

# Spherical Demons: Fast Surface Registration

B.T. Thomas Yeo<sup>1,\*</sup>, Mert Sabuncu<sup>1</sup>, Tom Vercauteren<sup>2</sup>, Nicholas Ayache<sup>3</sup>,  
Bruce Fischl<sup>4,1</sup>, and Polina Golland<sup>1</sup>

<sup>1</sup> Computer Science and Artificial Intelligence Laboratory, MIT, USA

<sup>2</sup> Mauna Kea Technologies, Paris, France

<sup>3</sup> Asclepios Group, INRIA, France

<sup>4</sup> Athinoula A. Martinos Center for Biomedical Imaging, MGH/HMS, USA

**Abstract.** We present the fast Spherical Demons algorithm for registering two spherical images. By exploiting spherical vector spline interpolation theory, we show that a large class of regularizers for the modified demons objective function can be efficiently implemented on the sphere using convolution. Based on the one parameter subgroups of diffeomorphisms, the resulting registration is diffeomorphic and fast – registration of two cortical mesh models with more than 100k nodes takes less than 5 minutes, comparable to the fastest surface registration algorithms. Moreover, the accuracy of our method compares favorably to the popular FreeSurfer registration algorithm. We validate the technique in two different settings: (1) parcellation in a set of in-vivo cortical surfaces and (2) Brodmann area localization in ex-vivo cortical surfaces.

## 1 Introduction

Motivated by the spherical representation of the cerebral cortex, this paper deals with the problem of registering two spherical images. Cortical folding patterns are correlated with both cytoarchitectural [13] and functional regions [11]. In group studies of cortical structure and function, determining corresponding folds across subjects is therefore important. There has been much effort focused on registering cortical surfaces in 3D [9,10,14]. Since cortical areas – both structure and function – are arranged in a mosaic across the cortical surface, an alternative approach is to model the surface as a 2D closed manifold in 3D and to warp the resulting spherical coordinate system [11,18,20,21,23].

Unfortunately, many spherical warping algorithms are computationally expensive. One reason is the need for invertible deformations that preserve the topology of structural or functional regions across subjects. Previously demonstrated methods for cortical registration [11,18,23] rely on soft regularization constraints to encourage invertibility. They require computationally expensive steps of unfolding the mesh triangles or small optimization steps to achieve invertibility [11,23]. Elegant regularization penalties to guarantee invertibility exist [2,16] but make certain assumptions valid only in Euclidean spaces.

---

\* Corresponding author: [ythomas@csail.mit.edu](mailto:ythomas@csail.mit.edu)

An alternative approach to achieve invertibility is to work in a Lie group of diffeomorphisms [3,4,9,15,22], the theory of which can be extended to manifolds [15]. The Large Deformable Diffeomorphic Metric Mapping (LDDMM) [4,9,15] is a popular framework under this paradigm that seeks diffeomorphisms parametrized by time-varying velocity fields. Because LDDMM optimizes over the entire path of the diffeomorphism, the resulting algorithm is slow.

In this paper, we take the approach, previously demonstrated in the Euclidean space [22], of restricting the deformation space to be a composition of diffeomorphisms, each of which is parameterized by a stationary velocity field. In each iteration, the algorithm greedily seeks the best diffeomorphism to be composed with the current transformation, resulting in much faster updates.

Another challenge in registration is the tradeoff between the image similarity measure and the regularization in the objective function. Since most regularizations favor smooth deformations, the gradient computation is complicated by the need to take into account the deformation in neighboring regions. For Euclidean images, the demons objective function [19,6,22] facilitates a fast two-step optimization where the second step handles the warp regularization via a single convolution with a smoothing filter.

Based on spherical vector spline interpolation theory [15] and other differential geometric tools, we show that the two-stage optimization procedure of the demons algorithm can be efficiently applied on the sphere. The problem is not trivial since tangent vectors at different points on the sphere are not comparable. We also emphasize that the extension of the demons algorithm to the sphere is *independent* of our choice of the space of admissible warps.

