Title

Voxelwise spectral diffusional connectivity and its applications to Alzheimer's disease and intelligence prediction.

Journal

Medical image computing and computer-assisted intervention : MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention, 16(Pt 1)

ISSN

0302-9743

Authors

Li, Junning
Jin, Yan
Shi, Yonggang
et al.

Publication Date

2013

DOI

10.1007/978-3-642-40811-3_82

Peer reviewed

Fast Local Trust Region Technique for Diffusion Tensor Registration Using Exact Reorientation and Regularization

Junning Li, Yonggang Shi, Giang Tran, Ivo Dinov, Danny J. J. Wang, and Arthur Toga*

***Abstract*—Diffusion tensor imaging is widely used in brain connectivity research. As more and more studies recruit large numbers of subjects, it is important to design registration methods which are not only theoretically rigorous, but also computationally efficient. However, the requirement of reorienting diffusion tensors complicates and considerably slows down registration procedures, due to the correlated impacts of registration forces at adjacent voxel locations. Based on the diffeomorphic Demons algorithm (Vercauteren *et al.*, 2009), we propose a fast local trust region algorithm for handling inseparable registration forces for quadratic energy functions. The method guarantees that, at any time and at any voxel location, the velocity is always within its local trust region. This local regularization allows efficient calculation of the transformation update with numeric integration instead of completely solving a large linear system at every iteration. It is able to incorporate exact reorientation and regularization into the velocity optimization, and preserve the linear complexity of the diffeomorphic Demons algorithm. In an experiment with 84 diffusion tensor images involving both pair-wise and group-wise registrations, the proposed algorithm achieves better registration in comparison with other methods solving large linear systems (Yeo *et al.*, 2009). At the same time, this algorithm reduces the computation time and memory demand tenfold.**

***Index Terms*—Diffeomorphisms, diffusion tensor imaging, image registration, partial differential equations, tensor reorientation, trust region methods.**

I. Introduction

WATER molecules in biological tissue tend to diffuse faster along obstacle structures than across them. Diffusion weighted imaging (DWI) noninvasively measures the water diffusivity along different directions, capturing orientation information on obstacles such as neural fibers. Depending on the angular resolution employed, the anisotropy of water diffusion can be approximated as second-order tensors with diffusion tensor imaging (DTI), or as a sphere domain function with high angular resolution diffusion imaging (HARDI) [3]. Different from the T1/T2 imaging modalities, which assess local intensity information, diffusion imaging essentially captures direction information in topological structures. Advanced technologies are necessary to process, analyze and interpret diffusion imaging data. A number of initiatives to acquire and model large amounts of DTI data, for example, the Human Connnectome Project,[1] require nonlinear normalization of DTI images with theoretically rigorous and computationally efficient tools.

Registration of DTI images is more complicated than that of univariate (scalar) images, because diffusion tensors must be reorientated to keep them aligned with the transformed space, as shown in Section IV-A. The tensor value at a point in the warped image is first taken from the original image by the composition operation involving interpolation, and is then reorientated according to the Jacobian matrix of the transformation field. The finite strain (FS) [4] and the preservation of principal direction (PPD) [4] are two widely used tensor-reorientation strategies. No matter the strategy employed, spatially displacing a voxel to a new location not only changes its own tensor value, but also reorientates its neighboring tensors. As a result, the impacts of deformation forces at adjacent voxels correlate, and such interaction largely complicates the registration of DTI images.

Because exact handling of tensor reorientation can significantly complicate the computation, methods based on rotation-invariant features have been developed [5]–[7]. In general, these methods first extract scalar-valued or multi-channel features from diffusion tensors or voxel neighborhoods, for instance, the fractional anisotropy (FA), mean diffusivity (MD), etc., and then spatially normalize DTI images by aligning these rotation-invariant features with registration methods for scalar or multi-channel images.

To achieve better accuracy, methods which directly work on tensors have been developed with similarity metrics defined on tensors, and indirect or direct involvement of reorientation into the registration procedure. Alexander and Gee [8] calculated the deformation force without the reorientation effect, but when transforming the image at every registration iteration, they applied the PPD reorientation. Ruiz-Alzola *et al.* [9] generalized

Manuscript received May 17, 2013; revised July 10, 2013; accepted July 12, 2013. Date of publication July 18, 2013; date of current version April 22, 2014. This work is supported by the National Institutes of Health under Grant K01EB013633, Grant R01MH080892 9P41EB015922, Grant P41RR013642, Grant R01MH71940, Grant U54RR021813, and Grant U24RR025736. *Asterisk indicates corresponding author.*

J. Li, Y. Shi, and I. Dinov are with the Laboratory of Neuro Imaging, Department of Neurology, University of California–Los Angeles, Los Angeles, CA 90095 USA.

G. Tran is with the Department of Mathematics, University of California–Los Angeles, Los Angeles, CA 90095 USA.

D. J. J. Wang is with the Brain Mapping Center, Department of Neurology, University of California–Los Angeles, Los Angeles, CA 90095 USA.

*A. Toga is with the Laboratory of Neuro Imaging, Department of Neurology, University of California–Los Angeles, Los Angeles, CA 90095 USA (e-mail: toga@loni.ucla.edu).

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.2013.2274051

[1]Available online: http://www.humanconnectomeproject.org/

the correlation coefficient to tensor images, and used it for landmark-based registration, yet ignored reorientation in the local-patch matching procedure. Zhang *et al.* [10] took the FS reorientation into exact account for affine registration. To achieve nonlinear registration, they applied piece-wise affine registration to image sub-blocks and fused these transformations together by smoothing. The optimal affine transformation is estimated for each sub-blocks, but it is not clear how the fusion step affects the total registration energy. Cao *et al.* [11] extended the large deformation diffeomorphic metric mapping (LDDMM) framework, to analytically embed the PPD reorientation [4] into the deformation force. In 2009, Yeo *et al.* [2] incorporated exact FS reorientation into the diffeomorphic Demons algorithm [1], [12]. Different from Cao *et al.*, Yeo *et al.* counted the effect of reorientation not only into the deformation force (the steepest descent direction), but also into the Hessian matrix for minimizing the registration energy function.

Yeo's work [2] demonstrated that modeling exact tensor reorientation into registration considerably improves the accuracy. The registration task is modeled as an alternative optimization procedure, and iterative updates of transformation are optimized by solving a large inseparable sparse linear system. Such a strategy improves the accuracy, yet impacts computation efficiency in the following ways.

First, it demands considerably more computation than the original diffeomorphic Demons algorithm [1]. For scalar images, the diffeomorphic Demons algorithm enjoys $O(n)$ complexity, where n is the number of voxels. With a large sparse linear system embedded into every registration iteration, the DTI version of the algorithm [2], does not preserve this linear complexity. For DTI images of size larger than $128 \times 128 \times 128$, the method can take over 7 h and may require more than 20 GB of memory on a desktop with an Intel Xeon 2.80 GHz CPU. Though some sparse linear systems can be solved efficiently with $O(n)$ or $O[n \ln(n)]$ complexity [13]–[16], their special requirements may not always be satisfied by tensor reorientation, as discussed in Section IV-B.

Second, the solution of the inseparable linear system may violate the assumption employed by its own construction. Typically, image registration is formulated as a nonlinear optimization problem regarding the transformation field. During the optimization procedure, at each iteration, around the current transformation field, the nonlinear registration energy function is approximated as a quadratic function with its first-order (Jacobian) and second-order (Hessian) derivatives. This intrinsically requires the transformation update to stay within a region where the Taylor approximation is valid. Such a region, from the perspective of optimization, is called the "trust region" [17]: a trust region for a function $f(x)$ around x_0 is a neighborhood $|x - x_0| < \gamma$ within which $f(x)$ can be approximated with its second-order Taylor expansion. For a large inseparable linear system, the solution can be a "large leap," violating the assumption behind its construction: the Taylor approximation. Yeo *et al.* [2] also suggested investigating this problem.

For image registration, we may further require the transformation update at every voxel location to be small enough to stay within its own local trust region for the following reasons. First, the registration energy is typically a spatial integration of local energies, and the Taylor approximation of each local energy has its own local trust region. For example, the deformation force often depends on the gradient of the input images. If the gradient at a voxel is estimated with the central difference method, its value is valid within its voxel neighborhood. If the voxel is displaced out of this neighborhood, it enters a region which cannot be approximated with the same Taylor expansion. Second, though violating local trust regions may not considerably impact the global registration energy, it can introduce local instability and defects, as shown in Section III-D. Third, some special circumstances also require limiting the transformation update locally. For example, to guarantee composite invertability, transformation updates must be smaller than 0.5 voxel spacing at every location, to avoid overlap conflicts with neighbors.

In this paper, we address the two computational problems introduced by exact tensor reorientation. Based on Yeo's work [2], we propose a DTI registration method which 1) takes exact FS tensor reorientation into account, 2) is of $O(n)$ complexity, and 3) guarantees that the transformation update at every voxel location stays within its local trust region, for example, 0.5 voxel spacing. In Section II, we briefly introduce the variational framework of registration. In Section III, we provide an interpretation on the diffeomorphic Demons algorithm from the perspective of partial differential equations (PDE). In Section IV, we discuss the computational challenges brought by tensor reorientation. In Section V, we design a regularized PDE procedure in which the velocity field, at any time and at every voxel location, is guaranteed to stay within local trust regions. The transformation update, as the temporal accumulation of the velocity, is then automatically restricted within the local trust regions given a unit time duration. Instead of solving a large inseparable linear system at every registration iteration, we estimate the transformation update by integrating the velocity with numerical techniques, the Runge Kutta methods [18]. In Section VII, we present the evaluation of the proposed method on a real DTI data set of 84 subjects. The proposed method achieved the same level of registration quality as Yeo's method [2], yet reduced the computation time and memory demand tenfold.

In summary, our work includes the following.

1) Providing an interpretation on the Demons algorithm from a PDE perspective.
2) Designing a PDE procedure in which the velocity field is guaranteed to stay within local trust regions, at any time point and at every voxel location.
3) Proposing a method for estimating the update from the perspective of integration, using adaptive computation according to the accuracy level, instead of solving a large inseparable linear system.
4) Evaluating the method with a real DTI data set of 84 subjects, using both pair-wise and group-wise registration.

II. Preliminaries to Image Registration

Given two images in a spatial domain Ω, the fixed image F (the target) and the moving image M (the data), a registration task is to find a spatial transformation φ, such that the warped image $M \diamond \varphi$ is similar to F, where the notation "$\diamond$" denotes

the transformation operation. A typical choice to measure the dissimilarity between two images is the sum of squared errors $\mathcal{E}_{\text{sim}}(\varphi|F,M)=\langle F-M\diamond\varphi, F-M\diamond\varphi\rangle$ where the notation "$\langle\bullet,\bullet\rangle$" denotes the inner product.

Normally, the composition notation "$\circ$" is used to denote transformation, because transforming a scalar image involves just composition. However, as shown in Section IV-A, transforming a tensor image involves not only composition, but also reorientation. Thus, we use the notation "$\diamond$" to denote the transformation operation including all the necessary operations to properly conduct a transformation.

Without restrictions on φ, a registration task is an ill-posed problem so regularizations have to be imposed. The two most widely used regularizations are diffeomorphism and smoothness. Diffeomorphism requires φ to be invertible and both φ and its inverse φ^{-1} to be differentiable. Diffeomorphism guarantees that the topology of the moving image is preserved. Typically, a diffeomorphic transformation is generated from a (possibly time-variant) velocity field v, with the following partial differential equation (PDE) [19]:

$$\frac{\partial\varphi}{\partial t}=(\nabla\varphi)v. \tag{1}$$

If v is a stationary velocity field, we define $\exp^{\tau}(v)$ as the solution of (1) at time $t=\tau$ with initialization $\varphi=I$ (the identity) at $t=0$. With variable substitution, we have $\exp^{\tau}(v)=\exp^{1}(\tau v)$. If $\tau=1$, we use $\exp(v)$ for short.

The following ordinary differential equation (ODE) is more widely used in the literature to describe the transformation-velocity relationship:

$$\frac{d\varphi_{\text{forward}}}{dt}=v_{\text{forward}}(\varphi_{\text{forward}}) \tag{2}$$

where φ_{forward} maps a point in the moving image to a point in the fixed image. In this paper, to better reflect the warping procedure as implemented in software, we define $\varphi=\varphi_{\text{forward}}^{-1}$ to map a point in the fixed image to a point in the moving image, and define $v=-v_{\text{forward}}$. Equations (1) and (2) are equivalent, as shown in Appendix A.

The smoothness requirement can be applied to either the transformation field φ, as an elastic-like regularization [20], [21], or to the velocity field v, as a fluid-like regularization [22]. When applied to the transformation field, it is typically implemented as a regularization energy $\mathcal{E}_{\text{trans}}(\varphi)=\langle R\varphi, R\varphi\rangle$, where R is a spatial differential operator. The same format of regularization can be applied to the velocity field, but must be integrated over time as $\mathcal{E}_{\text{velo}}(v)=\int_0^T\langle Rv, Rv\rangle dt$ where T represents the time duration and is typically 1.

In the variational framework, a registration task is typically formulated as minimizing a combination of the similarity energy (for matching two images) and the regularization energies (for smoothness)

$$\mathcal{E}(\varphi)=\mathcal{E}_{\text{sim}}(\varphi|F,M)+\mathcal{E}_{\text{trans}}(\varphi)+\mathcal{E}_{\text{velo}}(v) \tag{3}$$

subject to the diffeomorphism constraint (1) (to guarantee diffeomorphism).

Notations frequently used in the paper are listed in Table I.

TABLE I
NOTATIONS

F:	Fixed image	M:	Moving image
φ:	Transformation field		
u:	Update field	v:	Velocity field
t:	Artificial time	τ :	Time duration
p:	Point in space	i:	Voxel index
$\mathcal{E}$:	Registration energy	R:	Differential operator for regularizing φ with $\mathcal{E}_{\text{trans}}=\langle R\varphi, R\varphi\rangle$
f:	Registration force	H:	Registration Hessian matrix
b:	Registration difference	g:	Difference gradient
P:	Velocity regularizer	γ:	Positive real value
$\exp^{\tau}(v)$	Solution of Eq. (1) at time $t=\tau$ with initialization I		
$\exp(v)$	$\exp(u)=\exp^{1}(u)$		
$\bullet^t$, $\bullet_p$ or $\bullet_i$:	Value associated with time t, point p or voxel i.		
$\nabla_p\varphi$	Jacobian matrix of φ at a point p.		
r:	Rotation matrix. r satisfies $rr^{\intercal}=I$.		
$M\diamond\varphi$	Transformation, not just including the composition operation, but also other necessary operations to properly perform image transformation. For example, the transformation of tensor images also involves reorientation: $[M\diamond\varphi](p)=r(\nabla_p\varphi)M[\varphi(p)]r^{\intercal}(\nabla_p\varphi)$.		

