DT-REFinD: Diffusion Tensor Registration With Exact Finite-Strain Differential

B.T. Thomas Yeo*, Tom Vercauteren, Pierre Fillard, Jean-Marc Peyrat, Xavier Pennec, Polina Golland, Nicholas Ayache, and Olivier Clatz

***Abstract*—In this paper, we propose the DT-REFinD algorithm for the diffeomorphic nonlinear registration of diffusion tensor images. Unlike scalar images, deforming tensor images requires choosing both a reorientation strategy and an interpolation scheme. Current diffusion tensor registration algorithms that use full tensor information face difficulties in computing the differential of the tensor reorientation strategy and consequently, these methods often approximate the gradient of the objective function. In the case of the finite-strain (FS) reorientation strategy, we borrow results from the pose estimation literature in computer vision to derive an analytical gradient of the registration objective function. By utilizing the closed-form gradient and the velocity field representation of one parameter subgroups of diffeomorphisms, the resulting registration algorithm is diffeomorphic and fast. We contrast the algorithm with a traditional FS alternative that ignores the reorientation in the gradient computation. We show that the exact gradient leads to significantly better registration at the cost of computation time. Independently of the choice of Euclidean or Log-Euclidean interpolation and sum of squared differences dissimilarity measure, the exact gradient achieves better alignment over an entire spectrum of deformation penalties. Alignment quality is assessed with a battery of metrics including tensor overlap, fractional anisotropy, inverse consistency and closeness to synthetic warps. The improvements persist even when a different reorientation scheme, preservation of principal directions, is used to apply the final deformations.**

***Index Terms*—Diffeomorphisms, diffusion tensor imaging, finite-strain (FS), finite-strain differential, preservation of principal directions, registration, tensor reorientation.**

Manuscript received April 20, 2009; revised June 09, 2009. First published June 23, 2009; current version published November 25, 2009. This work was supported by the INRIA Associated Teams Program *Compu-Tumor* (http://www-sop.inria.fr/asclepios/projects/boston/), by the NAMIC under Grant NIH NIBIB NAMIC U54-EB005149, and by the NAC under Grant NIH NCRR NAC P41-RR13218). The work of T. Yeo was supported by the Agency for Science, Technology, and Research, Singapore. *Asterisk indicates corresponding author.*

*B.T. T. Yeo is with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: ythomas@csail.mit.edu).

P. Golland is with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: polina@csail.mit.edu).

T. Vercauteren is with the Mauna Kea Technologies, 75010 Paris, France (e-mail: tom.vercauteren@maunakeatech.com).

P. Fillard, J.-M. Peyrat, X. Pennec, N. Ayache, and O. Clatz are with the Asclepios Group, INRIA, 06902 Sophia Antipolis, France (e-mail: pierre.fillard@sophia.inria.fr; jean-marc.peyrat@sophia.inria.fr; xavier.pennec@sophia.inria.fr; nicholas.ayache@sophia.inria.fr; olivier.clatz@sophia.inria.fr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMI.2009.2025654

I. Introduction

DIFFUSION tensor imaging (DTI) noninvasively measures the diffusion of water in *in vivo* biological tissues [8]. The diffusion is anisotropic in tissues such as cerebral white matter. DTI is therefore a powerful imaging modality for studying white matter structures in the brain. The rate and anisotropy of diffusion at each voxel of a diffusion tensor image is summarized by an order 2 symmetric positive definite tensor, i.e., a positive definite 3×3 matrix. This is in contrast to scalar values in traditional magnetic resonance images. The eigenvectors of the tensor correspond to the three principal directions of diffusion while the eigenvalues measure the rate of diffusion in these directions.

To study the variability or similarity of white matter structures across a population or to track white matter changes of a single subject through time, registration is necessary to establish correspondences across different diffusion tensor (DT) images. Registration can be simplistically thought of as warping one image to match another. For scalar images, such a warp can be defined by a deformation field and an interpolation scheme. For DT images however, one also needs to define a tensor reorientation scheme. Reorientation of tensors is necessary to warp a tensor image consistently with the anatomy [3]. There are two commonly used reorientation strategies: the finite-strain (FS) reorientation and the preservation of principal directions (PPD) reorientation. In this paper, we derive an exact differential of FS reorientation strategy and show that incorporating the exact differential into the registration algorithm leads to significantly better registration than the common practice of ignoring the reorientation when computing the gradient [3]. Their empirical performance is similar [23], [34], [49].

Many DTI registration algorithms have been proposed [2], [15], [24], [28], [31], [36], [49], [50]. Because the reorientation strategies greatly complicate the computation of the gradient of the registration objective function [49], many of these registration techniques use scalar values or features that are invariant to image transformations. This includes the use of fractional anisotropy [31] and fibers extracted through tractography [50]. Leemans *et al.* [28] use mutual information to affinely align the diffusion weighted images from which the DT images are estimated. Nonlinear fluid registration of DT images based on information theoretic measures has since been introduced [16], [39].

Instead of using deformation invariant features, Alexander and Gee [2] perform elastic registration of tensor images by reorienting the tensors after each iteration using PPD reorientation. The reorientation is not taken into account when computing the gradient of the objective function. Cao *et al.* [15] propose a diffeomorphic registration of tensor images using PPD reorientation. The diffeomorphism is parameterized by a nonstationary velocity field under the large deformation diffeomorphic metric mapping (LDDMM) framework [11]. An exact gradient of the PPD reorientation is computed by a clever analytical reformulation of the PPD reorientation strategy. In this paper, we complement the work in [15] by computing the exact gradient of the FS reorientation.

For a general transformation, such as defined by *B*-splines or nonparametric free form displacement field, the FS reorientation [3] is defined through the rotation component of the deformation field. This rotation is estimated by the polar decomposition of the Jacobian of the deformation field using principles of continuum mechanics. The rotation produced by the polar decomposition of the Jacobian is the closest orthogonal operator to the Jacobian under any unitary invariant norm [26]. However, the polar decomposition requires computing the square root of a positive definite matrix, which replaces the eigenvalues of the original matrix with their square roots. The dependence of the rotation matrix on the Jacobian of deformation is therefore complex and the gradient of any objective function that involves reorientation is hard to compute.

Zhang *et al.* [48], [49] propose and demonstrate a piecewise local affine registration algorithm to register tensor images using FS reorientation. The tensor image is divided into uniform regions and the optimal affine transformation is then estimated for each such region. The rotation component of the deformation need *not* be estimated as a separate step. Instead, since rotation is already explicitly optimized in the affine registration, the gradient due to FS reorientation can be easily computed. These piecewise affine transformations are fused together to generate a smooth warp field. The algorithm is iterated in a multiscale fashion with smaller uniform regions. Unfortunately, it is unclear how much of the optimality is lost in fusing these locally optimal piecewise affine transformations.

In this paper, we borrow results from the pose estimation literature in computer vision [20] to compute the analytical differential of the rotation matrix with respect to the Jacobian of the displacement field. We propose a diffeomorphic DTI registration algorithm DT-REFinD, which extends the recently introduced diffeomorphic Demons registration of scalar images [42] to registration of tensor images. The availability of the exact analytical gradient allows us to utilize the Gauss–Newton method for optimization. Implemented within the Insight Toolkit (ITK) framework, registration of a pair of $128 \times 128 \times 60$ diffusion tensor volumes takes 15 min on a Xeon 3.2 GHz single processor machine. This is comparable to the nonlinear registration of scalar images whose runtime might range from a couple of minutes to hours. DT-REFinD has been incorporated into the freely available MedINRIA software.[1]

The diffeomorphic Demons registration algorithm [42] is an extension of the popular Demons algorithm [37]. It guarantees that the transformation is diffeomorphic. The space of transformations is parameterized by a composition of deformations, each of which is parametrized by a stationary velocity field. Such a representation is similar to that used by the large deformation diffeomorphic metric mapping (LDDMM) framework [11], [38]. However, unlike LDDMM, the diffeomorphic Demons algorithm does not seek a geodesic of the Lie group of diffeomorphism. At each iteration, the diffeomorphic Demons algorithm seeks the best diffeomorphism to be composed with the current transformation. Restricting each deformation update to belong to a one parameter subgroup of diffeomorphism results in a faster algorithm than the typical algorithm based on the LDDMM framework or algorithms that parameterize the entire diffeomorphic transformation by a stationary velocity field [7], [25].

In addition to DT-REFinD, we also propose a simpler and faster algorithm that ignores the reorientation during the gradient computation. Instead, reorientation is performed after each iteration. This faster algorithm is therefore a diffeomorphic variant of the method proposed by Alexander and Gee [2] with Gauss–Newton optimization. We compare the two algorithms and show that using the exact gradient results in significantly better registration at the cost of computation time.

While many methods for interpolating and comparing tensor images exist [27], [32], we use Euclidean interpolation and sum-of-squares difference (EUC-SSD) [2], [3], [49], as well as Log-Euclidean interpolation and sum-of-squares difference (LOG-SSD) [6], [21]. Regardless of the choice of interpolation and dissimilarity metric, we find the exact gradient achieves better alignment over an entire range of deformation regularization. Alignment quality is assessed with a set of seventeen different metrics including tensor overlap, fractional anisotropy and inverse consistency of the warps. We also find that the exact gradient method recovers synthetically generated warps with higher accuracy. Finally, we show that the improvements persist even when PPD is used to apply the final deformations.

We emphasize that there is no theoretical guarantee that using the true gradient will lead to a better solution. After all, the registration problem is nonconvex and any solution we find is a local optimum. In practice however, the experiments show that taking reorientation into account does significantly improve the registration results. We believe that the reorientation provides an additional constraint. The registration algorithm cannot arbitrarily pull in a faraway region for matching because this induces the reorientation of tensors in other regions (*cf.* the famous "C" example in large deformation fluid registration [18]). This additional constraint acts as a further regularization, leading to a better solution.

This paper extends a previously presented conference article [47] and contains detailed derivations, experiments and discussions left out in the conference version. The paper is organized as follows. The Section II describes the computation of the FS differential. We then present an overview of the diffeomorphic Demons algorithm in Section III and discuss certain conventions and numerical limitations of representing diffeomorphic transformations. We extend the diffeomorphic Demons to tensor images in Section IV using the exact FS differential. We also propose a simpler and faster algorithm that ignores the reorientation during the gradient computation. In Section V, we compare the

[1] MedINRIA can be downloaded at http://www-sop.inria.fr/asclepios/software/MedINRIA.

two algorithms on a set of 10 DT brain images. Further discussion is provided in Section VI.

To summarize, our contributions are as follows.

1) We derive the exact FS differential.
2) We incorporate the FS differential into a fast diffeomorphic DT image registration algorithm. We emphasize that the FS differential is useful, even if one were to use a different registration scheme with a different model of deformation or dissimilarity metric.
3) We demonstrate that the use of the exact gradient leads to better registration. In particular, we show that using the exact gradient leads to better tensor alignment over an entire range of deformation, regardless of whether we use LOG-SSD or EUC-SSD in the objective function. We also show that the exact gradient recovers synthetically generated deformation fields significantly better than when using an approximate gradient that ignores reorientation.
4) Our implementation allows for Euclidean interpolation and EUC-SSD metric, as well as Log-Euclidean interpolation and LOG-SSD metric.

II. Finite-Strain Differential

Deforming a tensor image by a transformation s involves tensor interpolation followed by tensor reorientation [3]. To compute a deformed tensor at a voxel n, one first interpolates the tensor to get the interpolated tensor $T(n)$. Interpolation schemes include Euclidean interpolation [3], Log-Euclidean interpolation [6], affine-invariant framework [10], [22], [29], [30], [32], Geodesic–Loxodromes [27], or other methods. In this work, we focus on Euclidean and Log-Euclidean interpolation since they are commonly used and computationally simple. The FS differential we compute in this section characterizes tensor reorientation. The following discussion is therefore independent of the interpolation strategy.