The Spherical Demons algorithm takes less than 5 minutes on a Xeon 3.2GHz processor, comparable to other non-linear cortical surface registration algorithms whose runtimes range from minutes [10,18] to more than an hour [11,23]. However, these fast algorithms [10,18] suffer from folding spherical triangles and intersecting triangles in 3D since only soft constraints are used.

Unlike [9,15], we do not assume the existence of corresponding landmarks. While landmark-free registration is harder, we demonstrate that our algorithm is accurate in both cortical parcellation and cyto-architectonic localization applications.

The contributions of this paper are multi-fold. First, we show that the demons algorithm can be efficiently applied on the sphere. Second, the use of a limited class of diffeomorphisms yields a speed gain of more than an order of magnitude compared with other landmark-free diffeomorphic spherical registration methods. Finally, we validate our algorithm by showing accuracy comparable to the popular FreeSurfer algorithm [11] on two data sets.

## 2 Background - Demons Algorithm

We choose to work with the modified demons objective function, essentially identical to [6,22], with a slightly different interpretation:

$$(s^*, c^*) = \operatorname{argmin}_{s, c} \|F - M \circ c\|^2 + \frac{1}{\sigma_x^2} \operatorname{dist}(s, c) + \frac{1}{\sigma_T^2} \operatorname{Reg}(s) \quad (1)$$

where  $F$  is the fixed image,  $M$  is the moving image,  $c$  is the desired registration and  $s$  is a hidden transformation that acts as a prior on  $c$ . The fixed image  $F$  and warped moving image  $M \circ c$  are treated as  $N \times 1$  vectors. Typically,  $\text{dist}(s, c) = \|s - c\|^2$ , encouraging the resulting transformation  $c$  to be close to the hidden transformation  $s$  and  $\text{Reg}(s) = \|\nabla s\|^2$ , i.e., the regularization penalizes the gradient magnitude of the hidden transformation  $s$ .  $\sigma_x$  and  $\sigma_T$  provide a tradeoff among the different terms of the objective function.

This formulation facilitates a two-step optimization procedure that alternately optimizes the first two and last two terms of Eq. (1). Starting from an initial displacement field  $s^0$ , the demons algorithm iteratively seeks an update transformation to be composed with the current estimate.

### Algorithm: Demons

**Step 1.** Given  $s^{(t)}$ , minimize the first two terms of Eq. (1)

$$u^{(t)} = \underset{u}{\operatorname{argmin}} \|F - M \circ \{s^{(t)} \circ u\}\|^2 + \frac{1}{\sigma_x^2} \text{dist}(s^{(t)}, \{s^{(t)} \circ u\}) \quad (2)$$

where  $u$  is any admissible transformation. Compute  $c^{(t)} = s^{(t)} \circ u^{(t)}$ .

**Step 2.** Given  $c^{(t)}$ , minimize the last two terms of Eq. (1):

$$s^{(t+1)} = \underset{s}{\operatorname{argmin}} \frac{1}{\sigma_x^2} \text{dist}(s, c^{(t)}) + \frac{1}{\sigma_T^2} \text{Reg}(s) \quad (3)$$

In the original demons algorithm [19], the space of admissible warps include all displacement fields:  $u$ ,  $s$  and  $c$  are 3D displacement fields, and the composition operator  $\circ$  corresponds to the addition of displacement fields. In the Diffeomorphic Demons algorithm [22],  $u$  is a diffeomorphism from  $\mathbb{R}^3$  to  $\mathbb{R}^3$  parameterized by a stationary velocity field  $v$ . Under certain smoothness conditions, a stationary velocity field  $v$  is related to a diffeomorphism through the exponential mapping  $u = \exp(v)$ . In this case,  $\exp(v)$  is the solution at time 1 of the stationary ODE  $\partial x(t)/\partial t = v(x(t))$ , with  $x(0) \in \mathbb{R}^3$ . Deformation  $\exp(v)(\cdot)$  maps point  $x(0)$  to point  $x(1)$ .

