Fast Diffusion Tensor Registration with Exact Reorientation and Regularization*

Junning Li¹, Yonggang Shi¹, Giang Tran³, Ivo Dinov¹,
Danny J.J. Wang², and Arthur W. Toga¹

¹ Laboratory of Neuro Imaging,

² Brain Mapping Center, Department of Neurology

³ Department of Mathematics

University of California, Los Angeles, CA, USA

{Junning.Li,yshi,ivo.dinov,JJ.Wang,toga}@loni.ucla.edu
giangtran@math.ucla.edu

Abstract. Diffusion tensor imaging is widely used in brain connectivity study. As more and more group studies recruit a large number of subjects, it is important to design registration methods that are not only theoretically rigorous, but also computationally efficient, for processing large data sets. However, the requirement of reorienting diffusion tensors complicates and slows down the registration, especially for those methods whose scalar-image versions have linear complexity, for example, the Demons algorithm. In this paper, we propose an extension of the Demons algorithm that incorporates exact reorientation and regularization into the calculation of deforming velocity, yet preserving its linear complexity. This method restores the computational efficiency of the Demons algorithm to diffusion images, but does not sacrifice registration goodness. In our experiments, the new algorithm achieved state-of-art performance at a ten-fold decrease of computational time.

1 Introduction

Water molecules in biology tissue tend to diffuse faster along, relative to across, obstacle structures. Diffusion tensor imaging (DTI) noninvasively measures this anisotropy (approximated with second-order tensors), providing information about structures such as neural fibers. As more and more large group studies, such as those sponsored by the Human Connectome Project [1], require nonlinear normalization of diffusion tensor images, it is important to design deformation registration methods that are not only theoretically rigorous, but also computationally efficient, for processing large data sets.

Registration of diffusion tensor images is more complicated than scalar images, because displacing a voxel to a new place does not only change its own diffusion tensor value, but also reorients those of its adjacent voxels [2]. As a result, the deformation force is no longer independent for adjacent voxels, but has a sparse 3D grid structure.

* This work is supported by grants K01EB013633, R01MH080892 9P41EB015922, P41RR013642, R01MH71940, U54RR021813, U24RR025736 from NIH.

Because reorientation significantly complicates the computation, methods based on rotation-invariant features have been developed. Many of these methods first extract scalar-valued or vector-valued features from tensors, for instance, the fractional anisotropy (FA), and then register diffusion images by aligning these features with multi-channel registration [3,4].

Some other methods directly work on tensors, with similarity metric defined on tensors, and indirect or direct involvement of reorientation into velocity field optimization. Alexander and Gee [5] reorientated tensors according to the new displacement field after each iteration, though not directly accounted the reorientation into optimization. Zhang *et al.* [6] applied piece-wise affine registration to image subblocks, and then fused these transformations together by smoothing. Since finite-strain (FS) reorientation [2] can be analytically incorporated into affine transformation, the method efficiently estimates the optimal local affine parameters, but it is not clear how the fusion step affects the total registration energy. Cao *et al.* [7] extended the Large Deformation Diffeomorphic Metric Mapping framework, to analytically embed the preservation-of-principal-direction (PPD) reorientation [2] into the deforming force. In 2009, Yeo *et al.* [8] embedded exact FS reorientation into the diffeomorphic Demons algorithm [9,10], showing that exact reorientation considerably improved registration accuracy.

Yeo *et al.*'s work [8] was based on the Demons algorithm, a fast algorithm with $O(n)$ complexity for scalar images, where n is the number of voxels. However, because a large sparse linear system was solved at every iteration, the algorithm became considerably slower. This effect becomes more exacerbated for such an algorithm whose scalar-image version enjoys linear complexity. For example, for our DTI images of size 128x128x128, the method took about 7 hours and more than 20 Giga Bytes (GB) memory on a desktop with an Intel Xeon 2.80 GHz CPU. On the other hand, the regularization on the displacement field was separated as a Gaussian smoothing step after updating the displacement field. This may not be always consistent with the diffeomorphic framework [10].