III. PDE Interpretation on Diffeomorphic Demons

In this section, we provide an interpretation on the diffeomorphic Demons algorithm from the perspective of partial differential equations (PDE). After a brief review of the algorithm, we present a PDE procedure for minimizing energy functions, and then explain the algorithm with this PDE. Such an insight into this popular algorithm is the foundation of the local-trust-region technique we propose in Section V.

A. The Algorithm

The diffeomorphic Demons algorithm by Vercauteren *et al.* [1] is a diffeomorphic variant of the Thirion's Demons algorithm [12]. The scalar-image version of the algorithm [1] is detailed in Algorithm 1, where $\bullet(i)$ and $\bullet_i$ denote a certain value associated with a voxel i. Vercauteren *et al.* in [1], [19] interpreted the algorithm as an alternative optimization procedure of an energy function.

B. Registration With PDE

Under the diffeomorphic perturbation along a velocity field v, the temporal derivative of $\mathcal{E}(\varphi)$ is $(d/d\tau)\mathcal{E}[\varphi\circ\exp^{\tau}(v)]=(\partial\mathcal{E}/\partial\varphi)(\nabla\varphi)v$. The most "effective" perturbation direction for minimizing $\mathcal{E}(\varphi)$ is $f(\varphi)\equiv-[(\partial\mathcal{E}/\partial\varphi)(\nabla\varphi)]^{\intercal}$, which is the registration force. The registration force equal to zero $f(\varphi)=0$ is a necessary condition for a diffeomorphic transformation to be a local minimum state of an energy function.

The following theorem whose proof is in Appendix B provides a PDE procedure for minimizing an energy function with the diffeomorphic constraint.

Theorem 1: Given a nonnegative energy function $\mathcal{E}(\varphi)$, and a self-adjoint, positive definite and possibly time-variant operator P whose eigenvalues are in a constant positive range

Algorithm 1 Diffeomorphic Demons for Scalar Images

Initialize $\varphi = I$.

Repeat until convergence

1) **For each** voxel i
   a) **Let** $b_i = F(i) - (M \circ \varphi)(i)$.
   b) **Let** $g_{ii} = \nabla(M \circ \varphi)(i)$.
   c) **Let** $u_i = \frac{g_{ii}^{\mathsf{T}} b_i}{g_{ii}^{\mathsf{T}} g_{ii} + 1/\tau_i}$, where $1/\tau_i$ is the update regularization coefficient at voxel i.
2) Optionally **smooth** the update field $u \leftarrow K_{\text{fluid}} \otimes u$, to achieve a fluid-like regularization. K_{fluid} is typically a Gaussian kernel.
3) **Update the transformation field** with the update field: $\varphi \leftarrow \varphi \circ \exp(u)$. $\exp(u)$ is defined in Table I.
4) Optionally **smooth** the transformation field $\varphi \leftarrow K_{\text{diff}} \otimes \varphi$ to achieve a diffusion-like regularization. K_{diff} is typically a Gaussian kernel.

$[\lambda_{\text{lower}}, \lambda_{\text{upper}})$, the stationary state of the partial differential equation (PDE) below

$$\begin{cases} f(\varphi) \equiv -\left[\frac{\partial \mathcal{E}}{\partial \varphi}(\nabla\varphi)\right]^{\mathsf{T}} = Pv, & (a) \\ \frac{\partial \varphi}{\partial t} = (\nabla\varphi)v, & (b) \end{cases} \tag{4}$$

satisfies $f(\varphi) = 0$, a necessary condition for a diffeomorphic transformation φ to be a local minimum state of $\mathcal{E}$.

It should be noted that the stationary state of the PDE can be a local minimum or a saddle point, and that the theorem does not require P to be stationary, but the eigenvalues of P to always be in a constant positive range. Thus, there are multiple choices for P (e.g., $P = I$), and it plays a crucial role in designing the registration algorithm.

The PDE can be solved as follows. According to (4b), for a short time duration τ, the transformation at time $t + \tau$ can be first-order approximated by that at t as $\varphi^{t+\tau} \approx \varphi^t + (\nabla\varphi^t)u(\tau)$ where $u(\tau) = \int_t^{t+\tau} v d\omega$ and ω is a dummy variable for integration. The registration force can be first order approximated as $f(\varphi^{t+\tau}) \approx f(\varphi^t) - Hu(\tau)$ where H is the Hessian operator of the energy function $\mathcal{E}$ for approximating $f(\varphi^{t+\tau})$ with $f(\varphi^t)$. H is self-adjoint and usually estimated in a certain way rather than being exact. These approximations allow us to convert the PDE around time t to an ODE regarding u as

$$\begin{cases} f(\varphi^{t+\tau}) \approx f(\varphi^t) - Hu \approx P\frac{du}{d\tau} \\ u|_{\tau=0} = 0 \end{cases} \tag{5}$$

and the semi-implicit solution of the update as

$$u(\tau) = (P/\tau + H)^{-1} f(\varphi^t). \tag{6}$$

Then the transformation is updated with $\varphi^{t+\tau} = \varphi^t \circ \exp(u)$.

C. Demons Interpreted as PDE

The Demons algorithm can be regarded as minimizing the energy function $\mathcal{E}(\varphi) = (1/2)\|F - M \diamond \varphi\|^2 + (1/2)\langle \nabla\varphi, \nabla\varphi \rangle$ in the PDE framework with the semi-implicit scheme.

$\mathcal{E}(\varphi)$ can be split as $\mathcal{E}_{\text{sim}}(\varphi|F, M) = (1/2)\|F - M \diamond \varphi\|^2$ and $\mathcal{E}_{\text{trans}}(\varphi) = (1/2)\langle R\varphi, R\varphi \rangle$. According to Propositions 3 and 4 (in Appendices C and D), the registration forces and the Hessian matrices of the two energy functions are

$$\begin{cases} f_{\text{sim}}(\varphi) = g_{\text{sim}}^{\mathsf{T}} b_{\text{sim}} \\ H_{\text{sim}}(\varphi) \approx g_{\text{sim}}^{\mathsf{T}} g_{\text{sim}} \end{cases}, \quad \begin{cases} f_{\text{trans}}(\varphi) = g_{\text{trans}}^{\mathsf{T}} b_{\text{trans}} \\ H_{\text{trans}}(\varphi) \approx g_{\text{trans}}^{\mathsf{T}} g_{\text{trans}}. \end{cases} \tag{7}$$

where the differences b_{sim} and b_{trans} and the gradients g_{sim} and g_{trans} are

$$\begin{cases} b_{\text{sim}} = F - M \diamond \varphi \\ g_{\text{sim}} = \frac{\partial (M \diamond \varphi)}{\partial \varphi} \nabla\varphi \end{cases}, \quad \begin{cases} b_{\text{trans}} = R\varphi \\ g_{\text{trans}} = -R\nabla\varphi \end{cases}. \tag{8}$$

Steps 1 and 4 of Algorithm 1 solve (4) with time-splitting of registration forces. In Step 1, the registration force derived from $\mathcal{E}_{\text{sim}}(\varphi|F, M)$ is active, the update calculated according to (6), excluding $f_{\text{trans}}(\varphi)$ and $H_{\text{trans}}(\varphi)$, is

$$u = \left(P/\tau + g_{\text{sim}}^{\mathsf{T}} g_{\text{sim}}\right)^{-1} g_{\text{sim}}^{\mathsf{T}} b_{\text{sim}}. \tag{9}$$

For scalar images, its transformation only requires composition, so g_{sim} equals the gradient of the warped image

$$g_{\text{sim}} = \nabla(M \circ \varphi). \tag{10}$$

In this case, g_{sim} is a block diagonal matrix, whose diagonal blocks are $g_{ii} = \nabla_i(M \circ \varphi)$, the gradient of the warped image at each voxel location i. If we choose $P/\tau = \text{diag}(I/\tau_1, \ldots, I/\tau_n)$ (where n is the number of voxels), a diagonal matrix whose diagonal blocks are I/τ_is, then the solution of (9) is $u_i = g_{ii}^{\mathsf{T}} b_i / (g_{ii}^{\mathsf{T}} g_{ii} + 1/\tau_i)$, as in Step 1c of the algorithm.

In Step 4, the registration force derived from $\mathcal{E}_{\text{trans}}(\varphi)$ is active, yet for simplicity the diffeomorphic constraint is neglected. Without the diffeomorphic constraint and with R being the gradient operator, the regularization force is $f_{\text{trans}} = -\triangle\varphi$. This force drives a diffusion process and the solution is a Gaussian smoothing of φ, as discussed in Section VI.

D. Comments on Demons

Two important technical points contribute to the success of the diffeomorphic Demons algorithm. First, its computational complexity is linear for scalar images. Inspired by diffusion processes, the algorithm determines the velocity at each voxel location with only local information, and calculates it in a constant time. This makes the total complexity of an iteration $O(n)$ where n is the number of voxels, very suitable for large image registration.

Second, the method can effectively control the update at each voxel location within a certain region $\|u_i\| < \gamma$ (where γ is a positive real value), guaranteeing that the assumption used in (5) is not violated. In (5), $f(\varphi^{t+\tau})$ is approximated by its first-order Taylor expansion, intrinsically requiring u_i to stay within a trust region $\|u_i\| < \gamma$ when $f(\varphi)$ is nonlinear, as discussed in Section IV-C. Thirion in [12] chose $1/\tau_i = \|b_i\|^2$ for Step 1c of the algorithm to guarantee $\|u_i\| < 0.5$.

IV. Computational Challenges Introduced by Tensor Reorientation

Registration of diffusion tensor images requires reorientation of tensors. In this section, after a summary on the FS reorienta-

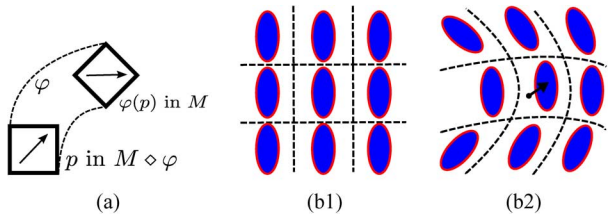

Fig. 1. (a) Vector reorientation. (b1) Tensors before transformation. (b2) Tensors after transformation. This figure is by no means to be mathematically rigorous, but just for illustration purposes.

tion, we discuss the computational challenges of incorporating tensor reorientation into registration procedure.

A. Tensor Reorientation

Transforming a DTI image M with a transformation field φ requires not only composition, but also reorientation of the tensors to keep them aligned with the transformed space, as illustrated in Fig. 1. The tensor value at a location p in the transformed image $M \diamond \varphi$ is $[M \diamond \varphi](p) = r_p M[\varphi(p)] r_p^{\mathsf{T}}$, where r_p is a rotation matrix derived from $J_p = \nabla_p \varphi$, the Jacobian matrix of the transformation φ at p.

The two most popular rotation strategies are the finite strain (FS) strategy [4] and the preservation of principal direction (PPD) strategy [4]. In our registration algorithm, we incorporate the FS strategy into the optimization procedure because it allows us to analytically derive the gradient and Hessian matrix for optimization. The FS strategy calculates the rotation matrix with

$$r_p(J_p) = J_p^{\mathsf{T}} \left(J_p J_p^{\mathsf{T}} \right)^{-\frac{1}{2}}. \tag{11}$$

As derived in [2], [10] and Appendix F, with the FS reorientation, the gradient g_{sim} in (8) is

$$g_{\text{sim}}(\varphi) \approx \nabla(M \diamond \varphi) + \frac{1}{2}(\nabla^{\mathsf{T}} - \nabla) \triangleleft (M \diamond \varphi) + \frac{1}{2}(M \diamond \varphi)(\nabla - \nabla^{\mathsf{T}}) \tag{12}$$

where "$\triangleleft$" is a functor on operators such that given two linear operators O_1 and O_2, $(O_1 \triangleleft O_2)$ satisfies $(O_1 \triangleleft O_2)\bullet = (O_1 \bullet) O_2$. The "$(\nabla^{\mathsf{T}} - \nabla)$" term is not applied to $M \circ \varphi$, but actually to the perturbation direction.

B. Large Inseparable Linear System

Because displacing a voxel will not only change its tensor value but also reorient those of its adjacent neighbors, the impacts of the velocity at adjacent voxels correlate. As a result, the gradient $g_{\text{sim}}(\varphi)$ involves the differential operator ∇, as shown in (12), and the optimization Hessian matrix H has an inseparable sparse block structure. In such a situation, calculating the update u with (9) demands cumbersome computation. Yeo's DTI registration work [2] solves a large inseparable linear system at every iteration, and does not preserve the linear complexity of the diffeomorphic Demons algorithm for scalar images.

Some special sparse linear systems can be solved efficiently, but tensor reorientation may not meet their special requirements. For example, the fast Fourier transform (FFT) [23] and the discrete cosine transform (DCT) [24] can respectively solve the problems of elastic regularization [20] and curvature regularization [13] at complexity $O[n \ln(n)]$, but they require periodical patterns of the linear system. The additive operator splitting (AOS) scheme [15], in combination with the Thomas algorithm [18], [25], can solve the problem of diffusion regularization [14], [16] at $O(n)$ complexity, but the Thomas algorithm requires diagonal dominance. Because the Hessian matrix H depends on input images, given the diversity of DTI images, the special requirements of these fast algorithms cannot always be satisfied.

C. Violation of Local Approximation

It is nontrivial to design a regularizer P such that the solution of (9) satisfies the local trust region condition $\|u_i\| < \gamma$. Limiting $\|u_i\|$ at each voxel location is crucial to the robustness of the algorithm, for the following reasons.