The demons algorithm and its variants are fast because for certain forms of  $\text{dist}(s, c)$  and  $\text{Reg}(s)$ , step 1 reduces to a non-linear least-squares problem that can be efficiently minimized via Gauss-Newton optimization and step 2 can be solved by a single convolution of the displacement field  $c$  with a smoothing filter.

## 3 Spherical Demons

In this section, we show that suitable choices of  $\text{dist}(s, c)$  and  $\text{Reg}(s)$  lead to efficient optimization on the sphere  $S^2$ . We work with updates  $u$  that are diffeomorphisms parameterized by a stationary velocity field  $v$ . We emphasize that unlike [22],  $v$  is tangent to the sphere and not an arbitrary 3D vector. It is also easy to extend our results to other transformations, e.g., spherical splines.

**Choice of  $\text{dist}(s, c)$ .** Suppose the transformation  $c$  maps a point  $x_n \in S^2$  to a point  $c(x_n) \in S^2$ . Let  $T_{x_n} S^2$  be the tangent space at  $x_n$ . We define  $\vec{c}_n \in T_{x_n} S^2$  to be the tangent vector at  $x_n$ , pointing along the great circle connecting  $x_n$  to  $c(x_n)$ , with length equal to the sine of the angle between  $x_n$  and  $c(x_n)$ . There is a 1-to-1 correspondence between  $c(x_n)$  and  $\vec{c}_n$ , assuming the angle is less than  $\pi/2$ , which is reasonable even for relatively large deformations. On a unit sphere,  $\vec{c}_n = -x_n \times (x_n \times c(x_n)) = -D_n^2 c(x_n)$ , where  $D_n$  is the  $3 \times 3$  skew-symmetric matrix representing the cross-product of  $x_n$  with another vector.

For a mesh of  $N$  vertices  $\{x_n\}_{n=1}^N$ , the set of transformed points  $\{c(x_n)\}_{n=1}^N$  (or equivalently tangent vectors  $\{\vec{c}_n\}_{n=1}^N$ ), together with a choice of an interpolation function, define the transformation  $c$  completely. Similarly, we can define  $s$  through  $\{\vec{s}_n\}_{n=1}^N$ . We emphasize that these tangent vector fields are just convenient representations of the transformations  $s$  and  $c$  and should not be confused with the velocity vector field  $v$ . We define  $\text{dist}(s, c) = \sum_{n=1}^N \|\vec{s}_n - \vec{c}_n\|^2$ , which is well-defined since both  $\vec{c}_n$  and  $\vec{s}_n$  belong to  $T_{x_n} S^2$ .

**Choice of  $\text{Reg}(s)$ .** With a slight abuse of notation, we use  $s$  to denote both the transformation and its equivalent tangent vector field representation. We assume  $s$  belongs to the Hilbert space  $V$  of vector fields obtained by the closure of the space of smooth vector fields on  $S^2$  via a choice of the so-called energetic norm denoted by  $\|\cdot\|_V$  [15]. We define  $\text{Reg}(s) = \|s\|_V$ . With a proper choice of the energetic norm, a smaller value of  $\|s\|_V$  corresponds to a smoother vector field. As we will see, the exact choice of the norm is unimportant for our purposes.

**Step 1.** With our choice of  $\text{dist}(s, c)$ , Step 1 of the demons algorithm is a minimization with respect to the velocity field  $v$  defined by  $\{\vec{v}_n \in T_{x_n} S^2\}_{n=1}^N$ :

$$\{\vec{v}_n^{(t)}\}_{\{\vec{v}_n\}} = \underset{\{\vec{v}_n\}}{\text{argmin}} \|F - M \circ \{s^{(t)} \circ \exp(v)\}\|^2 + \frac{1}{\sigma_x^2} \sum_{n=1}^N \|\vec{s}_n^{(t)} + D_n^2 \{s^{(t)} \circ \exp(v)\}(x_n)\|^2 \quad (4)$$

