

DTI Segmentation and Fiber Tracking Using Metrics on Multivariate Normal Distributions

Minyeon Han · F.C. Park

Published online: 6 December 2013
© Springer Science+Business Media New York 2013

Abstract Existing clustering-based methods for segmentation and fiber tracking of diffusion tensor magnetic resonance images (DT-MRI) are based on a formulation of a similarity measure between diffusion tensors, or measures that combine translational and diffusion tensor distances in some ad hoc way. In this paper we propose to use the Fisher information-based geodesic distance on the space of multivariate normal distributions as an intrinsic distance metric. An efficient and numerically robust shooting method is developed for computing the minimum geodesic distance between two normal distributions, together with an efficient graph-clustering algorithm for segmentation. Extensive experimental results involving both synthetic data and real DT-MRI images demonstrate that in many cases our method leads to more accurate and intuitively plausible segmentation results vis-à-vis existing methods.

Keywords Magnetic resonance imaging · Diffusion tensor · Image segmentation · Fiber Tracking · Multivariate normal distribution · Riemannian geometry

1 Introduction

Since the work of Basser et al. [4], the segmentation of diffusion tensor magnetic resonance images (DT-MRI) has typically been cast as a clustering problem involving a similarity measure between diffusion tensors. The diffusion tensors

characterize the spatial motion of water molecules; since water diffuses more rapidly in directions aligned with the internal tissue structure, by statistically characterizing the directions in which the water molecules diffuse, one can extract information about the underlying anatomical structure.

Once a reference frame and coordinates have been chosen, a diffusion tensor admits a representation as a symmetric positive-definite matrix—throughout we denote by $\mathcal{P}(n)$ the space of $n \times n$ symmetric positive-definite matrices. A similarity measure for diffusion tensors then corresponds to a choice of distance metric on $\mathcal{P}(n)$. The literature on $\mathcal{P}(n)$ distance metrics is extensive—see, e.g., [2, 9, 16, 17, 20, 23, 24]—and various distance metrics on $\mathcal{P}(n)$ have been used in the context of DT-MRI analysis: the Frobenius norm is used in [6] for DTI regularization and matching; the angular difference between principal eigenvectors is used in [27] for thalamic segmentation; the Riemannian distance based on the Fisher information metric is used in [13] for segmentation; the Log-Euclidean distance is used as an approximated version of the Riemannian distance in [3]; and the Kullback–Leibler divergence and its symmetrized version are used in [25] for segmentation. Despite some limitations of this approach, e.g., at voxels where many neuronal fibers cross, segmenting DT-MRI images based on a choice of distance metric on $\mathcal{P}(n)$ has proven quite effective in investigating tissue microstructure.

On the other hand it is easy to come up with situations in which using just a distance metric on $\mathcal{P}(n)$ can lead to nonintuitive results. In Fig. 1(i), for example, the shape of the ellipsoid at (b) would suggest that a water molecule at (b) is much more likely to diffuse toward (c) rather than (a). A metric on $\mathcal{P}(n)$, however, would indicate that the ellipsoids at (a) and (c) are both equidistant from the ellipsoid at (b). Similarly, for Fig. 1(ii), one would expect that a water molecule at (b) is more likely to diffuse toward (a) rather

M. Han (✉) · F.C. Park
Robotics Laboratory, Seoul National University, Seoul 151-744,
Korea
e-mail: alsdus@robotics.snu.ac.kr

F.C. Park
e-mail: fcp@snu.ac.kr

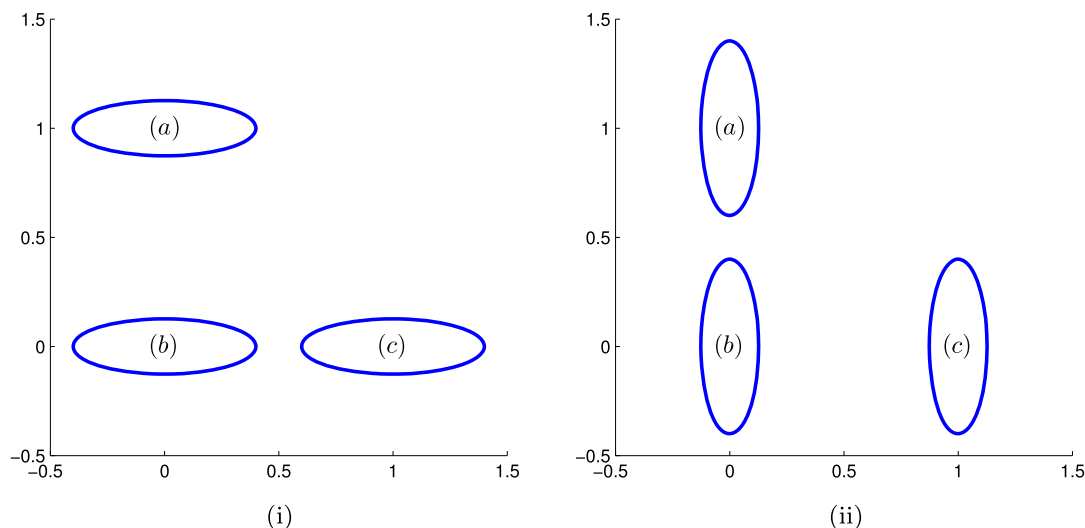


Fig. 1 (i) The ellipsoids represent covariances at different voxel locations; (ii) The ellipsoids at each voxel are now rotated ninety degrees

than (c). Here again a metric on $\mathcal{P}(n)$ would be unable to differentiate between these two cases. A reasonable distance metric should, for the left example, yield $d(a, b) > d(b, c)$, while for the right example it should yield $d(a, b) < d(b, c)$.

A simple remedy to this situation is to take into account not only the shape of the diffusion tensor, but also its spatial voxel location (typically \mathbb{R}^2 or \mathbb{R}^3 in our case) when calculating distances. This often entails a choice of weighting factor between Euclidean space and the space of diffusion tensors. Wiegell et al. [26] propose as a distance metric a linear combination of the Mahalanobis voxel distance and the Frobenius tensor distance with certain weight. Specifically, if x_a, x_b denote the spatial location of two voxels, and D_a, D_b denote their corresponding diffusion tensors, then the distance $d(a, b)$ between the diffusion tensors centered at x_a and x_b is evaluated as

$$d(a, b) = \|x_a - x_b\|_W + \|D_a - D_b\|, \tag{1}$$

where $\|\cdot\|_W$ is a vector norm taken with respect to some symmetric positive-definite matrix W (in [26] W is taken to be the spatial covariance associated with the voxels in the cluster containing x_b), and the norm on diffusion tensors is the usual Frobenius matrix norm. In Wiegell et al’s approach, the distance between two diffusion tensors of the same shape and orientation, located at voxels that are close to each other, will by construction be small. In [1], Abou-Moustafa and Ferrie propose two distance measures on multivariate normal distributions that are similar to those in [26] in terms of combining spatial and covariance distances:

$$d_{JR}(a, b) = ((\mu_a - \mu_b)^T (\Sigma_1^{-1} + \Sigma_2^{-1}) (\mu_a - \mu_b))^{\frac{1}{2}} + d_R(\Sigma_1, \Sigma_2), \tag{2}$$

$$d_{BR}(a, b) = \left((\mu_a - \mu_b)^T \left(\frac{1}{2} \Sigma_1 + \frac{1}{2} \Sigma_2 \right)^{-1} (\mu_a - \mu_b) \right)^{\frac{1}{2}} + d_R(\Sigma_1, \Sigma_2). \tag{3}$$

The first metric $d_{JR}(\cdot, \cdot)$ combines the mean component of the J-divergence with the $\mathcal{P}(n)$ Riemannian distance, while $d_{BR}(\cdot, \cdot)$ combines the mean part of Bhattacharyya distance with the $\mathcal{P}(n)$ Riemannian distance. While these metrics produce distances that agree with our intuition about the scenarios shown in Fig. 1, it is important to note that (2) and (3) are only pseudo-metrics; they fail to satisfy the triangle inequality, which has important consequences for, e.g., metric-based classification.

Another way to avoid the potential discrepancy described in Fig. 1 is via the method proposed in O’Donnell et al. [18]. Here the diffusion tensor at x , denoted D_x , defines an inverse Riemannian metric at x ; the distance between (x_a, D_a) and (x_b, D_b) is then measured to be the length of the shortest path in Euclidean space connecting x_a and x_b , with infinitesimal arclengths now measured according to $ds^2 = dx^T D_x^{-1} dx$. While this approach properly distinguishes between the two cases of Fig. 1, a smooth and continuous tensor field is required, and the choice of approximation and interpolation method will clearly have an influence on the final outcome. The metric of [18] also does not explicitly capture variations exclusively in the $\mathcal{P}(n)$ component along the Euclidean base curve, although this can be rectified straightforwardly with the addition of an appropriate $\mathcal{P}(n)$ metric term as detailed in Sect. 3.4.

A more natural and rigorous setting in which to formulate distance metrics like that of (1) is to identify each measured diffusion tensor with a normal distribution—its mean is given by the spatial location of the voxel, and its covariance by the $\mathcal{P}(n)$ representation of the diffusion tensor. This

is, in fact, precisely the geometric setting in which Lenglet et al. [13] formulate their DT-MRI segmentation algorithm, although for some reason they choose to ignore the mean and consider only the covariance in their distance calculations.

As is well-known, the space of n -dimensional normal distributions, which we denote $\mathcal{N}(n)$, has the structure of a Riemannian manifold, with the Fisher information metric serving as a natural choice of Riemannian metric. The Riemannian geometry of $\mathcal{N}(n)$ has been investigated in [5, 8, 10, 14, 22] where, among other things, the differential equations for minimal geodesics are derived. Finding the minimal geodesic (and thus the distance) between two normal distributions involves solving these equations with the two given endpoint conditions. This leads to a nonlinear two-point boundary value problem, which with very few exceptions must be solved numerically, typically via shooting or relaxation type methods. Analytic characterizations of the minimal geodesic are available in the one-dimensional case, but in the general n -dimensional case very little can be said.

In this paper we propose a method for DT-MRI segmentation that takes into account the natural Riemannian geometry of $\mathcal{N}(n)$ as induced from the Fisher information metric. As a primary contribution, we develop a fast and numerically stable shooting algorithm that calculates geodesic distances between any two given normal distributions. The method is coordinate-invariant in the sense that the endpoint fitting errors are translated to the initial point in a way that respects the intrinsic geometry of $\mathcal{N}(n)$ (specifically, we use parallel transport with respect to the Fisher information Riemannian metric). As a secondary contribution, we construct a graph structure of the DT-MRI image, and develop a graph-based clustering algorithm that considerably reduces the computational complexity vis-à-vis more conventional methods like spectral clustering.