Suppose the transformation s maps a point p to the point $s(p)$. Let $u \triangleq s - I$ be the displacement field associated with the transformation s. Then

$$s(p) = p + u. \tag{1}$$

Similarly, we denote $s \triangleq I + u$. Note that even for parametric representation of transformations, such as splines, one can always derive the equivalent displacement field representation.

According to the FS tensor reorientation strategy [3] for nonlinear deformation, one first computes the rotation component of the deformation at the nth voxel

$$R(n) = (J(n)J(n)^T)^{-1/2}J(n) \tag{2}$$

where $J(n)$ is the Jacobian of the spatial transformation s at the voxel n

$$\begin{aligned} J(n) &= \begin{pmatrix} \frac{\partial s_x(n)}{\partial x} & \frac{\partial s_x(n)}{\partial y} & \frac{\partial s_x(n)}{\partial z} \\ \frac{\partial s_y(n)}{\partial x} & \frac{\partial s_y(n)}{\partial y} & \frac{\partial s_y(n)}{\partial z} \\ \frac{\partial s_z(n)}{\partial x} & \frac{\partial s_z(n)}{\partial y} & \frac{\partial s_z(n)}{\partial z} \end{pmatrix} \\ &= I + \begin{pmatrix} \frac{\partial u_x(n)}{\partial x} & \frac{\partial u_x(n)}{\partial y} & \frac{\partial u_x(n)}{\partial z} \\ \frac{\partial u_y(n)}{\partial x} & \frac{\partial u_y(n)}{\partial y} & \frac{\partial u_y(n)}{\partial z} \\ \frac{\partial u_z(n)}{\partial x} & \frac{\partial u_z(n)}{\partial y} & \frac{\partial u_z(n)}{\partial z} \end{pmatrix} \end{aligned} \tag{3}$$

where u_x, u_y, u_z are the components of the displacement field in the x, y and z directions. $R(n)$ is called a polar decomposition of the matrix $J(n)$ and is therefore a function of the displacement field u in the neighborhood of n. Under the identity transformation, i.e., zero displacements, $J(n) = I$ and $R(n) = I$. Because of the matrix inverse in (2), to maintain numerical stability of the computations, the invertibility of the deformation (corresponding to $\det(J) > 0$) is important.

The interpolated tensor $T(n)$ is then reoriented, resulting in the final tensor $T'(n)$

$$T'(n) = R^T(n)T(n)R(n). \tag{4}$$

For registration based on the FS strategy, it is therefore necessary to compute the differential of rotation R with respect to the transformation s. Using chain rule, this reduces to computing the differential of rotation R with respect to the Jacobian J. Let dJ be the infinitesimal change in the Jacobian J. Then, as shown in Appendix A, the infinitesimal change in the rotation matrix R is computed as follows:

$$dR = -R\left[R^T(\mathrm{tr}(S)I - S)^{-1}R\sum_i (R^T)_i \times (dJ^T)_i\right]^{\oplus} \tag{5}$$

where $S = (JJ^T)^{1/2}$, $\times$ denotes the 3-D vector cross product, $(\cdot)_i$ denotes the ith column of $(\cdot)$ and $\oplus$ is the operator defined as

$$m^{\oplus} = \begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix}^{\oplus} \triangleq \begin{pmatrix} 0 & -m_3 & m_2 \\ m_3 & 0 & -m_1 \\ -m_2 & m_1 & 0 \end{pmatrix}. \tag{6}$$

This skew-symmetric operator is actually the matrix representation of cross-product, so that for two vectors a and b, $a \times b = a^{\oplus}b$. It is introduced to simplify the notation in the already complicated (5).

The detailed derivation, based on the pose estimation solution [20] is presented in Appendix A. Let J_{ij} be (i,j)th component of J. Equation (5) tells us the variation of the rotation R in terms of the components of the Jacobian J. In particular, $\partial R/\partial J_{ij}$ is computed by setting the matrix dJ in (5) to 0, except for $(dJ)_{ij}$ which is set to 1.

III. Background on Diffeomorphic Registration

In this section, we briefly review the diffeomorphic extension [42] of Thirion's Demons algorithm [37]. We also discuss numerical issues related to representing diffeomorphism by velocity fields and optimization methods we use in this paper.

A. Diffeomorphic Demons for Scalar Images

We consider the modified Demons objective function [14] for registering a moving scalar image M to a fixed scalar image F

$$E(c,s) = \|\Sigma^{-1}(F - M\circ c)\|^2 + \frac{1}{\sigma_x^2}\mathrm{dist}(s,c) + \frac{1}{\sigma_T}\mathrm{Reg}(s) \tag{7}$$

where c is the dense spatial transformation to be optimized, s is an auxiliary spatial transformation, $\circ$ denotes composition and $\|\cdot\|$ denotes the L_2-norm of a vector (or vector field, depending on the context). We can think of the fixed image F and warped moving image $M \circ c$ as 1-D vectors of length N voxels. Σ is a $N \times N$ diagonal matrix that defines the variability observed at a

```
Data: A fixed image $F$ and moving image $M$.
Result: Transformation $c$ so that $M \circ c$ is "close" to $F$.
Step 1. Choose a starting spatial transformation $s^{(0)}$ (represented by a displacement field)
Step 2. repeat
    i. Given $s^{(i)}$, compute a stationary velocity field update $v^{(i+1)}$ by minimizing the first two terms of Eq. (7):
       $$v^{(i+1)} = \underset{v}{\operatorname{argmin}} \|\Sigma^{-1}(F - M \circ s^{(i)} \circ \exp(v))\|^2 + \frac{1}{\sigma_x^2}\operatorname{dist}(s^{(i)}, s^{(i)} \circ \exp(v)),$$
       where $v$ is an element of the Lie algebra $g$ associated with the Lie group.
    ii. If a fluid regularization is used, let $v^{(i+1)} \leftarrow K_{\text{fluid}} \star v^{(i+1)}$. The convolution kernel is typically Gaussian.
    iii. Let $c^{(i+1)} \leftarrow s^{(i)} \circ \exp(v^{(i+1)})$
    iv. If a diffusion regularization is used, let $s^{(i+1)} \leftarrow I + K_{\text{diff}} \star (c^{(i+1)} - I)$; else let $s^{(i+1)} \leftarrow c^{(i+1)}$. The convolution kernel is also typically Gaussian.
until convergence ;
```

Algorithm 1. Diffeomorphic Demons Algorithm

particular voxel. σ_x and σ_s are parameters of the cost function. The instantiation of these parameters are further discussed in Section V-B.

This formulation enables a fast and simple optimization that alternately minimizes the first two terms and the last two terms of (7). Typically, $\operatorname{dist}(c, s) = \|c - s\|^2$, encouraging c and s to be close and $\operatorname{Reg}(s) = \|\nabla(s - \operatorname{Id})\|^2$, encouraging s to be smooth. The regularization can also be modified to handle a fluid model. We note that $\operatorname{dist}(c, s)$ and $\operatorname{Reg}(s)$ can together be interpreted probabilistically as a hierarchical prior on the deformation c [46].

For the classical Demons algorithm and its variants, the objective function is optimized over the complete space of nonparametric spatial transformations [14], [35], [37], [43], typically represented as displacement fields. Unfortunately, the resulting deformation might not be diffeomorphic. Instead, Vercauteren *et al.* [42] optimize over compositions of diffeomorphic deformations, each of which is parametrized by a stationary velocity field. At each iteration, the diffeomorphic Demons algorithm seeks the best diffeomorphism parameterized by the stationary velocity v, to be composed with the current transformation.

In this case, the velocity field v is an element of the Lie algebra g and $\exp(v)$ is the diffeomorphism associated with v. The operator $\exp(\cdot)$ is the group exponential relating the Lie Group G to its associated Lie algebra g. More formally, let $\Phi_{tv}(x_0)$ be the solution at time t of the following stationary ordinary differential equation (ODE):

$$\frac{dx}{dt} = v(x) \text{ with initial condition } x(0) = x_0. \tag{8}$$

We define

$$\exp(v)(x) \triangleq \Phi_{1v}(x) \triangleq w(x). \tag{9}$$

An image $M \circ \exp(v)$ is therefore a deformed version of image M obtained by transforming the coordinate system of M by $\exp(v)$: a point x in the deformed coordinate system corresponds to a point $\Phi_{1v}(x)$ in the old coordinate system.

The above formulation of the Demons objective function facilitates a fast iterative two-step optimization. We summarize the diffeomorphic Demons algorithm [42] in Algorithm 1 (see algorithm at the top of the page). Steps 2(ii) to 2(iv) essentially optimize the last two terms of (7). We refer the reader to [13], [14] for a detailed discussion of using convolution kernels to achieve elastic and fluid regularization. We also note that the above formulation is quite general, and in fact the diffeomorphic Demons algorithm can be extended to non-Euclidean domains, such as the sphere [46].

B. Numerical Details in Velocity Field Representations

While v and $\Phi_{1v}(x) = \exp(v)(x) = w(x)$ are technically defined on the entire continuous image domain, in practice, v and w are represented by vector fields on a discrete grid of image points, such as voxels [37], [42] or control points [7], [11]. From the theories of ODEs, we know that the integral curves $w = \exp(v)$ (or trajectories) of a velocity field $v(x, t)$ exist and are unique if $v(x, t)$ is Lipschitz continuous in x and continuous in t [12]. Uniqueness means that the trajectories do not cross, implying that the deformation is invertible. Furthermore, we know from the theories of ODEs that a C^r continuous velocity field v produces a C^r continuous deformation field $\Phi_{tv}(x)$. Therefore, a sufficiently smooth velocity field results in a diffeomorphic transformation.

Since the velocity field v is stationary in the case of the one parameter subgroup of diffeomorphism [5], v is clearly continuous (and in fact C^∞) in t. A smooth interpolation of v is continuous in the spatial domain and is Lipschitz continuous if we consider a compact domain, which holds since we only consider images that are closed and bounded.

To compute the final deformation of an image, we have to estimate $\exp(v)$ at least at the set of image grid points. For example, we can compute $\exp(v)$ by numerically integrating the smoothly interpolated velocity field v with Euler integration. In this case, the estimate becomes arbitrarily close to the true $\exp(v)$ as the number of integration time steps increases. With a sufficiently large number of integration steps, we expect the estimate to be invertible and the resulting transformation to be diffeomorphic.

The parameterization of diffeomorphism by stationary velocity field is made popular by the use of the fast "scaling and squaring" approach to computing $\exp(v)$ [5]. Instead of Euler integration, the "scaling and squaring" method works by multiple composition of displacement fields

$$\begin{aligned}
\Phi_{(1/2^N)v}(x) &= x + \frac{1}{2^N}v(x) \\
\Phi_{(1/2^{N-1})v}(x) &= \Phi_{(1/2^N)v} \circ \Phi_{(1/2^N)v}(x) \\
&\vdots \\
\Phi_{1v}(x) &= \Phi_{(1/2)v} \circ \Phi_{(1/2)v}(x).
\end{aligned} \tag{10}$$

While this method is correct in the continuous case, in the discrete case, composition of the displacement fields requires interpolation of displacement fields, introducing errors in the process. In particular, suppose $\Phi_{t_0 v}(x)$ and $\Phi_{2t_0 v}(x)$ are the true trajectories found by performing an accurate Euler integration up to time t_0 and $2t_0$ respectively. Then, there does not exist a trivial interpolation scheme that guarantees $\Phi_{2t_0 v}(x) = \Phi_{t_0 v} \circ \Phi_{t_0 v}(x)$. In practice however, it is widely reported that "scaling and squaring" tends to preserve invertibility even with rather large deformation [5], [7], [42]. In this work, we employ trilinear interpolation because it is fast. We find that in practice, the transformation is indeed diffeomorphic. Technically speaking, since we use linear interpolation for the displacement field, the transformation is only homeomorphic rather than diffeomorphic. However, we will follow the convention of [5], [7], [42] which call the resulting homeomorphic transformations diffeomorphisms.