$\vec{v}_n$  is a  $3 \times 1$  tangent vector on the sphere. Let  $\vec{e}_1, \vec{e}_2$  be  $3 \times 1$  orthogonal basis vectors of the tangent space  $T_{x_n} S^2$ . We can write  $\vec{v}_n = [\vec{e}_1 \vec{e}_2] \vec{v}_n'$  where  $\vec{v}_n'$  is a  $2 \times 1$  vector. We can thus optimize Eq. (4) with respect to  $\{\vec{v}_n'\}$ . The above non-linear least-squares form can be optimized efficiently with the Gauss-Newton method, which requires finding the gradient of both terms with respect to  $\{\vec{v}_n'\}$  at  $\{\vec{v}_n' = 0\}$  and solving a linearized least-squares problem. By switching back and forth between the tangent representation  $\vec{v}_n'$  and embedding space representation  $\vec{v}_n$ , we construct an update rule independent of the choice of coordinate frames.

To see that, let  $\overline{M}$  be a 3D image defined to be any smooth extension of  $M \circ s^{(t)}$ . For example, for all  $x \in \mathbb{R}^3 \setminus 0$ ,  $\overline{M}(x) = M \circ s^{(t)}(x/\|x\|)$ . Similarly, we extend  $s^{(t)}$  to  $\vec{s}$ . Let  $\vec{m}_n^T = \nabla \overline{M}(x_n)$  be the  $1 \times 3$  gradient of  $\overline{M}$  at  $x_n$  and  $B_n^T = \nabla \vec{s}(x_n)$  be the  $3 \times 3$  gradient of  $\vec{s}$ . Since the differential of  $\exp(v')$  at  $v' = 0$  is the identity, the derivative of the entries corresponding to vertex  $n$  in both terms of Eq. (4) with respect to  $\vec{v}_k'$  is 0 if  $n \neq k$ . With some algebra, we get:

$$\frac{\partial}{\partial \vec{v}_n} \left[ F(x_n) - M \circ \{s^{(t)} \circ \exp(\vec{v})\}(x_n) \right]_{\vec{v}=0} = -\vec{m}_n^T \left[ \vec{e}_1 \vec{e}_1^T + \vec{e}_2 \vec{e}_2^T \right] \quad (5)$$

$$\frac{\partial}{\partial \vec{v}_n} \left[ \vec{s}_n^{(t)} + D_n^2 \{s^{(t)} \circ \exp(\vec{v})\}(x_n) \right]_{\vec{v}=0} = D_n^2 B_n^T \left[ \vec{e}_1 \vec{e}_1^T + \vec{e}_2 \vec{e}_2^T \right] \quad (6)$$

The above equations involve the projection of the spatial gradients of  $\overline{M}$  and  $\overline{s}$  onto the tangent space  $T_{x_n}S^2$  and are therefore independent of the choices of  $\vec{e}_1$ ,  $\vec{e}_2$  or the extension mechanism for  $\overline{M}$  and  $\overline{s}$ . This leads to the following Gauss-Newton update for velocity vector  $\vec{v}'_n$ :

$$\vec{v}'_n{}^{(t)} = (F(x_n) - M \circ s^{(t)}(x_n)) \times \left( \begin{bmatrix} \vec{e}_1^T \\ \vec{e}_2^T \end{bmatrix} \left[ \vec{m}_n \vec{m}_n^T + \frac{1}{\sigma_x^2} B_n (D_n^2)^T D_n^2 B_n^T \right] \begin{bmatrix} \vec{e}_1 & \vec{e}_2 \end{bmatrix} \right)^{-1} \begin{bmatrix} \vec{e}_1^T \\ \vec{e}_2^T \end{bmatrix} \vec{m}_n \quad (7)$$

Eq. (7) is used to compute  $\vec{v}'_n{}^{(t)}$  for each vertex  $n$  independently and is therefore fast. We can then compute  $\vec{v}_n^{(t)} = [\vec{e}_1 \ \vec{e}_2] \vec{v}'_n{}^{(t)}$  and use scaling and squaring to estimate  $\exp(v^{(t)})$  [1], which is then composed with the current transformation estimate  $s^{(t)}$  to form  $c^{(t)} = s^{(t)} \circ \exp(v^{(t)})$ . It is less obvious here, but the update is independent of the choice of basis vectors  $\vec{e}_1$ ,  $\vec{e}_2$  and extensions  $\overline{s}^{(t)}$  and  $\overline{M}_s$ .

