbf{S}}(t))}\, dt \qquad (43)$$

qualifies as valid distance on $\mathrm{S}^+(3)$.

For the affine-invariant curve, this definition coincides with the Riemannian distance associated with the affine-invariant metric. It is globally defined and makes $\mathrm{S}^+(3)$ a complete metric space (Fletcher and Joshi, 2007). It has a closed form expression

$$d(\mathbf{S}_1, \mathbf{S}_2) = ||\log(\mathbf{S}_1^{-1/2}\mathbf{S}_2\mathbf{S}_1^{-1/2})||_2 \qquad (44)$$

which is a natural matrix analog of the scalar formula.

For the Log-Euclidean geometry, the distance has the closed form

$$d(\mathbf{S}_1, \mathbf{S}_2) = ||\log(\mathbf{S}_1) - \log(\mathbf{S}_2)||_2 \qquad (45)$$

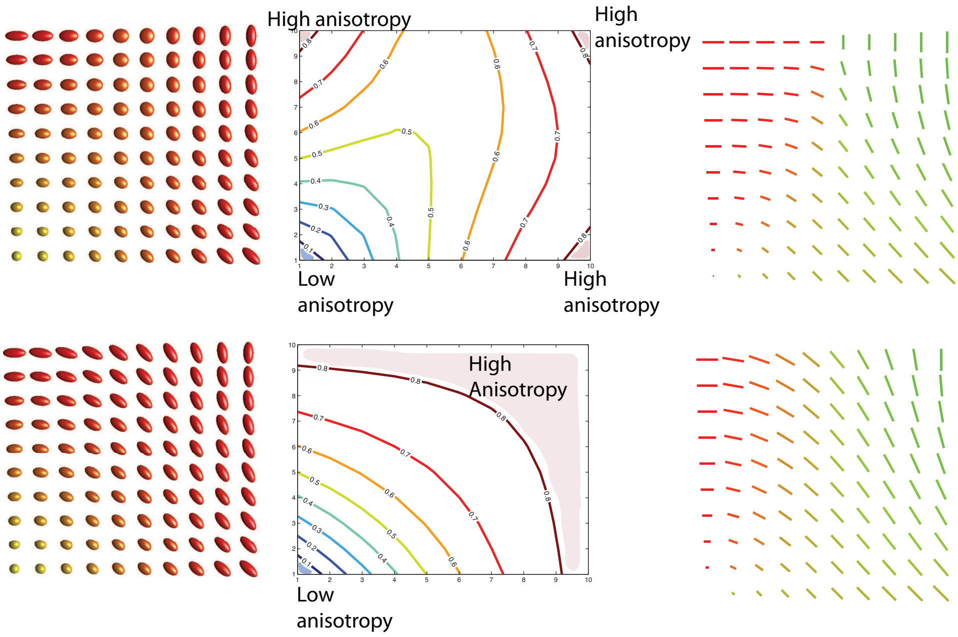


Fig. 5 Multi-dimensional interpolation between four tensors at the corners of the grids. *Top*: Log-Euclidean interpolation. *Bottom*: Spectral interpolation. Colors of the ellipsoids indicates their anisotropy from yellow (low anisotropy) to red for higher anisotropies. The subfigures in the middle represent a contour map of the Hilbert Anisotropy. Areas of constant anisotropy are represented by color lines. In the right, the orientation of the first eigenvector of tensors is represented. The lengths of the lines are proportional to the fractional anisotropy and the colors depend on the direction.

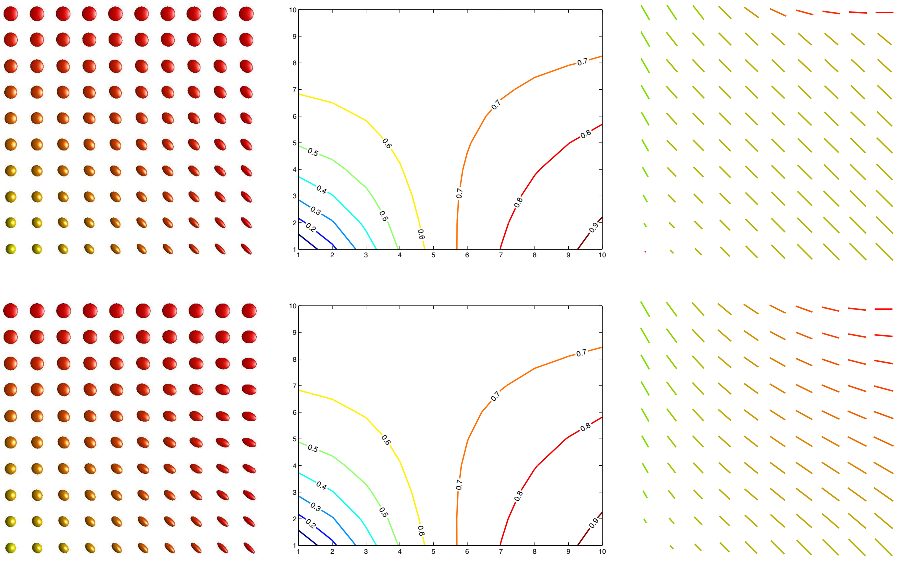


Fig. 6 Multi-dimensional interpolation between four tensors at the corners of the grids. *Top*: Log-Euclidean interpolation. *Bottom*: Spectral interpolation. Colors of the ellipsoids indicates their anisotropy from yellow (low anisotropy) to red for higher anisotropies. The subfigures in the middle represent a contour map of the Hilbert Anisotropy. Areas of constant anisotropy are represented by color lines. In the right, the orientation of the first eigenvector of tensors is represented. The lengths of the lines are proportional to the fractional anisotropy and the colors depend on the direction.

