

DTI REGISTRATION WITH EXACT FINITE-STRAIN DIFFERENTIAL

B.T. Thomas Yeo^{1*}, Tom Vercauteren^{2,3}, Pierre Fillard³, Xavier Pennec³,
Polina Golland¹, Nicholas Ayache³, Olivier Clatz³

(1) CSAIL, MIT, Boston, USA (2) Mauna Kea Technologies, Paris, France
(3) Asclepios Lab, INRIA Sophia-Antipolis, France

ABSTRACT

We propose an algorithm for the diffeomorphic registration of diffusion tensor images (DTI). Previous DTI registration algorithms using full tensor information suffer from difficulties in computing the differential of the Finite Strain tensor reorientation strategy. We borrow results from computer vision to derive an analytical gradient of the objective function. By leveraging on the closed-form gradient and the one-parameter subgroups of diffeomorphisms, the resulting registration algorithm is diffeomorphic and fast. Registration of a pair of $128 \times 128 \times 60$ diffusion tensor volumes takes 15 minutes. We contrast the algorithm with a classic alternative that does not take into account the reorientation in the gradient computation. We show with 40 pairwise DTI registrations that using the exact gradient achieves significantly better registration.

Index Terms— Biomedical image processing, image registration, diffusion tensor imaging

1. INTRODUCTION

Registering scalar images requires a deformation model, an interpolation scheme, a similarity metric and a regularization method. For DTI however, one also needs to define a tensor reorientation scheme in order to warp a tensor image consistently with the anatomy [1]. The two commonly used reorientation strategies, the Finite Strain (FS) reorientation and the Preservation of Principal Directions (PPD) reorientation, have similar empirical performances [2].

Need for reorientation substantially complicates the computation of the gradient of the registration objective function. Many DTI registration algorithms therefore only use scalar values or features that are invariant to image transformations [3].

Alexander and Gee [4] perform elastic registration of tensor images by reorientating the tensors after each iteration using PPD reorientation. The reorientation is not taken into account in the objective function. In the more recent work, Cao

et al. [5] register tensor images diffeomorphically using an exact gradient of the PPD reorientation.

For a general transformation, the FS reorientation is defined by the rotation component of the deformation field. Zhang *et al.* [2] propose a novel piecewise affine registration algorithm to register tensor images using FS reorientation. Because each piecewise transformation considered is affine, the rotation component of the deformation needs *not* be computed. Instead, since rotation is already explicitly optimized in affine registration, the gradient due to FS reorientation can be easily computed. These piecewise affine transformations are fused together to generate a smooth warp field. However, it is unclear how much of the optimality is lost through the fusion of the optimal piecewise affine transformations.

In this work, by borrowing results from pose estimation [6], we solve the open problem of computing the differential of the FS reorientation with respect to the Jacobian. Indeed, since the differential of the Jacobian with respect to the transformation parameters is usually easy to compute, this solves the general problem of finding the differential of the FS reorientation for any parametric transformation. We incorporate the exact FS gradient into a diffeomorphic DTI registration scheme, extending the recently introduced diffeomorphic demons algorithm [7] from scalar to tensor images. The diffeomorphic demons algorithm is a diffeomorphic variant of the original demons algorithm [8], using a fast exponential of stationary velocity fields [9]. The resulting registration is fast, taking about 15 minutes.

We compare our algorithm with an alternative that ignores the reorientation during the gradient computation but reorients tensors after each iteration. This algorithm is a diffeomorphic version of the one proposed by Alexander and Gee [4] using FS reorientation. We show that using the exact gradient results in significantly better registration. In particular, the Sum of Squares Difference (SSD) is improved regardless of the metric (Euclidean or Log-Euclidean) used for interpolation, objective function or evaluation.

To summarize, our contributions are: (*i*) derivation of the exact Finite Strain (FS) differential usable in any DTI registration scheme, (*ii*) incorporation of the FS differential into a fast diffeomorphic DTI registration algorithm, (*iii*) experiments showing that using the exact gradient leads to better

*This research is funded by the INRIA “associated teams” program *CompuTumor*: <http://www-sop.inria.fr/asclepios/projects/boston/>. B.T. Thomas Yeo is funded by the Agency for Science, Technology and Research, Singapore. He did this work while at INRIA. DTI data courtesy of Denis Ducreux, MD, PhD, Bicêtre Hospital, Paris.

registration, (*iv*) an implementation that allows for both Euclidean and Log-Euclidean approaches.