Experimental results are generated for a wide range of both synthetic and actual DT-MRI data, with experiments ranging from DTI segmentation to fiber tractography. Although the lack of ground truth information for actual DT-MRI images makes it difficult to draw strong and definitive conclusions, applying various quantitative criteria to measure segmentation performance, we find that performance is in many cases improved by using the $\mathcal{N}(n)$ metric over the $\mathcal{P}(n)$ metric or the ad hoc metric of [26]. Both visual inspection and qualitative assessment of the results also bears out our finding in a number of cases. Extensive numerical experiments with our shooting algorithm also suggest that $\mathcal{N}(n)$ is (at least in the cases $n = 2$ and $n = 3$) geodesically complete (that is, given any two points in $\mathcal{N}(n)$ a minimal geodesic between them exists), and that our shooting method always converges to a unique minimal geodesic (as of yet the question of existence and uniqueness of minimal geodesics on $\mathcal{N}(n)$ does not appear to have been formally addressed in the literature).

The paper is organized as follows. In Sect. 2 we review the Riemannian structure of the manifold of multivariate normal distributions, and explicitly derive the equations for parallel transport. In Sect. 3 we describe the numerical shooting algorithm for determining the minimal geodesics on $\mathcal{N}(n)$. In Sect. 4 we describe our graph model for DTI and the corresponding graph-based clustering algorithm. Results of experiments with synthetic and real DT-MRI data are described in Sect. 5.

2 Riemannian Geometry of Multivariate Normal Distributions

In this section we review the Riemannian structure of the space of multivariate normal distributions; we refer the reader to, e.g., [13, 22] for further details and references. The manifold $\mathcal{N}(n)$, is formally defined as follows:

$$\mathcal{N}(n) = \{N_n(\mu, \Sigma) \mid \mu \in \mathbb{R}^n, \Sigma \in \mathcal{P}(n)\}. \tag{4}$$

Here $N_n(\mu, \Sigma)$ denotes a normal distribution in \mathbb{R}^n with mean $\mu \in \mathbb{R}^n$ and covariance $\Sigma \in \mathcal{P}(n)$, with $\mathcal{P}(n)$ denoting the space of real symmetric $n \times n$ positive-definite matrices. $\mathcal{N}(n)$ is a differentiable manifold of dimension $n + \frac{1}{2}n(n + 1)$. A natural choice of local coordinate chart for $\mathcal{N}(n)$ is given by

$$\Psi(N_n(\mu, \Sigma)) = ((\mu_i)_{i=1,\dots,n}, (\sigma_{ij})_{i \leq j}), \tag{5}$$

where the local coordinates are $\mu = (\mu_1, \dots, \mu_n)^T$, $\Sigma = (\sigma_{ij})_{i,j=1,\dots,n}$.

$\mathcal{N}(n)$ can be turned into a Riemannian manifold by using the Fisher information as Riemannian metric. In terms of the above local coordinates, at a point (μ, Σ) the Riemannian metric $g(\cdot, \cdot)$ assumes the form

$$g(e_i, e_j) = e_i^T \Sigma^{-1} e_j \tag{6}$$

$$g(e_i, E_{kl}) = 0 \tag{7}$$

$$g(E_{ij}, E_{kl}) = \frac{1}{2} \text{tr}(\Sigma^{-1} E_{ij} \Sigma^{-1} E_{kl}), \tag{8}$$

where $\{e_1, \dots, e_n\}$ denote the standard basis vectors in \mathbb{R}^n (that is, the i -th element of e_i is 1, and the remaining elements are zero), and

$$E_{ij} = \begin{cases} 1_{(i,i)}, & i = j, \\ 1_{(i,j)} + 1_{(j,i)}, & i \neq j, \end{cases} \tag{9}$$

where $1_{(i,j)}$ represents the $n \times n$ matrix whose (i, j) entry is 1, and all other entries are zero.

Using above metric, we can define the inner product on $\mathcal{N}(n)$ space. The inner product of two tangent vector $V = (V_\mu, V_\Sigma)$ and $W = (W_\mu, W_\Sigma)$ defined at a point

$P = (\mu, \Sigma)$ on $\mathcal{N}(n)$ is given by

$$\langle V, W \rangle_P = V_\mu^T \Sigma^{-1} W_\mu + \frac{1}{2} \text{tr}(\Sigma^{-1} V_\Sigma \Sigma^{-1} W_\Sigma). \quad (10)$$

For a general Riemannian manifold with local coordinates $x \in \mathbb{R}^n$ and Riemannian metric $g_{ij}(x)$, the distance between two points on the manifold can be defined as the length of the shortest (twice-differentiable) path connecting the two points. The shortest path is known as the minimal geodesic, and must satisfy the Euler–Lagrange equations:

$$\frac{d^2 x_k(t)}{dt^2} + \sum_{i,j=1}^n \Gamma_{ij}^k \frac{dx_i(t)}{dt} \frac{dx_j(t)}{dt} = 0, \quad \forall k = 1, \dots, n. \quad (11)$$

Here the Γ_{ij}^k denote Christoffel symbols of the second kind, and are defined as

$$\Gamma_{ij}^k = \sum_{s=1}^m \frac{1}{2} g^{ks} \left(\frac{\partial g_{si}}{\partial x_j} - \frac{\partial g_{ij}}{\partial x_s} + \frac{\partial g_{js}}{\partial x_i} \right), \quad (12)$$

with g^{ks} denoting the (k, s) entry of the inverse of (g_{ij}) . Any solution to the above equations is a geodesic; the minimal geodesic is the geodesic with the shortest path, and requires the solution to a two-point boundary value problem.

On $\mathcal{N}(n)$ the geodesic equations are given explicitly by

$$\frac{d^2 \mu}{dt^2} = \left(\frac{d\Sigma}{dt} \right) \Sigma^{-1} \left(\frac{d\mu}{dt} \right) \quad (13)$$

$$\frac{d^2 \Sigma}{dt^2} = \left(\frac{d\Sigma}{dt} \right) \Sigma^{-1} \left(\frac{d\Sigma}{dt} \right) - \left(\frac{d\mu}{dt} \right) \left(\frac{d\mu}{dt} \right)^T. \quad (14)$$

It is worthwhile mentioning some special cases where analytical formulas are possible. As mentioned earlier, when $n = 1$ the minimal geodesics can be characterized analytically. Also, in the event that $\mu_0 = \mu_1$, Jensen (see [13] for a more detailed discussion) has derived the following formula for the length of the minimal geodesic:

$$\text{dist}(\Sigma_0, \Sigma_1) = \sqrt{\frac{1}{2} \sum_{i=1}^n \log^2(\lambda_i)}, \quad (15)$$

where the $\lambda_i, i = 1, \dots, n$ denote the eigenvalues of the matrix $\Sigma_0^{-1/2} \Sigma_1 \Sigma_0^{-1/2}$ (the square roots are taken to be symmetric positive-definite).

Henceforth we adopt the $\exp(\cdot)$ and $\log(\cdot)$ notation to denote time evolution along geodesics:

$$B = \exp_A(V) \quad (16)$$

$$V = \log_A(B). \quad (17)$$

Here $\exp_A(V)$ denotes the solution of the geodesic equations (13) and (14), while $\log_A(B)$ denotes the initial tangent

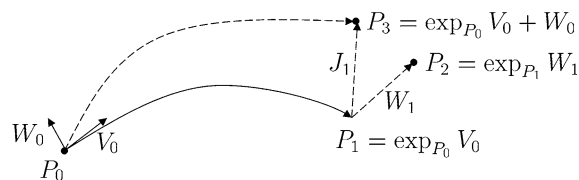


Fig. 2 Validation of using parallel transport instead of Jacobi field

vector at A corresponding to the minimal geodesic between A and B .

We close this section with a discussion of parallel transport and Jacobi field on $\mathcal{N}(n)$. On a general Riemannian manifold the Christoffel symbols of the second kind Γ_{ij}^k define a covariant derivative on the manifold, which in turn provides a means of transporting tangent vectors along curves on the manifold, in such a way that the tangent vectors remain parallel with respect to the covariant derivative. For our purposes, parallel transport offers a means of transporting tangent vectors from one end of a minimal geodesic to the other; this will prove useful in developing our geometric shooting method algorithm for determining minimal geodesics on $\mathcal{N}(n)$.

Given a smooth vector field V on $\mathcal{N}(n)$ (i.e., a smooth mapping V from $\mathcal{N}(n)$ to its tangent bundle of the form $(\mu, \Sigma) \mapsto (V_\mu, V_\Sigma)$), suppose we wish to transport this vector field along some curve on $\mathcal{N}(n)$. The transported vector field must then satisfy the following pair of differential equations:

$$\frac{dV_\mu}{dt} = \frac{1}{2} \left(\frac{d\Sigma}{dt} \right) \Sigma^{-1} V_\mu + \frac{1}{2} V_\Sigma \Sigma^{-1} \left(\frac{d\mu}{dt} \right) \quad (18)$$

$$\frac{dV_\Sigma}{dt} = \frac{1}{2} \left\{ \left(\frac{d\Sigma}{dt} \right) \Sigma^{-1} V_\Sigma + V_\Sigma \Sigma^{-1} \left(\frac{d\Sigma}{dt} \right) \right\} - \frac{1}{2} \left\{ \left(\frac{d\mu}{dt} \right) V_\mu^T + V_\mu \left(\frac{d\mu}{dt} \right)^T \right\}. \quad (19)$$

The Jacobi field is a vector field defined along a geodesic γ that describes the difference between the geodesic and infinitesimal variations of the geodesic. Specifically, the Jacobi field is defined to be

$$J(t) = \left. \frac{\partial \gamma_\tau(t)}{\partial \tau} \right|_{\tau=0}, \quad (20)$$

where τ denotes the variation of geodesic and $\gamma_0 = \gamma$. It is known that J then satisfies the following Jacobi equation:

$$\frac{D^2}{dt^2} J(t) + R(J(t), \dot{\gamma}(t)) \dot{\gamma}(t) = 0, \quad (21)$$

where R denotes Riemannian curvature tensor.

For example, given a geodesic curve γ between P_0 and P_1 with initial tangent vector V , suppose V is perturbed to $V + W$ for some infinitesimal tangent vector W . The variation of the final point P_1 can then be derived via the Jacobi

field, as the solution of (21) with $J(0) = 0$ and $\dot{J}(0) = W$. In terms of the exponential map, this relation can also be written

$$J(t) = \frac{d}{d\tau} \exp_{P_0}(t(V + \tau W)) \Big|_{\tau=0}. \tag{22}$$

3 Algorithm for Minimal Geodesics

3.1 Algorithm Description

To motivate our numerical procedure for finding minimal geodesics on $\mathcal{N}(n)$, consider the following scalar two-point boundary value problem: given the second-order differential equation $\ddot{x}(t) = f(t, x(t), \dot{x}(t))$ with boundary conditions $x(t_0) = x_0$ and $x(t_1) = x_1$, we seek the initial slope $\dot{x}(t_0) = a$ such that the solution satisfies the boundary conditions. Assuming a solution exists, the shooting method is essentially a numerical root-finding procedure for solving the equation

$$g(a) = x(t_1; a) - x_1 = 0, \tag{23}$$

where $x(t_1; a)$ denotes the trajectory at $t = t_1$ for the given initial slope $\dot{x}(t_0) = a$. Note that obtaining $x(t_1; a)$ requires numerical integration of the original differential equations. If the basic Newton–Raphson method is invoked, then intuitively each iteration in the shooting method amounts to adding $g(a)$, the error at the final time t_1 , to the initial slope a , and obtaining $x(t_1)$ for the new slope $a + g(a)/(t_1 - t_0)$. This procedure is repeated until the error converges to some prescribed ϵ . In actual implementations the method requires information about the derivatives of g with respect to a , which in turn involves repeated integrations of the differential equation for different initial values of a .