This raises the question "can the Demons algorithm still enjoy its $O(n)$ complexity when the deformation force is coupled between adjacent voxels and regularization is incorporated?" Though this question is raised from diffusion tensor images, it is generally applicable to other diffusion images, such as high-angular-resolution diffusion images (HARDI) and diffusion spectrum images (DSI). We are particularly interested in the Demons algorithm, because its original linear complexity in [9,10] is well suited for processing large group data sets.

In this paper, we extend the Demons algorithm to incorporate both exact reorientation and regularization into velocity calculation, but without directly solving a large non-separable linear system. This method restores the $O(n)$ computational efficiency of the original Demons algorithm to diffusion images, but does not sacrifice registration goodness, in comparison with solving a large linear system at each iteration. In our experiments, it introduced a 10-fold reduction of the computation time, and achieved state-of-art registration performance.

2 Method

2.1 Diffeomorphic Demons Registration for Diffusion Tensors

Let F and M respectively denote the fixed and moving image, u and v respectively denote the transformation field and velocity field, $\circ$ denote the compositive operator, $\bullet(i)$ denote the image value at voxel i , $\bullet_i$ and $\bullet_x$ denote the velocity, displacement or a certain value associated respectively with a voxel i or a point x . The diffeomorphic Demons algorithm for scalar images in [10] is as follows:

Repeat until convergence

1. **For each** voxel i , at iteration k , given current transformation field u^k :
 - (a) **Define** $b_i(v) = F(i) - M \circ [u^k \circ \exp(v)](i)$.
 - (b) **Let** $b_i = b_i(v) |_{v=0} = F(i) - M \circ u^k(i)$, and $g_i = \frac{\partial b_i(v)}{\partial v_i} |_{v=0}$.
 - (c) **Let** $v_i = \frac{g_i b_i}{g_i^2 + 1/\delta_i^2}$, where $1/\delta_i^2$ is the velocity regularizer at voxel i .
2. **Update displacement field** with the velocity field: $u^{k+1} = u^k \circ \exp(v)$.

The algorithm minimizes the energy function

$$\mathcal{E}(u) = \int [F(x) - M \circ u(x)]^2 dx + \int \left\| \frac{v_x}{\delta_x} \right\|^2 dx \quad (1)$$

with the displacement-velocity relationship $v = \frac{du}{dt} = \frac{\partial u}{\partial t} + (\nabla u)v$ (actually discretized as $u^{k+1} = u^k \circ \exp(v)$), and a discrete time interval $\tau = 1$. Smoothing can be applied to the displacement field or the velocity field, to impose regularization. The minimization is by approximating the energy function as $\mathcal{E} = \sum_i \left\| \left(\sum_j \frac{\partial b_i(v)}{\partial v_j} v_j \right) |_{v=0} - b_i \right\|^2 + \sum_i \left\| \frac{v_i}{\delta_i} \right\|^2$ with the Gauss-Newton method.

For scalar images, $\frac{\partial b_i(v)}{\partial v_j} |_{v=0}$ equals zero if $i \neq j$, so the optimization can be separated for each voxel as in Step (1) and has $O(n)$ complexity at each iteration, where n is the number of voxels. For details, please refer to [9,10].

For diffusion tensor images, tensors must be reorientated during transformation to align their directions with the transformed space. Consequently, the displacement at a voxel not only changes its own warped tensor value, but also reorientates those of its adjacent voxels, that is, $\frac{\partial b_i(v)}{\partial v_j} |_{v=0}$ is not zero for adjacent voxels i and j . In this case, the energy function cannot be separated for each voxel. For details on the formulas of tensor reorientation, please refer to [2].