C. Gauss–Newton Nonlinear Least-Squares Optimization

We now focus on the optimization of step 2(i) of the diffeomorphic Demons algorithm. We choose $\mathrm{dist}(s, s \circ \exp(v)) = \|s^{-1} \circ s \circ \exp(v)\|^2 = \|\exp(v)\|^2 = \|u\|^2$, where $u = \exp(v) - I$. We "subtract" the identity transformation from the resulting deformation field so that the identity transformation carries no penalty. The objective function in step 2(i) can then be written in a nonlinear least-squares form

$$E_s(v) = \left\| \begin{bmatrix} \Sigma^{-1}(F - M \circ s \circ \exp(v)) \\ \frac{1}{\sigma_x} \exp(v) \end{bmatrix} \right\|^2 \quad (11)$$

$$= \left\| \begin{bmatrix} \varphi^1(s \circ \exp(v)) \\ \varphi^2(v) \end{bmatrix} \right\|^2 \quad (12)$$

$$= \|\varphi_s(v)\|^2 \quad (13)$$

where we define $\varphi^1(s) = \Sigma^{-1}(F - M \circ s)$ and $\varphi^2(v) = (1/\sigma_x)\exp(v)$. Using Taylor series expansion around $v = 0$, we can write (13) as

$$E_s(v) = \left\| \varphi_s(0) + \begin{bmatrix} D_s^{\varphi^1}(0) \\ D_s^{\varphi^2}(0) \end{bmatrix} v + O(\|v\|^2) \right\|^2. \quad (14)$$

To interpret (14) for 3-D images with N voxels, let v be a $3N \times 1$ vector of components: $\{v_x(1), v_y(1), v_z(1), \ldots, v_x(N), v_y(N), v_z(N)\}$. Then $D_s^{\varphi^1}(0)$ is a $N \times 3N$ block diagonal matrix, whose nth block corresponds to a 1×3 matrix

$$\left[D_s^{\varphi^1}(0)\right]_n = \left.\frac{\partial \varphi_n^1(s \circ \exp(v))}{\partial v(n)}\right|_{v(n)=0} \quad (15)$$

$$= \left.\frac{\partial \varphi_n^1(s \circ \exp(v))}{\partial \exp(v)(n)} \cdot \frac{\partial \exp(v)(n)}{\partial v(n)}\right|_{v(n)=0} \quad (16)$$

$$= \left.\frac{\partial \varphi_n^1(s \circ \exp(v))}{\partial \exp(v)(n)}\right|_{v(n)=0} \quad (17)$$

$$= \left.\frac{\partial \varphi_n^1(s \circ w)}{\partial w(n)}\right|_{w(n)=n} \quad (18)$$

$$= -\Sigma^{-1}(n) \nabla(M \circ s)(n) \quad (19)$$

where $w(n) = \exp(v)(n) = \Phi_{1v}(n)$ is the transformation of voxel n and is the identity transformation when the velocity $v = 0$. In (17), we utilize the fact that the differential of the exponential map at $v = 0$ is the identity. $\nabla(M \circ s)(n)$ is the spatial derivative of the image intensity at voxel n of the warped moving image $M \circ s$.

Similarly, we can show that $D_s^{\varphi^2}(0) = (1/\sigma_x) I_{3N \times 3N}$ where I is a $3N \times 3N$ identity matrix. The Gauss–Newton optimization method ignores the $O(\|v\|^2)$ term within the norm in (14), leading to the classical linear least-squares problem. In particular, (14) can then be rewritten as

$$E_s(v) \approx \left\| \begin{bmatrix} \Sigma^{-1}(F - M \circ s) \\ 0 \end{bmatrix} + \begin{bmatrix} -\Sigma^{-1}(\nabla(M \circ s)) \\ \frac{1}{\sigma_x} I_{3N \times 3N} \end{bmatrix} v \right\|^2 \quad (20)$$

$$= \|b - Av\|^2 \quad (21)$$

which is a linear least-squares problem. Independently of the size of the matrices, it is easy to solve the resulting linear system $Av = b$ since the equations for each voxel can be decoupled from all other voxels. With the help of the Sherman-Morrison matrix inversion lemma, no matrix inversion is even needed to invert the resulting small system of linear equations at each voxel [42].

We note that the original Demons algorithm [37] replaced $\nabla(M \circ s)$ by ∇F. This is justified by the fact that at the optimum, the gradient of the warped moving image should be almost equal to the gradient of the fixed image.

IV. DT-REFinD: Tensor Image Registration

A. Diffeomorphic Demons for Vector Images

Before incorporating the FS differential for tensor registration, let us extend the diffeomorphic Demons algorithm to vector images. In addition to helping us explain our complete algorithm, the derivation will also be useful for computing update steps when ignoring tensor reorientation in Section IV-D. We define a vector image to be an image with a vector of intensities at each voxel. We can treat a vector image like a scalar image in the sense that each vector component is independent of the other components. Deformation of a vector image works just like a scalar image, by treating each component of the vector separately.

It is fairly straightforward to re-derive the results from the previous section for vector images. Let K be the dimension of the intensity vector at each voxel. For convenience, we define F_n to be the $K \times 1$ intensity vector of the nth voxel, $F_n = \{F_n(1), \ldots, F_n(K)\}$, and F to be the $NK \times 1$ vector of all image intensities $F = \{F_1, \ldots, F_N\}$. Then the diffeomorphic Demons algorithm from the previous section applies exactly to vector images except that in (20), $\nabla(M \circ s)$ is now a sparse $NK \times 3N$ block diagonal matrix, where each block is $K \times 3$. In particular, the nth block of $\nabla(M \circ s)$ contains spatial derivatives of $M \circ s$ at voxel n

$$\nabla(M \circ s)(n) = \begin{pmatrix} \frac{\partial(M \circ s)_n(1)}{\partial x} & \frac{\partial(M \circ s)_n(1)}{\partial y} & \frac{\partial(M \circ s)_n(1)}{\partial z} \\ \vdots & \vdots & \vdots \\ \frac{\partial(M \circ s)_n(K)}{\partial x} & \frac{\partial(M \circ s)_n(K)}{\partial y} & \frac{\partial(M \circ s)_n(K)}{\partial z} \end{pmatrix}. \quad (22)$$

The resulting least-squares linear system $Ax = b$ is slightly harder to solve than before. However, for each voxel n, we only have to solve a 3×3 linear system for the velocity vector update

$v(n)$. The solution of the system is also more stable as there are more constraints.

B. DT-REFinD: Diffusion Tensor Registration With Exact FS Differential

We will now extend the Demons algorithm to DT images. A DT image is different from a vector image because of the additional structure present in a tensor. In particular, the space of symmetric positive definite matrices (tensors) is not a vector space. When deforming a DT image, reorientation is also necessary. We extend the diffeomorphic Demons registration of vector images to tensor images.

In this work, we use the modified Demons objective function (7) and the FS reorientation strategy in our registration. The objective function in step 2(i) of the Demons algorithm, shown in (11) for scalars, becomes

$$E_s(v) = \left\| \begin{bmatrix} \Sigma^{-1}\left[F - R^T\left(M \circ s \circ \exp(v)\right)R\right] \\ \frac{1}{\sigma_x}\exp(v) \end{bmatrix} \right\|^2 \tag{23}$$

$$= \left\| \begin{bmatrix} \varphi^1(s \circ \exp(v)) \\ \varphi^2(v) \end{bmatrix} \right\|^2 \tag{24}$$

$$= \|\varphi_s(v)\|^2. \tag{25}$$

Here, $\left\|F - R^T\left(M \circ s \circ \exp(v)\right)R\right\|^2$ is the Euclidean sum of squares difference (EUC-SSD) between the tensor images. In particular, F can be seen as a $9N \times 1$ vector by "rasterizing" the 3×3 order 2 tensor at each voxel into a column vector. $M \circ s \circ \exp(v)$ should be interpreted as the interpolated tensor image. In practice, since the tensors are symmetric, we can work with $6N \times 1$ vectors to represent tensors and increase the weights of the entries of Σ^{-1} corresponding to the nondiagonal entries of the tensors by $\sqrt{2}$. Each interpolated tensor is then reoriented using the rotation matrix R of each voxel and "rasterized" into a column vector. Note that R is implicitly dependent on the transformation $s \circ \exp(v)$. The term $\left\|F - R^T\left(M \circ s \circ \exp(v)\right)R\right\|^2$ computes the SSD between each tensor of the fixed image and the corresponding reoriented and interpolated tensor in the warped moving image, by treating each tensor as a vector and adding the SSD for all voxels.

Equation (23) can also be interpreted as the LOG-SSD between tensors if F and M are the Log-Euclidean transforms of the original tensor images, obtained by converting each tensor T in the original image to a log-tensor $\log(T)$. Note that $\log(T)$ is simply a symmetric matrix [6]. $M \circ s \circ \exp(v)$ is then the interpolated log-tensor image. $R^T\left(M \circ s \circ \exp(v)\right)R$ is the interpolated and reoriented log-tensor image, since

$$\log(R^T T R) = R^T \log(T) R \tag{26}$$

for any rotation matrix R. Therefore, tensor reorientation followed by Log-Euclidean transformation is the same as Log-Euclidean transformation followed by reorientation. This is convenient since we can perform a one time Log-Euclidean transformation of the tensor images to log-tensor images before registration and convert the final warped log-tensor images to tensor images at the end of the registration.

In this case, $D_s^{\varphi^1}(0)$ is a sparse $9N \times 3N$ matrix. One can interpret $D_s^{\varphi^1}(0)$ as $N \times N$ blocks of 9×3 matrices. In particular, the (n, j)th block $\left[D_s^{\varphi^1}(0)\right]_{nj}$ is equal to $\left(\partial\varphi_n^1(s \circ w)/\partial v(j)\right)\Big|_{v(j)=0}$, where we remind the readers that $w = \exp(v) = \Phi_{1v}$ and n, j are also voxel indices. Using the chain rule, the product rule and the fact that the differential of the exponential map at $v = 0$ is the identity, we get

$$\left[D_s^{\varphi^1}(0)\right]_{nj} = \left.\frac{\partial\varphi_n^1(s \circ w)}{\partial v(j)}\right|_{v(j)=0} \tag{27}$$

$$= \sum_{k=1}^{N} \left.\frac{\partial\varphi_n^1(s \circ w)}{\partial w(k)} \frac{\partial w(k)}{\partial v(j)}\right|_{v(j)=0} \tag{28}$$

$$= \left.\frac{\partial\varphi_n^1(s \circ w)}{\partial w(j)}\right|_{w(j)=j} \tag{29}$$

$$= -\Sigma^{-1}(n)\left[\frac{\partial R^T(n)}{\partial w(j)}\left(M \circ s \circ w(n)\right)R(n) + R^T(n)\frac{\partial\left(M \circ s \circ w(n)\right)}{\partial w(j)}R(n) + R^T(n)\left(M \circ s \circ w(n)\right)\frac{\partial R(n)}{\partial w(j)}\right]\Bigg|_{w(j)=j} \tag{30}$$

where w is the identity transformation when the velocity $v = 0$.