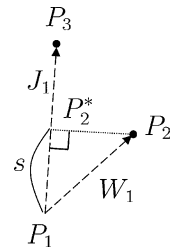
Since in our case the differential equations locally evolve on the manifold $\mathbb{R}^n \times \mathcal{P}(n)$, many of the previous vector space notions need to be suitably generalized. The numerical integration clearly needs to take into account the geometry of the underlying space, to ensure that the solution does not deviate from the manifold. More crucially, the question of how to update the initial slope a by “adding” the error $g(a)$ needs to be addressed. The most natural way is to find the initial value of the Jacobi field $\dot{J}(0)$ such that the error matches $J(1)$ (here $J(0) = 0$), and to add $\dot{J}(0)$ to the initial slope.

Using the Jacobi field, however, presents a number of practical difficulties. Although the Riemannian curvature tensor on $\mathcal{N}(n)$ is known, the Jacobi equation (21) has no known closed-form solution; solution via numerical integration is also computationally difficult. Finding $\dot{J}(0)$ to reach the desired $J(1)$ presents another two-point boundary value

Table 1 Comparison between Jacobi field and parallel transport in $\mathcal{N}(n)$

$\ v_0\ _{P_0}$	$\max(\text{dist}(P_1, P_3)/\text{dist}(P_1, P_2))$	$\max(\angle P_2 P_1 P_3)$
1	0.1373	4.3607
2	0.6589	16.325
3	1.9356	31.6088
4	4.6633	48.1440
5	9.3629	62.9973
7	35.1561	77.6493

Fig. 3 Appropriate update step size



problem that needs to be solved within the main shooting method procedure.

Our solution is to use parallel transport to approximate the Jacobi field. This approximation is valid in cases where the curvature is sufficiently close to zero and the geodesic length is reasonably short, as we now verify through numerical experiments. Referring to Fig. 2, we first define two random tangent vectors V_0 and W_0 at P_0 , and determine the geodesic curve from P_0 to P_1 with initial tangent vector V_0 . We then calculate W_1 at P_1 by parallel transporting W_0 along the geodesic. Denote by P_2 the endpoint of the geodesic from P_1 with initial tangent vector W_1 , and by P_3 the endpoint of the geodesic from P_0 with initial tangent vector $V_0 + W_0$. If W_0 is sufficiently small and parallel transport is a reasonable approximation to the Jacobi field, then $\text{dist}(P_2, P_3)$ should be much smaller than $\text{dist}(P_1, P_2)$. For our experiments, we set the mean and covariance of P_0 to zero and the identity, respectively (since our distance metric on $\mathcal{N}(n)$ is invariant with respect to affine transformations, i.e., $\text{dist}((\mu_1, \Sigma_1), (\mu_2, \Sigma_2)) = \text{dist}((A\mu_1 + b, A\Sigma_1A^T), (A\mu_2 + b, A\Sigma_2A^T))$ for all $A \in GL(n)$ and $b \in \mathbb{R}^n$, there is no loss of generality in the zero mean and identity covariance assumption).

Table 1 shows the worst case results for 1000 trials. When the distance is greater than 2, parallel transport does not approximate the Jacobi field very well. However, we still find that $\angle P_2 P_1 P_3$ is smaller than 90 degrees. Using the above result in our shooting method, we parallel transport the endpoint error to the initial point, and determine an appropriate update size by projecting the error on the numerically obtained Jacobi field.

In Fig. 3, we assume that P_1 is the endpoint of the current geodesic, and P_2 is the desired endpoint near P_1 . If we

Algorithm 1 Shooting method for minimal geodesics on $\mathcal{N}(n)$

Given: Initial point $P_0 = (\mu_0, \Sigma_0)$, final point $P_1 = (\mu_1, \Sigma_1)$.
Output: Minimal geodesic $P(t) = (\mu(t), \Sigma(t))$, $t \in [0, 1]$, such that $P(1) = (\mu_1, \Sigma_1)$.
Initialization: Choose initial velocities $V(0) = (\dot{\mu}(0), \dot{\Sigma}(0))$ (e.g., zeroes), initial values for ϵ (10^{-5}), error = 10^6 .
while error $\geq \epsilon$ **do**
 Numerically integrate the geodesic equations (13), (14) for given initial conditions $(\mu_0, \Sigma_0, \dot{\mu}_0, \dot{\Sigma}_0)$ from $t = 0$ to $t = 1$.
 Denote the solution by $(\mu(t), \Sigma(t))$;
 Set $W(1) = (W_\mu(1), W_\Sigma(1)) = (\mu_1 - \mu(1), \Sigma_1 - \Sigma(1))$;
 Calculate error = $\|W(1)\|_{P_1} = \sqrt{W_\mu(1)^T \Sigma_1^{-1} W_\mu(1) + \frac{1}{2} \text{tr}((\Sigma_1^{-1} W_\Sigma(1))^2)}$;
 Numerically integrate the parallel transport equations (18) and (19) for given trajectory $(\mu(t), \Sigma(t))$ and final velocities $W(1)$, backward in time from $t = 1$ to $t = 0$;
 Numerically calculate Jacobi field $J(1)$ from (22),
 $J(1) = \frac{\exp_{P_0}(V(0) + \alpha W(0)) - \exp_{P_0}(V(0))}{\alpha}$, where α is sufficiently small value and we use $\frac{\epsilon}{\|W(0)\|_{P_0}}$
 Determine proper update size s :
 $s_1 = \frac{\langle W(1), J(1) \rangle_{P(1)}}{\|J(1)\|_{P(1)}^2}$
if $\|W(1)\|_{P(1)} > 0.05$ **then**
 $s = 0.05 / \|W(1)\|_{P(1)} s_1$;
else
 $s = s_1$;
end if
 $V(0) \leftarrow V(0) + sW(0)$;
end while

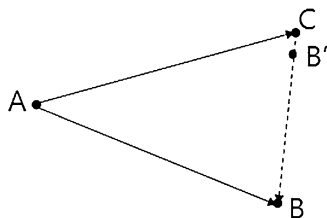


Fig. 4 An illustration of the shooting method on $\mathcal{N}(n)$

parallel transport the error vector W_1 to the initial point and add this to original initial vector $(V_0 + W_0)$, then the endpoint of the new geodesic becomes P_3 . J_1 is the Jacobi field with initial tangent vector V_0 and perturbation W_0 , and indicates the direction from P_1 to P_3 . In our algorithm, we project W_1 onto J_1 and determine an appropriate update size $s = \frac{\langle J_1, W_1 \rangle_{P_1}}{\|J_1\|_{P_1}^2}$. $V_0 + sW_0$ is our new initial tangent vector, and the endpoint of the geodesic approaches P_2^* , which is now closer to P_2 than P_3 .

The above derivation sheds further light on what it means intuitively to parallel transport the final position error back to the initial point. We now present our shooting method algorithm for finding the minimal geodesic between any two given endpoints in $\mathcal{N}(n)$. Given a minimal geodesic on $\mathcal{N}(n)$ connecting the initial point $A = (\mu_A, \Sigma_A)$ with the final point $B = (\mu_B, \Sigma_B)$, such that the initial tangent vector at A is $V = (V_\mu, V_\Sigma)$.

Referring to Fig. 4, suppose we seek the geodesic path between A and B . If our guess for the tangent vector at A results in a geodesic to C rather than B —in this case the tangent vector at A is given by $\log_A(C)$ —then this initial tangent vector needs to be corrected by taking into account the error at the endpoint, captured by a tangent vector at C that corresponds to the minimal geodesic from C to B , or $\log_C(B)$. This involves finding yet another minimal geodesic, this time from C to B . To avoid this complication we adopt the following iterative procedure to approximately determine $\log_C(B)$. First, we determine the straight line (in Euclidean space) between B and C , and choose a point B' on this line that is close to C . Provided B' is sufficiently close to C , then $\log_C(B')$ is closely approximated by $B' - C$.¹ This tangent vector is then parallel transported to A , and added to $\log_A(C)$ to form a new tangent vector at A with proper update size; the corresponding minimal geodesic should then generate an endpoint that is closer to B than the previously attained C . The above procedure is repeated until the resulting minimal geodesic reaches the desired endpoint B to some desired accuracy. This iterative procedure for finding the minimal geodesic is summarized in Algorithm 1.

Using Algorithm 1, we can determine the initial velocity vectors $\dot{\mu}(0)$ and $\dot{\Sigma}(0)$. The geodesic distance between

¹In our experiments, if $\text{dist}(B', C)$ is less than 0.05, $\log_C(B')$ is well approximated by $B' - C$. We use this constant value in Algorithm 1.

$(\mu(0), \Sigma(0))$ and $(\mu(1), \Sigma(1))$ is then evaluated from the formula

distance

$$= \int_0^1 \sqrt{\dot{\mu}^T(t) \Sigma^{-1}(t) \dot{\mu}(t) + \frac{1}{2} \text{tr}((\Sigma^{-1}(t) \dot{\Sigma}(t))^2)} dt. \tag{24}$$

Since geodesic curves have constant speed at every point, the above equation can be simplified to

$$\text{distance} = \sqrt{\dot{\mu}^T(0) \Sigma^{-1}(0) \dot{\mu}(0) + \frac{1}{2} \text{tr}((\Sigma^{-1}(0) \dot{\Sigma}(0))^2)}. \tag{25}$$

We remark that for the numerical evaluation of the geodesic equation, in principle one can use, e.g., a geometric integration scheme that ensures that each iterate remains on the manifold $\mathcal{N}(n)$. In practice such integration algorithms are computationally expensive, and our experience suggests that ordinary Runge–Kutta integration is sufficient for our purposes provided the integration time step is not too large.

Our algorithm performs well for geodesic distances up to 7; in cases where the geodesic distance exceeds 7, we observe poor convergence and numerical instability. The calculation of the geodesic distance between two very distant distributions, say

$$\mu_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \Sigma_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$\mu_1 = \begin{bmatrix} 10000 \\ 0 \end{bmatrix}, \quad \Sigma_1 = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix},$$

requires over 45,000 iterations using Algorithm 1. This problem can be resolved by extending our algorithm as follows. In order to calculate the geodesic distance between P_0 and P_1 on $\mathcal{N}(n)$, we first determine intermediate points $\{P^1, \dots, P^N\}$ between P_0 and P_1 , such that $P^1 = P_0$, $P^N = P_1$ and

$$\text{dist}(P^i, P^{i+1}) \leq 1.$$

This last condition $\text{dist}(P^i, P^{i+1}) \leq 1$ ensures fast performance of Algorithm 1. Next, we update $P^i = \exp_{P_{i-1}}(0.5 \times \log P_{i-1}(P_{i+1}))$ for even i , followed by an update for odd i ; both updates make use of Algorithm 1. Repeating this until $\sum_{i=1}^{N-1} \text{dist}(P^i, P^{i+1})$ converges, this value becomes the geodesic distance between P_0 and P_1 . For the above example, this distance is evaluated to be 23.4989.