**Step 2.** The optimization in Step 2 of the demons algorithm

$$s^{(t+1)} = \underset{s}{\operatorname{argmin}} \frac{1}{\sigma_x^2} \sum_{n=1}^N \|\vec{s}_n - \vec{c}_n^{(t)}\|^2 + \frac{1}{\sigma_T^2} \|s\|_V \quad (8)$$

seeks a smooth vector field  $s$  that approximates the tangent vectors  $\{\vec{c}_n^{(t)}\}_{n=1}^N$ . The optimum  $s^{(t+1)}$  is unique and is computed by solving a large system of linear equations [15]. We extend the results in [15], proving that the optimum vector field  $s^{(t+1)}$  at  $x_n$  always has the form

$$\vec{s}_n^{(t+1)} = \sum_{i=1}^N \lambda(x_i, x_n) T(x_i, x_n) \vec{c}_i^{(t)} \quad (9)$$

where  $T(x_i, x_n) \vec{c}_i$  is a linear transformation that parallel transports  $\vec{c}_i$  along the great circle from  $T_{x_i}S^2$  to  $T_{x_n}S^2$  and  $\lambda(x_i, x_n)$  is a non-negative scalar function that monotonically decreases as a function of the distance between  $x_i$  and  $x_n$ . We omit the proof due to space constraints.

In contrast to [15], Eq. (9) implies that we can avoid solving a large system of equations. The optimal tangent vector  $\vec{s}_n^{(t+1)} \in T_{x_n}S^2$  is given by a linear combination of  $\vec{c}_i^{(t)} \in T_{x_i}S^2$  parallel transported to  $T_{x_n}S^2$ , where the tangent vectors of closer points are given more weights via  $\lambda$ . One can interpret Eq. (9) as a spherical convolution of vector fields. This is the exact analogue of the convolution method of optimizing Step 2 in the demons algorithm [6,19,22] and is also similar to the convolution-based fast fluid registration in the Euclidean space [7].

The exact form of  $\lambda(\cdot, \cdot)$  is determined by the choice of the energetic norm, the relative locations of all the mesh points and the constant  $\sigma_x^2/\sigma_T^2$ . In particular, increasing  $\sigma_x^2/\sigma_T^2$  increases the “width” of  $\lambda$ . Rather than picking the energetic norm, we can simply choose a convenient  $\lambda$ . In practice, we replace the convolution operation with iterative smoothing: at each iteration, for each vertex  $x_n$ , tangent vectors of neighboring vertices are parallel transported to  $T_{x_n}S^2$  and linearly combined in a weighted fashion with the current estimate of  $\vec{s}_n$ . Using

more iterations of this process is equivalent to increasing the width of  $\lambda$ . Technically, the resulting  $\lambda$  might not correspond to any choice of the energetic norm. However, in practice, this does not appear to be a problem.

## 4 Experiments and Discussion

We use two sets of experiments to compare the accuracy of Spherical Demons and FreeSurfer [11]. The FreeSurfer registration algorithm uses the same similarity measure as Spherical Demons, but penalizes for metric and areal distortion. As mentioned earlier, its runtime is more than an hour while our runtime is less than 5 minutes.

**Parcellation of In-Vivo Cortical Surfaces.** We consider a set of 39 left and right cortical surface models extracted from in-vivo MRI. Each surface is spherically parameterized and represented as a spherical image with geometric features at each vertex (e.g., sulcal depth and curvature). Both hemispheres are manually parcellated by a neuroanatomist into 35 major sulci and gyri. We validate our algorithm in the context of automatic cortical parcellation.