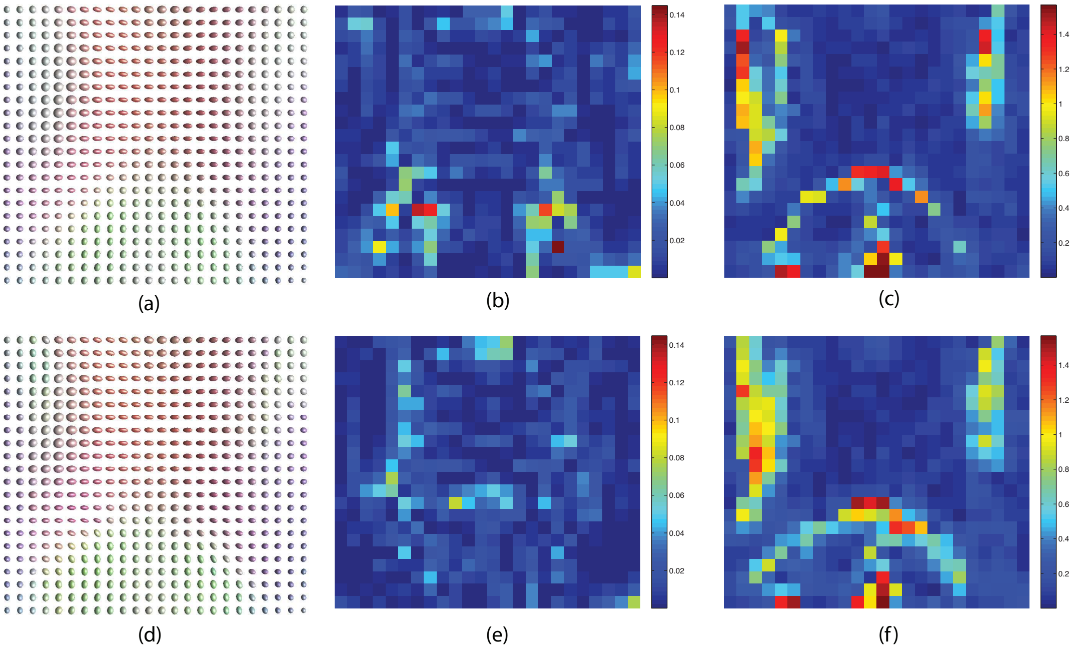


Fig. 7 Results of the successive rotations of $2\pi/15$ applied to a real image. *Top :* Means were computed with the Log-Euclidean framework. (a): Output of the protocol, (b) Map of the squared differences in FA between the original image and the output. (c) Map of the squared differences in the direction of principal diffusion between the original image and the output. *Bottom :* Idem to top with the Spectral Quaternion framework. It can be observed that the (localized) errors in FA are higher with the Log-Euclidean framework (color scales are identical). In both cases, the errors in orientation are concentrated in low anisotropy areas.

which is a chordal distance in the embedding space. It is simpler to calculate and a good approximation of (44) close to the identity matrix.

For the spectral geometry, the distance on the Lie group $SO(3)\times D^+(3)$ could be taken as given by formula (43) where the metric is defined by

$$g_{\mathbf{S}(t)}(\dot{\mathbf{S}}(t), \dot{\mathbf{S}}(t)) =$$

$$k^2(\mathbf{S}(t))\; g^{SO(3)}_{\mathbf{U}(t)}(\dot{\mathbf{U}}(t), \dot{\mathbf{U}(t)}) + \sum\Big(\frac{\dot{\Lambda}_i(\mathbf{S}(t))}{\Lambda_i(\mathbf{S}(t))}\Big)^2 \qquad (46)$$

which is a weighted sum of the (infinitesimal) Riemannian distances on $SO(3)$ and $D^+(3)$, and k is a weighting factor. The Riemannian distance on $SO(3)$ is not global because of cut-locus singularities. We propose to define the spectral-quaternion geometry replacing the geodesic distance on $SO(3)$ by a chordal distance in the space of quaternion, leading to the metric

$$g_{\mathbf{S}(t)}(\dot{\mathbf{S}}(t), \dot{\mathbf{S}}(t)) = k^2(\mathbf{S}(t))\; ||\dot{\mathbf{q}}(t)||^2 + \sum\Big(\frac{\dot{\Lambda}_i(\mathbf{S}(t))}{\Lambda_i(\mathbf{S}(t))}\Big)^2 . (47)$$

Building upon the interpolating curves (33), we propose to let the weighting factor in (47) be

$$k(\mathbf{S}(t)) = f(\mathrm{HA}(t)) \,, \qquad (48)$$

where f is defined in Eq. (31). Thus defined, it appears as a sigmoid that discards the (infinitesimal) distance with the orientation term when the considered tensor is isotropic (since this term contains no information), and fully accounts for orientation in the case of large anisotropy (that is, k is close to 1 when the ratio between the largest and smallest eigenvalue becomes larger than 20).