1) In (5), $f(\varphi^{t+\tau})$ is approximated by its first-order Taylor expansion, and the approximation takes place at each voxel location, which intrinsically requires each u_i to stay in a small region to make the approximation valid. The Hessian matrix H typically depends on the gradient of the input images. If the gradient is calculated with the central difference method, then the gradient value at a voxel location is valid within a cube whose radius is the voxel spacing. Outside this cube, the gradient may take different values and the approximation is invalid.
2) Violating the approximation locally at a certain voxel i may not considerably impact the energy function, but will lead to undesirable local inaccuracy. Fig. 3 shows the effect of lifting up the local trust region constraint.
3) If the composition rule $\varphi^{k+1} = \varphi^k \circ (I + u)$, instead of the diffeomorphic rule $\varphi^{k+1} = \varphi^k \circ \exp(u)$, is used to update the transformation, limiting $\|u_i\|$ to be smaller than 0.5 voxel spacing avoids overlapped displacements and guarantees invertability.

In general, it is very difficult to predict the solution of a large inseparable linear system without actually solving it. Although given γ we can choose $P = (1/n)(\|b\|/2\gamma)^2 I$ to achieve $(1/n)\sum_{i=1}^{n} \|u_i\|^2 < \gamma^2$, this does not guarantee $\|u_i\| < \gamma$, as illustrated in Fig. 2.

V. Velocity Regularized in Local Trust Regions

As responses to the computational challenges introduced by tensor reorientation, in this section, we propose a method for DTI image registration which 1) takes the FS tensor reorientation into exact account, 2) is of $O(n)$ complexity, and 3) guarantees the update u to satisfy $\|u_i\| < \gamma$ at every voxel location.

The method is based on a regularized partial differential equation (PDE) procedure whose velocity field v is guaranteed to stay within local trust regions $\|v_i\| < \gamma$ at any time and at every voxel location. With such a regularization, "particles" in the transformation model cannot move at a speed faster than γ, so given a fixed time duration $[t, t+\tau)$ (where τ typically is 1), any particle cannot move further than $\gamma\tau$. In this way, the transformation update is rigorously restricted within the local

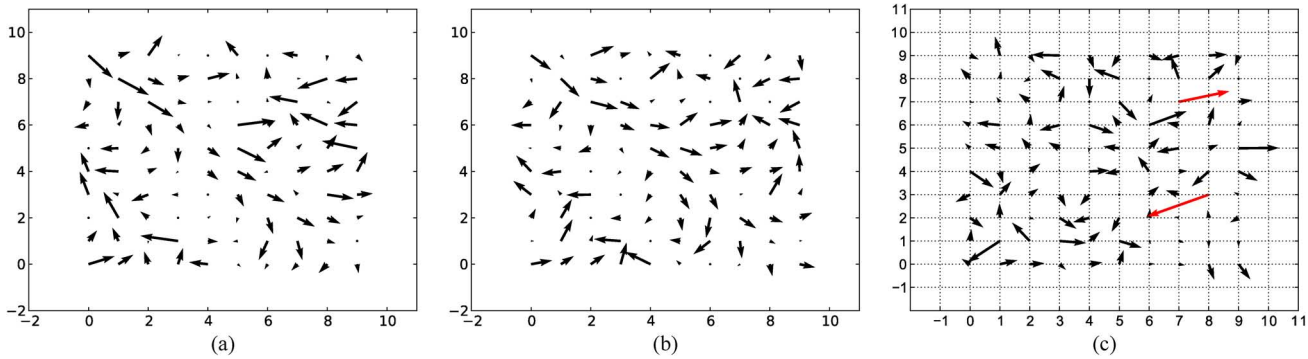

Fig. 2. Vector image registration. (a) Fixed image. (b) Moving Image. (c) Update field u calculated with the semi-implicit method, Eq.(9), where the regularizer P/τ is set to $(1/4n)\|b\|^2 I$ (n is the number of voxels), to guarantee that the mean squared local update $(1/n)\sum_{i=1}^{n}\|u_i\|^2 = \|u\|^2/n$ is not greater than 1, according to Proposition 2. As shown in the figure, the largest local update $\|u_i\|$ is greater than 2. This example shows that due to local velocity interactions, regularization on $\|u\|$ does not necessarily locally control $\|u_i\|$.

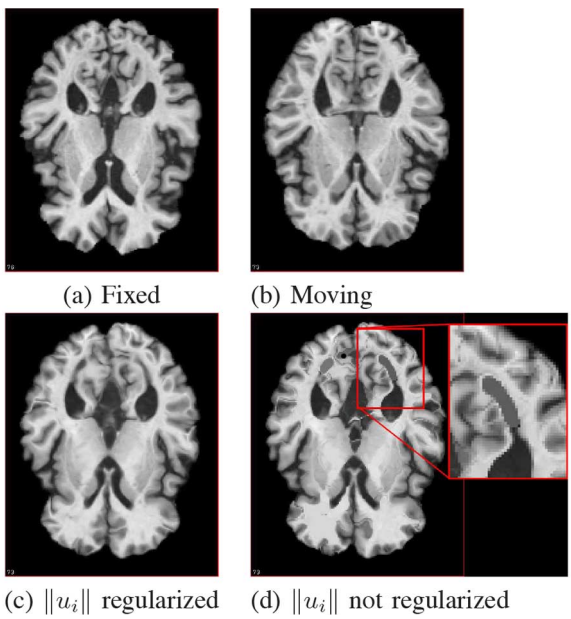

Fig. 3. Impacts of regularizing transformation update within local trust regions, demonstrated with the diffeomorphic Demons algorithm [1] with a fluid-like regularization. A moving image (b) is registered to a fixed image (a) with the algorithm, with the width of the update smoothing kernel set to 1.5 voxel units. By default, the scalar version of the diffeomorphic Demons algorithm automatically regularizes $\|u_i\|$ to be smaller than 0.5 voxel spacing. Its result (c), though imperfect, does not show strong defects. With the same registration parameters, but with the local trust region constraint turned off, the result (d) shows obvious defects.

trust regions $\|u_i\| < \gamma\tau$. At each iteration, the method does not solve a large sparse linear system. Rather, it employs numerical techniques to calculate u by integrating the velocity v over the time duration. We first discuss the intuitions behind this design, and then present the method and analyze its computational complexity.

A. Intuition 1

Shorter Time Steps Should Require Less Computation: As shown in (6), the update u derived from PDE (4) with the semi-implicit scheme is

$$u = (P + \tau H)^{-1} f\tau = (P/\tau + H)^{-1} f.$$

If we aggressively set $\tau \to \infty$, then the solution is $u \to H^{-1}f$, the same as the Newton's method. In this case, we need to solve the linear system $Hu = f$, which in general is computationally very expensive. If we choose $P = I$ and conservatively set $\tau \to 0$, then the solution is $u \to f\tau$, the same as the gradient descent method. In this case, we just need to calculate the registration force f, which usually is not very difficult. The smaller τ is, the closer u is to $f\tau$, so by intuition we expect that shorter time steps require less computation. If we restrict $\|u_i\| < \gamma$ to avoid violating the Taylor expansion used behind (6), we expect τ to be small.

B. Intuition 2

The Transformation φ is a Diffeomorphic Integration of Velocity v: Equation (4a) specifies the velocity, and (4b) specifies the integration rule. The exponential $\exp(u)$ is the transformation increment contributed by the time interval $[t, t+\tau)$. At each iteration, the PDE (4) is approximated as an ODE (5) where the update $u = \int_t^{t+\tau} v d\omega$ is the integration of the velocity. The update u can be calculated with numerical integration. In general, the longer the time step we use, the more computation is required because the integration is over a longer time duration. If τ approaches 0, $u \approx f\tau$ is the first-order approximation of the integration.

C. Intuition 3

Regularizing the Velocity at Different Voxel Locations Differently can Achieve Better Computational Efficiency Than Treating Them Uniformly: The exact solution of ODE (5) is $u = (I - e^{-P^{-1}H\tau})H^{-1}f$ and the eigenvalues of $P^{-1}H$ determine the convergence rate of the Taylor approximation of u. If H is a block-diagonal matrix $\mathrm{diag}(h_{11}, \cdots h_{nn})$ and P equals I, then the diagonal blocks of $P^{-1}H = H$ determine the convergence rate. If h_{ii}s are not uniform, different voxel locations will need different orders of Taylor expansion to achieve the same level of decimal accuracy. If P treats different voxel locations adaptively according to H so that $P^{-1}H$ is similar to a uniform block-diagonal matrix, for example $P^{-1}H \approx \lambda I$, then the Taylor expansion converges roughly at the same speed at all voxel locations.

D. Possible Solutions

Based on the aforementioned intuitions, the norm of u_i and the computational complexity are determined by τ and P, so we have the following options to restrict the update u within local trust regions.

1) gradually increase τ in small increments to find the best time duration τ that satisfies the local trust region condition, or
2) solve u with a predefined time duration τ and then scale it into local trust regions if it is too large, which is a popular method, or
3) design a velocity regularizer P to keep the velocity within local trust regions, that is $\|u_i\| < \gamma$, and integrate the velocity over a constant given time duration, for example $\tau = 1$.

We choose the third method for the following reasons.

1) With the first method, the time duration τ may need to be adjusted with an unpredictable number of trials, so the computational cost may be unpredictable.
2) With the second method, a considerable amount of unnecessary computation may be introduced. For example, with the Newton's method we first calculate $u = H^{-1}f$, and then scale it into local trust regions. This is equivalent to first integrating v over the time duration $[t, t+\infty)$, and then scaling u to be small enough.
3) With the third method, given a predefined time duration τ, the update always automatically satisfies $\|u_i\| < \tau\gamma$. This ensures that the computation cost for calculating u is fully determined by the desired accuracy order.

E. Adaptive Velocity Regularizer

The key point for regularizing the velocity within local trust regions at any time and at any voxel location is designing the regularizer P. As long as the registration force and the Hessian matrix are in the form of $f = g^{\mathsf{T}}b$ and $H = g^{\mathsf{T}}g$, we can design a block diagonal matrix $P_{\text{reg}}(g, b, \tau)$ with Algorithm 2 to satisfy the local trust region requirement according to the following proposition.

Proposition 2: Let g be an $m \times n$ matrix, b a column vector of dimension m. Let P be $P \equiv g^{\mathsf{T}}g + \langle b, b\rangle I$, an $n \times n$ square matrix, then $v = P^{-1}g^{\mathsf{T}}b$ satisfies $\|v\| \le 1/2$.

The proof of Proposition 2 is included in Appendix E. The regularizer designed in Algorithm 2 is a block diagonal matrix, and the local velocity regularized by $P_{\text{reg}}(g, b, \tau)$ is

$$v_i = P_{ii}^{-1}g_{\bullet i}^{\mathsf{T}}b = P_{ii}^{-1}g_{\bullet i}^{\mathsf{T}}\hat{b}_i = 2\gamma P_{ii}^{-1}g_{\bullet i}^{\mathsf{T}}\left(\frac{1}{2\gamma}\hat{b}_i\right) \tag{14}$$

and according to Proposition 2, we have $\|v_i\| \le 2\gamma(1/2) = \gamma$.

This design of P is inspired by Thirion's Demons algorithm: the "demon" at each voxel location does its best to minimize the energy function. For an energy function $\mathcal{E}(\varphi) = (1/2)\|F - M \circ \varphi\|^2 + (1/2)\langle R\varphi, R\varphi\rangle$, according to Propositions 3 and 4 in the Appendix, we have $\mathcal{E}(\varphi) = (1/2)\langle b, b\rangle$, and $g = -(\partial b/\partial\varphi)\nabla\varphi$. $g_{\bullet i}$ (the columns of g associated with velocity v_i) is the impact of v_i on b. $\hat{b}_i$ (the column vector derived from b by masking it with nonzero rows of $g_{\bullet i}$) is the contribution of b to $\mathcal{E}$ related to v_i. If v_i is optimized separately, it shall minimize $\langle \hat{b}_i, \hat{b}_i\rangle$, and consequently we have $g_{\bullet i}v_i = \hat{b}_i$, or equivalently $v_i = (g_{\bullet i}^{\mathsf{T}}g_{\bullet i})^{-1}g_{\bullet i}\hat{b}_i$. To regularize v_i within its local trust region $\|v_i\| < \gamma$ and to handle the possible singularity of $g_{\bullet i}$, $\langle \hat{b}_i/(2\gamma), \hat{b}_i/(2\gamma)\rangle I$ is added to $g_{\bullet i}^{\mathsf{T}}g_{\bullet i}$. Therefore, we have $P_{ii} = g_{\bullet i}^{\mathsf{T}}g_{\bullet i} + \langle \hat{b}_i/(2\gamma), \hat{b}_i/(2\gamma)\rangle I$.

Algorithm 2 Local Trust Region Velocity Regularizer

1) Let $g_{\bullet i}$ denote the columns of g which are associated with local velocity v_i. For an N-dimension image registration problem, $g_{\bullet i}$ is a matrix of N columns.
2) Let $\hat{b}_i$ be a masked column vector derived from b. If a row in $g_{\bullet i}$ has at least one nonzero value, then the corresponding row in $\hat{b}_i$ takes the same value as in b. Otherwise, a row in $\hat{b}_i$ takes 0 as its value.
3) We design P as the block diagonal matrix defined in Eq. (13)

$$\begin{cases} P_{\text{reg}}(g, b, \gamma) = \begin{bmatrix} P_{11} & & \\ & \ddots & \\ & & P_{nn} \end{bmatrix}, & (a) \\ P_{ii} = g_{\bullet i}^{\mathsf{T}}g_{\bullet i} + \left\langle \frac{1}{2\gamma}\hat{b}_i, \frac{1}{2\gamma}\hat{b}_i \right\rangle I, & (b) \end{cases} \tag{13}$$

where P_{ii} is an N-by-N matrix.

As shown in Theorem 1, the velocity regularizer P does not have to be stationary and can adaptively change. As long as its eigenvalues stay in a constant positive range, the stationary point of the PDE (4) satisfies $f(\varphi) = 0$, a necessary condition for φ to be a local minimum of $\mathcal{E}$.