We co-register all 39 spherical images of cortical geometry with Spherical Demons by iteratively building an atlas and registering the surfaces to the atlas. The atlas consists of the mean and variance of cortical geometry. One can easily modify the demons objective function (Eq. (1)) to use an atlas.

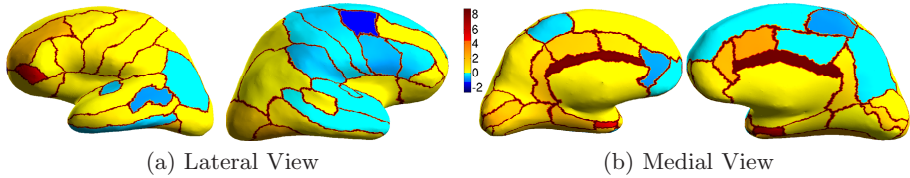
We then perform cross-validation parcellation 4 times, by leaving out subjects 1 to 10, training a classifier [8,12] using the remaining subjects, and using it to classify subjects 1 to 10. We repeat with subjects 11-20, 21-30 and 31-39.

We also perform registration and cross-validation with the FreeSurfer algorithm [11] using the same features and parcellation algorithm [8,12]. Once again, the atlas consists of the mean and variance of cortical geometry.

The average Dice measure (defined as the ratio of cortical surface area with correct labels to the total surface area averaged over the test set) on the left hemisphere is 88.9 for FreeSurfer and 89.6 for Spherical Demons. While the improvement is not big, the difference is statistically significant for a one-sided t-test with the Dice measure of each subject treated as an independent sample ( $p = 2 \times 10^{-6}$ ). On the right hemisphere, FreeSurfer obtains a Dice of 88.8 and Spherical Demons achieves 89.1. Here, the improvement is smaller, but still statistically significant ( $p = 0.01$ ).

Because the average Dice can be deceiving by suppressing small structures, we analyze the segmentation accuracy per structure. On the left (right) hemisphere, the segmentations of 16 (8) structures are statistically significantly improved by Spherical Demons with respect to FreeSurfer, while no structure got worse (False Discovery Rate = 0.05 [5]). Fig. 1 shows the percentage improvement of individual structures. Parcellation results suggest that our registration is at least as accurate as FreeSurfer.

**Brodmann Areas Localization on Ex-vivo Cortical Surfaces.** In this experiment, we evaluate the registration accuracy on ten human brains analyzed



**Fig. 1.** Percentage dice improvement over FreeSurfer. Yellow regions indicate structures scoring better than FreeSurfer. Blue regions correspond to decrease in accuracy. We note that no structure is statistically worse than FreeSurfer (see text).

**Table 1.** Average alignment errors of Brodmann areas in *mm* for the two registration methods. Lowest errors are shown in SD: Spherical Demons. FS: FreeSurfer.

|      | Left Hemisphere |            |            |            |            |            |            |            | Right Hemisphere |            |            |            |            |            |             |            |
|------|-----------------|------------|------------|------------|------------|------------|------------|------------|------------------|------------|------------|------------|------------|------------|-------------|------------|
|      | V1              | BA4a       | BA4p       | BA2        | V2         | BA6        | BA44       | BA45       | V1               | BA4a       | BA4p       | BA2        | V2         | BA6        | BA44        | BA45       |
| SD10 | <b>3.1</b>      | <b>3.3</b> | <b>3.3</b> | <b>5.4</b> | <b>3.7</b> | <b>6.4</b> | 7.7        | <b>6.4</b> | <b>3.2</b>       | <b>3.4</b> | <b>2.8</b> | <b>5.4</b> | <b>3.5</b> | <b>6.4</b> | 10.4        | <b>8.6</b> |
| FS10 | 3.8             | 4.4        | 3.8        | 6.3        | 4.6        | 7.0        | <b>7.4</b> | 6.8        | 3.8              | 3.8        | 3.1        | 5.9        | 4.0        | 6.5        | <b>11.5</b> | 9.9        |

histologically postmortem [17]. The histological sections were aligned to post-mortem MR with nonlinear warps to build a 3D volume. Eight manually labeled Brodmann areas from histology were sampled onto each hemispheric surface model and sampling errors were manually corrected [13]. Brodmann areas are cyto-architectonically defined regions closely related to cortical function.