It is hopeless to compute in explicit form the geodesic distance d associated to the metric (47). However, the geodesic distance satisfies

$$\sum |\log \frac{\lambda_i(\mathbf{S}_1)}{\lambda_i(\mathbf{S}_2)}| \leq d(\mathbf{S}_1, \mathbf{S}_2) \leq$$

$$\alpha(\mathbf{S}_1, \mathbf{S}_2) ||\mathbf{q}_1 - \mathbf{q}_2^r||_2 + \sum |\log \frac{\lambda_i(\mathbf{S}_1)}{\lambda_i(\mathbf{S}_2)}| := \varDelta(\mathbf{S}_1, \mathbf{S}_2) (49)$$

with

$$\alpha(\mathbf{S}_1, \mathbf{S}_2) = f(\min(\mathrm{HA}_1, \mathrm{HA}_2)) \,.$$

The lower bound is obvious since $k(\mathbf{S}(t)) \geq 0$ along the path; the upper bound is the (Riemannian) length in the sense of metric (47) of a curve originating at the tensor of minimal isotropy and made of two arcs: the first one linking the orientations at constant (minimal) anisotropy and the second one linking the eigenvalues while the orientation remains fixed. The geodesic distance being by definition smaller than the length of any path, the upper bound is obtained.

In the limit case of an isotropic tensor, the inequality above becomes an equality, and when it is close to identity, $\Delta(\mathbf{S}_1, \mathbf{S}_2)$ is thus a good approximation of the geodesic distance. It is meaningful, as the weighting factor goes to zero when the orientation information becomes uncertain. This measure captures the fact that the cost of rotation is weighted with the orientation information. We propose formula (49) as a convenient substitute for the geodesic distance associated to the Riemannian metric (46-48). This similarity measure is not a distance because it does not satisfy the triangular inequality. However, the bounds in (49) show that it approximates the distance when the anisotropy is small and weights the distance between quaternion only when the anisotropy is high, that is, when the orientation of tensors carries physical information.

It should be noted that the proposed interpolating curve (33), metric (47), and similarity measure (49) are not very sensitive to the form of the sigmoid f, nor to the parameter β. The rationale for the term $k(\mathbf{S}(t))$ is simply to let the orientation quaternion $\mathbf{q}$ play a role only when the anisotropy exceeds a certain threshold.

We close this section with a conceptual remark for the interested reader. The interpolating curve (33), metric (47), and similarity measure (49) approximate a Riemmanian framework in $\mathrm{S}^+(3)$ by properly combining a Riemmanian framework in $\mathrm{D}^+(3)$ and in $\mathrm{SO}(3)$. This means that distances between ellipsoids are approximated by the weighted sum of distances between rotations and distances between dilations. This idea is in close analogy with the framework recently developed by the authors (Bonnabel and Sepulchre, 2009; Bonnabel et al, 2013) to define distances between flat ellipsoids, that is, positive semidefinite tensors of fixed rank: the proposal being to approximate the distance between the flat ellipsoids by the weighted sum of distances between the subspaces that support them and distances between full ellipsoids in a common subspace. The Riemmanian framework proposed for flat ellipsoids is probably not directly relevant for the DTI application but it is of interest to notice that different applications motivate a separate weighting of rotations and dilations, leading to similar theoretical and computational issues about reconciling the geometry of the product space to the geometry of the original space.

Algorithm 1 : Computation of interpolation

```
Inputs : two tensors $\mathbf{S}_1$ and $\mathbf{S}_2$, a parameter $t$
Output : the interpolated tensor for the value $t$, $\mathbf{S}_t$

1. $\mathbf{S}_1 = \mathbf{U}_1 \Lambda_1 \mathbf{U}_1^T$, $\mathbf{S}_2 = \mathbf{U}_2 \Lambda_2 \mathbf{U}_2^T$, with the eigenvalues
   in decreasing order.
2. Compute the quaternions representing the orientations,
   $\mathbf{q}_1$ and $\mathbf{q}_2$, and the associated set $\mathcal{Q}_2$. This set contains
   the four elements
   $$\left\{ \begin{pmatrix} \mathbf{q}_2(1) \\ \mathbf{q}_2(2) \\ \mathbf{q}_2(3) \\ \mathbf{q}_2(4) \end{pmatrix} \begin{pmatrix} \mathbf{q}_2(2) \\ -\mathbf{q}_2(1) \\ -\mathbf{q}_2(4) \\ \mathbf{q}_2(3) \end{pmatrix} \begin{pmatrix} \mathbf{q}_2(3) \\ \mathbf{q}_2(4) \\ -\mathbf{q}_2(1) \\ -\mathbf{q}_2(2) \end{pmatrix} \begin{pmatrix} \mathbf{q}_2(4) \\ -\mathbf{q}_2(3) \\ \mathbf{q}_2(2) \\ -\mathbf{q}_2(1) \end{pmatrix} \right\}$$
   and their opposites.
3. Compute the interpolated eigenvalues matrix, using
   $\lambda_i(t) = \exp((1-t)\log(\lambda_{i,1}) + t\log(\lambda_{i,2}))$.
4. Select the 'realigned' quaternion $\mathbf{q}_2^a$ with respect to $\mathbf{q}_1$
   using $\mathbf{q}_2^a = \arg\max_{\mathbf{q}_2 \in \mathcal{Q}_2} \mathbf{q}_1^T . \mathbf{q}_2$
5. Compute the interpolated quaternion as
   $\mathbf{q}_m(t) = (1-t)\mathbf{q}_1 + t\mathbf{q}_2^a$
   $\mathbf{q}(t) = \mathbf{q}_m(t)/||\mathbf{q}_m(t)||$
6. Compute $\mathbf{U}(t)$ as the rotation matrix corresponding to
   $\mathbf{q}(t)$.
7. The interpolated tensor is given by
   $\mathbf{S}(t) = \mathbf{U}(t)\Lambda(t)\mathbf{U}(t)^T$ .
```