Recall that $R(n)$ is a function of the Jacobian of displacement field $J(n)$ at the voxel n and that (3) gives an analytical expression of $J(n)$. In practice, $J(n)$ is defined numerically using finite central difference as shown in (31) at the bottom of the page where $\{n_{x-}, n_{x+}, n_{y-}, n_{y+}, n_{z-}, n_{z+}\}$ are the neighbors of voxel n in the x, y and z directions, respectively. Therefore $w_y(n_{x+})$ denotes the y-coordinate of n_{x+} after transformation $w(n_{x+})$ and $w_y(n_{x-})$ denotes the y-coordinate of n_{x-} after transformation $w(n_{x-})$. $\triangle x$, $\triangle y$ and $\triangle z$ are the voxel spacings in the x, y and z directions respectively. Using the differential of R (5) and the expression of J (31), we can compute $\partial R(n)/\partial w(j)$ using the chain rule. Appendix B provides the detailed derivation. This definition of J implies

$$\left[D_s^{\varphi^1}(0)\right]_{nn} = -\Sigma^{-1}(n)R^T(n)\nabla(M \circ s)(n)R(n) \tag{32}$$

$$J(n) = \begin{pmatrix} \frac{s \circ w_x(n_{x+}) - s \circ w_x(n_{x-})}{2\triangle x} & \frac{s \circ w_x(n_{y+}) - s \circ w_x(n_{y-})}{2\triangle y} & \frac{s \circ w_x(n_{z+}) - s \circ w_x(n_{z-})}{2\triangle z} \\ \frac{s \circ w_y(n_{x+}) - s \circ w_y(n_{x-})}{2\triangle x} & \frac{s \circ w_y(n_{y+}) - s \circ w_y(n_{y-})}{2\triangle y} & \frac{s \circ w_y(n_{z+}) - s \circ w_y(n_{z-})}{2\triangle z} \\ \frac{s \circ w_z(n_{x+}) - s \circ w_z(n_{x-})}{2\triangle x} & \frac{s \circ w_z(n_{y+}) - s \circ w_z(n_{y-})}{2\triangle y} & \frac{s \circ w_z(n_{z+}) - s \circ w_z(n_{z-})}{2\triangle z} \end{pmatrix} \tag{31}$$

and for neighbor j of voxel n, we get

$$\left[D_s^{\varphi^1}(0)\right]_{nj} = -\Sigma^{-1}(n)\left[\frac{\partial R^T(n)}{\partial w(j)}\Big(M \circ s\Big)(n)R(n) + R^T(n)\Big(M \circ s\Big)(n)\frac{\partial R(n)}{\partial w(j)}\right]\Bigg|_{w(j)=j}. \tag{33}$$

Note that the first and second terms in the above expression are transpose of each other. Therefore, for $n \neq j$, the (n, j)th block of $D_s^{\varphi^1}(0)$ is zeros if voxels n and j are not neighbors.

As before, $D_s^{\varphi^2}(0) = (1/\sigma_x)I_{3N\times 3N}$. In summary, we have computed the full gradient of our objective function

$$D_s^{\varphi}(0) = \begin{bmatrix} D_s^{\varphi^1}(0) \\ \frac{1}{\sigma_x^2} I_{3N\times 3N} \end{bmatrix} \tag{34}$$

where $\left(1/\sigma_x^2\right)I$ is a constant diagonal matrix, while $D_s^{\varphi^1}(0)$ is a sparse $9N \times 3N$ matrix.

C. Gauss–Newton Nonlinear Least-Squares Optimization

From the previous sections, we can now write

$$E_s(v) \approx \left\| \begin{bmatrix} \Sigma^{-1}(F - R^T(M \circ s)R) \\ 0 \end{bmatrix} + \begin{bmatrix} D_s^{\varphi^1}(0) \\ \frac{1}{\sigma_x} I \end{bmatrix} v \right\|^2 \tag{35}$$

$$= \|b - Av\|^2. \tag{36}$$

The resulting least-squares problem is harder to solve than before, since the linear systems of equations cannot be separated into voxel-specific set of equations. However, the sparsity of the matrix makes the problem tractable. In practice, we solve the linear systems of equations using Gmm++, a free generic C++ template library for solving linear sparse systems.[2] At the finest resolution, solving the sparse linear system requires about 60 s. This is the bottleneck of the algorithm. However, due to the fast convergence of Gauss–Newton method, we typically only need to solve the linear systems 10 times per multiresolution level. The resulting registration takes about 15 min on a Xeon 3.2 GHz single processor machine.

The efficiency of the Demons algorithm for scalar images comes from separating the optimization into two phases: optimization of the dissimilarity measure and optimization of the regularization term. This avoids the need to solve a nonseparable system of linear equations when considering the two phases together. Because of the reorientation in tensor registration, we have to solve a sparse system of linear equations anyway. In this case, we could have incorporated the optimization of the regularization term together with the optimization of the dissimilarity measure without much loss of efficiency. In this work, we keep the two phases separate to allow for fair comparison with the case of ignoring the reorientation of tensors in the gradient computation (see Section IV-D) by using almost the same implementation. Any improvement must then clearly come from the use of the true gradient and not from using a one-phase optimization scheme versus a two-phase optimization scheme.

D. Classical Alternative: Ignoring the Reorientation of Tensors

Previous work [2] performs tensor registration by not including the reorientation in the gradient computation, but reorienting the tensors after each iteration using the current estimated displacement field. To evaluate the utility of the true gradient, we modify our algorithm to ignore the reorientation part of the objective function in the gradient computation. In particular, we can simplify the Gauss–Newton optimization in the previous section by setting $\left[D_s^{\varphi^1}(0)\right]_{nj} = 0$ and $\left[D_s^{\varphi^1}(0)\right]_{nn} = -\Sigma(n)^{-1}\nabla\left(R^T(n)(M \circ s)R(n)\right)$, effectively ignoring the effects of the displacement field of a voxel on the reorientation of its neighbors. Note that $\left[D_s^{\varphi^1}(0)\right]_{nn}$ is slightly different from before because we directly use the gradient of the warped and reoriented image. In each iteration, we treat the tensor like a vector, except when deforming the moving image. The resulting least-squares problem degenerates to that in Section IV-A. The algorithm is thus much faster since we only need to invert a 3×3 matrix per voxel at each iteration. Registration only takes a few minutes on a Xeon 3.2 GHz single processor machine.

V. EXPERIMENTS

We now compare the DT-REFinD algorithm that uses the exact FS differential with the classical alternative that uses an approximate gradient and a basic Demons algorithm that uses the fixed image gradient.

A. Data and Preprocessing

We use 10 DT images acquired on a Siemens 1.5T scanner using an EPI sequence, consisting of healthy volunteers with the following acquisition parameters: echo time $= 110$ ms; 25 diffusion gradients, image dimensions $= 128 \times 128 \times 60$; image resolution $= 1.875 \text{ mm} \times 1.875 \text{ mm} \times 4 \text{ mm}$. These images are kindly contributed by Dr. Ducreux, Bicêtre Hospital, Paris, France.

We first use morphological operations to extract a foreground mask from the diffusion weighted (DW) images of each of the 10 DT images. This involves an automatic thresholding of any single DW image, except the baseline, so that the skull and the eyes do not interfere in the mask calculation. The threshold is chosen so that the mask contains the entire brain. This inevitably contains some outliers in the background. Then, a sequence of erosions with a ball of radius 1 voxel is performed to remove outliers (3 to 4 iterations are sufficient), which results in a set of connected components ensured to lie within the brain. Finally, we use conditional reconstruction to create the final mask. This involves dilating the connected components while intersecting the result with the initial mask, and repeating this process until convergence. Doing so allows the connected components to grows within the brain while ensuring the background outliers are canceled. If holes are still present in the mask, a hole filling algorithm can be applied (a simple morphological closing is generally sufficient).

[2] http://home.gna.org/getfem/gmm_intro

B. Implementation Details

We perform pairwise registration of DT images via a standard multiresolution optimization, by smoothing and downsampling the data for initial registration and using the resulting registration from a coarser resolution to initialize the registration of a finer resolution. We find that 10 iterations per multiresolution level were sufficient for convergence. When computing the SSD objective function, only voxels corresponding to the fixed image foreground are included.

There are three main parameters in the algorithm: the diagonal variability matrix Σ, and the tradeoff parameters σ_x and σ_T. Σ could be, in principle, estimated from a set of diffusion tensor images via coregistration. Since we deal with pairwise registration, we set Σ to be a constant diagonal matrix. Consequently, because the local optimum is determined by the relative weighting of Σ, σ_x and σ_T, we simply set Σ to be the identity matrix. σ_x determines the step-size taken at each iteration, which affects the stability of the registration algorithm [42]. Therefore, we empirically set σ_x so that the update at each iteration is about 2 voxels. The relative values of σ_x and σ_T determine the width of the kernel used to smooth the deformation field. Once σ_x is determined, the value of σ_T determines the warp smoothness.

As previously shown [14], it does not make sense to compare two registration algorithms with a fixed tradeoff between the dissimilarity measure and regularization, especially when the two algorithms use different dissimilarity measures and/or regularizations. Furthermore, one needs to be careful with the tradeoff selection for optimal performance in a given application [44].

In this work, we compare the algorithms over a broad range of kernel sizes. We note that larger kernel sizes lead to more smoothing and thus smoother warps. Because kernel sizes are not comparable across the different algorithms we consider, we use harmonic energy as a more direct measure of warp smoothness. We define the harmonic energy to be the average over all voxels of the squared Frobenius norm of the Jacobian of the displacement field. Note that the Jacobian of the displacement field corresponds to the Jacobian of the transformation defined in (3) without the identity. Therefore lower harmonic energy corresponds to smoother deformation.

C. Evaluation Metrics

To assess the alignment quality of two registered DT images, we use a variety of tensor metrics [2], [9], [21]. Let D be a diffusion tensor. We let $\lambda_1, \lambda_2, \lambda_3$ be its eigenvalues in descending order with corresponding eigenvectors e_1, e_2, e_3. We denote $\langle\lambda\rangle$ the average eigenvalues. Similarly, let D' denote another diffusion tensor with corresponding eigenvalues $\lambda_1', \lambda_2', \lambda_3'$ and eigenvectors e_1', e_2', e_3'. The following measures are averaged over the foreground voxels of the fixed image. This in turn allows us to average results across different registration trials by normalizing for brain sizes.

1) Euclidean Mean Squared Errors (EUC-MSE): squared Frobenius norm of $D - D'$.
2) Log Euclidean Mean Squared Errors (LOG-MSE): squared Frobenius norm of $\log(D) - \log(D')$.
3) 1—Overlap: $1 - \left(\sum_{i=1}^{3} \lambda_i \lambda_i' (e_i \cdot e_j)^2 / \sum_{i=1}^{3} \lambda_i \lambda_i'\right)$.

Note that the EUC-MSE and LOG-MSE correspond to the SSD and LOG-SSD dissimilarity metrics we employ during registration. In additional to these tensor metrics, we also consider the following scalar measures [1], [9]. The dissimilarity between two tensors is defined to be the sum of squared differences between these scalar measures, averaged over the foreground voxels.

1) $\mathrm{FA(Fractional\ Anisotropy)}$: $\left(\sqrt{3\left((\lambda_1 - \langle\lambda\rangle)^2 + (\lambda_2 - \langle\lambda\rangle)^2 + (\lambda_3 - \langle\lambda\rangle)^2\right)} / 2\left(\lambda_1^2 + \lambda_2^2 + \lambda_3^2\right)\right)$.
2) LFA (Logarithmic Anisotropy): FA computed for $\log(D)$.
3) ADC (Apparent Diffusion Coefficient): $\mathrm{Trace}(D) = 3\langle\lambda\rangle$.
4) VOL (Volume): $\det(D) = \lambda_1\lambda_2\lambda_3$.
5) CL (Linear Anisotropic Diffusion): $(\lambda_1 - \lambda_2/3\langle\lambda\rangle)$.
6) CP (Planar Ansiotropic Diffusion): $(2(\lambda_2 - \lambda_3)/3\langle\lambda\rangle)$.
7) CS (Spherical Anisotropic Diffusion): $(3\lambda_3/3\langle\lambda\rangle)$.
8) RA (Relative Anisotropy): $\left(\sqrt{(\lambda_1 - \langle\lambda\rangle)^2 + (\lambda_2 - \langle\lambda\rangle)^2 + (\lambda_3 - \langle\lambda\rangle)^2} / \sqrt{6}\langle\lambda\rangle\right)$.
9) VR (Volume Ratio): $\left(\lambda_1\lambda_2\lambda_3/\langle\lambda\rangle^3\right)$.
10) DISP (Dispersion): $\sqrt{(\lambda_2 + \lambda_3/2\lambda_1)}$.
11) λ_1.
12) λ_2.
13) λ_3.