2.2 Regularization

Standard regularizers for displacement fields include the elastic, diffusion, and curvature regularizers [11] which all can be formatted as the squared norm of a certain derivative value of the displacement field. When discretized, these regularizers can be formulated as $\sum_{i=1}^n \|A_{\text{reg } i} u\|^2$ where $A_{\text{reg } i}$ is a differential operator at voxel i , represented as a matrix. In this paper, we used $\sum_l \int \|H_l u\|^2 dx$ as the regularizer, where H_l is the Hessian operator applied to the l th dimensional component of u . We call it affineness regularizer because the affine transformation is the only non-trivial kernel not punished by this regularizer.

2.3 Local Gauss-Newton Optimization

As shown in Sections 2.1 and 2.2, the energy function of the Demons algorithm, including both the dissimilarity and regularization terms, is the sum of squares of many linear functions. Therefore, the energy function can be written as $\mathcal{E} = \|A_{img}v - b_{img}\|^2 + \|A_{reg}v - b_{reg}\|^2 = \|Av - b\|^2$. Due to tensor reorientation, A is not separable for adjacent voxels, and the optimization cannot be separated as in Step (1) of the Demons algorithm for scalar images. In [8], Yeo minimized the energy by solving this large sparse linear system at each iteration, excluding the regularization term. Though the system is sparse, the computational cost still goes up sharply as the image size increases. For images of size 128x128x128, we observed 20 GB memory usage and about 7 hours computation. This raises the question “can the Demons algorithm still have its $O(n)$ complexity when the deformation force is coupled between adjacent voxels?”

In this paper, we propose a method that incorporates both tensor reorientation and displacement regularization into its optimization iterations, while restores the $O(n)$ complexity of the Demons algorithm. The algorithm is as follows, where we rewrite the energy function as $\mathcal{E} = \|Av - b\|^2 = \|(\sum A_i v_i) - b\|^2$ and A_i is the columns of A related to the velocity at voxel i :

1. **For each voxel i :**

- (a) If $\|A'_i b\| < \text{stop threshold}$, then let $d_i = 0$, else let $d_i = \underset{z}{\operatorname{argmin}} \|A_i z - b\|^2 = (A'_i A_i)^{-1} A'_i b$.

2. Let $d = \begin{bmatrix} d_1 \\ \vdots \\ d_n \end{bmatrix} = \begin{bmatrix} (A'_1 A_1) & & \\ & \ddots & \\ & & (A'_n A_n) \end{bmatrix}^{-1} A' b$.

3. Let $\tau = \underset{t}{\operatorname{argmin}} (\|A d t - b\|^2) = \frac{b' A d}{\|A d\|^2} = \frac{b' A d}{\|A_{img} d\|^2 + \|A_{reg} d\|^2}$.

4. Let $v = d\tau$.

This algorithm first chooses a descent direction d with local Gauss-Newton optimizations at Step (1a), and then determines the step length τ with a higher-level Gauss-Newton optimization of the energy function given direction d . The algorithm can be understood as the following procedure. Each particle (or voxel) tries to take the shortest path to minimize the global energy as much as possible (Step (1a)), without knowing other particles' movements. Because their movements change the energy function jointly, rather than independently, a global step length τ is used to coordinate their movement. This strategy is fully compatible with Thirion's Demons algorithm [9] when A is separable.

At each optimization iteration, we do not choose as the descent direction the steepest direction $A'b$, for the following reason. The steepest direction $A'b$ sometimes is dominated by large $A'_i b$ at some voxels (as illustrated in Fig. 1), suppressing other voxels from displacing. Theoretically this phenomenon will disappear as $A'_i b$ approaching zero. However, in practice, on a discrete grid, this cannot be perfectly achieved. We noticed this phenomenon when we initially tried to use the steepest descent optimization.

This local Gauss-Newton strategy converges slower than the Gauss-Newton strategy, but it takes much less computation at each iteration. As shown in Section 2.4, its computational complexity is linear. In our experiments, it achieved equal registration goodness as the Gauss-Newton algorithm in [8], but significantly reduced the registration time and memory usage.