It has been shown that nonlinear surface registration of cortical folds can significantly improve Brodmann area overlap across different subjects [13,24]. Registering the ex-vivo surfaces is more difficult than in-vivo surfaces because the reconstructed volumes are extremely noisy, resulting in noisy geometric features.

We co-register the ten surfaces to each other by iteratively building an atlas and registering the surfaces to the atlas. We compute the average distance between the boundaries of the Brodmann areas for each pair of registered subjects. For future reference, we call this experiment **SD10**. We repeat this process using FreeSurfer, referred to as **FS10** in the remainder of this section.

Table 1 presents the summary of the results. We see that SD10 outperforms FS10. We perform a permutation test to test for statistical significance. Compared with FS10, SD10 improves the alignment of 5 (2) Brodmann areas on the left (right) hemisphere (False Discovery Rate = 0.05 [5]) and no structure gets worse. These results suggest that the Spherical Demons algorithm is at least as accurate as FreeSurfer in aligning Brodmann areas. Another experiment (not shown) which involves registering the ex-vivo surfaces to an in-vivo atlas also results in comparable performance between Spherical Demons and FreeSurfer.

### 5 Conclusion

In this paper, we presented the fast Spherical Demons algorithm for registering spherical images. We showed that the characteristic two-step optimization of the



Demons algorithm can also be applied on the sphere. A clear future challenge is to take into account the original metric properties of the cortical surface in the registration process [11,20]. We tested the algorithm extensively on two data sets and show that the accuracy of the algorithm compares favorably with the widely used FreeSurfer registration algorithm [11], while offering more than one order of magnitude speedup.

**Acknowledgments.** We would like to thank Hartmut Muhlberg, Katrin Amunts and Karl Zilles for the histological dataset. Support for this research is provided in part by the NIMIC (NIH NIBIB NIMIC U54-EB005149), the NAC (NIT CRR NAC P41-RR13218), the mBIRN (NIH NCRR mBIRN U24-RR021382), the NIH NINDS R01-NS051826 grant, the NSF CAREER 0642971 grant, NCRR (P41-RR14075, R01 RR16594-01A1), the NIBIB (R01 EB001550, R01EB006758), the NINDS (R01 NS052585-01) and the MIND Institute. Additional support was provided by The Autism & Dyslexia Project funded by the Ellison Medical Foundation. B.T. Thomas Yeo is funded by the A\*STAR, Singapore.