A question then arises over whether these scalar measures should be computed after deforming the moving image or computed on the unwarped moving image and then interpolated with the deformation field. The latter is attractive because the result is independent of the tensor reorientation and interpolation strategies. On the other hand, since registration is almost never an end-goal—the deformed tensor images are presumably used for other tasks, one could argue that it is important to measure the quality of the deformed tensors. Therefore, in this work, we consider both strategies. Finally, we define the average distance between two deformation fields c_1 and c_2 to be

$$\frac{1}{N} \sum_{n \in \mathrm{foreground}} ||c_1(n) - c_2(n)||_{L^2} \qquad (37)$$

where N is the number of foreground voxels. We compute the average difference in the deformation fields obtained by the different methods to evaluate how different the deformation fields are. We find that the average difference in the deformation fields between the exact gradient method and the approximate gradient method ranges from 2 mm at low harmonic energy to 5 mm at high harmonic energy (*cf.* image resolution $= 1.875\ \mathrm{mm} \times 1.875\ \mathrm{mm} \times 4\ \mathrm{mm}$). The average difference in the deformation fields between the fixed image gradient method and the other two methods ranges from 3 mm at low harmonic energy to 8 mm at high harmonic energy.

The average distance can also be used to measure inverse consistency. Without the availability of ground truth deformation, inverse consistency [17] can be used as an indirect assessment of deformation quality. In particular, given a deformation c_{ij} from subject i to subject j and c_{ji} from subject j to subject i, inverse

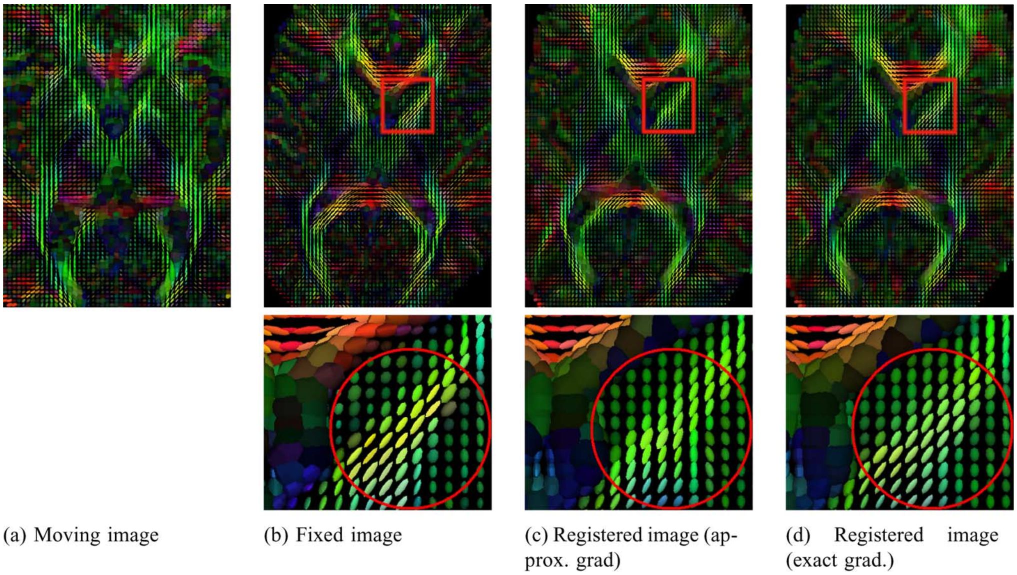

(a) Moving image
(b) Fixed image
(c) Registered image (approx. grad)
(d) Registered image (exact grad.)

Fig. 1. Qualitative comparison between the exact FS gradient and the approximated gradient for registering a pair of subjects using the Log-Euclidean framework and the same parameters in the registration. (a) Moving image. (b) Fixed image. (c) Registration using the approximated moving image gradient. (d) Registration using the exact FS gradient. Volumes were slightly cropped for better display. Exact gradient achieves better alignment of fiber tracts with a smoother displacement field. Tensors in the anterior limb of the internal capsule, highlighted in (b) and (d) are coherently oriented in a north-east direction. However, in (c), the directions of the tensors are more scattered. Furthermore, the volume of the tensors in (c) is swollen relative to (b) and (d). Numerically, the exact FS gradient has lower SSD with a smoother deformation field (not shown).

consistency is defined to be the average distance between the displacement field associated with the composed warp $c_{ij} \circ c_{ji}$ and a zero displacement field.

D. Qualitative Evaluation

Fig. 1 shows an example registration of two subjects from our data set. Visually, DT-REFinD results in better tract alignment, such as the anterior limb of the internal capsule highlighted in the figure. See figure caption for more discussion. In this particular example, DT-REFinD also achieves a better Log-Euclidean mean-square-error (LOG-MSE) and smoother deformation as measured by the harmonic energy.

E. Quantitative Evaluation I

To quantitatively compare the performance of the exact FS gradient, the approximate gradient and the fixed image gradient, we consider pairwise registration of the 10 DT images. Since our registration is not symmetric between the fixed and moving images, there are 90 possible pairwise registration experiments. We randomly select 20 pairs of images for pairwise registration. By swapping the roles of the fixed and moving images, we obtain 40 pairs of image registration. From our experiments, we find that the statistics we compute appear to converge after about 30 pairwise registrations, hence 40 pairwise registrations are sufficient for our purpose.

Even though we are considering algorithms with the same dissimilarity measure and regularization (and effectively the same implementation) but different optimization schemes, we find that for a fixed-size smoothing kernel, using the exact FS differential tends to converge to a solution of lower harmonic energy, i.e., a smoother displacement field. Smaller harmonic energy implies a smoother deformation, providing evidence that the reorientation acts as an additional constraint for the registration problem. To properly compare the algorithms, we consider smoothing kernels of sizes from 0.5 to 2.0 in increments of 0.1. In particular, we perform the following experiment.

For each pair of subjects *and* for each kernel size

i) Run the diffeomorphic Demons registration algorithm using Euclidean interpolation and EUC-SSD using:
   (a) Exact FS gradient (DT-REFinD).
   (b) Approximate gradient by ignoring reorientation.
   (c) Fixed image gradient. This is the gradient proposed in Thirion's original Demons algorithm [37].

ii) Repeat (i) using Log-Euclidean interpolation and LOG-SSD.

iii) Use the estimated deformation fields to compute the tensor and scalar measures discussed in Section V-C using FS reorientation or PPD reorientation.

iv) Compute the inverse consistency of the deformations from subject i to subject j and from subject j to subject i.

For a given smoothing kernel and registration strategy, registering different pairs of images leads to a set of error metric values corresponding to different harmonic energies. To average the error metric values across different pairs of images and to compare registration results among different strategies, for each registration, we linearly interpolate the dissimilarity metric (EUC-MSE, LOG-MSE, and so on) over a fixed set of harmonic energies sampled between 0.03 to 0.3. This allows us to average the error metric across different pairs of images and compare different strategies at a given harmonic energy.

1) Tensor Alignment: Fig. 2 shows the error metrics (averaged over 40 pairwise registrations) with respect to the harmonic energies when using the dissimilarity metric EUC-SSD, euclidean interpolation and FS reorientation for registration. The final deformations were applied using FS reorientation.

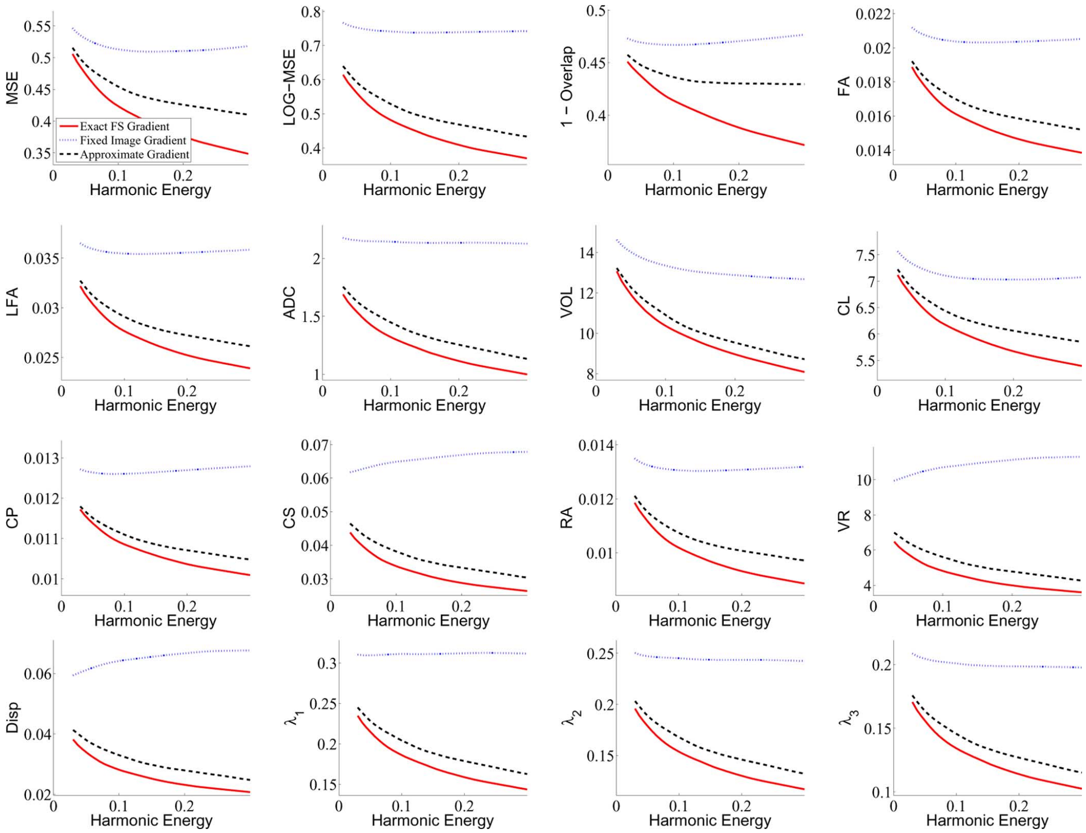


Fig. 2. Comparison of exact FS gradient, approximate gradient and fixed image gradient over an entire spectrum of harmonic energy (x-axis) using EUC-SSD, Euclidean interpolation and FS reorientation for registration. The final deformations are applied using FS reorientation. We find that the exact FS gradient method achieves the best performance.

Fig. 3 shows the corresponding plot when applying the final warps using PPD reorientation. In both cases, we find that at all harmonic energy levels, the exact FS gradient method achieves the lowest errors. The approximate gradient method outperforms the fixed image gradient method.

As mentioned earlier, the scalar measures, such as FA, can be computed after deforming the moving tensor image or computed on the unwarped moving image and then deformed. Figs. 2 and 3 show results based on the former strategy. We obtain similar results using the latter strategy, but omit them here for brevity.

The amount of improvement increases as the harmonic energies increase. In our experiments, a harmonic energy of 0.3 corresponds to severe distortion (pushing the limits of the numerical stability of scaling and squaring), while a harmonic energy of 0.03 corresponds to very smooth warps. In previous work, we showed in the context of image segmentation that extreme distortion causes overfitting, while extremely smooth warps might result in insufficient fitting [44]. Only a concrete application can inform us of the optimal amount of distortion and is the subject of future studies. For now, we assume a “safe” range for assessing the algorithm’s behavior to be between harmonic energies 0.1 and 0.2. From the values in Figs. 2 and 3, we conclude that the exact FS gradient provides an improvement of between 5% to 10% over the approximate gradient in this “safe” range of harmonic energies.