Algorithm 2 : Computation of weighted mean

```
Inputs : A set of tensors $\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_N$ and their associate
weights $w_i, i = 1, \ldots, N$.
Output : the mean tensor $\mathbf{S}_\mu$

1. Perform the spectral decomposition of each of the $N$
   tensors.
2. Compute the weighted mean of eigenvalues, through
   $\lambda_{\mu,k} = \exp(\sum_{i=1}^{N} w_i \log(\lambda_{i,k})), \quad k = 1, 2, 3.$
3. Select as the reference tensor the one that maximizes
   the product $w_i \mathrm{HA}_i$. The reference quaternion is $\mathbf{q}^r$.
4. For each tensor, select the 'realigned' quaternion $\mathbf{q}_i^a$
   with respect to $\mathbf{q}^r$ using $\mathbf{q}_i^a = \arg\max_{\mathbf{q}_i \in \mathcal{Q}_i} \mathbf{q}^r . \mathbf{q}_i$
5. The weighted mean of quaternions is
   $\mathbf{q}_m = \sum_i w_i \mathbf{q}_i^a$
   $\mathbf{q}_\mu = \frac{\mathbf{q}_m}{||\mathbf{q}_m||}$
6. Compute $\mathbf{U}_\mu$ from $\mathbf{q}_\mu$ according to (13).
7. The mean tensor is $\mathbf{S}_\mu = \mathbf{U}_\mu \Lambda_\mu \mathbf{U}_\mu^T$.
```

Algorithm 3 : Computation of similarity measure

```
Inputs : Two tensors $\mathbf{S}_1$ and $\mathbf{S}_2$.
Output : the measure of their similarity $\Delta(\mathbf{S}_1, \mathbf{S}_2)$

1. Perform the spectral decomposition of each tensor, and
   compute the quaternions corresponding to the orientation matrices.
2. The distance between eigenvalues is given by
   $d_\Lambda = \sum |\log \frac{\lambda_i(\mathbf{S}_1)}{\lambda_i(\mathbf{S}_2)}|$
3. The weighting factor is given by
   $\alpha(\mathbf{S}_1, \mathbf{S}_2) = \frac{(\beta \min(\mathrm{HA}_1, \mathrm{HA}_2))^4}{1+(\beta \min(\mathrm{HA}_1, \mathrm{HA}_2))^4}$, with $\beta = 0.6$.
4. Compute the chordal distance between quaternions, using one of the quaternion
   as the reference and realigning the other one to this reference, i.e. if $\mathbf{q}_1$
   is the reference, select the 'realigned' quaternion $\mathbf{q}_2^a$ with respect to $\mathbf{q}_1$
   using $\mathbf{q}_2^a = \arg\max_{\mathbf{q}_2 \in \mathcal{Q}_2} \mathbf{q}_1.\mathbf{q}_2$. The distance
   $d_\mathbf{q} = ||\mathbf{q}_1 - \mathbf{q}_2^a||_2$
5. $\Delta(\mathbf{S}_1, \mathbf{S}_2) = \alpha d_\mathbf{q} + d_\Lambda$.
```

6 Filtering and denoising

In this section, we will illustrate how the framework introduced in this paper affects the results of some important processing methods. All these methods are based on the principal notion of weighted means. We focus on filtering methods, however means appear in many other applications, such as in transformation of images, registration and template construction. It should be noted that in those cases, the concept of the reorientation of the tensors (Alexander et al, 2001) is crucial. Many image processing tasks imply the convolution of the image with some kernels. This is the case of denoising, smoothing, edge detection, image sharpening and filtering. The convolution of diffusion tensor images is straightforward form the definition of means.

Gaussian filtering

In a continuous setting, the convolution of a vector field $F_0(x)$ by a Gaussian G_σ is given by

$$F(x) = \int_y G_\sigma(y-x) F_0(y) dy \,.$$

In the case of images (*i.e* pixels or voxels on a discrete grid), the coefficients are renormalized since the neighborhood $\mathcal{V}$ is usually limited to points within one to three times the standard deviation σ. The discretization is given by

$$F(x) = \frac{\sum_{u \in \mathcal{V}(x)} G_\sigma(u) F_0(x+u)}{\sum_{u \in \mathcal{V}(x)} G_\sigma(u)} \,, \tag{50}$$

which is nothing else than a weighted averaging, where the weights are given by the coefficients G_σ. We have applied a Gaussian filter to a true image (Figure 8,(a)), corrupted by additive noise in Figure 8, (b). The size of the Gaussian filter was 7 × 7, and $\sigma = 3$ pixels. The Log-Euclidean filtering (Figure 8, (c)) is compared to a convolution based on the spectral-quaternions framework (Figure 8, (d)). It can be observed that both methods produce smoothened images. The corresponding FA maps show that tensors which are highly anisotropic in the true image become much more isotropic with the Log-Euclidean filtering (Figure 8, (g)). This expected degradation of anisotropy is less important with the spectral-quaternion framework (Figure 8, (h)).