3.2 Examples of Minimal Geodesics

We now consider several examples on $\mathcal{N}(2)$. First, suppose we seek the minimal geodesic connecting the following end-

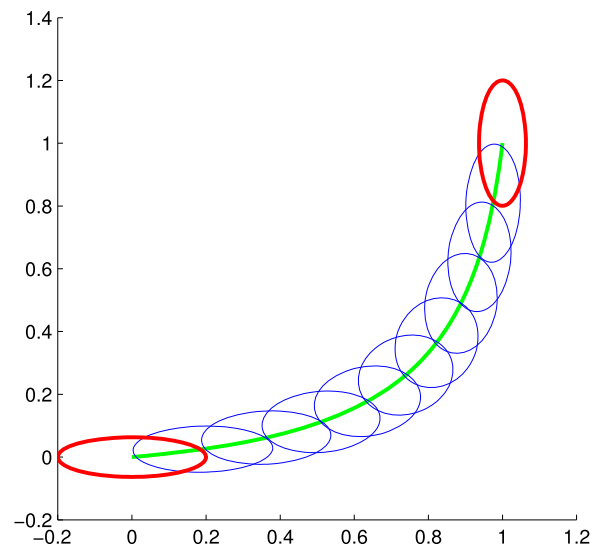


Fig. 5 A minimal geodesic on $\mathcal{N}(2)$

points:

$$\mu_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \Sigma_0 = \begin{bmatrix} 1 & 0 \\ 0 & 0.1 \end{bmatrix},$$

$$\mu_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \Sigma_1 = \begin{bmatrix} 0.1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{26}$$

We take the starting initial tangent vector to be the zero vector, and use the Runge–Kutta fourth-order method to integrate both the geodesic and parallel transport equations, with an integration stepsize of 0.01. The algorithm requires 77 iterations to converge to an endpoint fitting error under 10^{-5} . The resulting geodesic curve is shown in Fig. 5; the geodesic distance is evaluated to be 3.1329.

Figure 5 displays the geodesic curve between the two given distributions. Observe that the geodesic curve between two normal distributions tends to move in directions of greatest uncertainty, i.e., along the major principal axes of the ellipsoid. Figure 6 graphs the error as a function of the number of iterations. We observe that the error decreases in a statistically consistent manner, and converges logarithmically to zero.

Figure 7 displays several more examples of minimal geodesics on $\mathcal{N}(2)$. In examples (i)–(iii) the means follow straight line trajectories, which from symmetry considerations is not altogether surprising. In examples (iv)–(vi) the mean trajectories are nonlinear, and the ellipsoids deform in shape along the direction of the trajectory; this is particularly visible in (vi).

Before proceeding further, a remark on the existence and uniqueness of minimal geodesics on $\mathcal{N}(n)$ is appropriate. We are not aware of any theoretical results confirming the existence and uniqueness of minimal geodesics

on $\mathcal{N}(n)$, and for this purpose we perform a set of numerical experiments to examine existence and uniqueness. First, for randomly selected pairs of multivariate normal distributions that are distant from each other, in all cases our algo-

rithm was able to calculate a geodesic. To test whether the geodesic is unique, we first construct a reference geodesic curve from the geodesic equation, and use our algorithm to calculate geodesics for various random initial tangent vectors (recall that in Algorithm 1 all initial tangent were zero). If the geodesic happens to not be unique, then there should exist two distinct vectors, V_1 and V_2 , that satisfy $\exp_P V_1 = \exp_P V_2$ on $\mathcal{N}(n)$. For random initial tangent vectors close to V_1 one would expect our algorithm to produce V_1 as the result (and similarly for V_2). In all our experiments we always obtain the same initial tangent vector. These numerical experiments lead us to conjecture that, at least for the cases $n = 2$ and $n = 3$, the minimal geodesic exists and is unique for any arbitrary pair of normal distributions.

3.3 Minimal Geodesics for DT-MRI

In calculating minimal geodesics on $\mathcal{N}(3)$ for DT-MRI applications, the relation between the diffusion time scale and the choice of Riemannian metric first needs to be examined in some detail. Recall that the diffusion tensor in DT-MRI data represents the rate of diffusion of water molecules,

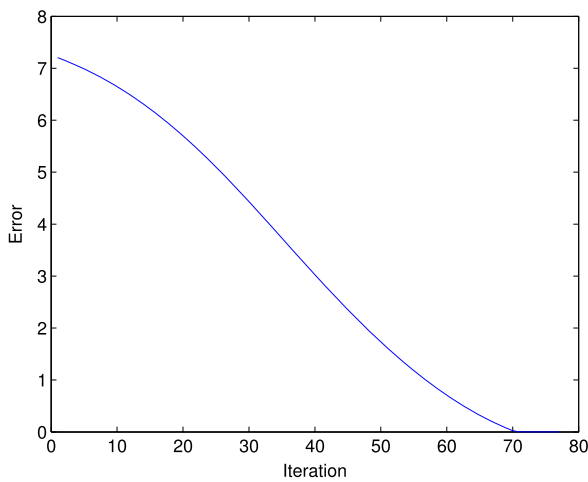


Fig. 6 Endpoint error as a function of number of iterations

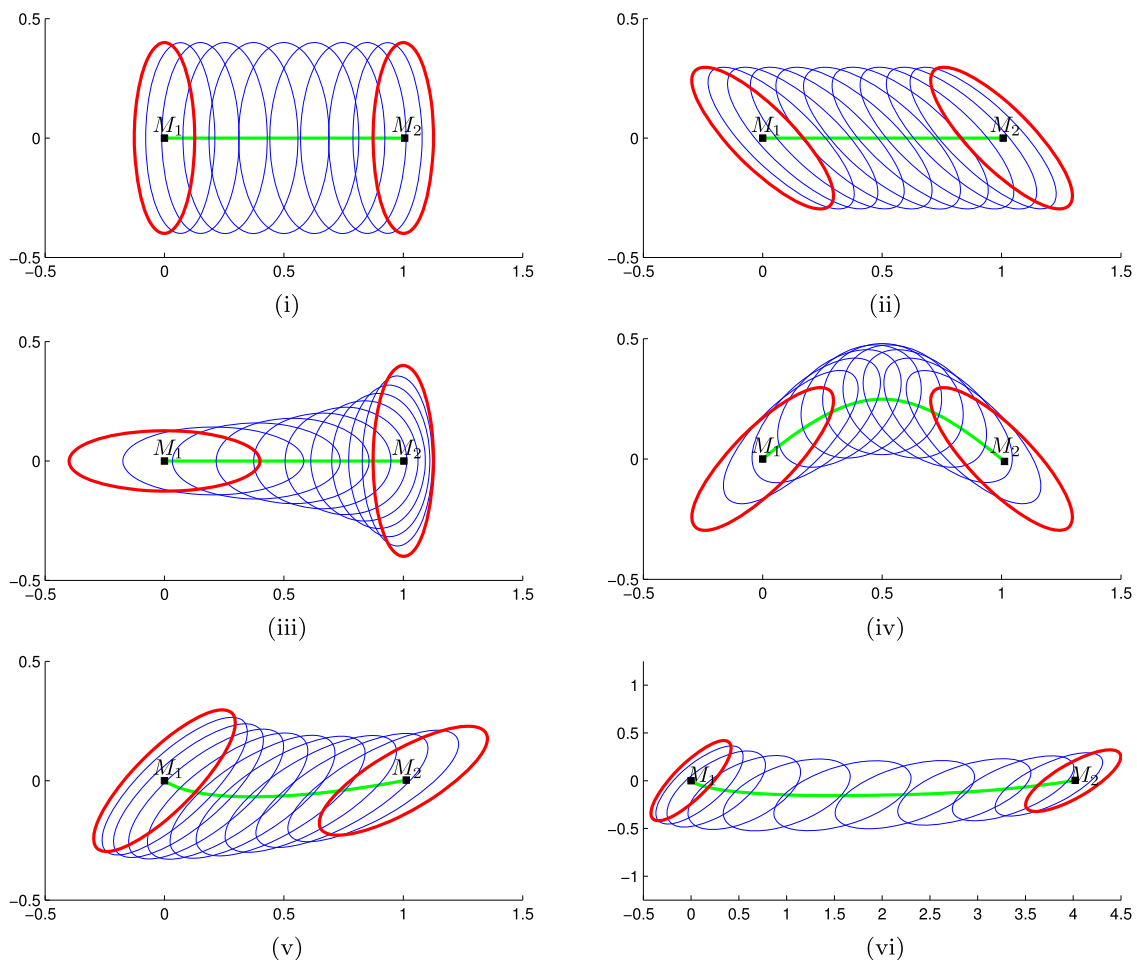


Fig. 7 Further examples of minimal geodesics on $\mathcal{N}(2)$

while the magnitude of the diffusion tensor elements depends on some diffusion time scale τ . The diffusion of water molecules is modelled according to a normal distribution with mean μ and covariance $\Sigma = 2D\tau$ (D is diffusion tensor):

$$\rho(x | \tau) = \frac{1}{\sqrt{(4\pi\tau)^3 \det D}} e^{-\frac{(x-\mu)^T D^{-1} (x-\mu)}{4\tau}}. \tag{27}$$

In the case of the covariance-only geodesic distance on $\mathcal{P}(n)$, the diffusion time scale turns out to be irrelevant due to the transformation invariance property of the distance metric, i.e., $\text{dist}(A, B) = \text{dist}(XAX^T, XBX^T)$ for all $A, B \in \mathcal{P}(3)$ and $X \in GL(3)$.

On $\mathcal{N}(n)$ the diffusion time scale needs to be specified prior to constructing a distance metric. We observe that as the diffusion time scale approaches infinity, the $\mathcal{N}(n)$ geodesic distance and the $\mathcal{P}(n)$ geodesic distance become equal. If on the other hand the diffusion time scale approaches zero, the $\mathcal{N}(n)$ geodesic distance approaches infinity. This diffusion time scale can thus be considered a tunable parameter that emphasizes the relative weights of the spatial locations (means) versus the diffusion tensor shapes (covariances).

For our experiments, we scale the spatial location using a scale factor c . That is, we assume $(c\mu, D)$, where μ denotes spatial coordinates (in mm units) and D is diffusion tensors (in mm^2/s units), is a normal distribution in some DT-MRI image. In the case $c = 0$, the $\mathcal{N}(n)$ metric is equivalent to the $\mathcal{P}(n)$ metric. Since there is no single intrinsic scale factor, we seek scale factors that make the voxel volume nearly identical to the diffusion tensor volume; with this criterion we determine the range of the scale factor to be $0.0026 \sim 0.083$ for our dataset. Using this prior knowledge, we compute the $\mathcal{N}(3)$ geodesic distance with a scale factor in the range of $0.0 \sim 0.1$.