To better appreciate the improvements, Fig. 4 shows the difference in errors by subtracting the error metric values of the approximate gradient method from the error metric values of the exact gradient method when using the dissimilarity metric EUC-SSD, Euclidean interpolation and FS reorientation for registration. The final deformations were applied using FS reorientation. The error bars indicate that the exact gradient method is statistically significantly better than the approximate gradient method over the entire range of harmonic energies and all the error metrics ($p < 10^{-5}$ for almost entire range of harmonic energies). We emphasize that the improvements persist even when we evaluate a different dissimilarity measure or use a different reorientation strategy from those used during registration.

Similarly, we find that the exact FS gradient method achieves the lowest errors when using LOG-SSD similarity metric and Log-Euclidean interpolation for registration, regardless of whether FS or PPD reorientation was used to apply the final deformation. We omit the results here in the interest of space.

2) Inverse Consistency: Fig. 5 shows the inverse consistency errors (averaged over 20 sets of forward and backward pairwise registrations) with respect to the harmonic energies. Once again, we find that all harmonic energy levels, the exact FS gradient method achieves the lowest errors, regardless of whether EUC-SSD and Euclidean interpolation or LOG-SSD and Log-Euclidean interpolation were used. Similarly, the approximate gradient method outperforms the fixed image gradient method.

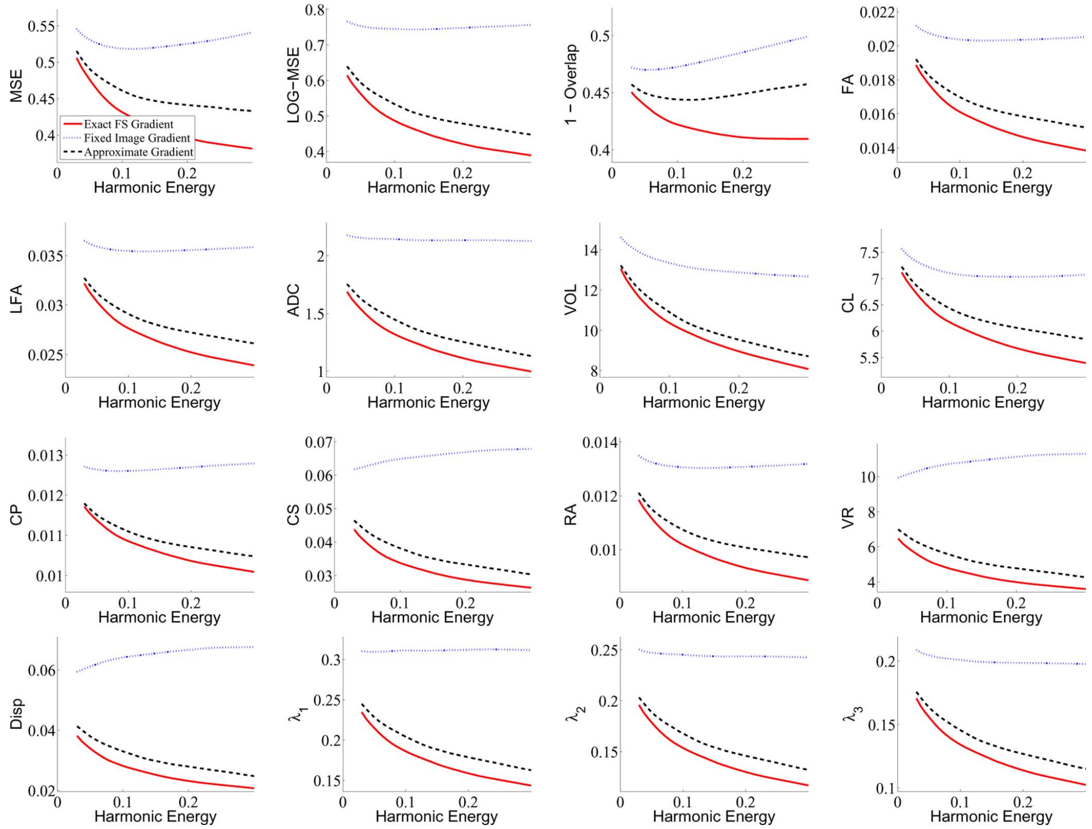


Fig. 3. Comparison of exact FS gradient, approximate gradient and fixed image gradient over an entire spectrum of harmonic energy (x-axis) using EUC-SSD, Euclidean interpolation and FS reorientation for registration. The final deformations are applied using PPD reorientation. We find that the exact FS gradient method achieves the best performance.

F. Quantitative Evaluation II

We perform a second set of experiments to evaluate the algorithm's ability to recover randomly generated synthetic warps. Given a DT image, we first generate a set of random warps by sampling a random velocity at each voxel location from an independent and identically distributed (I.I.D.) Gaussian. The foreground mask is then used to remove the velocity field from the background voxels. The resulting velocity field is smoothed spatially with a Gaussian filter. We compute the resulting displacement field by "scaling and squaring." This displacement field is used to warp the given DT image using Log-Euclidean interpolation. We use either FS or PPD to reorient the tensors. I.I.D. Gaussian noise is added to the warped DT image.

We pick a single DT image and generate 40 sets of random warps. We obtain an average displacement of 9.4 mm over the foreground voxels. The average harmonic energy is 0.15. We then perform pairwise registration between the DT image and the warped DT image using LOG-SSD. Once again, we consider a wide range of smoothing kernel sizes. We also compute the registration error defined to be the average difference between the ground truth random warps and the estimated deformation field specified in (37). Note that without registration, i.e., under the identity transformation, the average registration error is 9.4 mm.

Fig. 6 shows the registration errors (averaged over 40 trials) of the three gradients we are considering. From the plots, when the synthetic warps were applied using FS reorientation, the exact FS gradient recovers the ground truth warps up to 1.56 mm or 17% error with respect to the average 9.4 mm random warps. The approximate gradient achieves 2.34 mm or 25% error. Finally, the fixed image gradient achieves 2.73 mm or 29% error. Therefore, the exact FS gradient achieves an average of $(2.34 - 1.56)/2.34 \times 100 = 33\%$ and 43% reduction in registration errors compared with the approximate gradient and fixed image gradient, respectively.

When the synthetic warps were applied using PPD reorientation, the exact FS recovers the ground truth warps up to 2.80 mm or 30% error with respect to the average 9.4 mm random warps. The approximate gradient achieves 3.20 mm or 34% error. Finally, the fixed image gradient achieves 3.50 mm or 37% error. Therefore, the exact FS gradient achieves an average of 13% and 20% reduction in registration errors compared with the approximate gradient and fixed image gradient respectively.

Consistent with the previous experiments, using the exact FS gradient leads to the lowest registration errors regardless of whether FS or PPD reorientation were used to apply the synthetic deformation fields. We note that the registration errors inevitably increase when PPD were used to apply the synthetic deformation field, since we use FS reorientation during registration.

VI. Discussion and Future Work

Since Gauss–Newton optimization allows the use of "big steps" in the optimization, it might cause the approximate gradient to be more sensitive to the reorientation. It is possible

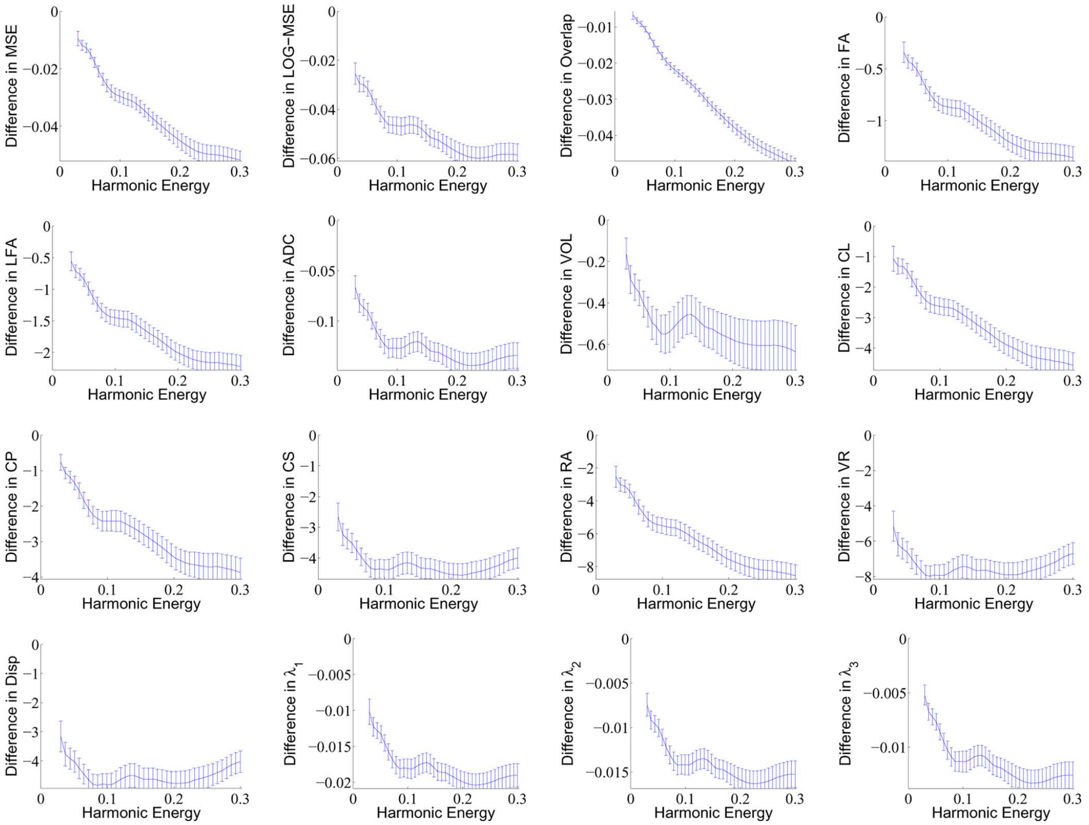
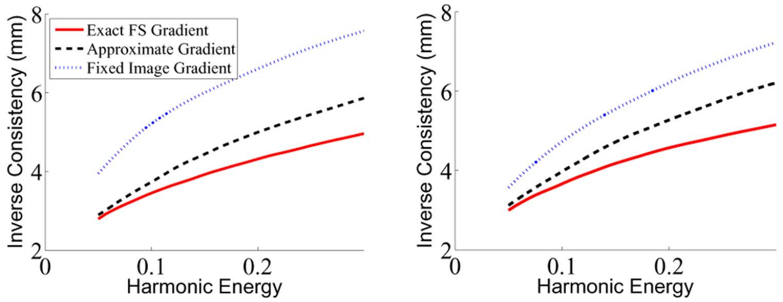


Fig. 4. Comparison of exact FS gradient and approximate gradient over an entire spectrum of harmonic energy (x-axis) using EUC-SSD, Euclidean interpolation and FS reorientation for registration. The final deformations are applied using PPD reorientation. Y-axis shows the difference in errors obtained by subtracting the error metrics of the approximate gradient method from the error metrics of the exact gradient method. Negative values imply that the exact gradient method outperforms the approximate gradient method. The error bars show the statistical variability (and thus significance) of the results.