Anisotropic filtering

Anisotropic regularization of images corrupted by noise is very important in medical image processing, since it allows for a reduction of the noise level while preserving boundaries and structures. The main idea of anisotropic filtering is to 'penalize' the smoothing in the directions where the derivative is important (Perona and Malik, 1990). As proposed in (Pennec et al, 2006), if $c(\cdot)$ is a weighting function decreasing from $c(0) = 1$ to $c(\infty) = 0$, anisotropic filtering can be realized using a discrete implementation of the Laplacian operator. The contribution $\Delta_u \mathbf{S}$ of the spatial direction u to the Laplace-Beltrami operator is weighted by the decreasing function, according to the norm of the gradient in that direction. The norm of the gradient should of course be evaluated according to the corresponding metric or measure. Finite difference approximations of the derivatives are explained in (Pennec et al, 2006) for the affine-invariant metric. In the case of Log-Euclidean metric, the schemes are simpler since the dependence of the gradient on the current tensor disappears on the logarithms of tensors. The norm of gradient is then easily computed, as explained in (Arsigny et al, 2006). A similar comment can be made about spectral quaternions method : all the computations can be done in an Euclidean way by using the quaternions and the logarithms of eigenvalues. Not entering into the details, the numerical implementations of an anisotropic filtering with the spectral-quaternions method is based on the following equations, where $\mathcal{V}$ represents the 4 used directions used for the computations. For each pixel at position x and at each time step, both the orientation and the eigenvalues have to be computed.

$$\mathbf{q}_{\mathrm{temp}}(x) = \mathbf{q}_n(x) + 2\varepsilon_q \Delta\mathbf{q}(x) \tag{51}$$

$$\mathbf{q}_{n+1}(x) = \frac{\mathbf{q}_{\mathrm{temp}}(x)}{||\mathbf{q}_{\mathrm{temp}}(x)||} \tag{52}$$

where

$$\Delta\mathbf{q}(x) = \sum_{u \in \mathcal{V}} c(\sqrt{\alpha}||\mathbf{q}_n^*(x+u) - \mathbf{q}_n(x)||) \Delta_u \mathbf{q}(x) \tag{53}$$

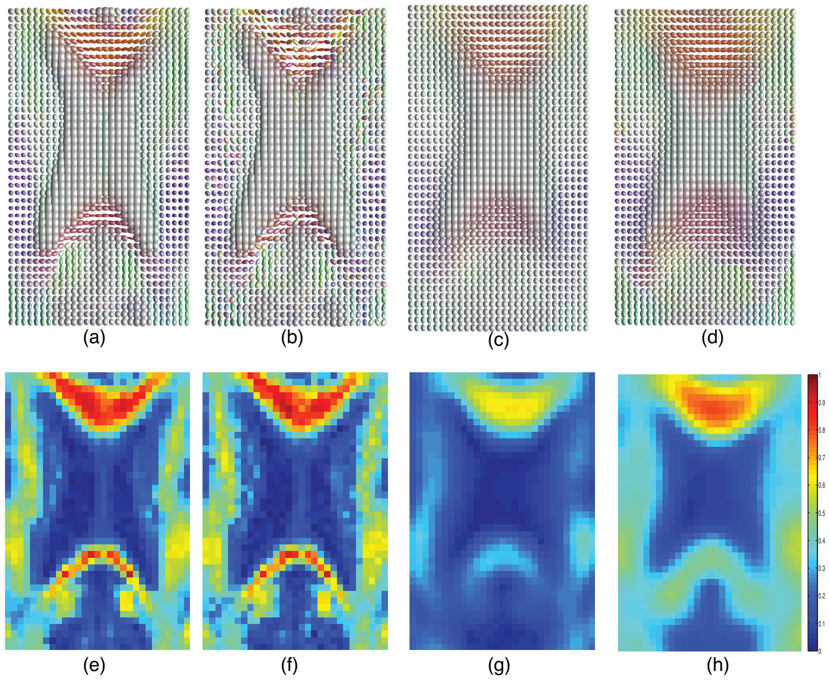


Fig. 8 Gaussian filtering of a true image. *Top:* Slices of diffusion tensor images, *bottom :* corresponding FA maps. (a): Slice of an image constructed from real data,(b): noisy image, (c): result of the Gaussian filtering with the Log-Euclidean method, (d): Gaussian filtering with the spectral-quaternion framework. For both methods, boundaries are less visible after the filtering. It can be observed that tensors which are highly anisotropic in the true image are more isotropic with the Log-Euclidean framework. This degradation is less important with the spectral-quaternions method.

with $\mathbf{q}_n^*(x+u)$ the realigned quaternion with respect to $\mathbf{q}_n(x)$ and $\Delta_u\mathbf{q}(x) = \mathbf{q}_n^*(x+u) - \mathbf{q}_n(x)$. The parameter α in the argument of function $c(\cdot)$ is the one used in the similarity measure.