We do not choose $P = I$, which corresponds to the steepest descent direction $v_i = g_{\bullet i}^{\mathsf{T}}\hat{b}_i$, for the following reason. Because g equals $-(\partial b/\partial\varphi)\nabla\varphi$, image regions with large $\|\hat{b}_i\|$ tend to also have large $\|g_{\bullet i}\|$. As a result, $\|v_i\| = \|g_{\bullet i}^{\mathsf{T}}\hat{b}_i\|$ in these regions will be much larger than those in other regions, suppressing the displacement of voxels in other regions. Though this suppressing effect in theory will gradually disappear as $\|\hat{b}_i\|$ approaches zero, with $g_{\bullet i}$ calculated on a discrete grid it may never be achieved, and the suppressing effect will constantly prevent the displacement of voxels in other regions. We observed this phenomenon when choosing $P = I$.

F. Integration Instead of Inversion

For energy functions whose registration force has the form of $f = g^{\mathsf{T}}b$ and whose Hessian matrix has the form of $H = g^{\mathsf{T}}g$, (5) becomes $g^{\mathsf{T}}(b + gu) = P(du/d\tau)$. By applying the local-trust-region velocity regularizer designed in Algorithm 2 to it, we have the following ODE about the update u

$$\begin{cases} \frac{du}{d\tau} = v(u) = P_{\text{reg}}^{-1}(g, b + gu, \gamma)g^{\mathsf{T}}(b + gu), & (a) \\ u|_{\tau=0} = 0 & (b) \end{cases}. \tag{15}$$

We can estimate the value of u at time τ with the Runge Kutta methods [18]. The mth-order Runge Kutta method evaluates equation (15a) m times and offers mth-order accuracy regarding τ. We call this method the fast local trust region (fLTR) method. Algorithm 3 illustrates the fLTR method with the second-order Runge Kutta (RK2) integration.

We choose the Runge Kutta methods instead of the semi-implicit Euler method for the following reason. First, though the semi-implicit method may provide more stability than the

Algorithm 3 Fast Local Trust Region Method with RK2 Integration.

In each registration iteration:

1) Calculate b and g according to Eq. (8). To incorporate tensor reorientation, g_{sim} is calculated according Eq. (12).
2) Define $v(\bullet)$ as Eq. (15a) with the adaptive velocity regularizer P_{reg} defined in Algorithm 2, and γ specified by the user, for example, 0.5 voxel spacing.
3) Let $\tau = 1$.
4) Let $v^0 = v(0)$.
5) Let $v^1 = v(v^0\tau)$.
6) Let $u = \frac{1}{2}(v^0 + v^1)\tau$.

Runge Kutta methods, because we have rigorously limited u_i in local trust regions, we can expect Algorithm 3 to be robust. Second, the semi-implicit method solves a large linear system and demands much more computation.

Yeo *et al.* in [2] solved u with the semi-implicit method and an empirical choice of P/τ. In Section VII, we compare the proposed local trust region method with the semi-implicit method in [2].

G. Computational Complexity

The computational complexity of the algorithm is $O(nm)$ within one registration iteration, where n is the number of voxels and m is the order of the Runge Kutta method. For the mth-order Runge Kutta method, equation (15a) is evaluated m times. For each evaluation, function (14) is evaluated for every voxel. For a particular voxel, its transformation φ_i only impacts the warped voxel values of itself and those in its neighborhood, as well as the regularization energy at itself and those in its neighborhood. Consequently, the number of nonzero entries in $g_{\bullet i}$ is a constant number c independent of n. Therefore, each evaluation of (14) can be finished in a constant time depending only on c. Therefore, the computational complexity for an iteration is $O(nm)$.

VI. Regularization

The diffeomorphic Demons algorithm (Algorithm 1) separates the regularization as an individual smoothing procedure applied to the transformation, which is very elegant for software development, and very efficient in computation. This implementation is theoretically rigorous for the additive transformation update rule, yet not consistent with the diffeomorphic framework.

With the additive update rule $d\varphi/dt = v$, the registration force derived from transformation energy $\mathcal{E}_{\text{trans}}(\varphi) = 1/2\langle\nabla\varphi, \nabla\varphi\rangle$ is $f_{\text{trans}}(\varphi) = -\triangle\varphi$ where $\triangle$ is the Laplace operator. Because this force drives a homogeneous and isotropic diffusion process, its contribution can be implemented as smoothing the transformation with its Green's function, the Gaussian smoothing kernel, as shown in [14] by Fisher and Modersitzki, and in [26] by Cahill *et al.*

However, with the diffeomorphic update rule $d\varphi/dt = (\nabla\varphi)v$, the registration force for $\mathcal{E}_{\text{trans}}(\varphi) = (1/2)\langle R\varphi, R\varphi\rangle$ is $f_{\text{trans}}(\varphi) = -(\nabla\varphi)^{\mathsf{T}}(R^{\mathsf{T}}R\varphi)$ which cannot be implemented as a separated Gaussian smoothing procedure. If this force is included in the calculation of velocity, its Hessian matrix $H_{\text{trans}} = \langle R\nabla\varphi, R\nabla\varphi\rangle$ is an inseparable sparse linear system, which in general cannot satisfy the special requirements of the fast algorithms discussed in Section IV-B. Ashburner in [27] incorporated the regularization on transformation into the calculation of velocity, but assumed a stationary rather than time-variant velocity field, which is not suitable for large deformation.

Vercauteren *et al.* in [1] noticed this incompatibility, but kept it due to its elegance for software implementation and efficiency in computation, though it may prevent the algorithm from generating the inverse transformation in one registration. In [28], the authors proposed a symmetric version of the Demons algorithm which is able to generate the inverse,but they assumed a stationary velocity field.

The fast local trust region (fLTR) algorithm is able to embed the regularization term $\mathcal{E}_{\text{trans}}(\varphi)$ directly into the calculation of the update, because its registration force f_{trans} and Hessian matrix H_{trans} are respectively in the format of $g^{\mathsf{T}}b$ and $g^{\mathsf{T}}g$, consistent with the requirement of Algorithms 2 and 3. We implemented a second-order regularization which we call the "affinity" regularization.

The second derivative of an affine transformation is zero. If we punish the second derivative of a transformation, we have the regularization energy defined in (16), which we call the affinity regularization, where $\text{Hessian}(\varphi, p, d)$ is the Hessian matrix of the dth component of φ at location p, and $\|\bullet\|$ denotes the Frobenius norm of a matrix

$$\mathcal{E}_{\text{affinity}}(\varphi) = \frac{1}{2}\int_{\Omega}\sum_{d=1}^{dim}\|\text{Hessian}(\varphi, p, d)\|^2\, dp. \tag{16}$$

Its discrete format, as illustrated with a 2-D unit grid, is (17), where $p = (x, y)$

$$\begin{aligned}\mathcal{E}_{\text{affinity}}(\varphi) = &\sum_{d=1}^{2}\sum_{x,y}\left(\varphi^d_{x,y} + \varphi^d_{x+1,y+1} - \varphi^d_{x+1,y} - \varphi^d_{x,y+1}\right)^2 \\ &+ \frac{1}{2}\sum_{d=1}^{2}\sum_{x,y}(\varphi^d_{x+1,y} + \varphi^d_{x-1,y} - 2\varphi^d_{x,y})^2 \\ &+ \frac{1}{2}\sum_{d=1}^{2}\sum_{x,y}(\varphi^d_{x,y+1} + \varphi^d_{x,y-1} - 2\varphi^d_{x,y})^2. \end{aligned}\tag{17}$$

If a differentiable transformation satisfies $\mathcal{E}_{\text{affinity}}(\varphi) = 0$, then it is an affine transformation.

Our energy function involving $\mathcal{E}_{\text{affinity}}(\varphi)$ has the format $\mathcal{E} = \mathcal{E}_{\text{sim}}(\varphi|F, M) + w_{\text{affinity}}\mathcal{E}_{\text{affinity}}(\varphi)$, where the weight parameter w_{affinity} adjusts the importance of $\mathcal{E}_{\text{affinity}}(\varphi)$ relative to $\mathcal{E}_{\text{sim}}(\varphi|F, M)$.

VII. Experiments

A. Data Acquisition and Preprocessing

Diffusion weighted images (DWI) of 84 pediatric subjects, 7–17 years old, were collected with 30-direction isotropic DTI

sequences ($\mathrm{b} = 1000\ \mathrm{s/mm^2}$, voxel size $= 2 \times 2 \times 2\ \mathrm{mm^3}$, $\mathrm{dimension} = 128 \times 128 \times 128$). During the adolescence period, white matter volume or FA increases significantly throughout multiple regions of the central neural system [29]–[32], so the registration of this age group is very challenging. The FSL brain extraction tool (BET) was applied to the B0 images to mask brain regions, with the "-R" option turned on and the fractional intensity threshold (the "-f" option) set to 0.3. Diffusion tensor images (DTI) were reconstructed with the "DiffusionTensor3DReconstructionImageFilter" implemented in the "Insight Segmentation and Registration Toolkit" (ITK).[2] The diffusivity unit used is $10^{-3}\ \mathrm{mm^2 \cdot s^{-1}}$. With this unit, the mean diffusivity for cerebral spinal fluid is roughly 3. Negative eigenvalues of the reconstructed tensors were rectified to their absolute values. Before deformation registration, all the images were linearly aligned to the International Consortium for Brain Mapping (ICBM) fractional anisotropy (FA) template[3] based on their FA images. Tensors were linearly interpolated and reorientated with the preservation-of-principal-direction (PPD) method [4] when the affine transformations were applied.

B. Evaluated Methods

The proposed local trust region method was compared with a semi-implicit method, which does not enforce the local trust region constraint directly into its velocity regularization [2]. In [2], the authors proposed a DTI registration method which directly incorporates FS tensor reorientation into its velocity calculation. In each iteration, the update u is calculated with the semi-implicit method as $u = (P/\tau + H)^{-1} f$, where P/τ is an empirically specified diagonal matrix. As shown in Section IV-A, in such a situation the Hessian matrix H is an inseparable large sparse matrix. The algorithm regularizes the velocity field by smoothing it, as in Step (2) of Algorithm 1, to achieve a fluid-like regularization. Because this method involves matrix inversion and applies a fluid-like regularization, we refer to it as the "MatInv-fluid" method. It should be noted that even with the P designed according to (13), the solution of $(P/\tau + H)^{-1} f$ is not guaranteed to be within local trust regions because it involves the inversion of an inseparable linear system whose result is very difficult to predict. A C++ implementation of the algorithm is provided by the authors in the tensor toolkit (TTK) available online.[4]

The proposed local trust region method was implemented in a similar way to the MatInv-fluid method, except the following differences. The velocity regularizer P is designed according to Algorithm 2 to guarantee that at any time point t and for any voxel location i, the amplitude of local velocity v_i is smaller than a predefined constant γ. The update u is calculated by integrating the ODE defined in (15) over a predefined time duration τ, with the Runge Kutta methods [33]. As shown in Section V, such a strategy automatically enforces the amplitude of u_i to be smaller than $\gamma\tau$ at every voxel location at any time point. A fluid-like regularization on velocity is implemented in the same way as the "MatInv-fluid" method. As the proposed local trust region method is able to incorporate the regularization on transformation directly into the calculation of velocity, we also implemented the affinity regularization proposed in Section VI. In the following section, we abbreviate the fast Local Trust Region method with the fluid-like regularization as the "fLTR-fluid" method and that with the affinity regularization on transformation as the "fLTR-affinity" method. An implementation of the fLTR methods is available online.[5]

C. Evaluation Strategy

The "MatInv-fluid," the "fLTR-fluid," and the "fLTR-affinity" methods were evaluated with two registration strategies: pair-wise registration and group registration. In the pair-wise registration, smoothness parameters were tuned over a range for each method to explore the smoothness-similarity relationship. For the MatInv-fluid and the fLTR-fluid method, their smoothness parameters are the width of the smoothing kernel applied to the update u. For the fLTR-affinity method, its smoothing parameter is the weight of the affinity regularization in the energy function.

In the group registration, unbiased templates were iteratively constructed with the framework proposed by Joshi *et al.* in [34], and the cross-subject variance/consistence was evaluated. Each of the methods was used as the pair-wise registration module in the template construction procedure.

All the registration tasks were conducted with three-level multi-resolution procedures. Coarse levels were generated by smoothing and downsampling, and their registration results were forwarded to finer levels as the initial transformations. At each multi-resolution level, iterations were repeated until the metric decrement was smaller than 1%. The local trust region was defined as $\|u_i\| < 0.5$ voxel spacing. After transformation fields were calculated, they were applied with either the PPD or the FS reorientation [4] for evaluation. Because the results with the two reorientation methods are very similar, due to page limit we present in the paper only the PPD results.

D. Pair-Wise Registration: Configuration

100 pairs of images were randomly sampled from the pool of 84 DTI images to evaluate the three registration methods: the MatInv-fluid, the fLTR-fluid and the fLTR-affinity. We tuned the smoothness parameters for each of the methods to let their smoothness, as evaluated with the curvature harmonic energy [13], span roughly the same range. For the MatInv-fluid method, we increased the width of the velocity smoothing kernel from 0.7 to 2.0 voxel units, with an incremental step of 0.1. The smallest smoothing kernel width available in the TTK implementation of the MatInv-fluid method is 0.5. With the kernel width equal to or smaller than 0.6, we experienced numerical instability, so the smallest kernel width we used for the MatInv-fluid method is 0.7. For the fLTR-fluid method, we increased the width of the velocity smoothing kernel from 1.0 to 2.5 voxel units, with an incremental step of 0.1. For the fLTR-affinity method, we used the following values for the regularization weight parameter w_{affinity}: 0.005, 0.025, 0.05, 0.10, 0.15, 0.2, 0.3, 0.4, 0.5, 0.6, 0.9, 1.2, 1.6, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0.

[2]Available online: http://www.itk.org/

[3]Available online: http://www.loni.ucla.edu/atlases

[4]Available online: https://gforge.inria.fr/projects/ttk

[5]Available online: http://ucla.in/SPFgVS

E. Pair-wise Registration: Evaluation Metrics

Two categories of metrics, similarity and smoothness, are used to evaluate the goodness of registration. For pair-wise registration, the similarity is measured between the warped image and the fixed image by directly comparing their tensors, or by comparing scalar indexes derived from the tensors.