3.4 Comparison with Other Geodesic Concepts in DT-MRI

Several approaches to fiber tractography [11, 18, 21] also make use of geodesic distances that are related to our $\mathcal{N}(n)$ geodesic distance. Given a diffusion tensor field D_x , where $x \in \mathbb{R}^3$ denotes the spatial location, the geodesic distance between two data points (x_a, D_a) and (x_b, D_b) is defined to be the length of the shortest curve $x(t)$ in \mathbb{R}^3 that minimizes

$$\int_0^T (\dot{x}^T D_x^{-1} \dot{x})^{1/2} dt, \tag{28}$$

with $x(0) = x_a$ and $x(T) = x_b$, i.e., the inverse diffusion tensor field is taken to be the underlying Riemannian metric ([18] and [11] calculate geodesic connectivity based on the Eikonal equations, while the Euler–Lagrange equations are

integrated to obtain the geodesic path in [21]). Comparing this with our $\mathcal{N}(n)$ geodesic distance, i.e.,

$$\int_0^T \left(\dot{x}^T D_x^{-1} \dot{x} + \frac{1}{2} \text{tr}(D_x^{-1} \dot{D}_x D_x^{-1} \dot{D}_x) \right)^{1/2} dt, \tag{29}$$

observe that (29) contains an additional term $\frac{1}{2} \text{tr}(D_x^{-1} \dot{D}_x D_x^{-1} \dot{D}_x)$ corresponding to the metric on $\mathcal{P}(n)$. If the first term $\dot{x}^T D_x^{-1} \dot{x}$ reflects water diffusivity, then the second term in (29) also takes into consideration the variations in shape of the diffusion tensors. Whereas geodesics of (28) are spatial curves in \mathbb{R}^3 , geodesics of (29) consist of both a spatial curve in \mathbb{R}^3 and a curve in diffusion tensor space.

Moreover, since DT-MRI diffusion tensors are defined at discrete voxels, to evaluate (28) typically requires some procedure for smooth interpolation through an ordered set of diffusion tensors. The choice of interpolation method, particularly for diffusion tensors that are far apart, introduces an ad hoc element that can influence the results. Such choices can be mitigated with our $\mathcal{N}(n)$ geodesic distance, which involves only the choice of a scalar scaling factor c that reflects the relative magnitudes of the voxel and diffusion tensor volumes. Choosing a scaling factor that equally balances these two notions of volume for the given datasets, smooth diffusion tensor curves satisfying the boundary conditions are automatically generated. Later in Sect. 5 we provide numerical results that contrast these approaches.

4 Graph-Based Clustering Algorithm

4.1 Graph Representation

We first begin with an examination of the distance formula on $\mathcal{N}(n)$ when applied to two distant diffusion tensors. Referring to Fig. 8, it is straightforward enough to determine the minimal geodesic, and calculate distance, between the

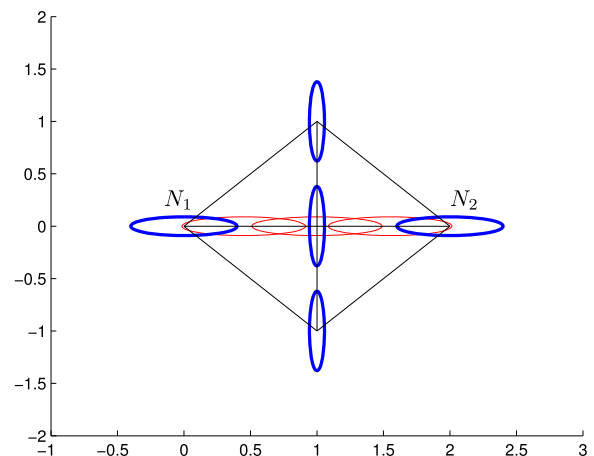


Fig. 8 A simple graph representation on $\mathcal{N}(n)$

two points N_1 and N_2 in $\mathcal{N}(n)$. More often than not the covariance along the geodesic curves will not coincide with the actual measured diffusion tensors at the specific voxel locations. Since a diffusion tensor only provides local information about the diffusion properties of water, when evaluating minimal geodesics between two diffusion tensors it is quite reasonable to use the information provided by intermediate diffusion tensors.

In the light of this observation, in our approach we map a DT-MRI image into an undirected graph: each vertex corresponds to a diffusion tensor, and each edge—placing each vertex at the center of a $3 \times 3 \times 3$ cube, its neighbors are the 26 nodes constituting this cube—represents the distance from the vertex to its adjacent neighbors in three-dimensional space. The distance between any two nodes in the graph is defined to be the length of the shortest path on the graph connecting the two nodes (this notion of distance on graphs is also called geodesic distance, not to be confused with our earlier notion of geodesic distance between two normal distributions). Given this graph representation of DT-MRI data, the weight of each graph edge contains all the information regarding the mean and covariance. We make use of this graph representation, and also Dijkstra's algorithm, for our later DTI clustering and fiber tractography experiments.

4.2 Algorithm Description

For our clustering algorithm we present an adaptation of the k -medoids clustering algorithm [12]. Like the k -means algorithm upon which it is based, the k -medoids algorithm partitions the dataset into clusters by designating a center point (the medoid) for each cluster, and clustering the points so that the sum of the distances of each point from its medoid is minimized. The objective function in k -medoids clustering is given by

$$\sum_{i=1}^n \sum_{k=1}^K I(x_i \in \text{cluster}_k) \text{dist}(m_k, x_i), \quad (30)$$

where x are the data points, m_k denotes the medoid of cluster $_k$, and $I(x_i \in \text{cluster}_k)$ is an indicator function specifying whether x_i belongs to cluster $_k$.

Our clustering algorithm consists of two phases: (i) assigning each point to a cluster, and (ii) updating the medoids for each cluster. In the assignment step, each cluster is characterized as follows:

$$S_i^{(t)} = \{x_j \mid \text{dist}(x_j, m_i^{(t)}) \leq \text{dist}(x_j, \mathbf{m}_{i^*}^{(t)}) \text{ for all } i^* = 1, \dots, k\}, \quad (31)$$

where $\text{dist}(\cdot, \cdot)$ refers to the geodesic distance on the graph. Intuitively, each vertex is assigned to the cluster corresponding to its nearest medoid. To evaluate graph geodesic distances we use a modified form of Dijkstra's algorithm for

Algorithm 2 Cluster assignment step

Given: V (vertex set), E (edge set, E_{ij} is a distance between adjacent vertices i and j), initial medoids.
Output: cluster(v), distance(v), parent(v)
for all $v \leftarrow V$ **do**
 if v is medoid **then**
 cluster(v) = cluster index of v and distance(v) = 0
 and parent(v) = null
 else
 cluster(v) = null and distance(v) = ∞
 end if
end for
 Q = the set of all nodes in graph
while Q is not empty **do**
 $v \leftarrow \arg \min_{u \in Q} \text{distance}(u)$
 Remove v from Q
 for all $u \leftarrow v$'s neighborhood **do**
 if distance(v) + $E_{uv} < \text{distance}(u)$ **then**
 distance(u) = distance(v) + E_{uv} and cluster(u) = cluster(v)
 parent(u) = v
 Reordering u in Q
 end if
 end for
end while

finding the shortest paths in the graph, to accommodate multiple sources without measurably increasing the calculation costs compared to the usual single source case. Algorithm 2 describes the cluster assignment procedure, following standard graph-theoretic notation. We also define two additional functions: cluster(v) indicates the cluster index of vertex v , distance(v) denotes the length of the shortest path between vertex v and its medoid, and parent(v) is the previous vertex of the shortest path.

As is well-known, the performance of Dijkstra's algorithm depends on the data set Q . Using Fibonacci heap, the algorithm has complexity $\mathcal{O}(|E| + |V| \log |V|)$, where $|E|$ and $|V|$ denote the number of edges and vertices, respectively. The medoid update step can be characterized by

$$\min_{v \in \text{cluster}_i} \sum_{v_j \in \text{cluster}_i} \text{dist}(v, v_j). \quad (32)$$

Here we first evaluate the objective function at the initial medoids and their neighbors. Comparing these values, we determine a new set of medoids with minimum values, and repeat this process until the medoids no longer change. Dijkstra's algorithm is also used to evaluate the objective function; the detailed algorithm is described in Algorithm 3. Here we define the function obj(v) to denote the sum of the distance of each vertex in the cluster from v , i.e., the objective function value given in (32).

Algorithm 3 Medoid update step for cluster i

Given: V_i (vertex set of cluster i), E (edge set), initial medoid.
Output: new medoid.
Initialization: $\text{obj}(v) = \infty$ for all $v \in V_i$.
while medoid is not fixed **do**
 $W = \{\text{Current medoid and its neighborhood}\}$
 for all $w \in W$ and $\text{obj}(w) \neq \infty$ **do**
 for all $v \leftarrow V_i$ **do**
 if $v = w$ **then**
 $\text{distance}(v) = 0$
 else
 $\text{distance}(v) = \infty$
 end if
 end for
 $Q = \text{set of all vertices in } V_i$
 while Q is not empty **do**
 $v \leftarrow \arg \min_{u \in Q} \text{distance}(u)$
 Remove v from Q
 for all $u \leftarrow v$'s neighborhood **do**
 if $\text{distance}(v) + E_{uv} < \text{distance}(u)$ **then**
 $\text{distance}(u) = \text{distance}(v) + E_{uv}$
 Reordering u in Q
 end if
 end for
 end while
 $\text{obj}(w) = \sum_{v \in V_i} \text{distance}(v)$
 end for
 new medoid = $\arg \min_{w \in W} \text{obj}(w)$
 end while

We remark that in Algorithm 3, greedy search rather than full search is applied. While greedy search does not guarantee a global minimum, provided the initial medoids are not chosen unreasonably far from their final selections, the computational performance is far superior to using full search. This algorithm is of complexity $\mathcal{O}((|E| + |V| \log |V|)gl)$, where g is the number of medoid searches performed for each cluster, and l is the number of iterations. Typically g will be much smaller than $|V|$ if greedy search is applied; if we apply full search then g becomes $|V|$. The number of iterations is also much smaller compared with k -means clustering, since only data points can be cluster centers. In our later clustering experiments for whole brain DT-MRI data with 200 clusters, using the full search method required over 2800 seconds, with a resulting objective function value of 2.834×10^6 . On the other hand when greedy search is applied, only 93 seconds are required, and the final objective function value is 2.844×10^6 . While the clustering performance of full search is slightly better than that of greedy search, using greedy search is on average about thirty times faster than using full search.

In general, k -means type clustering algorithms (including ours) tend to be sensitive to the choice of initial points. In our later experiments, the objective function values for a whole brain DT-MRI image using 200 clusters is about $2.842 \times 10^6 \sim 2.870 \times 10^6$ using uniform or random initial points. However, in the worst initial case (i.e., all the initial points are stuck together), the objective function value is 4.968×10^6 , which is considerably larger than the uniform or random initial case.