The logarithms of eigenvalues are used to compute the filtering.

$$\Lambda_{n+1}(x) = \exp(\Lambda_n(x) + 2\varepsilon_\Lambda \Delta\Lambda(x)) \tag{54}$$

where

$$\Delta\Lambda(x) = \sum_{u\in\mathcal{V}} c(||\Delta_u\Lambda(x)||)\Delta_u\Lambda(x) \tag{55}$$

with $\Delta_u\Lambda(x) = \log(\Lambda_n(x+u)) - \log(\Lambda_n(x))$. The function $c(\cdot)$ used in both cases is $c(t) = \exp(-t^2/\sigma^2)$, with different σ for the orientation and the eigenvalues ($\sigma_q = 0.1$ and $\sigma_\Lambda = 1$). The time steps used are also different, with $\varepsilon_q = 0.1$ and $\varepsilon_\Lambda = 0.001$ in our example. This algorithm has been applied to the same noisy image that had been used for the Gaussian filtering, Figure 8, and the results are shown in Figure 9. In this figure it can be observed that both methods perform well. Compared to the results of the Gaussian filtering, the structures are better preserved. Some outliers are not regularized in both figures. Moreover, the results suggest that the spectral quaternions method produces better orientation results, for highly anisotropic tensors. In the subfigures showing the angular difference between the principal eigenvectors, low errors areas are observable for the spectral quaternion framework. Those structures correspond to high anisotropy areas.

7 Conclusion

In this paper, we have introduced a novel geometric framework for the Diffusion Tensor Images processing. This framework is based on interpolating curves that interpolate intensities and orientation separately. The main advantage of this method is to control the spectral interpolation of tensors during the processing. Moreover, it possesses all the important properties of existing metrics, such as the invariances and the preservation of other information as the determinant and the orientation of tensors.

Computational obstacles previously described in (Tschumperlé and Deriche, 2001; Chefd'hotel et al, 2004) are circumvented by embedding the set of rotation matrices in the space of quaternions, long used for its numerical efficiency in robotics.

The resulting interpolation method retains the computational tractability and the geometry of the Log-

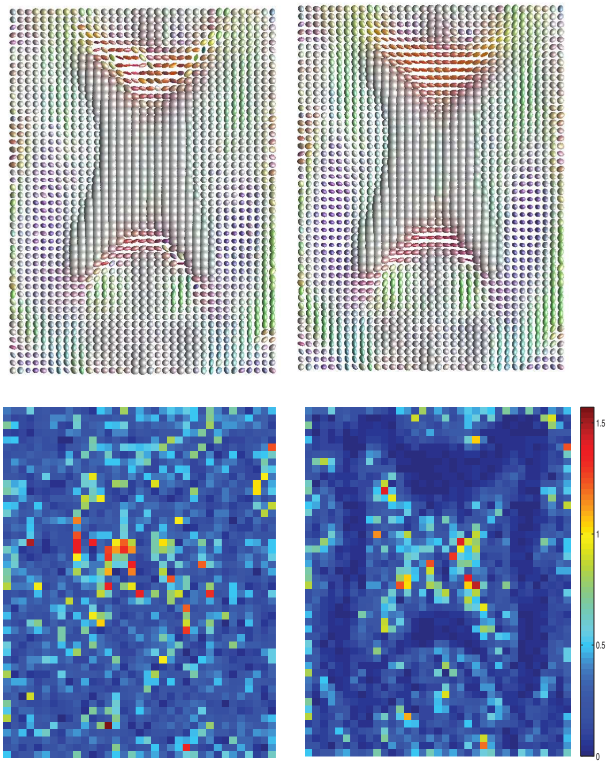


Fig. 9 Anisotropic filtering of a true image. *Top left*: Log-Euclidean filtering. *Top right*: Filtering based on the spectral quaternions framework. *Bottom*: angular difference between the first eigenvector of the filtered tensor and the one of the 'true' tensor. It can be observed that some 'very noisy' tensors are not well regularized. Compared to the Gaussian filtering, the structures are well preserved. Moreover, the orientation of highly anisotropic tensors is better regularized with the spectral quaternion framework, as can be observed from the visible 'low errors areas'.

Euclidean framework but addresses a limitation of this framework regarding the degradation of anisotropy.
Although several illustrations of the paper exemplify the potential benefit of preserving anisotropy through averaging and interpolation operation encountered in statistical process, the benefits of the proposed framework for clinical applications remain to be demonstrated. Registration and tractography are two particular areas where the advantages of the proposed method should be evaluated quantitatively.
Future research will extend the proposed framework to more complex models of diffusion, such as HARDI. In particular, Riemannian methods used to process those models suffer the same drawback of degradation of the anisotropy (Cheng et al, 2009; Goh et al, 2011). It is interesting to note that recent contributions have advocated a decoupling between orientation and other parts of the diffusion model to correct those artifacts (Cetingül et al, 2012; Ncube and Srivastava, 2011).

References

Alexander, D., Gee, J. (2000). Elastic matching of Diffusion Tensor Images. *Computer Vision and Image Understanding*, 77(2), 233 – 250.

Alexander, D., C., P., Basser, P., Gee, J. (2001). Spatial transformations of diffusion tensor magnetic resonance images. *IEEE Trans Med Imaging*, 20(11), 1131–1139.

Ando, T., Li, C. K., Mathias, R. (2004). Geometric means. *Linear Algebra and its Applications*, 385(0), 305 – 334.

Arsigny, V., Fillard, P., Pennec, X., Ayache, N. (2006). Log-Euclidean metrics for fast and simple calculs on diffusion tensors. *Magnetic Resonance in Medicine*, 56, 411–421.