In the case of directly comparing tensors, a difference metric is calculated at each voxel location between the warped image and the fixed image, and then all the metric values within the brain region of the fixed image are averaged as

$$\mathrm{Ave} = \frac{1}{|\mathrm{Brain}|} \sum_{i \in \mathrm{Brain}} \mathrm{Diff}\left(D_i^{\mathrm{Warped}}, D_i^{\mathrm{Fixed}}\right) \tag{18}$$

where D_i^{Warped} and D_i^{Fixed} are the diffusion tensors at a voxel location i respectively in the warped image and the fixed image, "Diff" is a difference metric function, "Brain" is the brain region defined on the fixed image, and $|\mathrm{Brain}|$ is the volume of the brain region. The difference metrics we used include the following.

- Squared error (SQE)

$$\mathrm{SQE}_{\mathrm{tensor}}(X, Y) = \|X - Y\|^2. \tag{19}$$

  The squared error is the numerical difference between two tensors and it is a part of the energy function.

- Symmetrized Kullback–Leibler divergence (SymKLD)

$$\mathrm{SymKLD}(X, Y) = \frac{1}{4}\left[\mathrm{Trace}(X^{-1}Y) + \mathrm{Trace}(Y^{-1}X)\right] - 3. \tag{20}$$

  The symmetrized Kullback–Leibler divergence [35], rooted in the information theory, compares the diffusion density distributions represented by tensors, and is able to capture the difference between two tensors in their shapes, sizes, and directions. The symmetrized Kullback–Leibler divergence is not a part of the energy function.

In the case of comparing tensor-based scalar indexes, scalar index images are derived from the warped image and the fixed image, and then the Pearson correlation coefficient between the two derived scalar images are calculated, as shown in (21) at the bottom of page where $x_i^{\bullet}$ is a tensor-based scalar index at voxel i. Squared errors between the derived scalar images were also calculated, and they showed the same trend of goodness among the three registration methods as the correlation coefficient does. Due to space limit, we only present the correlation coefficient results in the paper. The tensor-based scalar indexes we used include the following.

- Mean diffusivity (MD)

$$\mathrm{MD}(D) = \frac{\lambda_1 + \lambda_2 + \lambda_3}{3}. \tag{22}$$

- Fractional anisotropy (FA)

$$\mathrm{FA}(D) = \frac{1}{\sqrt{2}} \frac{\sqrt{(\lambda_1 - \lambda_2)^2 + (\lambda_2 - \lambda_3)^2 + (\lambda_1 - \lambda_3)^2}}{\sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}}. \tag{23}$$

- Tensor volume (TV)

$$\mathrm{TV}(D) = \lambda_1 \lambda_2 \lambda_3 \tag{24}$$

where λ_1, λ_2, and λ_3 are the eigenvalues of a tensor in descending order.

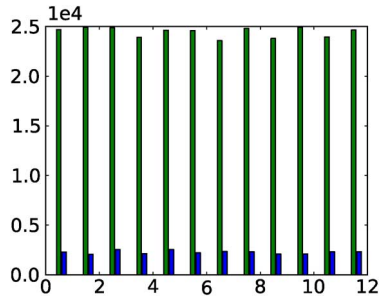

Fig. 4. Estimated computation time of the MatInv-fluid (green) and the fLTR-affinity method (blue). The computation time (in seconds) was estimated with 12 pairwise registration of $128 \times 128 \times 128$ DTI images, on an Intel Xeon 2.80-GHz CPU and single-threaded. The computation time was calculated from the programs' output messages.

F. Pair-Wise Registration: Results

The computation time and memory use of the MatInv-fluid method and the fLTR-affinity method were estimated with 12 registration cases, image of size $128 \times 128 \times 128$, on an Intel Xeon 2.80 GHz CPU, and single-threaded. For each registration case, the MatInv-fluid method took about 2.0×10^4 s (Fig. 4) and more than 20 GB of memory. The fLTR-affinity method took about 2.5×10^3 s (Fig. 4) and about 2 GB of memory. The computation time was calculated from the programs' output messages. The memory usage was manually monitored by using the Linux "top" command when the algorithms were working at the finest resolution of their multi-resolution registration. The fLTR-affinity method reduces the computation time and memory demand about tenfold. Because the fLTR-fluid

$$\mathrm{CC}_x = \frac{\sum_{i \in \mathrm{BrainFixed}} \left(x_i^{\mathrm{Fixed}} - \bar{x}^{\mathrm{Fixed}}\right)\left(x_i^{\mathrm{Warped}} - \bar{x}^{\mathrm{Warped}}\right)}{\sqrt{\sum_{i \in \mathrm{BrainFixed}} \left(x_i^{\mathrm{Fixed}} - \bar{x}^{\mathrm{Fixed}}\right)^2} \sqrt{\sum_{i \in \mathrm{BrainFixed}} \left(x_i^{\mathrm{Warped}} - \bar{x}^{\mathrm{Warped}}\right)^2}}$$
$$\bar{x}^{\bullet} = \frac{1}{|\mathrm{BrainFixed}|} \sum_{i \in \mathrm{BrainFixed}} x_i^{\bullet} \tag{21}$$

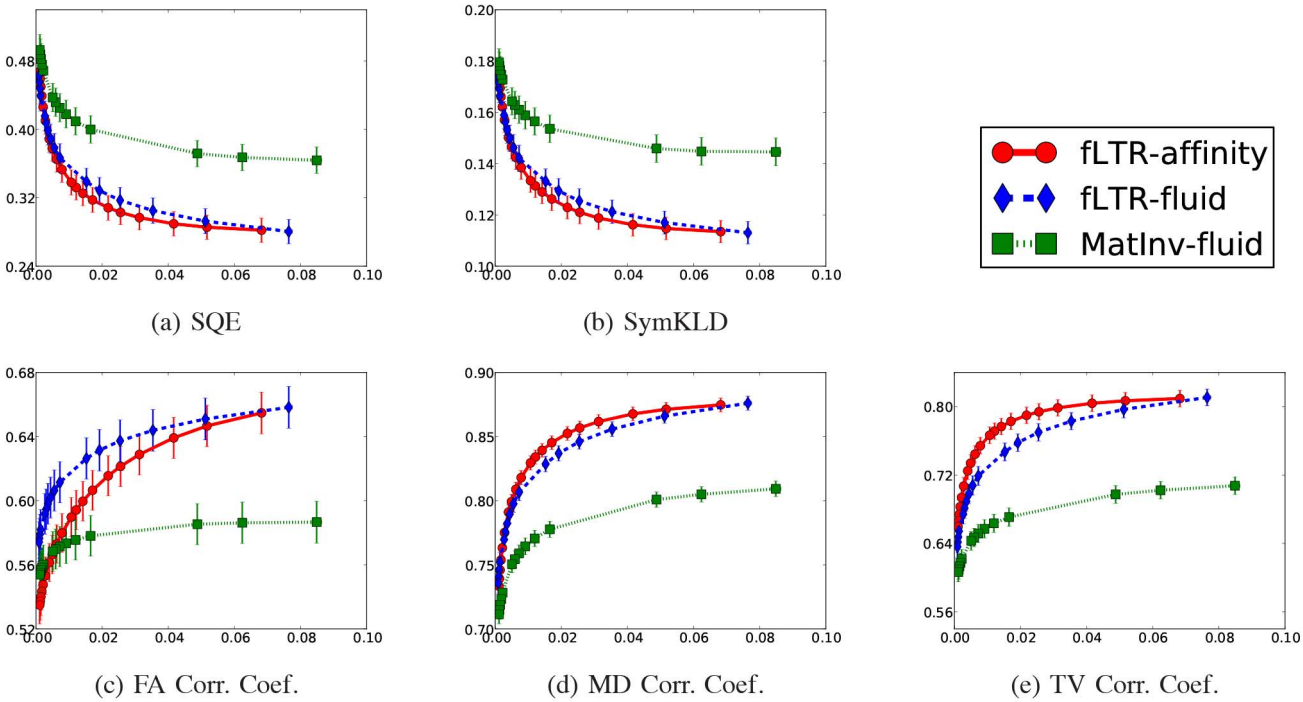

Fig. 5. Quantitative evaluation of pair-wise registration performance, showing the trade-off between the smoothness of transformation and the similarity/dissimilarity between the warped images and the fixed images. In all the sub-figures, the smoothness index (x-axis) is the curvature harmonic energy of transformation [13]. The similarity/dissimilarity index (y-axis) (defined in Section VII-E) of each sub-figure is labeled beneath it. The values of these indexes are the average of the 100 registration pairs, with error bars showing the 95% confidence intervals. For sub-figures (a) and (b), smaller values indicate better matching between the warped and the fixed images; for sub-figures (c)–(e), larger values indicate better matching. As presented in Section VII-B, "fLTR-affinity" stands for the fast local trust region (fLTR) method with the affinity regularization on transformation; "fLTR-fluid" stands for the fLTR method with a fluid-like regularization; "MatInv-fluid" stands for the "Matrix Inversion" method with a fluid-like regularization.

method does not incorporate transformation regularization into its velocity calculation as the fLTR-affinity method does, the fLTR-fluid method took less computation time and memory use.

The quantitative performance comparison of the pair-wise registration is shown in Fig. 5. In all the sub-figures, the smoothness index (x-axis) is the curvature harmonic energy of transformation [13]. The similarity/dissimilarity index (y-axis) of each sub-figure is labeled beneath it. The values of these indexes are the average of the 100 registered pairs, with error bars showing the 95% confidence intervals of these statistics. For the SQE index and the SymKLD index, smaller values indicate better matching between the warped images and the fixed images. For tensor-based scalar indexes FA, MD, and TV, larger correlation coefficients indicate better matching between the warped images and the fixed images. Regarding the indexes shown in Fig. 5, at the same smoothness level, the fLTR methods achieved better matching between the warped images and the fixed images than the MatInv-fluid method did. The 95% confidence-interval bars show that the improvement is statistically significant.

The fLTR-affinity method outperformed the fLTR-fluid method on all indexes except the FA index. The FA index captures tensor properties different from the other indexes do. The tensor SQE and the SymKLD compare tensors' orientation, size and sharpness jointly; the MD and TV indexes compare tensors' size, respectively using the sum and the product of eigenvalues. On the other hand, the FA compares tensors' sharpness. Therefore it is possible that the fLTR-affinity performs better with other indexes but underperforms with the FA index. To optimize the registration performance with the FA index, Zhang *et al.* [10] proposed using deviatoric tensors, as discussed in Section VIII.

The fLTR-affinity method and fLTR-fluid method perform very similarly in this experiment. As a second-order regularization, the affinity regularization does not punish local linear transformations and allows large displacements. On the other hand, the fluid regularization can theoretically achieve any deformation given enough time. This might explain their similar performance. In comparison, the fLTR-fluid method runs a little bit faster because its regularization can be efficiently implemented as a Gaussian smoothing procedure.

Fig. 6 shows examples of the pair-wise registration results. The MatInv-fluid method, which potentially allows large leaps, may introduce abrupt displacements and "break" the continuity of fibers.

G. Group Registration: Configuration

In the group registration, we constructed three DTI templates from the 84 subject DTI images with the unbiased template construction framework proposed by Joshi in [34], respectively using the MatInv-fluid method, the fLTR-fluid method, and the fLTR-affinity method as the pair-wise registration module.

The unbiased template construction framework proposed by Joshi in [34] builds a template in the following way. First, all the input images are averaged to generate an initial intermediate template image. Then all the input images are warped to the

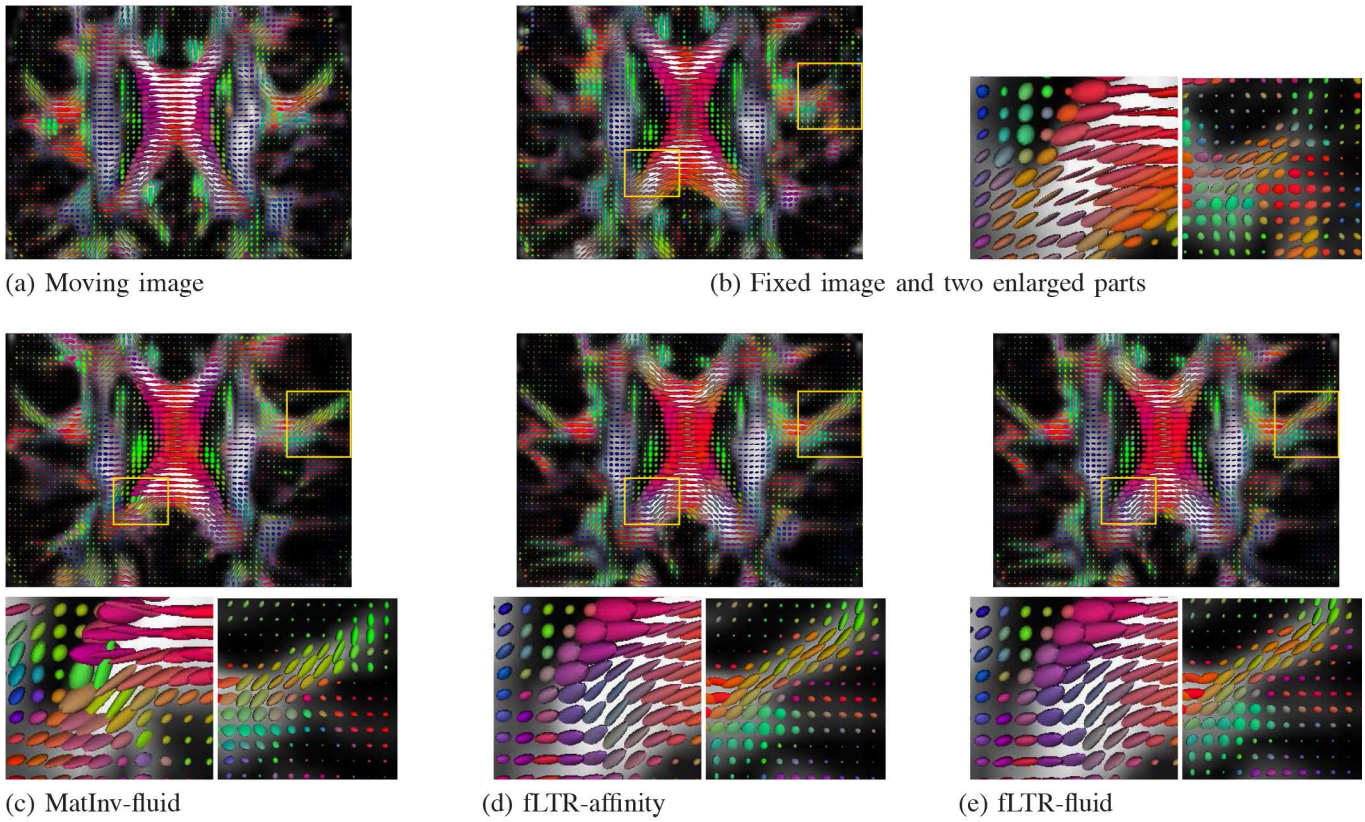

Fig. 6. Examples of pair-wise registration results. (a) Moving image. (b) Fixed image. (c) MatInv-fluid warped method. (d) fLTR-affinity warped image. (e) fLTR-fluid warped image. Images were warped with the PPD reorientation. Two highlighted regions are enlarged.

intermediate template. After the warping, the warped images are averaged to update the intermediate template, and the input images are warped again to the updated intermediate template. This procedure repeats iteratively. We took 30 iterations in our template construction. The warping from an input image to the intermediate template is a pair-wise registration task.