Though this local Gauss-Newton strategy is motivated to solve the problem of diffusion-tensor image registration, its application is not limited to diffusion-tensor images. Generally, it restores the $O(n)$ complexity to the Demons algorithm for situations where the gradient matrix is not separable. It also allows the Demons algorithm to directly incorporate regularization without increasing computational complexity. Application to HARDI image registration, a computation intensive problem, is another interesting extension.

Fig. 1. Descent Directions. The steepest can be dominated by certain components.

2.4 Computational Complexity

The computational complexity of the algorithm is $O(n)$ where n is the number of voxels of interest. For a particular voxel, its tensor reorientation as well as its local energy function involves only the displacement of the voxels in its neighborhood, so A has a block sparse structure and the number and size of the non-zero blocks in A_i are constant numbers fully determined by image dimension and the regularizer's differential operator. Therefore, Step (1a) has $O(1)$ complexity, Step (3) has $O(n)$ complexity and the total complexity is $O(n)$.

3 Experiment

3.1 Data Acquisition and Preprocessing

Diffusion weighed images (DWI) of 120 pediatric subjects were collected with 30-direction isotropic DTI sequence ($b = 1000\text{s/mm}^2$, voxel size = $2 \times 2 \times 2 \text{ mm}^3$, dimension = $128 \times 128 \times 128$). FSL brain extraction tool (BET) was applied to the B_0 images to mask the brain region, with the “R” option turned. Due to the long computation time (about 7 hours) and high memory demand (about 20 GB) of the algorithm in [8], we randomly selected 12 pairs of the subjects for pair-wise registration.

3.2 Image Registration

For the diffeomorphic Demons algorithm proposed by Yeo *et al.* in [8], its open source implementation, the Tensor Toolkit (TTK) (<https://gforge.inria.fr/projects/ttk>) was used, with the parameters recommended in [8]: three level multi-scaling and each with ten iterations. For the Local Gauss-Newton Demons algorithm, we applied the affineness regularization to the displacement field. We adjusted the regularization weight to match the displacement smoothness of TTK’s outputs. In Section 3.4, we evaluated the algorithms with three types of harmonic energy, include those that our algorithm did not directly optimize.

3.3 Evaluation Metric

Tensor rooted-mean-square (RMS) difference, FA correlation coefficients (Corr. Coef.), and MD Corr. Coef., between the fixed image and the transformed moving image, were used to indicate registration similarity. Three types of harmonic energy, affineness, curvature and diffusion, were used to indicate displacement smoothness. We used different harmonic energy to give a comprehensive evaluation, avoiding the bias introduced by algorithms’ preference. All these metrics were estimated within the brain masks.

3.4 Results

For each of the registration cases, the two algorithms achieved similar matching between the fixed images and the warped images, with similar regularization energy, while the Local Gauss-Newton algorithm took significantly less computation time and memory usage, as shown in Table 1 and Figures 2 and 3. Regarding FA Corr. Coef, both algorithms performed almost equally well; regarding MD Corr. Coef and tensor RMS, the Local Gauss-Newton algorithm performed better, with smoother displacement fields.

Table 1. Summary Statistics of the Experiment

	Euclidean RMS	FA Corr. Coef.	MD Corr. Coef.	Time [†]
Local Gauss-Newton	0.700 ± 0.037	0.668 ± 0.027	0.707 ± 0.032	2272 ± 165
Gauss-Newton	0.741 ± 0.046	0.667 ± 0.033	0.656 ± 0.034	24441 ± 485
	Affineness	Curvature	Diffusion	Memory [‡]
Local Gauss-Newton	0.145 ± 0.008	0.108 ± 0.005	0.2680 ± 0.021	≈ 2 GB
Gauss-Newton	0.165 ± 0.006	0.119 ± 0.005	0.3056 ± 0.029	≈ 20 GB

[†]Computation time was estimated on an Intel Xeon 2.80 GHz CPU and single-threaded.

[‡] Memory usage was manually monitored by using the Linux “top” command, during the algorithms worked on the finest resolution of the multi-scale registration.