## References

1. Arsigny, V., et al.: A Log-Euclidean Framework for Statistics on Diffeomorphisms. In: Larsen, R., Nielsen, M., Sporrang, J. (eds.) MICCAI 2006. LNCS, vol. 4190, pp. 924–931. Springer, Heidelberg (2006)
2. Ashburner, J., Andersson, J., Friston, K.: High-dimensional Image Registration using Symmetric Priors. *NeuroImage* 9, 619–628 (1999)
3. Ashburner, J.: A Fast Diffeomorphic Image Registration Algorithm. *NeuroImage* 38, 95–113 (2007)
4. Beg, M., et al.: Computing Large Deformation Metric Mapping via Geodesic Flows of Diffeomorphisms. *International Journal of Computer Vision* 61(2), 139–157 (2005)
5. Benjamini, Y., Hochberg, Y.: Controlling the False Discovery Rate: A Practical and Powerful Approach to Mult. Testing. *J. Roy. Stats. Soc.* 57(1), 289–300 (1995)
6. Cachier, P.: Iconic Feature Based Non-rigid Registration: The PASHA Algorithm. *CVIU* 89(2-3), 272–298 (2003)
7. Bro-Nielsen, M., Gramkow, C.: Fast Fluid Registration of Medical Images Visualization in Biomedical Computing, 267–276 (1996)
8. Desikan, R., et al.: An automated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based regions of interest. *NeuroImage* (2006)
9. Durrleman, S., et al.: Measuring Brain Variability via Sulcal Lines Registration: a Diffeomorphic Approach. In: Ayache, N., Ourselin, S., Maeder, A. (eds.) MICCAI 2007, Part I. LNCS, vol. 4791, pp. 675–682. Springer, Heidelberg (2007)
10. Eckstein, I., et al.: Generalized Surface Flows for Deformable Registration and Cortical Matching. In: Ayache, N., Ourselin, S., Maeder, A. (eds.) MICCAI 2007, Part I. LNCS, vol. 4791, pp. 692–700. Springer, Heidelberg (2007)
11. Fischl, B., et al.: High-resolution intersubject averaging and a coordinate system for the cortical surface. *HBM* 8(4), 272–284 (1999)
12. Fischl, B., et al.: Automatically Parcellating the Human cerebral Cortex. *Cerebral Cortex* 14, 11–22 (2004)



13. Fischl, B., et al.: Cortical Folding Patterns and Predicting Cytoarchitecture. *Cerebral Cortex* (2007)
14. Geng, X., et al.: Transitive Inverse-Consistent Manifold Registration. In: Christensen, G.E., Sonka, M. (eds.) *IPMI 2005*. LNCS, vol. 3565, pp. 468–479. Springer, Heidelberg (2005)
15. Glaunès, J., et al.: Landmark Matching via Large Deformation Diffeomorphisms on the Sphere. *Journal of Mathematical Imaging and Vision* 20, 179–200 (2004)
16. Nielsen, M., et al.: Brownian Warps: A Least Committed Prior for Non-rigid Registration. In: Dohi, T., Kikinis, R. (eds.) *MICCAI 2002*. LNCS, vol. 2489, pp. 557–564. Springer, Heidelberg (2002)
17. Schleicher, A., et al.: Observer independent method for microstructural parcellation of cerebral cortex: a quantitative approach to cytoarchitectonics. *NeuroImage* 9, 165–177 (1999)
18. Tosun, D., Prince, J.: Cortical Surface Alignment Using Geometry Driven Multi-spectral Optical Flow. In: Christensen, G.E., Sonka, M. (eds.) *IPMI 2005*. LNCS, vol. 3565, pp. 480–492. Springer, Heidelberg (2005)
19. Thirion, J.: Image Matching as a Diffusion Process: an Analogy with Maxwell's Demons. *Medical Image Analysis* 2(3), 243–260 (1998)
20. Thompson, P., et al.: Mathematical/Computational Challenges in Creating Deformable and Probabilistic Atlases of the Human Brain. *HBM* 9(2), 81–92 (2000)
21. Van Essen, D., et al.: Functional and structural mapping of human cerebral cortex: solutions are in the surfaces. *PNAS* 95(3), 788–795 (1996)
22. Vercauteren, T., et al.: Non-parameteric Diffeomorphic Image Registration with the Demons Algorithm. In: Ayache, N., Ourselin, S., Maeder, A. (eds.) *MICCAI 2007, Part II*. LNCS, vol. 4792, pp. 319–326. Springer, Heidelberg (2007)
23. Yeo, B.T.T., et al.: Effects of Registration Regularization and Atlas Sharpness on Segmentation Accuracy. In: Ayache, N., Ourselin, S., Maeder, A. (eds.) *MICCAI 2007, Part I*. LNCS, vol. 4791, pp. 683–691. Springer, Heidelberg (2007)
24. Yeo, B.T.T., et al.: What Data to Co-register for Computing Atlases. In: *MMBIA, Proc. ICCV*, pp. 1–8 (2007)