For the MatInv-fluid method, we set the width of the velocity smoothing kernel to 0.8, because with this parameter the SymKLD index defined in (20) became stable in the pair-wise registration evaluation. We chose according to the SymKLD index because it compares tensors via the diffusion probability distribution they represent and is able to capture tensor differences in shapes, directions and sizes. For the fLTR-fluid method, we set the width of the velocity smoothing kernel to 1.1, and for the fLTR-affinity, method we set the weight of the affinity regularization to 0.025, because in the pair-wise registration evaluation, these parameters provided the most transformation freedom without exceeding that chosen for the Matins-fluid method.

H. Group Registration: Evaluation Metrics

In the group registration, we evaluated the registration methods by measuring the dispersion or coherence among all the warped images, within the brain region derived from the template image. In general, at each voxel location we calculated a dispersion or coherence index across all the subject warped images with a function $h(x_i^1, \cdots, x_i^S)$, where x_i^s is a feature, for instance, a diffusion tensor, at a voxel location i in subject s's warped image, and S is the number of subjects. Then we compared the statistical distribution of h in the template images generated by the three registration methods. We also calculated summary statistics by spatially averaging h within the template brain, as

$$\text{AVE} = \frac{1}{|\text{Brain}|} \sum_{i \in \text{Brain}} h\left(x_i^1, \cdots, x_i^S\right) \tag{25}$$

where "Brain" is the brain region in the template image and $|\text{Brain}|$ is its volume.

The metrics we used include the following.

1) Alignment of Tensor Principal Directions: With the second-order approximation, a diffusion probability density function is represented by a diffusion tensor whose principal eigenvector, in the case of anisotropy, is regarded as the most probable direction of the neural fibers. (Readers should note that the second-order approximation is unable to handle the situation of crossing fibers.) We used dyadic tensors to assess the coherence of principal eigenvectors among the warped subject images, as used in [36].

The dyadic tensor E_i at a voxel location i is defined as

$$E_i = \frac{1}{S} \sum_{s=1}^{S} e_i^s \left(e_i^s\right)^{\mathsf{T}} \tag{26}$$

where e_i^s is the principle eigenvector at voxel i in subject s's warped image. The better the principal eigenvectors align across

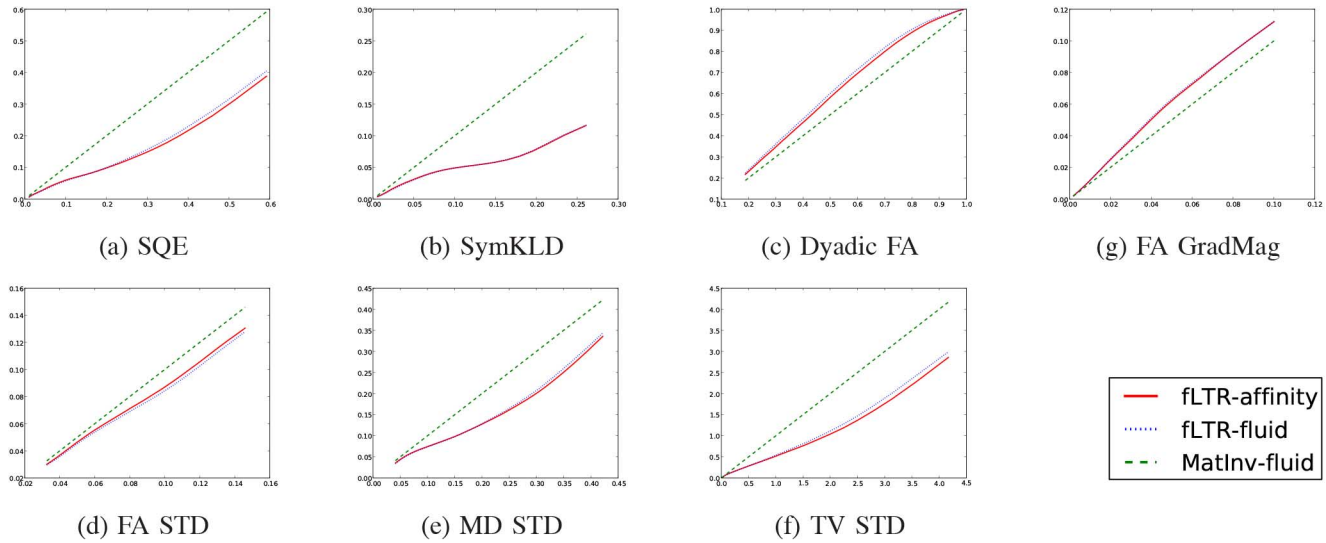

Fig. 7. Quantitative comparison of the dispersion/coherence of the warped subjects images generated for template construction by the fLTR-affinity, the fLTR-fluid, and the MatInv-fluid methods. The coherence/dispersion index used in each sub-figure is labeled beneath it. The distributions of the coherence indexes within the foreground of the constructed template are compared with the QQ plot. The x-axis is the percentiles of the distributions of the MatInv-fluid results. The y-axis is the corresponding percentiles of the distributions under comparison, including the results of the fLTR-affinity, the fLTR-fluid, and the MatInv-fluid methods. The MatInv-fluid versus MatInv-fluid comparison is a self comparison, so it is a line of 45° (green), the baseline of the comparisons. In the QQ plot, distributions with lower values show lower shift, and those with higher values show upper shift. Lower values of the following indexes indicate better coherence among the warped images: the SQE, the SymKLD, the STD of FA, MA, and TV. Higher values of the dyadic FA indicate better coherence among the warped images. Higher values of the template GradMag indicate that the template preserves more details. Regarding the indexes shown in this figure, the warped subject images generated with the fLTR-affinity and fLTR-fluid methods are more coherent than those generated with the MatInv-fluid method.

subjects, the more anisotropic the dyadic tensor is. If the principal eigenvectors of the warped tensors align perfectly, $e_i^s(e_i^s)^{\mathsf{T}}$ should be the same for all the subjects, and the dyadic tensor E_i should be singular and its rank should be 1. If the principal eigenvectors are completely random, E_i should be isotropic, that is, the identity matrix I. We used the FA defined in (23), to estimate the anisotropy of dyadic tensors, that is, we used

$$h_{\text{dyadic}}\left(e_i^1,\cdots,e_i^S\right)=\text{FA}\left[\frac{1}{S}\sum_{s=1}^{S}e_i^s\left(e_i^s\right)^{\mathsf{T}}\right]. \quad (27)$$

2) Template-Subject Similarity: The template image, as a representation of a group of images, is expected to be as similar to all the subject images as possible. We measure this similarity/dissimilarity with the symmetrized Kullback–Leibler divergence and the squared error. The tensor at a voxel location i in the template image is defined as $\bar{D}_i = (1/S)\sum_{s=1}^{S} D_i^s$, where D_i^s is the tensor in subject s's warped image, and S is the number of subjects. The dispersion index function based on the SymKLD is

$$h_{\text{SymKLD}}\left(D_i^1,\cdots,D_i^S\right)=\frac{1}{S}\sum_{s=1}^{S}\text{SymKLD}\left(D_i^s,\bar{D}_i\right). \quad (28)$$

The dispersion index function based on the squared error is

$$h_{\text{SQE}}\left(D_i^1,\cdots,D_i^S\right)=\frac{1}{S}\sum_{s=1}^{S}\left\|D_i^s-\bar{D}_i\right\|^2. \quad (29)$$

3) Deviation of Scalar Indexes: We use the cross-subject deviation of tensor-based scalar indexes to estimate the similarity between the warped subject images. This is based on the assumption that misregistration of nonhomologous fiber tracts will increase voxel-level deviation of these indexes across subjects. The deviation index function we plug into (25) is

$$h_{\text{std}}\left(x_i^1,\cdots,x_i^S\right)=\text{std}\left(x_i^1,\cdots,x_i^S\right) \quad (30)$$

where x_i^s is a scalar index derived from the tensor at a voxel location i in subject s's warped image, including the FA, the MD, and the TV, as defined in (23), (22), and (24), respectively.

4) Template Sharpness: In addition to the aforementioned metrics defined among the warped subject images, the sharpness of the constructed template is also evaluated. By intuition, mismatching among the warped images will result in blurry templates. Though such an idea is backed more by intuition than theory, templates preserving more details are still preferred. We define the sharpness of a template as its gradient magnitude. The larger the gradient magnitude is, the more high frequency information is in an image and the visually sharper the image is. The gradient magnitude index is calculated with the FA images derived from the template DTI images.

I. Group Registration: Results

Fig. 7 shows the quantitative comparison of the dispersion/coherence among the warped subject images generated for template construction by the fLTR-affinity, the fLTR-fluid, and the MatInv-fluid methods. The statistical distributions of the following indexes within the foreground of the constructed template are compared with the quantile-quantile (QQ) plot: the SQE, the dyadic FA, the SymKLD, the standard deviation (STD) of FA, MA, and TV, and the gradient magnitude (GradMag) of the FA image derived from the template DTI image. As shown in Fig. 8, these indexes have become stable

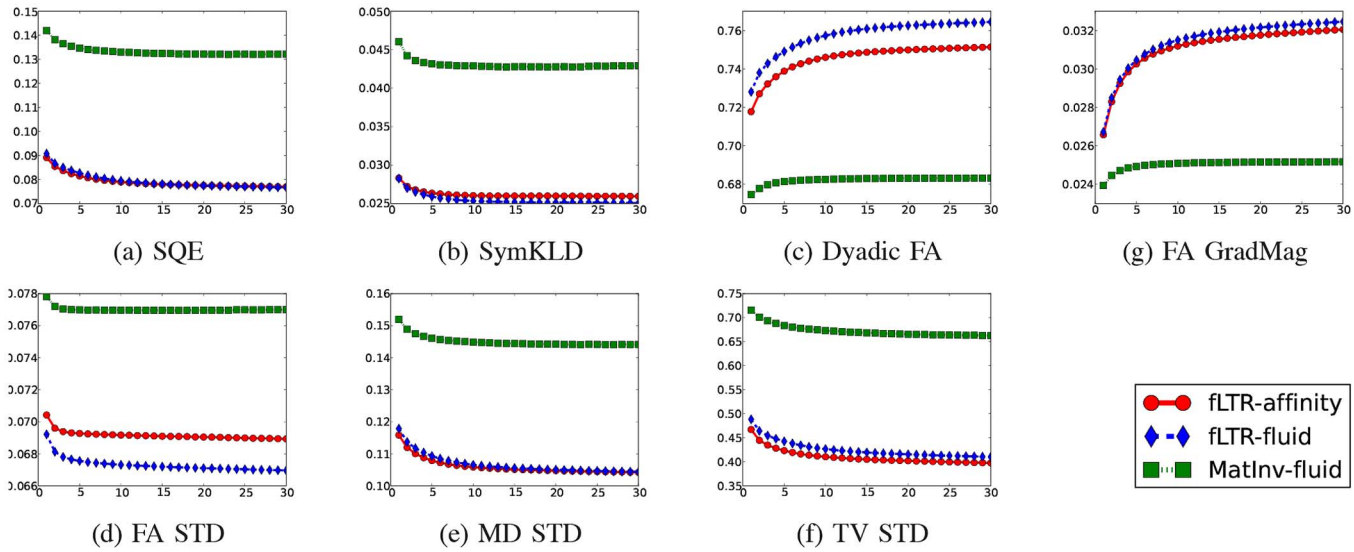

Fig. 8. Coherence/dispersion indexes of the warped subject images as the template construction procedure progresses. The x-axis shows the number of iterations, and the y-axis shows the coherence/dispersion indexes. The indexes are averaged within the foreground of the template image. Lower values of the following indexes indicate better coherence among the warped images: the SQE, the SymKLD, the STD of FA, MA, and TV. Higher values of the dyadic FA indicate better coherence among the warped images. Higher values of the GradMag indicate that the template preserves more details.

after 30 iterations. Fig. 7 shows the statistics of the 30th iteration. In the figure, the x-axis is percentiles of the distributions of the MatInv-fluid results. The y-axis, is the corresponding percentiles of the distributions under comparison, including the results of the fLTR-affinity, the fLTR-fluid, and the MatInv-fluid methods. The MatInv-fluid versus MatInv-fluid comparison is a self comparison, so it is a line of 45° (green), the baseline for other comparisons. In the QQ plot, distributions with lower values show lower shift, and those with higher values show upper shift. Lower values of the following indexes indicate better coherence among the warped subject images: the SQE, the SymKLD, the STD of FA, MA, and TV. Higher values of the dyadic FA indicate better coherence. Regarding the indexes shown in Fig. 7, the warped subject images generated by the fLTR-affinity and fLTR-fluid methods are more coherent than those generated by the MatInv-fluid method.

The DTI template images constructed with the fLTR-affinity and the fLTR-fluid methods preserve more details than that constructed with the MatInv-fluid method. Visually, as shown in Fig. 9, the fLTR templates are sharper than the MatInv-fluid template, preserving more details. Quantitatively, as shown in Fig. 7(g), the FA images derived from the DTI templates constructed with the fLTR methods have higher gradient magnitude than that constructed with the MatInv-fluid method.

VIII. Discussion and Conclusion

When a spatial transformation is applied to a DTI, to keep tensor directions aligned with the transformed space, the tensors must be reoriented according to the Jacobian matrices of the transformation. This reorientation procedure makes the registration force correlate between adjacent voxels, complicating the optimization and raising computational challenges, as discussed in Section IV.