In k -means clustering, there are several well-known approaches to initialization that attempt to get near to a global minimum solution [15]. Despite the similarities between our clustering algorithm and k -means clustering, these initialization methods cannot be applied to our clustering methods due to the graph structure. We thus propose “remove” and “split” methods for improving clustering performance. In this approach we compare the increasing in the objective function value from the cluster being removed, with the decreasing objective function value from the cluster being split. If the decreasing value is larger than the increasing value, the remove-and-split procedure is then performed. Since accurately measuring changes in the objective function values accurately accrues considerable computational cost, we use as an approximation the maximum increasing value and minimum decreasing value of the objective function. Details of our **method** are described in Algorithm 4. Here, $\text{closestmedoid}(k)$ represents the minimum distance between the medoid of cluster k and the medoid of other clusters, and $\text{cnum}(v)$ is the number of child nodes when the graph is converted to a tree structure using the shortest path obtained from Algorithm 2. This procedure should be performed between the assignment and update steps.

Though our remove-and-split method only roughly adjusts the objective function value, it shows good performance even for the worst initial case. There was little change in the objective function values for the uniform and random initial cases. However, using the remove-and-split method for the worst initial case, the objective function values decreased from 4.968×10^6 to 2.892×10^6 .

Since our clustering method is not a boundary extraction method such as in, e.g., [25], in some cases oversegmentation can result. However, our algorithm evaluates the distance between diffusion tensors without losing any information, and in this regard our clustering method is an appropriate means of comparing segmentation results for different distance measures.

5 Experimental Results

To test the performance of our segmentation algorithm, we perform experiments involving both synthetic data and real MR diffusion tensor images of the human brain.

Algorithm 4 Remove-and-Split step

Given: V (vertex set), E (edge set), $\text{cluster}(v)$, $\text{distance}(v)$, $\text{depth}(v)$, $\text{parent}(v)$.
Output: New medoid.
Initialization: $\text{cnum}(v) = 0$ for all $v \in V$.
for all $v \leftarrow V$ **do**
 $w \leftarrow \text{parent}(v)$
 while w is not null **do**
 $\text{cnum}(w) = \text{cnum}(w) + 1$
 $w \leftarrow \text{parent}(w)$
 end while
end for
 $x = \arg \min_v (\text{cnum}(v) + 1) \times \text{closestmedoid}(\text{cluster}(v))$ such that v is a medoid
 $y = \arg \max_v (\text{cnum}(v) + 1) \times (\text{distance}(v) - \text{distance}(\text{parent}(v)))$ such that v is not a medoid
 $\text{obj_increase} = (\text{cnum}(x) + 1) \times \text{closestmedoid}(\text{cluster}(x))$
 $\text{obj_decrease} = (\text{cnum}(y) + 1) \times (\text{distance}(y) - \text{distance}(\text{parent}(y)))$
if $\text{obj_increase} < \text{obj_decrease}$ **then**
 Remove x in medoid
 Add y in medoid
end if

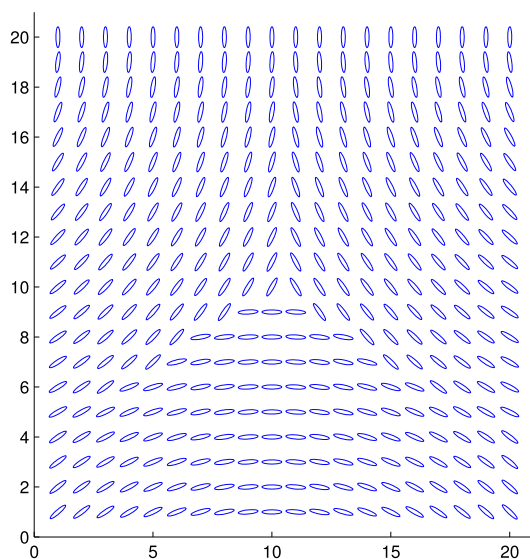


Fig. 9 The synthetic data used in the experiments

5.1 Clustering Experiments with Synthetic Data

Segmentation experiments with synthetic data have been performed for both the covariance-only geodesic distance metric and the geodesic distance on $\mathcal{N}(2)$, using our graph-based clustering algorithm and also more conventional spectral clustering. The synthetic data, shown in Fig. 9, consists of a uniformly spaced 20×20 grid of two-dimensional multivariate normal distributions. We generate three diffusion tensor fields that respectively rotate around the points $(10, -10)$, $(-7, 20)$ and $(27, 20)$. As a result, we can expect

to find similar tensors near the points $(10, 20)$, $(1.5, 5)$ and $(18.5, 5)$.

Figure 10 shows the clustering results for the covariance-only geodesic distance, using both spectral clustering and our graph-based k -medoids clustering algorithm. Both clustering algorithms produce similar results, in which diffusion tensors near $(10, 20)$, $(1.5, 5)$ and $(18.5, 5)$ are grouped into distinct clusters. However, using the $\mathcal{N}(2)$ geodesic distance, diffusion tensors near $(10, -10)$, $(-7, 20)$ and $(27, 20)$ are grouped into three distinct clusters in Fig. 11. In the former case, it is the similarity of the diffusion tensors associated with the water molecules that is the criterion of choice, whereas in the latter case it is the mobility with which a water molecule can diffuse into a neighboring voxel that is the key factor in clustering. Intuitively the case can be made that the $\mathcal{N}(2)$ segmentation results are more in line with what we would expect.

5.2 Fiber Tractography Experiments with Synthetic Data

We now present results of a geodesic-based fiber tractography using synthetic data. Using the synthetic data shown in Fig. 12, we compare results obtained using the $\mathcal{N}(2)$ geodesic distance with those obtained using the Mahalanobis-like distance metric of (28). The distance in (28) is calculated by integrating the infinitesimal Mahalanobis distance with respect to the corresponding diffusion tensor along curve x . Since the diffusion tensors are prescribed at discrete points, smooth interpolation between the discrete tensors is typically performed in most integration schemes. For our purposes the edge weight shall simply be

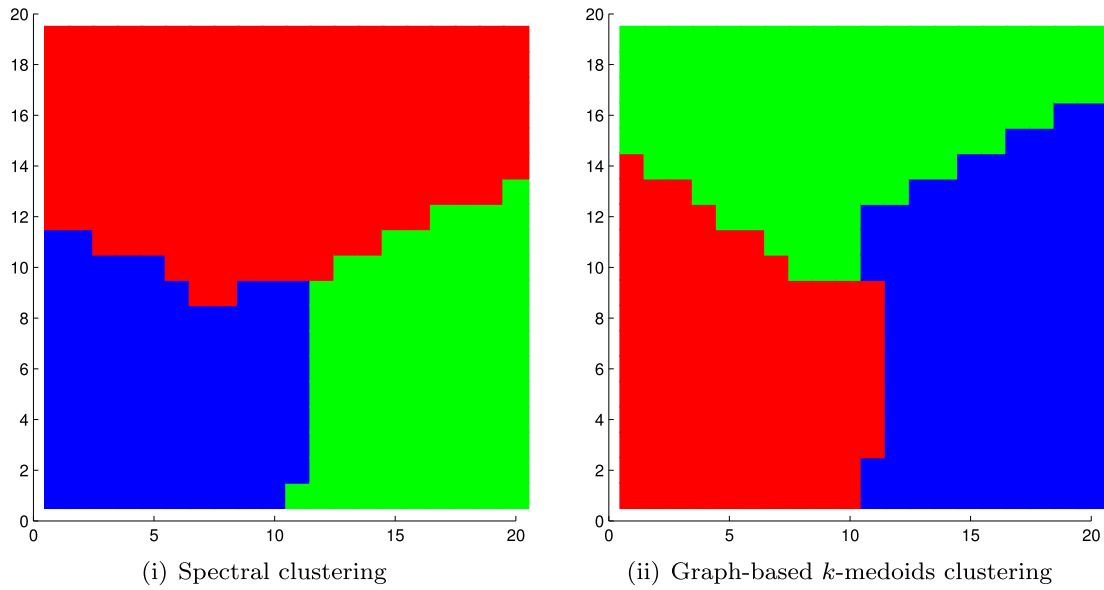


Fig. 10 Clustering result for synthetic data using covariance-only geodesic distance

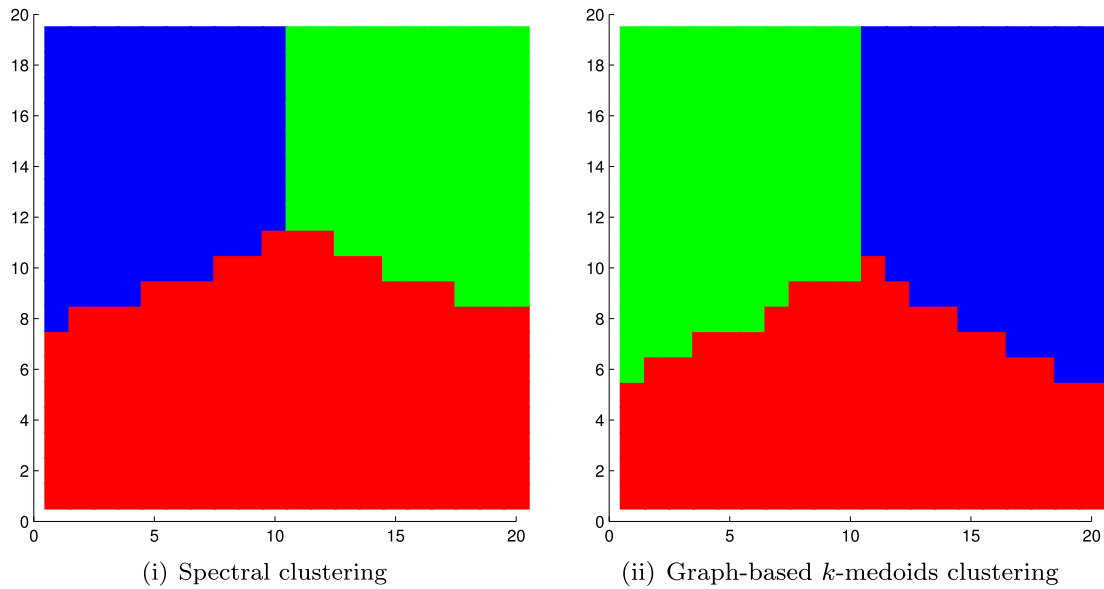


Fig. 11 Clustering result for synthetic data using geodesic distance on $\mathcal{N}(2)$

defined to be the Mahalanobis distance with respect to the Riemannian average of the diffusion tensors at each node. After constructing the graph, the Dijkstra algorithm is then used to find the shortest path between two points. This procedure is nearly the same as that of Algorithm 3, and the details are omitted.