This experiment with twelve cases suggests that the Local Gauss-Newton method can achieve similar goodness at similar harmonic energy as the method

Fig. 2. Statistics of Each Registration Case. The two algorithms achieved similar goodness with similar regularization energy for each of the registration cases while the Local Gauss-Newton algorithm took much less computation time.

Fig. 3. Registration Results Example. Visually, the registration results of the Gauss-Newton and the Local Gauss-Newton methods look similar. Since they optimize the same energy function, as expected, they yield similar results.

in [8] does. However, the Local Gauss-Newton algorithm significantly reduced the computation time from about 7 hours to about 45 minutes, and decreased memory demand from 20 GB to 2 GB. The smoother displacement fields of the Local Gauss-Newton algorithm was possibly achieved by directly incorporating the affineness regularization into its optimization of the velocity fields.

4 Conclusion

As more and more brain-connectivity studies recruit a large number of subjects, it is important to design registration methods that are not only theoretically

rigorous, but also computationally efficient, for processing large data sets. However, the requirement of reorienting diffusion tensors complicates the calculation of deforming force, and significantly slows computation.

In this paper, we propose a method for the Demons algorithm to exactly incorporate both reorientation and regularization into deforming velocity calculation, but without directly solving a large non-separable linear system. This method restores the $O(n)$ computational efficiency of the original Demons algorithm to diffusion images, without sacrificing registration goodness, in comparison with directly solving a large linear system at each iteration. In our experiments, the new algorithm reduced the computation time and memory demand for ten times.

References

1. The Human Connectome Project, <http://www.humanconnectomeproject.org/>
2. Alexander, D.C., Pierpaoli, C., Basser, P.J., Gee, J.C.: Spatial transformations of diffusion tensor magnetic resonance images. *IEEE Transactions on Medical Imaging* 20(11), 1131–1139 (2001)
3. Park, H.J., Kubicki, M., Shenton, M.E., Guimond, A., McCarley, R.W., Maier, S.E., Kikinis, R., Jolesz, F.A., Westin, C.F.: Spatial normalization of diffusion tensor MRI using multiple channels. *Neuroimage* 20(4), 1995–2009 (2003)
4. Yang, J., Shen, D., Davatzikos, C., Verma, R.: Diffusion Tensor Image Registration Using Tensor Geometry and Orientation Features. In: Metaxas, D., Axel, L., Fichtinger, G., Székely, G. (eds.) *MICCAI 2008, Part II*. LNCS, vol. 5242, pp. 905–913. Springer, Heidelberg (2008)
5. Alexander, D.C., Gee, J.C.: Elastic matching of diffusion tensor images. *Computer Vision and Image Understanding* 77(2), 233–250 (2000)
6. Zhang, H., Yushkevich, P.A., Alexander, D.C., Gee, J.C.: Deformable registration of diffusion tensor MR images with explicit orientation optimization. *Med. Image Anal.* 10(5), 764–785 (2006)
7. Cao, Y., Miller, M., Mori, S., Winslow, R., Younes, L.: Diffeomorphic matching of diffusion tensor images. In: *Conference on Computer Vision and Pattern Recognition Workshop, CVPRW 2006*, p. 67 (June 2006)
8. Yeo, T., Vercauteren, T., Fillard, P., Peyrat, J.M., Pennec, X., Golland, P., Ayache, N., Clatz, O.: DT-REFinD: Diffusion tensor registration with exact finite-strain differential. *IEEE Transactions on Medical Imaging* 28(12), 1914–1928 (2009)
9. Thirion, J.P.: Image matching as a diffusion process: an analogy with Maxwell’s demons. *Med. Image Anal.* 2(3), 243–260 (1998)
10. Vercauteren, T., Pennec, X., Perchant, A., Ayache, N.: Diffeomorphic demons: Efficient non-parametric image registration. *Neuroimage* 45(1 suppl.), S61–S72 (2009)
11. Fischer, B., Modersitzki, J.: A unified approach to fast image registration and a new curvature based registration technique. *Linear Algebra and its Applications* 380, 107–124 (2004)