The diffeomorphic Demons algorithm [1], which enjoys linear computational complexity, when adapted to DTI registration in [2] loses its linear efficiency, as discussed in Section IV. The Demons algorithm can be interpreted as a PDE, as shown in Section III-B. Such a perspective provides insight on how computational efficiency can be regained even with inseparable registration force. At each iteration, the PDE (4) regarding the transformation field is approximated by its Taylor expansion around a time point, and converted to an ODE (5) regarding the update field. Consequently, the update field can be calculated with an integration procedure. In general, the longer the integration time, the more computation is required, and the larger the update field leaps. On the other hand, the Taylor approximation intrinsically requires the update at each voxel to stay within its local trust region (as shown in Section III-D), which suggests small local leaps and short time steps. Therefore, it is possible to jointly achieve both computational efficiency and local stability.

Directly solving a large inseparable linear system at every registration, as conducted in [2], may not only yield an update field which violates the assumption behind the Taylor approximation, but also introduces a considerable unnecessary amount of computation, as discussed in Section V. It takes the same amount of computation as integrating an ODE over an infinitely long time duration, much more than the necessary for finding the update field within local trust regions.

The proposed fast local-trust-region method provides several advantages by adaptively regularizing the velocity within local trust regions, as discussed in Section V. First, it increases stability of the registration procedure because it effectively limits update leaps locally within stable ranges; for example, a half of voxel spacing. Second, the time step length does not need to be adjusted with an unpredictable number of trials, and a predefined time length is able to keep the local-trust-region feature. Third, the computation load for calculating the transformation update is fully determined by the desired accuracy order. For example, the mth-order Runge Kutta method [18] can achieve

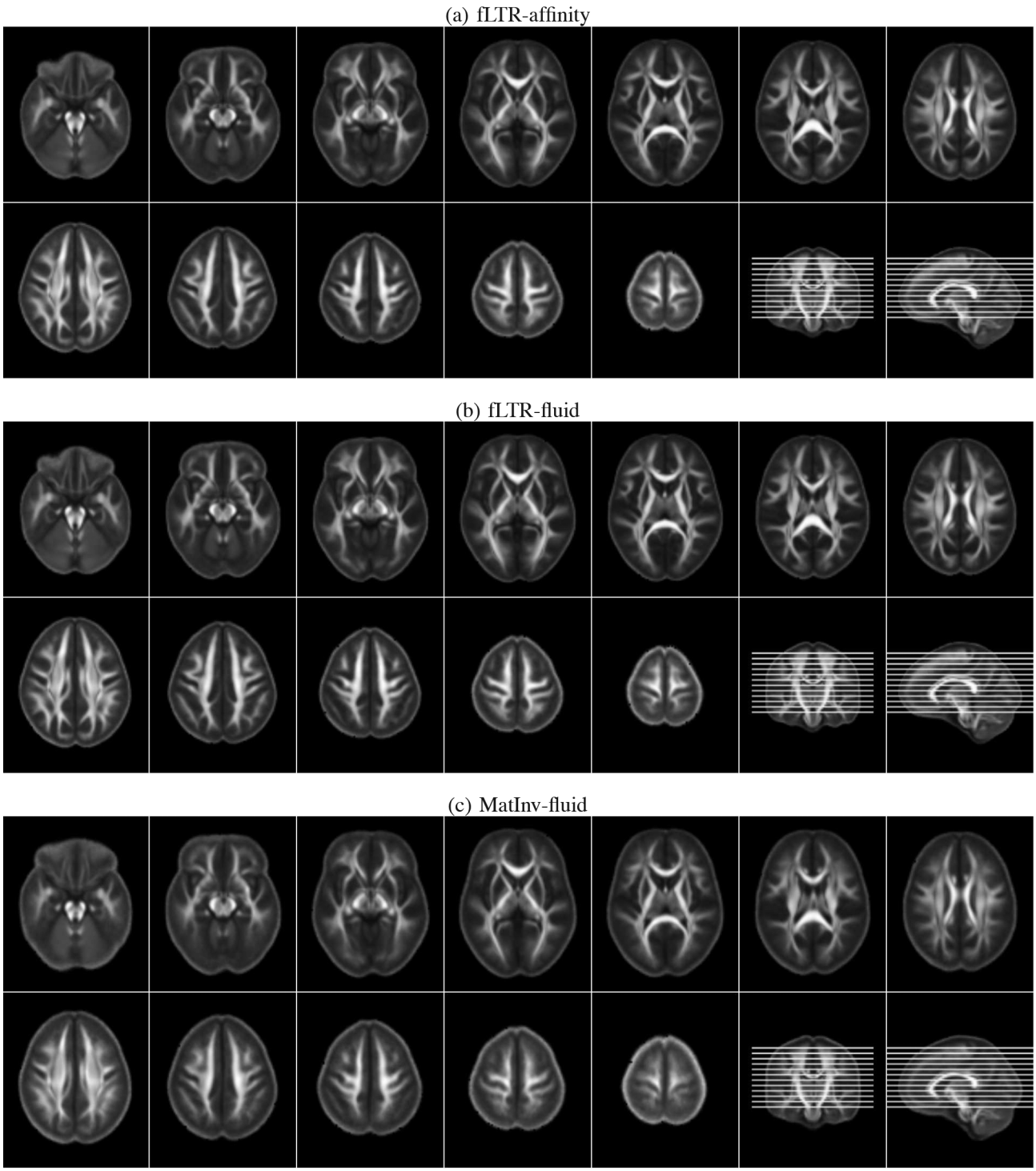

Fig. 9. FA images of the DTI templates constructed with the fLTR-affinity, fLTR-fluid, and the MatInv-fluid methods. The templates constructed with the fLTR-affinity and the fLTR-fluid method are visually sharper than that constructed with the MatInv-fluid method. This is consistent with the gradient magnitude index shown in Fig. 7(g).

the mth-order accuracy. Local velocity regularization achieves both stability and computation efficiency simultaneously. It reduces computation time and memory demand about tenfold, and yields better registration goodness.

Zhang *et al.* [10] suggested using similarity metrics defined on deviatoric tensors to achieve better FA matching. Because tensor image transformation followed by deviatoric transformation is the same as deviatoric transformation followed by image transformation, the proposed method can optimize the squared error metric of deviatoric tensors by using deviatoric tensor images as its input. Similarly, the proposed method can also use matrix-logarithm transformed tensors as its input to take the advantage of the log-Euclidean tensor interpolation [37] proposed by Arsigny *et al.*

Though in this paper the fast local-trust-region method is applied to DTI image registration, its principle is generally applicable to different diffusion models related to other acquisition schemes, for example, the multi-tensor model [38] related to the HARDI [3], and the nonparametric model related to the diffusion spectrum imaging (DSI) [39]. As shown in Section V-E, the local-trust-region technique only requires that the registration force and the Hessian operator have the format of (7) and (8). As long as the g_{sim} term in the equations can be derived for the diffusion models, the proposed technique is applicable. Extending the local-trust-region scheme to these diffusion models is worth further investigation.

APPENDIX

A. Transformation-Velocity Relationship

Given a point y, let $x(y,t)$ denote the trajectory starting from it. The particle is at position $x(y,t)$ at time t, and $x(y,0)$ equals y. Given y, dx/dt is the velocity of the particle at time t. With the Eulerian frame of reference, we denote the velocity with $v_{\text{forward}}(x,t) = dx/dt$.

If we define $\varphi_{\text{forward}}(y,t) = x$ as the mapping from the starting point y to its current position x, then we have the following forward relationship:

$$\frac{d\varphi_{\text{forward}}}{dt} = v_{\text{forward}}(\varphi_{\text{forward}}, t).$$

To study the backward relationship, let us define $\varphi_{\text{backward}}$ as the inverse of φ_{forward}, that is $\varphi_{\text{backward}}(x,t) = y$ and define $v_{\text{backward}}(x,t) = -v_{\text{forward}}(x,t)$. Given y, we have

$$\begin{aligned}
0 = \frac{dy}{dt} &= \frac{d}{dt}\varphi_{\text{backward}}(x,t) \\
&= (\nabla\varphi_{\text{backward}})\frac{dx}{dt} + \frac{\partial}{\partial t}\varphi_{\text{backward}}(x,t) \\
&= (\nabla\varphi_{\text{backward}})v_{\text{forward}} + \frac{\partial}{\partial t}\varphi_{\text{backward}}(x,t) \\
&= -(\nabla\varphi_{\text{backward}})v_{\text{backward}} + \frac{\partial}{\partial t}\varphi_{\text{backward}}(x,t)
\end{aligned}$$

which implies

$$\frac{\partial}{\partial t}\varphi_{\text{backward}}(x,t) = (\nabla\varphi_{\text{backward}})v_{\text{backward}}.$$

B. Proof of Theorem 1

Theorem 1 can be proved without much difficulty. The outline is as follows. First, the transformation model, (4b), guarantees that φ is diffeomorphic. Second, the temporal derivative of $\mathcal{E}$ is

$$\frac{d\mathcal{E}}{dt} = \frac{\partial\mathcal{E}}{\partial\varphi}\frac{\partial\varphi}{\partial t} = \frac{\partial\mathcal{E}}{\partial\varphi}(\nabla\varphi)v = -f^{\mathsf{T}}(\varphi)v = -v^{\mathsf{T}}Pv \le 0$$

which guarantees that $\mathcal{E}$ as a function of t monotonically decreases. Third, because $\mathcal{E}$ is nonnegative and monotonically decreases as a function of t, it converges to a value, rather than diverges to the negative infinite. Fourth, because $\mathcal{E}$ monotonically converges, its temporal derivative $d\mathcal{E}/dt = -v^{\mathsf{T}}Pv$ converges to 0. Fifth, because the eigenvalues of P are in a constant positive range $[\lambda_{\text{lower}}, \lambda_{\text{upper}})$, the convergence of $-v^{\mathsf{T}}Pv$ to zero implies the convergence of v to zero. Last, $v \to 0$ implies, with a constantly bounded operator P, that the registration force $f(\varphi) = Pv$ converges to zero.

C. Similarity Energy Optimization

Proposition 3: For an energy function of the format $\mathcal{E}(\varphi) = (1/2)\|F - M \diamond \varphi\|^2$, its diffeomorphic registration force and Hessian operator are

$$\begin{cases} f(\varphi) = g^{\mathsf{T}}b \\ H(\varphi) \approx g^{\mathsf{T}}g \end{cases} \text{where } \begin{cases} b = F - M \diamond \varphi \\ g = \frac{\partial(M\diamond\varphi)}{\partial\varphi}\nabla\varphi. \end{cases}$$

Proof: The diffeomorphic registration force at φ is

$$\begin{aligned}
f(\varphi) \equiv -\left[\frac{\partial\mathcal{E}}{\partial\varphi}\nabla\varphi\right]^{\mathsf{T}} &= -\left[\frac{1}{2}\frac{\partial\|F - M\diamond\varphi\|^2}{\partial\varphi}\nabla\varphi\right]^{\mathsf{T}} \\
&= -\left[\frac{1}{2}\frac{\partial\|F - M\diamond\varphi\|^2}{\partial(M\diamond\varphi)}\frac{\partial(M\diamond\varphi)}{\partial\varphi}\nabla\varphi\right]^{\mathsf{T}} \\
&= \left[(F - M\diamond\varphi)^{\mathsf{T}}\frac{\partial(M\diamond\varphi)}{\partial\varphi}\nabla\varphi\right]^{\mathsf{T}} \\
&= \left[\frac{\partial(M\diamond\varphi)}{\partial\varphi}\nabla\varphi\right]^{\mathsf{T}}(F - M\diamond\varphi) = g^{\mathsf{T}}(\varphi)b(\varphi).
\end{aligned}$$

The registration force at $\varphi \circ \exp(u)$ can be approximated as

$$\begin{aligned}
f[\varphi\circ\exp(u)] &= g^{\mathsf{T}}[\varphi\circ\exp(u)]\, b[\varphi\circ\exp(u)] \\
&\approx g^{\mathsf{T}}(\varphi) b[\varphi + (\nabla\varphi)u] \\
&\approx g^{\mathsf{T}}(\varphi)\left[b(\varphi) + \frac{\partial b(\varphi)}{\partial\varphi}(\nabla\varphi)u\right] \\
&= g^{\mathsf{T}}(\varphi)b(\varphi) + g^{\mathsf{T}}(\varphi)\frac{\partial(F - M\diamond\varphi)}{\partial\varphi}(\nabla\varphi)u \\
&= f(\varphi) - g^{\mathsf{T}}(\varphi)\frac{\partial M\diamond\varphi}{\partial\varphi}(\nabla\varphi)u \\
&= f(\varphi) - g^{\mathsf{T}}(\varphi)g(\varphi)u = f(\varphi) - H(\varphi)u.
\end{aligned}$$

D. Regularization Energy Optimization

Proposition 4: For a regularization energy function of the format $\mathcal{E}(\varphi) = (1/2)\langle R\varphi, R\varphi\rangle$, where R is a differential operator, its diffeomorphic registration force and Hessian operator are

$$\begin{cases} f(\varphi) = g^{\mathsf{T}}b \\ H(\varphi) \approx g^{\mathsf{T}}g \end{cases} \text{where } \begin{cases} b = R\varphi \\ g = -R\nabla\varphi. \end{cases}$$

Proof: The diffeomorphic registration force at φ is

$$\begin{aligned} f(\varphi) &\equiv -\left[\frac{\partial \mathcal{E}}{\partial \varphi}\nabla\varphi\right]^{\mathsf{T}} = -\left[\frac{1}{2}\frac{\partial \langle R\varphi, R\varphi\rangle}{\partial \varphi}\nabla\varphi\right]^{\mathsf{T}} \\ &= -\left[\frac{1}{2}\frac{\partial \langle R\varphi, R\varphi\rangle}{\partial R\varphi}\frac{\partial R\varphi}{\partial \varphi}\nabla\varphi\right]^{\mathsf{T}} \\ &= -\left[(R\varphi)^{\mathsf{T}} R\nabla\varphi\right]^{\mathsf{T}} \\ &= \left[-R\nabla\varphi\right]^{\mathsf{T}} R\varphi = g^{\mathsf{T}}(\varphi)b(\varphi). \end{aligned}$$