The diffusion tensors located at A and B of Fig. 12 are given by the quadratic form

$$\begin{bmatrix} 1 & 0 \\ 0 & 0.2 \end{bmatrix}.$$

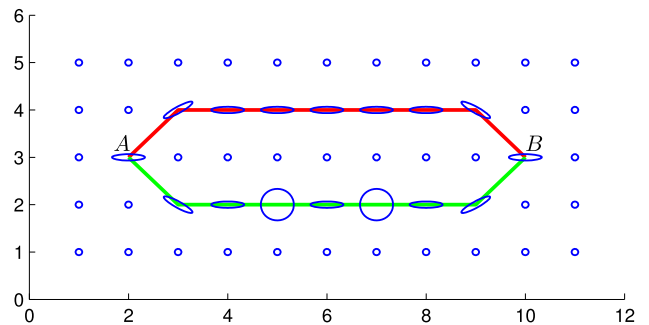


Fig. 12 The synthetic data for fiber tractography

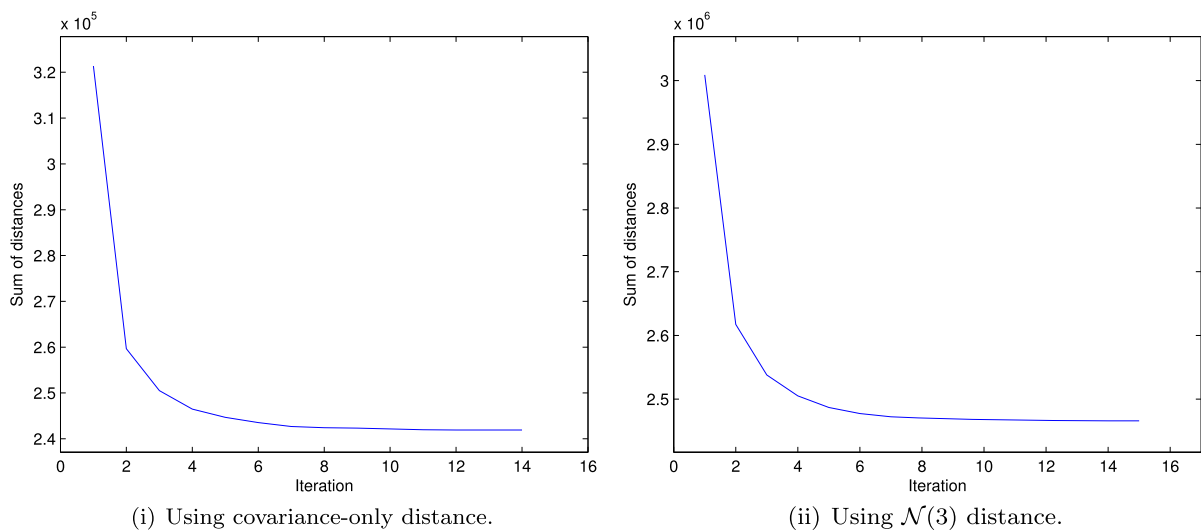


Fig. 13 Convergence of clustering algorithm

The two spherical diffusion tensors on the green path correspond to the identity quadratic form, while the remaining diffusion tensors on the green and red paths correspond to appropriately rotated versions of the quadratic form

$$\begin{bmatrix} 1 & 0 \\ 0 & 0.2 \end{bmatrix}.$$

Using the distance metric of [18] and [21], the lengths of both the red and green paths are identically 8.7854. Using our $\mathcal{N}(2)$ geodesic distance in conjunction with the graph structure and Dijkstra's algorithm described above, the length of the red path is 10.8760, while the length of the green path is 12.8732. For the applications considered in this paper, a convincing argument can be made that it is desirable for the red path to have a shorter length than the green path.

5.3 Experiments with DT-MRI Brain Data

We now present results of our segmentation algorithm applied to DT-MRI images of the human brain. Our brain DT-MRI image consists of a $144 \times 144 \times 85$ lattice, of which 300, 498 points contain valid diffusion tensor data. Each voxel is of size $1.67 \times 1.67 \times 1.7 \text{ mm}^3$. We first construct three graphs from this dataset: one obtained using the covariance-only geodesic distance, another obtained using the $\mathbb{R}^3 \times \mathcal{P}(3)$ distance (i.e. sum of the Euclidean distance of voxels and the $\mathcal{P}(3)$ geodesic distance) and the other obtained using the geodesic distance on $\mathcal{N}(3)$. Of the 26 adjacent voxels, only those that contain valid data are considered when making connections. The resulting graph has 3, 755, 762 edges. The scale factor is set to 3 for the $\mathbb{R}^3 \times \mathcal{P}(3)$ distance and 0.05 for $\mathcal{N}(3)$ distance. A total of 3, 755, 762 $\mathcal{N}(3)$ geodesic distances are computed. Our algorithm is

programmed in C with four threads running on a 2.2 GHz quad core CPU; the entire algorithm takes approximately thirty minutes to complete.

As is well-known, typical brain DT-MRI images contain voxels filled with water, e.g., the cerebral ventricle area. Since at such voxels water diffuses uniformly in all directions, measured geodesic distances within such areas are typically smaller, and clusters tend to form around such water-filled voxels. To prevent such undesirable clustering, we construct a mean diffusivity histogram for the DT-MRI image, and apply Otsu's algorithm [19] to eliminate nodes that have a mean diffusivity over $1.84 \times 10^{-3} \text{ mm}^2/\text{s}$. The water-filled regions can therefore be regarded as an outlier of the dataset.

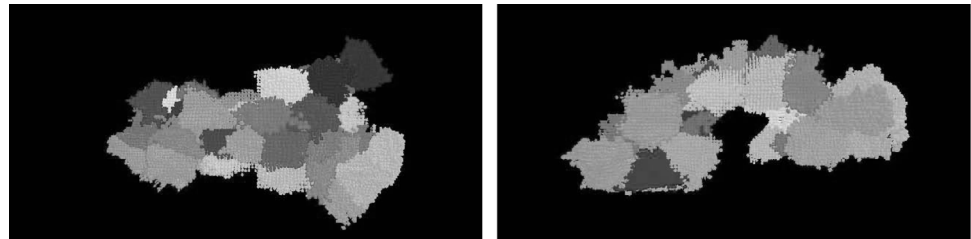
Seeds for clustering is generated by first indexing the DTI data according to position, and then to extract a seed in a manner that is uniform and consistent with this indexing. Uniform seeds are set for the initial medoids, and clustering is performed on our graph structure using three distance metrics. For $100 \sim 500$ clusters, our graph-based k -medoids clustering algorithm converges in less than 21 iterations, with computation times on the order of $40 \sim 120$ seconds. Even for this reasonably large graph, our clustering algorithm demonstrates good performance. With the number of clusters set at 200, Fig. 13 plots the objective function value as a function of the number of iterations; for both the $\mathcal{P}(n)$ and $\mathcal{N}(n)$ metrics our clustering algorithm shows stable convergence behavior even after only 10 iterations.

In order to compare the actual segmentation performance of our proposed distance measure on $\mathcal{N}(3)$ with the covariance only-based geodesic distance and the $\mathbb{R}^3 \times \mathcal{P}(3)$ distance, we focus on the corpus callosum region. To more objectively compare the segmentation results, we manually identify the corpus callosum from the DT-MRI image of

Fig. 14 Segmentation of corpus callosum (number of clusters = 200)



(i) Manually extracted corpus callosum: top (left) and side (right) views



(ii) Segmentation using covariance-only distance



(iii) Segmentation using combined Euclidean and $\mathcal{P}(3)$ distance



(iv) Segmentation using $\mathcal{N}(3)$ distance

Fig. 14(i). Setting the number of clusters at 200, results of our clustering algorithm using the covariance only, the $\mathbb{R}^3 \times \mathcal{P}(3)$ and $\mathcal{N}(3)$ geodesic distance metrics are shown in Figs. 14(ii) ~ (iv). In these cases several clusters contain some part of the corpus callosum.

Denoting the set of all diffusion tensors contained in the corpus callosum region by A , and the set of diffusion tensors belonging to the i -th cluster by C_i , we propose the following measure of clustering performance:

$$\text{ClusteringPerformance} = \frac{n(A)}{n(B)}, \tag{33}$$

where $n(\cdot)$ denotes the number of elements in the set, and $B = \{x \in C_i | A \cap C_i \neq \emptyset, \forall i\}$. This ratio measures the number of points in A to the total number of points in the image that belong to some cluster; a ratio close to the maximal

value of 1 implies that the entire corpus callosum has been segmented into distinct clusters, with no points outside the corpus callosum belonging to any of these clusters.

Varying the number of clusters from 100 to 500, results of clustering performance using this ratio are shown in Fig. 15. In the case of using a covariance-only distance metric with 200 clusters, the corpus callosum is included in 42 clusters, and our performance metric yields 18.74 %; that is, only 18.74 % of the points contained within these 42 clusters lie inside the corpus callosum. Using the $\mathbb{R}^3 \times \mathcal{P}(3)$ metric, the corpus callosum is included in 33 clusters and 24.22 % of the points contained within these 23 clusters lie inside the corpus callosum. Similar results are obtained as the number of clusters is varied. Using our $\mathcal{N}(3)$ geodesic distance metric, the corpus callosum is included in 28 clusters. Moreover, 28.55 % of the points contained within these 26 clusters lie

inside the corpus callosum. Similar results are obtained as the number of clusters is varied.

Clearly none of the segmentation results are entirely satisfactory. Brain DT-MRI segmentation, however, is a highly involved, multi-faceted problem that simultaneously draws upon a collection of methods and techniques, of which geodesic distances are but one (albeit important) tool. The experiment results obtained above offer some justification to our claim that the $\mathcal{N}(3)$ distance metric offers several advantages over using the traditional covariance-only metric.

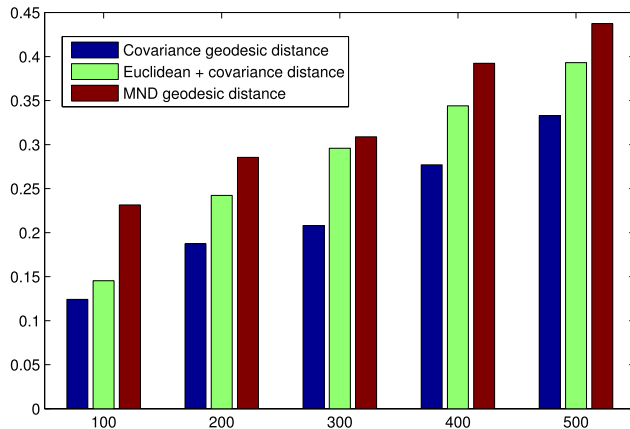
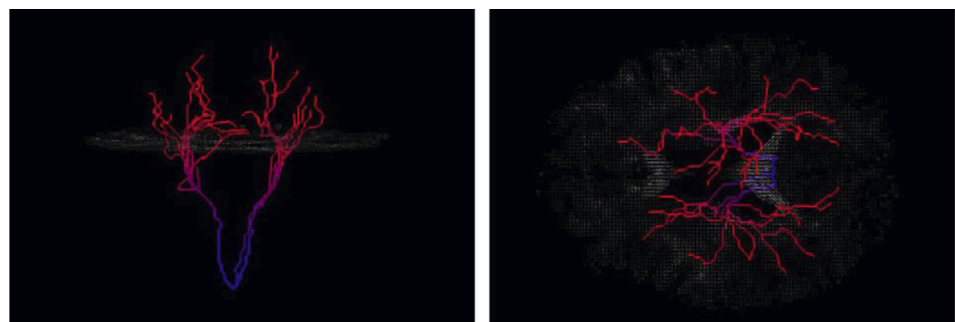
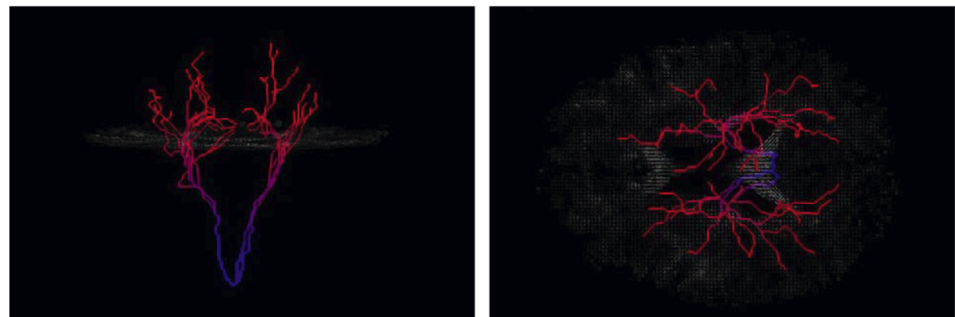


Fig. 15 Measure of clustering performance

Fig. 16 Geodesic tractography of brain DT-MRI



(i) Fiber tractography of brain DT-MRI using Mahalanobis distance: front (left) and top (right) views



(ii) Fiber tractography of brain DT-MRI using $\mathcal{N}(3)$ geodesic distance: front (left) and top (right) views

5.4 Geodesic Tractography Experiments with Brain Data

We now perform geodesic-based fiber tractography experiments with the previous brain DT-MRI dataset. Here we find shortest paths from the spinal cord to points in the brain white matter using both the $\mathcal{N}(3)$ and Mahalanobis metrics. For these experiments the DT-MRI images are modified to emphasize the principal eigenvectors, using the tensor sharpening method introduced in [7] and [21].