2. FINITE STRAIN DIFFERENTIAL

Deforming a tensor image by a transformation s involves tensor interpolation followed by tensor reorientation [1]. To compute a deformed tensor at a voxel n , one first computes the interpolated tensor $T(n)$. In this work, we will use Euclidean [1] and Log-Euclidean interpolation [10].

According to the FS reorientation strategy [1], we compute the rotation component of the deformation at pixel n :

$$R(n) = (J(n)J(n)^T)^{-\frac{1}{2}}J(n) \quad (1)$$

where $J(n)$ is the Jacobian of the spatial transformation $s = \text{Id} + \mathbf{u}$ at pixel n , \mathbf{u} being a dense non-parametric displacement field. The presence of the matrix square root makes any gradient computation difficult. The interpolated tensor $T(n)$ is then reoriented, resulting in the final tensor $T'(n) = R^T(n) \cdot T(n) \cdot R(n)$. For registration using the finite strain strategy, we see from the chain rule that we need to compute the differential of R with respect to J . Defining $S = (JJ^T)^{\frac{1}{2}}$ and using the results of pose estimation methods [6], we get:

$$dR = -R[R^T(\text{tr}(S)I - S)^{-1}R \sum_i (R^T)_i \otimes (dJ^T)_i]^\oplus \quad (2)$$

where \otimes denotes 3D cross product, $(\cdot)_i$ denotes the i -th column of (\cdot) and \oplus is the operator that converts any 3×1 vector $m = [m_1 \ m_2 \ m_3]^T$ into the skew symmetric matrix $\begin{pmatrix} 0 & -m_3 & m_2 \\ m_3 & 0 & -m_1 \\ -m_2 & m_1 & 0 \end{pmatrix}$. Let J_{ij} be the ij -th component of J . $\frac{\partial R}{\partial J_{ij}}$ is then given by dR in (2) if we set dJ to 0, except for $(dJ)_{ij}$ which is set to 1.

3. DIFFEOMORPHIC DEMONS FOR DTI

3.1. Algorithm overview

The diffeomorphic demons algorithm for scalar images [7] uses a modified demons objective function, equivalent to that of the original demons algorithm [8]. We use the same demons objective function, but have to incorporate the reorientation matrix R into the deformation model.

$$E_c(s) = \|\Sigma^{-1} [F - R^T (M \circ c) R]\|^2 + \sigma_x^{-2} \text{dist}(s, c)^2 + \sigma_T^{-2} \text{Reg}(s) \quad (3)$$

F is the fixed tensor image and M is the moving tensor image. s is the spatial transformation to be optimized, c is an auxiliary variable and $\|\cdot\|$ denotes the Euclidean Sum of Squares Difference (EUC-SSD) similarity measure between the fixed image and warped moving image [4, 2]. R is the rotation matrix computed for each voxel and is implicitly dependent

on the transformation c . Σ is a diagonal matrix that defines the variability at a particular voxel. σ_x and σ_T provide the tradeoffs between the terms in the objective function.

This formulation leads to a fast and simple optimization procedure by alternately optimizing the first two terms and the last two terms of (3) [7]. Typically, one has $\text{dist}(c, s) = \|c - s\|$ and $\text{Reg}(s) = \|\nabla s\|^2$. The regularizations can be modified to handle fluid-like constraints.

F and M can be seen as a $9N \times 1$ vector by ‘‘rasterizing’’ each of the N (3×3) tensors into a column vector. But F and M can also be taken as the Log-Euclidean transforms of the original tensor images (done by converting each tensor T to its logarithm $\log(T)$ [10]). This applies because $\log(R^T T R) = R^T \log(T) R$ for any rotation matrix R . This is convenient, since we only need to perform a one-time logarithm transformation of the tensor images to work under the Log-Euclidean framework [10]. The resulting similarity measure will be referred to as the Log-Euclidean SSD (LOG-SSD).

Under certain smoothness conditions, a stationary velocity field \mathbf{v} parameterizes a diffeomorphism via the exponential map $\exp(\mathbf{v})$ [9]. Formally, $\exp(\mathbf{v})$ is the solution at time 1 of the stationary ODE $\partial c / \partial t = \mathbf{v}(x)$, with $c(0) = \text{Id}$. Thus, $c(1) = \exp(\mathbf{v})$. The scaling and squaring approach [9] allows us to compute the exponential without having to explicitly solve the ODE.