Arsigny, V., Fillard, P., Pennec, X., Ayache, N. (2007). Geometric Means in a Novel Vector Space Structure on Symmetric Positive-Definite Matrices. *SIAM Journal on Matrix Analysis and Applications*, 29(1), 328–347.

Awate, S., Zhang, H., Gee, J. (2007). Fuzzy Nonparametric DTI Segmentation for Robust Cingulum-Tract Extraction. In: *MICCAI* (pp. 294–301).

Basser, P., Pierpaoli, C. (1996). Microstructural and Physiological Features of Tissues Elucidated by Quantitative-Diffusion-Tensor MRI. *Journal of Magnetic Resonance, Series B*, 111(3), 209 – 219.

Basser, P., Mattiello, J., Bihan, D. L. (1994). MR diffusion tensor spectroscopy and imaging. *Biophysical Journal*, 66(1), 259 – 267.

Batchelor, P. G., Moakher, M., Atkinson, D., Calamante, F., Connelly, A. (2005). A rigorous framework for diffusion tensor calculus. *Magnetic Resonance in Medicine*, 53(1), 221–225.

Birkhoff, G. (1957). Extensions of jentzsch's theorem. *Transactions of the American Mathematical Society*, 85(1), 219–227.

Bonnabel, S., Sepulchre, R. (2009). Riemannian metric and geometric mean for positive semidefinite matrices of fixed rank. *SIAM Journal on Matrix Analysis and Applications*, 31(3), 1055–1070.

Bonnabel, S., Collard, A., Sepulchre, R. (2013). Rank-preserving geometric means of positive semi-definite matrices. *Linear Algebra and its Applications*, 438(8), 3202–3216.

Burbea, J., Rao, C. (1982). Entropy differential metric, distance and divergence measures in probability spaces: A unified approach. *Journal of Multivariate Analysis*, 12(4), 575 – 596.

Castaño-Moraga, C., Lenglet, C., Deriche, R., Ruiz-Alzola, J. (2006). A fast and rigorous anisotropic smoothing method for DT-MRI. In: *IEEE Inter-*

national Symposium on Biomedical Imaging: From Nano to Macro (pp. 93–96).

Castro, F., Clatz, O., Dauguet, J., Archip, N., Thiran, J. P., Warfield, S. (2007). Evaluation of brain image nonrigid registration algorithms based on Log-Euclidean MR-DTI consistency measures. In: *IEEE International Symposium on Biomedical Imaging: From Nano to Macro* (pp. 45–48).

Cetingül, H. E., Afsari, B., Wright, M. J., Thompson, P. M., Vidal, R. (2012). Group action induced averaging for HARDI processing. In: *Proc IEEE Int Symp Biomed Imaging* (pp. 1389–1392).

Chefd'hotel, C., Tschumperlé, D., Deriche, R., Faugeras, O. (2004). Regularizing flows for constrained matrix-valued images. *Journal of Mathematical Imaging and Vision*, 20, 147–162.

Cheng, J., Ghosh, A., Jiang, T., Deriche, R. (2009). A Riemannian framework for orientation distribution function computing. In: *MICCAI* (pp. 911–8), vol 12.

Chiang, M. C., Leow, A. D., Klunder, A. D., Dutton, R. A., Barysheva, M., Rose, S., McMahon, K., de Zubicaray, G., Toga, A., Thompson, P. M. (2008). Fluid registration of Diffusion Tensor Images using information theory. *IEEE Transactions on Medical Imaging*, 27.

Dai, Y., Trumpf, J., Li, H., Barnes, N., Hartley, R. (2010). Rotation Averaging with Application to Camera-Rig Calibration. In: *Computer Vision – ACCV 2009* (pp. 335–346).

Dryden, I. L., Koloydenko, A., Zhou, D. (2009). Non-Euclidean statistics for covariance matrices, with applications to diffusion tensor imaging. *The Annals of Applied Statistics*, 3(3), 1102–1123.

Fillard, P., Pennec, X., Arsigny, V., Ayache, N. (2007). Clinical DT-MRI estimation, smoothing and fiber tracking with Log-Euclidean metrics. *IEEE Transactions on Medical Imaging*, 26.

Fletcher, P. T., Joshi, S. (2007). Riemannian Geometry for the Statistical Analysis of Diffusion Tensor Data. *Signal Processing*, 87, 250–262.

Fletcher, P. T., Venkatasubramanian, S., Joshi, S. (2009). The geometric median on Riemannian manifolds with application to robust atlas estimation. *NeuroImage*, 45(1, Supplement 1), S143 – S152.

Goh, A., Lenglet, C., Thompson, P. M., Vidal, R. (2011). A nonparametric Riemannian framework for processing high angular resolution diffusion images and its applications to ODF-based morphometry. *Neuroimage*, 56(3), 1181–201.

Goodlett, C., Fletcher, P. T., Gilmore, J. H., Gerig, G. (2009). Group analysis of DTI fiber tract statistics with application to neurodevelopment. *NeuroImage*, 45(1, Supplement 1), S133 – S142.

Gur, Y., Sochen, N. (2007). Fast invariant Riemannian DT-MRI regularization. In: *International Conference on Computer Vision* (pp. 1–7).

Ingalhalikar, M., Yang, J., Davatzikos, C., Verma, R. (2010). DTI-DROID: Diffusion tensor imaging-deformable registration using orientation and intensity descriptors. *International Journal of Imaging Systems and Technology*, 20(2), 99–107.