Euclidean dissimilarity and interpolation

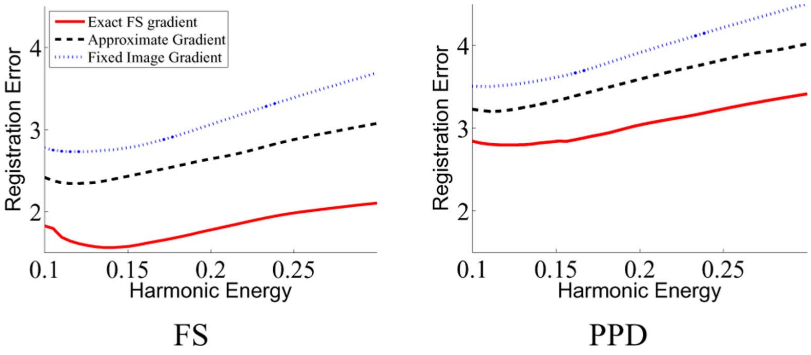
Log-Euclidean dissimilarity and interpolation

Fig. 5. Comparison of exact FS gradient, approximate gradient and fixed image gradient over an entire spectrum of harmonic energy (x-axis). Y-axis shows the inverse-consistency errors averaged over 20 sets of forward and backward pairwise registrations. We find that the exact FS gradient method achieves the lowest errors.

that other optimization methods, such as the conjugate gradient, might improve the results of using the approximate gradient, by allowing for "smaller steps" and reorient after each "small step." Possible future work would involve comparing the exact gradient and approximate gradient under an optimization framework that takes small steps in the optimization procedure.

However, from optimization theory and from our experience, Gauss–Newton method requires much fewer iterations to converge than conjugate gradient. Furthermore, conjugate gradient requires a line search, resulting in many function evaluations. Function evaluations are quite expensive in our case, because of the need to reorient and perform "scaling and squaring" of the velocity field. On the other hand, we find that in practice, line search is not necessary with Gauss–Newton optimization.

We should also emphasize that ignoring the gradient of the reorientation term can lead to registration errors that cannot be recovered regardless of any gradient-based optimization scheme. For example, consider the registration of a 2-D diffusion tensor image consisting of only horizontal tensors and a 2-D diffusion tensor image consisting of tensors orientated at



FS

PPD

Fig. 6. Comparison of exact FS gradient, approximate gradient and fixed image gradient over an entire spectrum of harmonic energy (x-axis) using synthetic images with ground truth deformations. Y-axis shows the registration errors (in millimeters) averaged over 40 pairwise registrations. We find that the exact FS gradient method achieves the lowest errors regardless of whether FS or PPD reorientation were used to generate the warps.

a 10° angle. In this case, ∇F and ∇M are both zeros, so the approximate gradient and fixed image gradient updates are both zeros. In contrast, the exact gradient update is nonzero due to the reorientation.

In our experiments, we show that using the exact gradient results in better inverse consistency than the approximate gradient. However, the inverse consistency is not perfect. Incorporating an inverse consistency constraint as suggested by the recent extension of the diffeomorphic Demons algorithm [41] should not be difficult.

An interesting observation from the synthetic warp experiment in Section V-F is that the best registration occurs when the kernel size is such that the harmonic energy is about 0.14, which is close to the average harmonic energy of the synthetic warps. In practice, no ground truth deformation is available, making selection of the optimal kernel size difficult. Furthermore, we believe the amount of deformation required is dependent on the application of interest; the appropriate kernel size is likely to vary with applications.

It may also be the case that different anatomical regions might require different optimal warp smoothness. While using a different kernel at each spatial location is possible, this would reduce the efficiency of the demons algorithm. More importantly, it becomes unclear whether step 2 of the demons algorithm (spatial smoothing) is justified. To get around such a situation, one could instead shift the burden to step 1 of the demons algorithm. In this paper, the variability matrix Σ is set to be the identity matrix. Allowing for a nonconstant diagonal matrix Σ will effectively result in spatially varying warp smoothness, since smaller values of the nth diagonal entry place greater emphasis on matching the nth tensor of the fixed image to the moving image. Estimating Σ and an optimal registration regularization tradeoff is an active area of research [4], [19], [33], [40], [44], [45] that we do not deal with in this paper.

The exact FS differential is useful even with a different model of deformation or dissimilarity metric from the ones we employ in this paper. Mutual Information (MI) has been proposed as a criterion to register diffusion images [16], [28], [39]. Because MI can handle nonlinear change in intensities across images, it can potentially handle diffusion image registration without any reorientation. In fact, [39] suggests that MI without reorientation results in better registration than MI with iterative reorientation with either FS or PPD. Future work could involve testing their observation when reorientation is properly taken into account using the analytical differential we presented in this paper.

VII. Conclusion

In this work, we derive the exact differential of the FS reorientation. We propose a fast diffeomorphic DT image registration algorithm DT-REFinD using the exact FS differential. We show that the use of the exact gradient achieves better tensor alignment than the approximate gradient which ignores reorientation, over an entire spectrum of harmonic energies. The improvements persist even if we use an error metric different from the objective function we optimize and if we use PPD reorientation for applying the final deformation. We also show that the exact gradient method recovered randomly generated warps significantly better than the approximate gradient method—1.56 mm versus 2.34 mm error on average. DT-REFinD has been incorporated into the freely available MedINRIA software, which can be downloaded at http://www-sop.inria.fr/asclepios/software/MedINRIA.

Appendix A
FS Differential

In [20], the differential of the matrix $r = A(A^TA)^{-(1/2)}$ is derived, where $A = YX^T$ and Y and X are $3 \times n$ matrices. In the context of [20], X contains the measured coordinates of a set of labeled points and Y contains their measured positions after rigid body motion. X and Y can be used to estimate the rotation component of the rigid motion r using the least-squares estimate $r = A(A^TA)^{-(1/2)}$. Finding the differential dr in terms of X and Y therefore allows the error analysis of the estimate r when the measurements X and Y are noisy.

From $r^Tr = I$ and $(r + dr)^T(r + dr) = I$, we get $drr^T = -rdr^T$ by ignoring second order terms [20]. Defining $\delta r \triangleq drr^T$, we have

$$\delta r \triangleq drr^T = -rdr^T = -\delta r^T. \tag{38}$$

From (38), δr is a skew symmetric matrix, and therefore takes the form

$$\delta r = \begin{pmatrix} 0 & -m_3 & m_2 \\ m_3 & 0 & -m_1 \\ -m_2 & m_1 & 0 \end{pmatrix} \triangleq m^{\oplus}. \tag{39}$$

We define $vec(\delta r) \triangleq (m_1, m_2, m_3)^T$ and $S \triangleq (A^T A)^{1/2}$. Then, the major result of [20] can be expressed as follows:

$$vec(\delta r) = r(\mathrm{tr}(S)I - S)^{-1} r^T \cdot \left(\sum_i (rX)_i \times dY_i + (rdX)_i \times Y_i\right) \tag{40}$$

where $(\cdot)_i$ denotes the ith column of $(\cdot)$ and $\times$ denotes the cross product operator.

Recall that we are interested in dR, where $R = (JJ^T)^{-(1/2)}J$. By setting $J \triangleq A^T$ and $S \triangleq (A^T A)^{1/2} = (JJ^T)^{1/2}$, we obtain $R = r^T$ and $dR = dr^T$. Therefore

$$\delta r \overset{(38)}{=} -r dr^T = -r dR. \tag{41}$$

Since $r^T r = I$, by multiplying (41) by r^T, we obtain

$$dR = -r^T \delta r = -R\delta r. \tag{42}$$

By setting, $X = I$ and $A = YI^T = J^T$, we finally arrive at the expression for dR

$$\begin{aligned} dR &\overset{(42)}{=} -R\delta r \\ &\overset{(40)}{=} -R\left[r(\mathrm{tr}(S)I - S)^{-1} r^T \cdot \left(\sum_i (rX)_i \times dY_i + (rdX)_i \times Y_i\right)\right]^{\oplus} \quad (43) \\ &= -R\left[R^T(\mathrm{tr}(S)I - S)^{-1} R \sum_i (R^T)_i \times (dJ^T)_i\right]^{\oplus} \quad (44) \end{aligned}$$

where we have used the fact that $dX = dI = 0$.

Appendix B
Rotation Derivatives

For completeness, we now derive the expressions for $(\partial R(n)/\partial w(j))\Big|_{w(j)=j}$, where j are the neighboring voxels of voxel n. Recall that $\{n_{x-}, n_{x+}, n_{y-}, n_{y+}, n_{z-}, n_{z+}\}$ are the neighbors of voxel n in the x, y and z directions respectively. u_x, u_y, u_z are the components of the displacement field in the x, y and z directions. For convenience, we denote $(\partial R(n)/\partial w(j))\Big|_{w(j)=j} = \{(\partial R(n)/\partial w_x(j)), (\partial R(n)/\partial w_y(j)), (\partial R(n)/\partial w_z(j))\} = \{(\partial R(n)/\partial w_k(j))\}$. Using the chain rule, we have

$$\begin{aligned} \frac{\partial R(n)}{\partial w_k(n_{x+})} &= \sum_m \sum_{ij} \frac{\partial R(n)}{\partial J_{ij}(n)} \frac{\partial J_{ij}(n)}{\partial (s \circ w_m(n_{x+}))} \cdot \frac{\partial (s \circ w_m(n_{x+}))}{\partial w_k(n_{x+})}\Bigg|_{w_k(n_{x+})=n_{x+}} \quad (45) \\ &= \sum_m \frac{\partial R(n)}{\partial J_{m1}(n)} \frac{\partial J_{m1}(n)}{\partial (s \circ w_m(n_{x+}))} \cdot \frac{\partial (s \circ w_m(n_{x+}))}{\partial w_k(n_{x+})}\Bigg|_{w_k(n_{x+})=n_{x+}} \quad (46) \\ &= \frac{1}{2\triangle x} \sum_m \frac{\partial R(n)}{\partial J_{m1}(n)} J_{mk}(n_{x+}). \quad (47) \end{aligned}$$

The second and third equalities come from evaluating $(\partial J_{ij}(n)/\partial(s \circ w_m(n_{x+})))$, which are mostly zeros. Notice that $J_{m1}(n)$ and $J_{mk}(n_{x+})$ are evaluated at two different voxels. Similarly, we have

$$\begin{aligned} \frac{\partial R(n)}{\partial w_k(n_{x-})} &= -\frac{1}{2\triangle x} \sum_m \frac{\partial R(n)}{\partial J_{m1}(n)} J_{mk}(n_{x-}) \\ \frac{\partial R(n)}{\partial w_k(n_{y+})} &= \frac{1}{2\triangle y} \sum_m \frac{\partial R(n)}{\partial J_{m2}(n)} J_{mk}(n_{y+}) \\ \frac{\partial R(n)}{\partial w_k(n_{y-})} &= -\frac{1}{2\triangle y} \sum_m \frac{\partial R(n)}{\partial J_{m2}(n)} J_{mk}(n_{y-}) \\ \frac{\partial R(n)}{\partial w_k(n_{z+})} &= \frac{1}{2\triangle z} \sum_m \frac{\partial R(n)}{\partial J_{m3}(n)} J_{mk}(n_{z+}) \\ \frac{\partial R(n)}{\partial w_k(n_{z-})} &= -\frac{1}{2\triangle z} \sum_m \frac{\partial R(n)}{\partial J_{m3}(n)} J_{mk}(n_{z-}). \end{aligned} \tag{48}$$

Acknowledgment

The authors would like to thank D. Ducreux, M.D., Ph.D., Bicêtre Hospital, Paris, for the DTI data and the reviewers for their many helpful suggestions. B. T. T. Yeo would like to thank C.-F. Westin, G. Kindlmann, and M. Sabuncu for useful discussions and feedback.

References

[1] A. Alexander, K. Hassan, G. Kindlmann, D. Parker, and J. Tsuruda, "A geometric analysis of diffusion tensor measurements of the human brain," *Magn. Reson. Med.*, vol. 44, no. 2, pp. 283–291, Aug., 2000.

[2] D. Alexander and J. Gee, "Elastic matching of diffusion tensor images," *Comput. Vis. Image Understand.*, vol. 77, pp. 233–250, 2000.