At each iteration, the diffeomorphic demons algorithm looks for an update transformation in the space of diffeomorphisms parameterized by stationary velocity fields and compose the update with the current transformation:

1. Choose a starting spatial transformation s (represented by a displacement field)
2. Iterate until convergence:
 - (i) Given s , compute a velocity field update \mathbf{v} by minimizing the first two terms of (3):
$$E_s(\mathbf{v}) = \|\Sigma^{-1} [F - R^T (M \circ s \circ \exp(\mathbf{v})) R]\|^2 + \sigma_x^{-2} \|\text{dist}(s, s \circ \exp(\mathbf{v}))\|^2$$
 wrt \mathbf{v}
 - (ii) If a fluid like regularization is used, let $\mathbf{v} \leftarrow K_{\text{fluid}} \star \mathbf{v}$. The kernel will typically be Gaussian.
 - (iii) Let $c \leftarrow s \circ \exp(\mathbf{v})$
 - (iv) If a diffusion-like regularization is used, let $s \leftarrow \text{Id} + K_{\text{diff}} \star (c - \text{Id})$ (else let $s \leftarrow c$).

Steps 2(ii) to 2(iv) essentially optimize the last terms of (3).

3.2. Objective Function Derivative

We now focus on the optimization of step 2(i) of the DTI diffeomorphic demons algorithm, which objective function can be written in a non-linear least-squares form:

$$E_s(\mathbf{v}) = \left\| \begin{bmatrix} \Sigma^{-1} [F - R^T (M \circ s \circ e^{\mathbf{v}}) R] \\ \sigma_x^{-1} e^{\mathbf{v}} \end{bmatrix} \right\|^2 = \left\| \begin{bmatrix} \varphi^1(s \circ e^{\mathbf{v}}) \\ \varphi^2(s \circ e^{\mathbf{v}}) \end{bmatrix} \right\|^2 \quad (4)$$

The derivative of φ^2 is a $3N \times 3N$ matrix: $D_s^{\varphi^2}(\mathbf{v} = 0) = \sigma_x^{-1} \text{Id}$. $D_s^{\varphi^1}(0)$ is a sparse $9N \times 3N$ matrix. It is easier to interpret $D_s^{\varphi^1}(0)$ as $N \times N$ blocks of 9×3 matrices. In particular, the (n, j) -th block $[D_s^{\varphi^1}(0)]_{nj}$ is equal to $\frac{\partial \varphi_n^1(s \circ \epsilon^{\mathbf{v}})}{\partial \mathbf{v}(j)} \Big|_{\mathbf{v}=0}$. Using the chain rule and the fact that the derivative of the exponential at the identity is the identity, we get:

$$[D_s^{\varphi^1}(0)]_{nj} = \frac{\partial \varphi_n^1(s \circ (\text{Id} + \mathbf{u}))}{\partial \mathbf{u}(j)} \Big|_{\mathbf{u}=0} \quad (5)$$

Using the product rule, (5) can be decomposed into a sum of:

- $A_{nj} = \frac{\partial R^T(n)}{\partial \mathbf{u}(j)} \Big|_{\mathbf{u}=0} \cdot M \circ s(n) \cdot R(n)$ and A_{nj}^T
- $R^T(n) \cdot \frac{\partial M \circ s \circ (\text{Id} + \mathbf{u})(n)}{\partial \mathbf{u}(j)} \Big|_{\mathbf{u}=0} \cdot R(n)$

$R(n)$ is a function of the Jacobian $J(n)$ of spatial transformation at the n -th pixel. In practice, $J(n)$ is defined numerically using finite central difference, and depends only on the displacement of the 6 neighbors. Therefore,

$$[D_s^{\varphi^1}(0)]_{nn} = -\Sigma^{-1}(n)R^T(n)\nabla(M \circ s)(n)R(n) \quad (6)$$

and for neighbors j of voxel n , we get

$$[D_s^{\varphi^1}(0)]_{nj} = -\Sigma^{-1}(n)(A_{nj} + A_{nj}^T) \quad (7)$$

Using the differential of R from (2), we can compute $\frac{\partial R(n)}{\partial \mathbf{u}(j)}$ using the chain rule.

In summary, we have computed the full gradient of our objective function: $D_s^{\varphi}(0) = \begin{bmatrix} D_s^{\varphi^1}(0) \\ \sigma_x^{-2} \text{Id} \end{bmatrix}$ where $\sigma_x^{-2} \text{Id}$ is a $3N \times 3N$ matrix, while $D_s^{\varphi^1}(0)$ is a sparse $9N \times 3N$ matrix.