Jensen, J. H., Helpern, J. A. (2010). MRI quantification of non-Gaussian water diffusion by kurtosis analysis. *NMR in Biomedicine*, 23(7), 698–710.

Karcher, H. (1977). Riemannian center of mass and mollifier smoothing. *Communications on Pure and Applied Mathematics*, 30(5), 509–541.

Kindlmann, G., San Jose Estepar, R., Niethammer, M., Haker, S., Westin, C. F. (2007). Geodesic-loxodromes for diffusion tensor interpolation and difference measurement. In: *MICCAI* (pp. 1–9).

Lenglet, C., Rousson, M., Deriche, R., Faugeras, O. (2006). Statistics on the manifold of multivariate normal distributions: theory and application to diffusion tensor MRI processing. *Journal of Mathematical Imaging and Vision*, 25, 423–444.

Lenglet, C., Campbell, J., Descoteaux, M., Haro, G., Savadjiev, P., Wassermann, D., Anwander, A., Deriche, R., Pike, G., Sapiro, G., Siddiqi, K., Thompson, P. (2009). Mathematical methods for diffusion MRI processing. *NeuroImage*, 45(1, Supplement 1), S111 – S122.

Lepore, N., Brun, C., Chiang, M. C., Chou, Y. Y., Dutton, R., Hayashi, K., Lopez, O., Aizenstein, H., Toga, A., Becker, J., Thompson, P. (2006). Multivariate Statistics of the Jacobian Matrices in Tensor Based Morphometry and Their Application to HIV/AIDS. In: *MICCAI* (pp. 191–198).

Moakher, M. (2005). A differential geometric approach to the geometric mean of symmetric positive-definite matrices. *SIAM Journal on Matrix Analysis and Applications*, 26(3), 735–747.

Moakher, M., Zéraï, M. (2011). The Riemannian Geometry of the Space of Positive-Definite Matrices and Its Application to the Regularization of Positive-Definite Matrix-Valued Data. *Journal of Mathematical Imaging and Vision*, 40, 171–187.

Ncube, S., Srivastava, A. (2011). A novel Riemannian metric for analyzing HARDI data. In: *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series* (p 7), vol 7962.

Nesterov, Y., Nemirovskii, A. S., Ye, Y. (1994). *Interior-point polynomial algorithms in convex programming*, vol 13. SIAM.

Parker, G. J. M., Haroon, H. A., Wheeler-Kingshott, C. A. M. (2003). A framework for a streamline-based probabilistic index of connectivity (PICo) using a structural interpretation of MRI diffusion measurments. *Journal of Magnetic Resonance Imaging*, 18, 245–254.

Pennec, X., Fillard, P., Ayache, N. (2006). A Riemannian framework for tensor computing. *International Journal of Computer Vision*, 66(1), 41–66.

Perona, P., Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(7), 629–639.

Petz, D., Temesi, R. (2005). Means of positive numbers and matrices. *SIAM J Matrix Anal Appl*, 27(3), 712–720.

Sarlette, A., Sepulchre, R., Leonard, N. (2007). Cooperative attitude synchronization in satellite swarms: a consensus approach. *Proceedings of the 17th IFAC Symposium on Automatic Control in Aerospace.*

Skovgaard, L. T. (1984). A Riemannian Geometry of the Multivariate Normal Model. *Scandinavian Journal of Statistics*, 11(4), pp. 211–223.

Smith, S. T. (2005). Covariance, subspace, and intrinsic Cramér-Rao bounds. *IEEE Transactions on Signal Processing*, 53, 1610–1630.

Thévenaz, P., Blu, T., Unser, M. (2000). Interpolation revisited [medical images application]. *Medical Imaging, IEEE Transactions on*, 19(7), 739–758.

Tschumperlé, D., Deriche, R. (2001). Diffusion tensor regularization with constraints preservation. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1, 948.

Tuch, D. (2004). Q-ball imaging. *Magnetic Resonance in Medicine*, 52(6), 1358–1372.

Weldeselassie, Y., Hamarneh, G. (2007). DT-MRI segmentation using graph cuts. (In: *SPIE*)vol 6512.

Weldeselassie, Y., Hamarneh, G., Beg, F., Atkins, S. (2009). Novel decomposition of tensor distance into shape and orientation distances. In: *MICCAI* (pp. 173–180).

Yeo, B., Vercautern, T., Fillard, P., Pennec, X., Gotland, P., Ayache, N., Clatz, O. (2008). DTI registration with exact finite-strain differential. (In: *IEEE Internation Symposium on Biomedical Imaging: From Nano to Macro*).

Yeo, B., Vercauteren, T., Fillard, P., Peyrat, J. M., Pennec, X., Golland, P., Ayache, N., Clatz, O. (2009). DT-REFinD: Diffusion Tensor registration with exact finite-strain differential. *IEEE Transactions on Medical Imaging*, 28.

Zhang, H., Yushkevich, P. A., Alexander, D., Gee, J. (2006). Deformable registration of diffusion tensor MR images with explicit orientation optimization. *Medical Image Analysis*, 10(5), 764 – 785.

Zhou, D. (2010). Statistical analysis of diffusion tensor imaging. PhD thesis, University of Nottingham.

Zhou, D., Dryden, I. L., Koloydenko, A., Bai, L. (2013). Procrustes analysis for diffusion tensor image processing. *International Journal of Computer Theory and Engineering*, 5(1), 108–113.