[3] D. Alexander, C. Pierpaoli, and J. Gee, "Spatial transformations of diffusion tensor magnetic resonance images," *IEEE Trans. Med. Imag.*, vol. 20, no. 11, pp. 1131–1139, Nov. 2001.

[4] S. Allassonniere, Y. Amit, and A. Trouvé, "Toward a coherent statistical framework for dense deformable template estimation," *J. R. Stat. Soc. B*, vol. 69, no. 1, pp. 3–29, 2007.

[5] V. Arsigny, O. Commowick, X. Pennec, and N. Ayache, "A Log-Euclidean framework for statistics on diffeomorphisms," in *Proc. Int. Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*. New York: Springer, 2006, vol. 4190, Lecture Notes Computer Science, pp. 924–931.

[6] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, "Log-Euclidean metrics for fast and simple calculus on diffusion tensors," *Magn. Reson. Med.*, vol. 56, no. 2, pp. 411–421, 2006.

[7] J. Ashburner, "A fast diffeomorphic image registration algorithm," *NeuroImage*, vol. 38, pp. 95–113, 2007.

[8] P. Basser and S. Pajevic, "Estimation of the effective self-diffusion tensor from the NMR spin echo," *J. Magn. Reson.*, vol. 103, no. 3, pp. 247–254, 1994.

[9] P. Basser and S. Pajevic, "Statistical artifacts in diffusion tensor MRI (DT-MRI) caused by background noise," *Magn. Reson. Med.*, vol. 44, pp. 41–50, 2000.

[10] P. Batchelor, M. Moakher, D. Atkinson, F. Calamante, and A. Connelly, "A rigorous framework for diffusion tensor calculus," *Magn. Reson. Med.*, vol. 53, pp. 221–225, 2005.

[11] M. Beg, M. Miller, A. Trouvé, and L. Younes, "Computing large deformation metric mappings via geodesic flows of diffeomorphisms," *Int. J. Comput. Vis.*, vol. 61, no. 2, pp. 139–157, 2005.

[12] G. Birkhoff and G. Rota*, Ordinary Differential Equations*. New York: Wiley, 1978.

[13] M. Bro-Nielsen and C. Gramkow, "Fast fluid registration of medical images," in *Proc. Visualizat. Biomed. Comput.*, 1996, vol. 1131, pp. 267–276.

[14] P. Cachier, E. Bardinet, D. Dormont, X. Pennec, and N. Ayache, "Iconic feature based non-rigid registration: The PASHA algorithm," *Comput. Vis. Image Understand.*, vol. 89, no. 2-3, pp. 272–298, 2003.

[15] Y. Cao, M. I. Miller, S. Mori, R. L. Winslow, and L. Younes, "Diffeomorphic matching of diffusion tensor images," presented at the Workshop Math. Methods Biomed. Image Anal., Int. Conf. Comput. Vis. Pattern Recognit., New York, 2006.

[16] M. Chiang, A. Leow, A. Klunder, R. Dutton, M. Barysheva, S. Rose, K. McMahon, G. de Zubicaray, A. Toga, and P. Thompson, "Fluid registration of diffusion images using information theory," *IEEE Trans. Med. Imag.*, vol. 27, no. 4, pp. 442–456, Apr., 2008.

[17] G. Christensen and H. Johnson, "Consistent image registration," *IEEE Trans. Med. Imag.*, vol. 20, no. 7, pp. 568–582, Jul. 2001.

[18] G. Christensen, R. Rabbit, and M. Miller, "Deformable templates using large deformation kinematics," *IEEE Trans. Image Process.*, vol. 5, pp. 1435–1447, Oct. 1996.

[19] O. Commowick, R. Stefanescu, P. Fillard, V. Arsigny, N. Ayache, X. Pennec, and G. Malandain, "Incorporating statistical measures of anatomical variability in atlas-to-subject registration for conformal brain radiotherapy," in *Proc. Int. Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*. New York: Springer, 2005, vol. 3750, Lecture Notes Computer Science, pp. 927–934.

[20] L. Dorst, "First order error propagation of the procrustes method for 3D attitude estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 2, pp. 221–229, Feb. 2005.

[21] P. Fillard, X. Pennec, V. Arsigny, and N. Ayache, "Clinical DT-MRI estimation, smoothing, and fiber tracking with Log-Euclidean metrics," *IEEE Trans. Med. Imag.*, vol. 26, no. 11, pp. 1472–1482, Nov. 2007.

[22] P. Fletcher and S. Joshi, "Riemannian geometry for the statistical analysis of diffusion tensor data," *Signal Process.*, vol. 87, no. 2, pp. 250–262, 2007.

[23] J. Gee and D. Alexander, "Diffusion tensor image registration," in *Visualization and Image Processing of Tensor Fields*. New York: Springer, 2005.

[24] A. Guimond, C. Guttmann, S. Warfield, and C.-F. Westin, "Deformable registration of DT-MRI data based on transformation invariant tensor characteristics," in *Proc. Int. Symp. Biomed. Imag.: From Nano Macro*, 2002, pp. 761–764.

[25] M. Hernandez, M. Bossa, and S. Olmos, "Registration of anatomical images using geodesic paths of diffeomorphisms parameterized with stationary velocity fields," in *Proc. Workshop Math. Methods Biomed. Image Anal. Int. Conf. Comput. Vis.*, 2007, pp. 1–9.

[26] J. Keller, "Closest unitary, orthogonal and hermitian operators to a given operator," *Math. Mag.*, vol. 48, no. 4, pp. 192–197, 1975.

[27] G. Kindlmann, R. Estepar, M. Niethammer, S. Haker, and C. Westin, "Geodesic-Loxodromes for diffusion tensor interpolation and difference measurement," in *Proc. Int. Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*. New York: Springer, 2007, vol. 4791, Lecture Notes Computer Science, pp. 1–9.

[28] A. Leemans, J. Sijbers, S. Backer, E. Vandervliet, and P. Parizel, "Affine coregistration of diffusion tensor magnetic resonance images using mutual information," in *Proc. Advanced Concepts for Intelligent Vision Systems*. New York: Springer, 2005, vol. 3708, Lecture Notes Computer Science, pp. 523–530.

[29] C. Lenglet, M. Rousson, R. Deriche, and O. Faugeras, "Statistics on the manifold of multivariate normal distributions: Theory and application to diffusion tensor MRI processing," *J. Math. Imag. Vis.*, vol. 25, no. 3, pp. 423–444, 2006.

[30] M. Moakher, "A differential geometric approach to the geometric mean of the symmetric positive-definite matrices," *SIAM J. Matrix Anal. Appl.*, vol. 26, no. 3, pp. 735–747, 2005.

[31] H.-J. Park, M. Kubicki, M. Shenton, A. Guimond, R. McCarley, S. Maier, R. Kikinis, F. Jolesz, and C.-F. Westin, "Spatial normalization of diffusion tensor MRI using multiple channels," *NeuroImage*, vol. 20, no. 4, pp. 1995–2009, 2003.

[32] X. Pennec, P. Fillard, and N. Ayache, "A Riemannian framework for tensor computing," *Int. J. Comput. Vis.*, vol. 66, pp. 41–66, 2005.

[33] X. Pennec, R. Stefanescu, V. Arsigny, P. Fillard, and N. Ayache, "Riemannian elasticity: A statistical regularization framework for non-linear registration," in *Proc. MICCAI*. New York: Springer, 2005, vol. 3750, Lecture Notes Computer Science, pp. 943–950.

[34] J.-M. Peyrat, M. Sermesant, X. Pennec, H. Delingette, C. Xu, E. McVeight, and N. Ayache, "A computational framework for the statistical analysis of cardiac diffusion tensors: Application to a small database of canine hearts," *IEEE Trans. Med. Imag.*, vol. 26, no. 11, pp. 1500–1514, Nov. 2007.

[35] P. Rogelj and S. Kovacic, "Symmetric image registration," *Med. Image Anal.*, vol. 10, no. 3, pp. 484–493, 2006.

[36] J. Ruiz-Alzola, C.-F. Westin, S. Warfield, C. Alberola, S. Maier, and R. Kikinis, "Nonrigid registration of 3D tensor medical data," *Med. Image Anal.*, vol. 6, pp. 143–161, 2002.

[37] J. Thirion, "Image matching as a diffusion process: An analogy with Maxwell's demons," *Med. Image Anal.*, vol. 2, no. 3, pp. 243–260, 1998.

[38] A. Trouvé, "Diffeomorphisms, groups and pattern matching in image analysis," *Int. J. Comput. Vis.*, vol. 28, no. 3, pp. 213–221, 1998.

[39] W. Van Hecke, A. Leemans, E. D'Agostino, S. De Backer, E. Vandervliet, P. M. Parizel, and J. Sijbers, "Nonrigid coregistration of diffusion tensor images using a viscuous fluid model and mutual information," *IEEE Trans. Med. Imag.*, vol. 26, no. 11, pp. 1598–1612, Nov. 2007.

[40] K. Van Leemput, "Probabilistic brain atlas encoding using Bayesian inference," in *Proc. Int. Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*. New York: Springer, 2006, vol. 4190, Lecture Notes Computer Science, pp. 704–711.

[41] T. Vercauteren, X. Pennec, A. Perchant, and N. Ayache, "Symmetric log-domain diffeomorphic registration: A demons-based approach," *Med. Imag. Comput.*, vol. 5241, pp. 754–761, 2008.

[42] T. Vercauteren, X. Pennec, A. Perchant, and N. Ayache, "Diffeomorphic demons: Efficient non-parametric image registration," *NeuroImage*, vol. 45, no. 1, pp. S61–S72, 2009.

[43] H. Wang, L. Dong, J. O'Daniel, R. Mohan, A. Garden, K. Ang, D. Kuban, M. Bonnen, J. Chang, and R. Cheung, "Validation of an accelerated 'Demons' algorithm for deformable image registration in radiation therapy," *Phys. Med. Biol.*, vol. 50, no. 12, 2005.

[44] B. T. Yeo, M. Sabuncu, R. Desikan, B. Fischl, and P. Golland, "Effects of registration regularization and atlas sharpness on segmentation accuracy," *Med. Image Anal.*, vol. 12, no. 5, pp. 603–615, 2008.

[45] B. T. Yeo, M. Sabuncu, P. Golland, and B. Fischl, "Task-optimal registration cost functions," in *Proc. Int. Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 2009, vol. 5761, Lecture Notes Computer Science, pp. 598–606.

[46] B. T. Yeo, M. Sabuncu, T. Vercauteren, N. Ayache, B. Fischl, and P. Golland, "Spherical demons: Fast diffeomorphic landmark-free surface registration," *IEEE Trans. Med. Imag.*, to be published.

[47] B. T. Yeo, T. Vercauteren, P. Fillard, X. Pennec, P. Golland, N. Ayache, and O. Clatz, "DTI registration with exact finite-strain differential," in *Proc. Int. Symp. Biomed. Imag.: From Nano to Macro*, 2008, pp. 700–703.

[48] H. Zhang, B. Avants, P. Yuschkevich, J. Woo, S. Wang, L. McCluskey, L. Elman, E. Melhem, and J. Gee, "High-dimensional spatial normalization of diffusion tensor images improves the detection of white matter differences: An example study using amyotrophic lateral sclerosis," *IEEE Trans. Med. Imag.*, vol. 26, no. 11, pp. 1585–1597, Nov. 2007.

[49] H. Zhang, P. Yushkevich, D. Alexander, and J. Gee, "Deformable registration of diffusion tensor MR images with explicit orientation optimization," *Med. Image Anal.*, vol. 10, no. 5, pp. 764–785, 2006.

[50] U. Ziyan, M. Sabuncu, L. Donnell, and C.-F. Westin, "Nonlinear registration of diffusion MR images based on fiber bundles," in *Proc. Int. Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*. New York: Springer, 2007, vol. 4791, Lecture Notes Computer Science, pp. 351–358.