3.3. Optimization: Gauss-Newton Method

From the previous sections, we can now write

$$E_s(\mathbf{v}) \approx \left\| \begin{bmatrix} \Sigma^{-1}(F - R^T(M \circ s)R) \\ 0 \end{bmatrix} + \begin{bmatrix} D_s^{\varphi^1}(0) \\ \sigma_x^{-1} \text{Id} \end{bmatrix} \mathbf{v} \right\|^2 \quad (8)$$

Solving this sparse linear system of equations for \mathbf{v} provides an update for a single iteration of the Gauss-Newton optimization method. At the finest multiresolution level, solving the sparse linear system requires about 60 seconds. This is the bottleneck of the algorithm. Due to the fast convergence of the Gauss-Newton method, we typically only need to solve the linear systems 10 times per multi-resolution level. The resulting registration takes about 15 minutes.

4. EXPERIMENTS AND DISCUSSIONS

We consider pairwise registration among 10 DTI images ($128 \times 128 \times 60$, 25 gradient directions). To illustrate the utility of

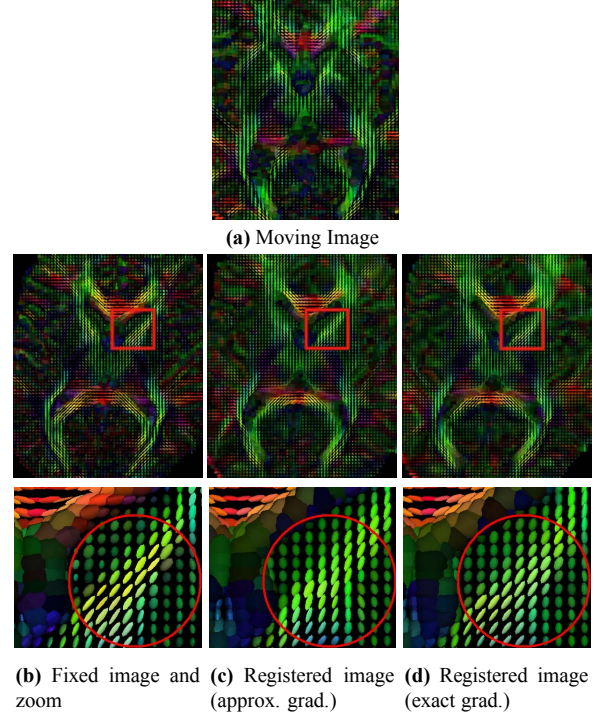


Figure 1: Qualitative comparison between exact FS gradient and approximated gradient for registering a pair of subjects (same parameters) using the Log-Euclidean framework (a) Moving image. (b) Fixed image. (c) Registration using approximated moving image gradient. (d) Registration using exact FS gradient.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

the exact FS gradient, we adapt our algorithm to ignore the re-orientation part of the objective function in the gradient computation. Tensors are treated like vectors, but they are reoriented at every resampling step. This approximated-gradient algorithm is thus a diffeomorphic variant of [4], except we use FS reorientation. In this case, the linear system (8) can be decoupled into 3×3 independent sub-systems. A further simplification of the algorithm uses only the gradient of the fixed image during the entire registration process, similarly to the most common variant of the classical demons algorithm [8].

Figure 1 shows the registration of two subjects. Visually, the exact FS gradient results in better tract alignment with smoother displacement field than the use of the approximated gradient.

To quantitatively compare the performance of the exact FS gradient and the approximated gradient, we consider pairwise registration of 10 DTI. Among the 90 possible registrations, we randomly select 40 pairs of images. The statistics appear to converge after about 30 pairs of registrations.

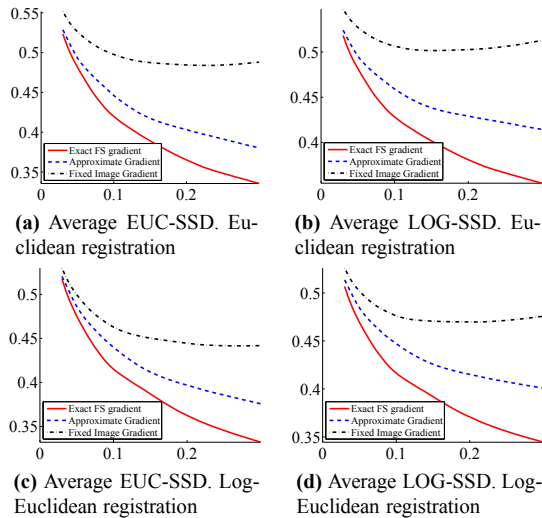


Figure 2: Average SSD (y-axis) over an entire spectrum of harmonic energy (x-axis). Harmonic energy is increased by decreasing the size of the smoothing kernel.