The registration force at $\varphi \circ \exp(u)$ can be approximated as

$$\begin{aligned} f\left[\varphi\circ\exp(u)\right] &= g^{\mathsf{T}}\left[\varphi\circ\exp(u)\right] b\left[\varphi\circ\exp(u)\right] \\ &\approx g^{\mathsf{T}}(\varphi) b\left[\varphi + (\nabla\varphi)u\right] \\ &\approx g^{\mathsf{T}}(\varphi)\left[b(\varphi) + \frac{\partial b(\varphi)}{\partial \varphi}(\nabla\varphi)u\right] \\ &= g^{\mathsf{T}}(\varphi)b(\varphi) + g^{\mathsf{T}}(\varphi)\frac{\partial R\varphi}{\partial \varphi}(\nabla\varphi)u \\ &= f(\varphi) + g^{\mathsf{T}}(\varphi)R(\nabla\varphi)u \\ &= f(\varphi) - g^{\mathsf{T}}(\varphi)g(\varphi)u = f(\varphi) - H(\varphi)u. \end{aligned}$$

E. Proof of Proposition 2

Let $U\Lambda V^{\mathsf{T}}$ be the singular value decomposition (SVD) of g where U and V are unitary matrices and Λ is diagonal matrix. Then P equals $V(\Lambda^2 + \|b\|^2 I)V^{\mathsf{T}}$, $g^{\mathsf{T}}b$ equals $V\Lambda U^{\mathsf{T}}b$, and v equals $V(\Lambda^2 + \|b\|^2 I)^{-1}\Lambda U^{\mathsf{T}}b$. Because V is unitary, we have

$$\|v\| = \left\| \left(\Lambda^2 + \|b\|^2 I\right)^{-1} \Lambda U^{\mathsf{T}} b \right\|.$$

Let $\lambda_1, \cdots, \lambda_n$ denote the diagonal elements of Λ. Then $(\Lambda^2 + \|b\|^2 I)^{-1}\Lambda$ is a diagonal matrix whose diagonal elements are $(\lambda_1/(\lambda_1^2 + \|b\|^2)), \cdots, (\lambda_n/(\lambda_n^2 + \|b\|^2))$. The largest possible amplitude of v is

$$\begin{aligned} \max_i \left|\frac{\lambda_i}{\lambda_i^2 + \|b\|^2}\right| \|U^{\mathsf{T}}b\| &= \max_i \left|\frac{\lambda_i}{\lambda_i^2 + \|b\|^2}\right| \|b\| \\ &= \max_i \frac{|\lambda_i|\|b\|}{\lambda_i^2 + \|b\|^2}. \end{aligned}$$

With the triangular inequality, we have $(|\lambda_i|\|b\|/(\lambda_i^2 + \|b\|^2)) \leq 0.5$ as long as $\lambda_i^2 + \|b\|^2 > 0$.

F. Relationship Between Update Field and Warped Image

1) Approximate $g_{\mathrm{sim}}(\varphi)$ From $M \diamond \varphi$:

$$\begin{aligned} g_{\mathrm{sim}}(\varphi) &\equiv \frac{\partial M\diamond\varphi}{\partial\varphi}\nabla\varphi \\ &= \frac{\partial M\diamond\varphi}{\partial\varphi}\frac{\partial \varphi\circ\exp(u)}{\partial u}\Big|_{u=0} \\ &= \frac{\partial M\diamond[\varphi\circ\exp(u)]}{\partial u}\Big|_{u=0} \\ &\approx \frac{\partial (M\diamond\varphi)\diamond\exp(u)}{\partial u}\Big|_{u=0}. \end{aligned}$$

2) Decompose $(M \diamond \varphi) \diamond \exp(u)$:

$$(M\diamond\varphi)\diamond\exp(u) = r(u)W(u)r^{\mathsf{T}}(u)$$

where

$$\begin{aligned} W(u) &= (M\diamond\varphi)\circ\exp(u) \\ r(u) &= J^T\left[\exp(u)\right]\left\{J\left[\exp(u)\right]J^T\left[\exp(u)\right]\right\}^{-1/2} \end{aligned}$$

and $J^T[\exp(u)]$ is the Jacobian matrix of $\exp(u)$.

With this decomposition, we have

$$\begin{aligned} d\left[(M\diamond\varphi)\diamond\exp(u)\right] &= r(u)d\left[W(u)\right]r^{\mathsf{T}}(u) \\ &\quad + d\left[r(u)\right]W(u)r^{\mathsf{T}}(u) + r(u)W(u)d\left[r^{\mathsf{T}}(u)\right] \end{aligned}$$

3) $(\partial W(u)/\partial u)|_{u=0}$:

$$\begin{aligned} \frac{\partial W(u)}{\partial u}|_{u=0} &= \frac{\partial (M\diamond\varphi)\circ\exp(u)}{\partial u}|_{u=0} \\ &= \frac{\partial (M\diamond\varphi)\circ\exp(u)}{\partial\exp(u)}|_{u=0}\frac{\partial\exp(u)}{\partial u}|_{u=0} \\ &= \nabla(M\diamond\varphi). \end{aligned}$$

4) Infinitesimal Difference of $r(u)$ Around $u = 0$: The perturbation relationship between r and u can be derived via $J[\exp(u)]$. r and J have the following relationship:

$$rJr = J^T(JJ^T)^{-1/2}JJ^T(JJ^T)^{-1/2} = J^{\mathsf{T}}$$

which implies $(dr)Jr + r(dJ)r + rJ(dr) = dJ^{\mathsf{T}}$. When $u = 0$, both J and r equal I, and the equation is reduced to $dr + dJ + dr = dJ^{\mathsf{T}}$. Thus,

$$dr = \frac{1}{2}(dJ^{\mathsf{T}} - dJ) \text{ if } u = 0.$$

On the other hand, we have

$$dJ = \nabla d\left[\exp(u)\right] = \nabla d[I + u] = \nabla du$$

which implies

$$dr = \frac{1}{2}(\nabla^{\mathsf{T}} - \nabla)du \text{ if } u = 0.$$

5) Putting the Results in F1, F2, F3, and F4 Together, We Have:

$$g_{\mathrm{sim}}(\varphi) \approx \nabla(M\diamond\varphi) + \frac{1}{2}(\nabla^{\mathsf{T}} - \nabla)\triangleleft(M\diamond\varphi) + \frac{1}{2}(M\diamond\varphi)(\nabla - \nabla^{\mathsf{T}})$$

where "$\triangleleft$" is a functor on operators such that given two linear operators O_1 and O_2, $(O_1 \diamond O_2)$ satisfies $(O_1\bullet)O_2 = (O_1 \triangleleft O_2)\bullet$.

REFERENCES

[1] T. Vercauteren, X. Pennec, A. Perchant, and N. Ayache, "Diffeomorphic demons: Efficient non-parametric image registration," *Neuroimage*, vol. 45, pp. S61–S72, Mar. 2009.

[2] T. Yeo, T. Vercauteren, P. Fillard, J.-M. Peyrat, X. Pennec, P. Golland, N. Ayache, and O. Clatz, "DTREFinD: Diffusion tensor registration with exact finite-strain differential," *IEEE Trans. Med. Imag.*, vol. 28, no. 12, pp. 1914–1928, Dec. 2009.

[3] D. S. Tuch, T. G. Reese, M. R. Wiegell, N. Makris, J. W. Belliveau, and V. J. Wedeen, "High angular resolution diffusion imaging reveals intravoxel white matter fiber heterogeneity," *Magn. Reson. Med.*, vol. 48, no. 4, pp. 577–582, 2002.

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

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

[6] J. Yang, D. Shen, C. Davatzikos, and R. Verma, "Diffusion tensor image registration using tensor geometry and orientation features," in *Medical Image Computing and Computer-Assisted Intervention MICCAI 2008*. Berlin, Germany: Springer, 2008, vol. 5242, Lecture Notes in Computer Science, pp. 905–913.

[7] P.-T. Yap, G. Wu, H. Zhu, W. Lin, and D. Shen, "F-timer: Fast tensor image morphing for elastic registration," *IEEE Trans. Med. Imag.*, vol. 29, no. 5, pp. 1192–1203, May 2010.

[8] D. C. Alexander. J. C. Gee, "Elastic matching of diffusion tensor images," *Comput. Vis. Image Understand.*, vol. 77, no. 2, pp. 233–250, 2000.

[9] J. Ruiz-Alzola, C.-F. Westin, S. K. Warfield, C. Alberola, S. Maier, and R. Kikinis, "Nonrigid registration of 3d tensor medical data," *Med. Image Anal.*, vol. 6, no. 2, pp. 143–161, Jun. 2002.

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

[11] Y. Cao, M. I. Miller, S. Mori, R. L. Winslow, and L. Younes, "Diffeomorphic matching of diffusion tensor images," in *Proc. Conf. Computer Vis. Pattern Recognit. Workshop*, Jun. 2006, p. 67.

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

[13] B. Fischer and J. Modersitzki, "A unified approach to fast image registration and a new curvature based registration technique," *Linear Algebra Appl.*, vol. 380, pp. 107–124, 2004.

[14] B. Fischer and J. Modersitzki, "Fast diffusion registration," *Contemp. Math.*, vol. 313, pp. 117–128, 2002.

[15] J. Weickert, B. M. T. H. Romeny, and M. A. Viergever, "Efficient and reliable schemes for nonlinear diffusion filtering," *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 398–410, Mar. 1998.

[16] R. Stefanescu, X. Pennec, and N. Ayache, "Grid powered nonlinear image registration with locally adaptive regularization," *Med. Image Anal.*, vol. 8, no. 3, pp. 325–342, Sep. 2004.

[17] R. H. Byrd, R. B. Schnabel, and G. A. Shultz, "A trust region algorithm for nonlinearly constrained optimization," *SIAM J. Numer. Anal.*, vol. 24, no. 5, pp. 1152–1170, 1987.

[18] W. H. Press, S. A. Teukolsky, W. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, 3rd ed. Cambridge, U.K.: Cambridge Univ. Press, 2007.

[19] T. Vercauteren, X. Pennec, E. Malis, A. Perchant, and N. Ayache, "Insight into efficient image registration techniques and the demons algorithm," in *Information Processing in Medical Imaging*. New York: Springer, 2007, pp. 495–506.

[20] C. Broit, "Optimal registration of deformed images," Ph.D. dissertation, Univ. Pennsylvania, Philadelphia, 1981.

[21] J. C. Gee and R. K. Bajcsy, "Elastic matching: Continuum mechanical and probabilistic analysis," *Brain Warp.*, vol. 183, p. 197, 1999.

[22] G. E. Christensen, "Deformable shape models for anatomy," Ph.D. dissertation, Washington Univ., St. Louis, MO.

[23] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Math. Computat.*, pp. 297–301, 1965.

[24] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. 100, no. 1, pp. 90–93, Jan. 1974.

[25] L. H. Thomas, Elliptic problems in linear difference equations over a network Watson Sci. Comput. Lab. Rep., Columbia Univ., New York, Tech. Rep., 1949.

[26] N. D. Cahill, J. A. Noble, and D. J. Hawkes, "Demons algorithms for fluid and curvature registration," in *Proc. IEEE Int. Symp. Biomed. Imag.: From Nano to Macro*, 2009, pp. 730–733.

[27] J. Ashburner, "A fast diffeomorphic image registration algorithm," *Neuroimage*, vol. 38, no. 1, pp. 95–113, Oct. 2007.

[28] T. Vercauteren, X. Pennec, A. Perchant, and N. Ayache, "Symmetric logdomain diffeomorphic registration: A demons-based approach," in *Proc. Med. Image Comput. Comput.-Assist. Intervent. (MICCAI 2008)*, 2008, pp. 754–761.

[29] N. Barnea-Goraly, V. Menon, M. Eckert, L. Tamm, R. Bammer, A. Karchemskiy, C. C. Dant, and A. L. Reiss, "White matter development during childhood and adolescence: A cross-sectional diffusion tensor imaging study," *Cereb Cortex*, vol. 15, no. 12, pp. 1848–1854, Dec. 2005.

[30] C. Lebel, L. Walker, A. Leemans, L. Phillips, and C. Beaulieu, "Microstructural maturation of the human brain from childhood to adulthood," *Neuroimage*, vol. 40, no. 3, pp. 1044–1055, Apr. 2008.

[31] D. Bonekamp, L. M. Nagae, M. Degaonkar, M. Matson, W. M. A. Abdalla, P. B. Barker, S. Mori, and A. Horská, "Diffusion tensor imaging in children and adolescents: Reproducibility, hemispheric, age-related differences," *Neuroimage*, vol. 34, no. 2, pp. 733–742, Jan. 2007.

[32] T. J Eluvathingal, K. M Hasan, L. Kramer, J. M Fletcher, and L. Ewing-Cobbs, "Quantitative diffusion tensor tractography of association and projection fibers in normally developing children and adolescents," *Cereb Cortex*, vol. 17, no. 12, pp. 2760–2768, Dec. 2007.

[33] J. C. Butcher, *Numerical Methods for Ordinary Differential Equations*, 2nd ed. New York: Wiley, 2008.

[34] S. Joshi, B. Davis, M. Jomier, and G. Gerig, "Unbiased diffeomorphic atlas construction for computational anatomy," *NeuroImage*, vol. 23, pp. S151–S160, 2004.

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

[36] P. J. Basser and S. Pajevic, "Statistical artifacts in diffusion tensor MRI (DT-MRI) caused by background noise," *Magn. Reson. Med.*, vol. 44, no. 1, pp. 41–50, Jul. 2000.

[37] 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, Aug. 2006.

[38] M. Taquet, B. Scherrer, O. Commowick, J. Peters, M. Sahin, B. Macq, and S. K. Warfield, "Registration and analysis of white matter group differences with a multi-fiber model," *Med. Image Comput. Comput. Assist. Interv.*, vol. 15, pt. . 3, pp. 313–320, 2012.

[39] V. J. Wedeen, P. Hagmann, W.-Y. I. Tseng, T. G. Reese, and R. M. Weisskoff, "Mapping complex tissue architecture with diffusion spectrum magnetic resonance imaging," *Magn. Reson. Med.*, vol. 54, no. 6, pp. 1377–1386, Dec. 2005.