From the results shown in Fig. 16, it can be seen that the results obtained for both metrics are quite similar; there are only small differences in the white matter fiber tracts. Since the $\mathcal{N}(3)$ distance essentially adds a $\mathcal{P}(3)$ distance term to the integrand of the Mahalanobis distance, our results would seem to imply that the contributions of the $\mathcal{P}(3)$ variations are quite small overall for this dataset.

6 Conclusion

In this paper we have proposed the idea that, for the segmentation of DT-MRI images, it is better to take into account both the mean and covariance of the multivariate normal distribution attached to each spatial voxel. Practical implementations of this approach require, as a basic computational element, the computation of geodesic distances between two multivariate normal distributions with different

means and covariances. We have developed an efficient, numerically robust algorithm for calculating the required minimal geodesics. Our algorithm can be viewed as a geometric generalization of the classical shooting method for solving two-point boundary value problems. In particular, the notion of parallel transport (with respect to the Riemannian connection) plays a fundamental role in transporting velocity errors at the final point back to the initial point, in a coordinate-invariant way that respects the geometry of the underlying manifold of multivariate normal distributions.

As a secondary minor contribution, we also develop a graph-based clustering algorithm that can be viewed as a geometric extension of the k -medoids algorithm using greedy search. Experiments with both synthetic data and real brain DT-MRI images confirm that, independent of the chosen clustering algorithm—in our experiments we consider both our graph-based k -medoids algorithm and the more conventional spectral clustering method—using the general metric on $\mathcal{N}(n)$ leads to qualitatively superior segmentation results than using a covariance-only based distance metric, or weighted Mahalanobis distance metrics that combine diffusion tensor and spatial voxel distances in an ad hoc way.

Acknowledgements This research was supported in part by the Center for Advanced Intelligent Manipulation, the Biomimetic Robotics Research Center, the BK21+ program at SNU-MAE, and SNU-IAMD.

Appendix: Triangle Inequality Counterexample

In [1] it is claimed that the distance metrics (2) and (3) satisfy all the metric axioms including the triangle inequality. The following simple counterexample shows that this is not the case. Consider the following three normal distributions a, b and c on $\mathcal{N}(1)$:

$$\begin{aligned} \mu_a &= 0, & \sigma_a &= 1, \\ \mu_b &= 1, & \sigma_b &= 1, \\ \mu_c &= 2, & \sigma_c &= 0.01. \end{aligned} \tag{34}$$

The distances calculated using the proposed metrics are given in Table 2.

Both metrics violate the triangle inequality, i.e., $\text{dist}(a, b) + \text{dist}(b, c) \not\leq \text{dist}(c, a)$.

Table 2 Distances between a, b and c

Distance metric	$\text{dist}(a, b)$	$\text{dist}(b, c)$	$\text{dist}(a, c)$
$d_{JR}(\cdot, \cdot)$	1.4142	13.3062	23.3561
$d_{BR}(\cdot, \cdot)$	1	4.66991	6.0834

References

1. Abou-Moustafa, K.T., Ferrie, F.P.: A note on metric properties for some divergence measures: the Gaussian case. *J. Mach. Learn. Res.* **25**, 1–15 (2012)
2. Alexander, D., Gee, J., Bajcsy, R.: Similarity measures for matching diffusion tensor images. In: *British Machine Vision Conference*, vol. 99, pp. 93–102 (1999)
3. Arsigny, V., et al.: Log-Euclidean metrics for fast and simple calculus on diffusion tensors. *Magn. Reson. Med.* **56**, 411–421 (2006)
4. Basser, P.J., Mattiello, J., LeBihan, D.: MR diffusion tensor spectroscopy and imaging. *J. Biophys.* **66**, 259–267 (1994)
5. Calvo, M., Oller, J.: A distance between multivariate normal distributions based in an embedding into a Siegel group. *J. Multivar. Anal.* **35**, 223–242 (1990)
6. Coulon, O., Alexander, D.C., Arridge, S.: Diffusion tensor magnetic resonance image regularization. *Med. Image Anal.* **8**(1), 47–68 (2004)
7. Descoteaux, M., Lenglet, C., Deriche, R.: Diffusion tensor sharpening improves white matter tractography. *Proc. SPIE* **6512**, 65121J (2007)
8. Eriksen, P.S.: Geodesics connected with the Fisher metric on the multivariate normal manifold. In: *Proc. GST Workshop*, Lancaster, pp. 225–229 (1987)
9. Fletcher, P.T., et al.: Riemannian geometry for the statistical analysis of diffusion tensor data. *Signal Process.* **87**(2), 250–262 (2007)
10. Imai, T., Takaesu, A., Wakayama, M.: Remarks on geodesics for multivariate normal models. *Surv. Math. Ind.* **B**(6), 125–130 (2011)
11. Jbadi, S., Bellec, P., Toro, R., Daunizeau, J., Pélérini-Issac, M., Benali, H.: Accurate anisotropic fast marching for diffusion-based geodesic tractography. *Int. J. Biomed. Imaging* **2008**, 320195 (2008)
12. Kaufman, L., Rousseeuw, P.J.: Clustering by means of medoids. In: Dodge, Y. (ed.) *Statistical Data Analysis Based on the L_1 Norm and Related Methods*, pp. 405–416. North-Holland, Amsterdam (1987)
13. Lenglet, C., Rousson, M., Deriche, R., Faugeras, O.: Statistics on the manifold of multivariate normal distributions: theory and application to diffusion tensor MRI processing. *J. Math. Imaging Vis.* **25**, 423–444 (2006)
14. Lovric, M., Min-Oo, M., Ruh, E.A.: Multivariate normal distributions parametrized as a Riemannian symmetric space. *J. Multivar. Anal.* **74**(1), 36–48 (2000)
15. Maitra, R., Peterson, A.D., Ghosh, A.P.: A systematic evaluation of different methods for initializing the k -means clustering algorithm. In: *IEEE Trans. Knowledge and Data Engineering* (2010)
16. Moakher, M.: A differential geometric approach to the geometric mean of symmetric positive-definite matrices. *SIAM J. Matrix Anal. Appl.* **26**, 735–747 (2005)
17. Moakher, M., Batchelor, P.G.: Symmetric positive-definite matrices: from geometry to applications and visualization. In: *Visualization and Processing of Tensor Fields*, pp. 285–298 (2006)
18. O’Donnell, L., Haker, S., Westin, C.-F.: New approaches to estimation of white matter connectivity in diffusion tensor MRI: elliptic PDEs and geodesics in a tensor-warped space. In: *Proc. Med. Image Comput. Comp. Assisted Intervention*, vol. 2488, pp. 459–466 (2002)
19. Otsu, N.: A threshold selection method from gray-level histogram. *IEEE Trans. Syst. Man Cybern.* **SMC-9**(1), 62–66 (1979)
20. Pennec, X., Fillard, P., Ayache, N.: A Riemannian framework for tensor computing. *Int. J. Comput. Vis.* **66**(1), 41–46 (2006)
21. Sepasian, N., ten Thije Boonkkamp, J.H.M., Ter Haar Romeny, B.M., Vilanova, A.: Multivalued geodesic ray-tracing for computing brain connections using diffusion tensor imaging. *SIAM J. Imaging Sci.* **5**, 483–504 (2012)

22. Skovgaard, L.T.: A Riemannian geometry of the multivariate normal model. *Scand. J. Stat.* **11**, 211–233 (1984)
23. Smith, S.T.: Covariance, subspace, and intrinsic Cramér–Rao bounds. *IEEE Trans. Signal Process.* **53**(5), 1610–1630 (2005)
24. Tsuda, K., Akaho, S., Asai, K.: The em algorithm for kernel matrix completion with auxiliary data. *J. Mach. Learn. Res.* **4**, 67–81 (2003)
25. Wang, Z., Vemuri, B.: DTI segmentation using an information theoretic tensor dissimilarity measure. *IEEE Trans. Med. Imaging* **24**(10), 1267–1277 (2005)
26. Wiegell, M., Tuch, D., Larson, H., Wedeen, V.: Automatic segmentation of thalamic nuclei from diffusion tensor magnetic resonance imaging. *NeuroImage* **19**, 391–402 (2003)
27. Ziyang, U., Tuch, D., Westin, C.F.: Segmentation of thalamic nuclei from DTI using spectral clustering. In: *Proc. Med. Image Comput. Comp. Assisted Intervention*, pp. 807–814 (2006)



Minyeon Han received the B.S. and Ph.D. degrees in mechanical engineering from Seoul National University, Seoul, Korea, in 2003 and 2013, respectively. He is currently a postdoctoral researcher in the Robotics Laboratory at Seoul National University.



F.C. Park received the B.S. degree in electrical engineering from MIT in 1985, and the Ph.D. degree in applied mathematics from Harvard University in 1991. From 1991 to 1994, he was an assistant professor of mechanical engineering at the University of California, Irvine. Since 1995 he has been a professor at the School of Mechanical and Aerospace Engineering at Seoul National University, Seoul, Korea. From 2009 to 2012 he was also an adjunct professor in the Department of Interactive Computing at Georgia Tech. He was a 2007–2008 IEEE Robotics and Automation Society Distinguished Lecturer, served as Secretary of the IEEE Robotics and Automation Society from 2007–2008 and 2012–2013, and as an editorial board member for the Springer Handbook of Robotics, STAR Tracts in Robotics, and Robotica. He is editor-in-chief of the IEEE Transactions on Robotics. His research interests are in robotics, vision, and related areas of applied mathematics.