As implied by previous literature, comparing registration algorithms using only one fixed tradeoff between the similarity measure and regularization can potentially bias the comparison. We therefore use smoothing kernels of sizes $\{0.5, 0.6, \dots, 1.9, 2.0\}$ to span a wide range of tradeoffs between similarity and regularization. We find that, for a given smoothing kernel size, using the exact FS differential tends to converge to a solution with a lower harmonic energy. Smaller harmonic energy implies a smoother deformation, providing some evidence that the reorientation provides additional constraint on the registration problem. For each pair of subjects and each kernel size, we perform four registrations, choosing from approximate or exact FS gradient, and Euclidean or Log-Euclidean framework.

For each pair of subjects, we obtain a set of SSD with corresponding harmonic energies. The harmonic energies and SSD across different pair of subjects are different. To average across trials, we linearly interpolate the SSD over a set of harmonic energies for each pair of subjects. We can then compute the mean as well as the standard deviation of SSD across trials for a particular harmonic energy.

Regardless of whether we use the Euclidean framework (Fig. 2a) or the Log-Euclidean one (Fig. 2d), the exact FS gradient leads to an improvement over the entire range of harmonic energies. Moreover, the improvement is statistically significant (one-sided paired-samples t-test p-value less than 10^{-5} for the entire range of harmonic energies). The amount of improvement increases as the harmonic energies increase. In our experiments, a harmonic energy of 0.3 corresponds to severe distortion (pushing the limits of the numerical stability of diffeomorphism), while a harmonic energy of 0.03 corresponds to very smooth warps.

We emphasize that the improvements persist in Fig. 2b and Fig. 2c even though we evaluate a different similarity

measure from the one that is optimized (e.g. evaluating EUC-SSD whereas Log-Euclidean registration is used).

5. CONCLUSION

In this work, we derive the exact differential of the FS reorientation. We propose a fast diffeomorphic DTI registration algorithm using the exact FS differential. We show that the use of the exact differential improves the image similarity metric by 5 to 10 percent over an entire spectrum of harmonic energies. The improvements persist even if we evaluate a different similarity measure from the objective function we optimize.

Taking the reorientation into account therefore allows the algorithm to match two tensor images more easily. The reorientation also provides an additional constraint: the registration algorithm cannot arbitrarily pull in a far-away region for matching because this induces reorientation of tensors in other regions. This additional constraint acts as a further regularization, leading to a better solution.

6. REFERENCES

- [1] D. C. Alexander, C. 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.
- [2] H. Zhang, P. A. Yushkevich, D. C. Alexander, and J. C. Gee, "Deformable registration of diffusion tensor MR images with explicit orientation optimization," *Medical Image Analysis*, vol. 10, no. 5, pp. 764–785, Oct. 2006.
- [3] 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.
- [4] D. C. Alexander and J. C. Gee, "Elastic matching of diffusion tensor images," *Comput. Vis. Image Underst.*, vol. 77, no. 2, pp. 233–250, Feb. 2000.
- [5] Y. Cao, M. I. Miller, S. Mori, R. L. Winslow, and L. Younes, "Diffeomorphic matching of diffusion tensor images," in *Proc. CVPRW'06 - MMBIA*, 2006, p. 67.
- [6] L. Dorst, "First order error propagation of the Procrustes method for 3D attitude estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 2, pp. 221–229, Feb. 2005.
- [7] T. Vercauteren, X. Pennec, A. Perchant, and N. Ayache, "Non-parametric diffeomorphic image registration with the demons algorithm," in *Proc. MICCAI'07*, ser. LNCS, vol. 4792, 2007, pp. 319–326.
- [8] J.-P. Thirion, "Image matching as a diffusion process: An analogy with Maxwell's demons," *Medical Image Analysis*, vol. 2, no. 3, pp. 243–260, Sep. 1998.
- [9] V. Arsigny, O. Commowick, X. Pennec, and N. Ayache, "A Log-Euclidean framework for statistics on diffeomorphisms," in *Proc. MICCAI'06*, ser. LNCS, vol. 4190, 2006, pp. 924–931.
- [10] 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.