

# Template Estimation for Large Database : a Diffeomorphic Iterative Centroid Method Using Currents

Claire Cury<sup>12346</sup>, Olivier Colliot<sup>1234</sup>, and Joan A. Glaunès<sup>5</sup>

<sup>1</sup> Université Pierre et Marie Curie-Paris 6, Centre de Recherche de l'Institut du Cerveau et de la Moëlle épinière, Paris, France

<sup>2</sup> Inserm, UMR-S975, CNRS, UMR 7225, Paris, France

<sup>3</sup> ICM Institut du Cerveau et de la Moëlle épinière, 47 boulevard de l'hôpital, 75013 Paris, France

<sup>4</sup> Aramis project-team, Inria Paris-Rocquencourt, Pitié Salpêtrière Hospital, Paris, France

<sup>5</sup> MAP5, Université Paris Descartes, Sorbonne Paris Cité, France

<sup>6</sup> Corresponding author: [claire.cury@gmail.com](mailto:claire.cury@gmail.com)

**Abstract.** Computing a template in the Large Deformation Diffeomorphic Metric Mapping (LDDMM) framework is a key step for the study of large databases of anatomical surfaces, but can lead to very computationally expensive algorithms in the case of large databases. Here we present an iterative method to compute a better initialization for the template estimation method proposed in [1]. The method provides quickly a centroid of the population in shape space. Using this centroid as initialization for template estimation can save up to 72% of computation times.

## 1 Introduction

Computational anatomy aims at developing tools for the analysis of variability of anatomical structures and its variation in healthy and pathological cases [2]. The Large Deformation Diffeomorphic Metric Mapping (LDDMM) framework [3][4] has been widely used for the study of intra-population variability and inter-population differences. It focuses the study on the spatial transformations which can match subjects' anatomies one to each other, or one to a template structure which needs to be estimated. Several methods have been proposed to create template in the LDDMM framework [5][1][6]. The LDDMM framework is of great interest for anatomical structures analysis but it is computationnally expensive. Here we are interested in the hippocampus brain structure which is implicated in neurodegenerative diseases. We use the method presented by Glaunès et al. [1] which proceeds by optimizing a template composed with all shapes of the population co-registered, and therefore does not require a priori on the template topology. However, for heterogeneous populations the method

will need many iterations to converge. For large databases, the method can take months of computation using a standard CPU implementation. Here we are using a GPU implementation, that increases the speed by a factor 200, but for large database a few days is still needed. A way to lower the number of iterations is to start with a better initialization. In this paper we propose a new method for the initialization of template estimation algorithms, which compute iteratively a sort of centroid of the population. We will show that this new method called Iterative Centroid, provides a centroid which is already correctly centered within the population of shapes, which drastically improves performance and convergence of the subsequent template estimation algorithm. This method could be used as it is, because shape analysis template-based it is a linear approximation of the population geometry from a shape space point of view. The paper is organized as follows: first we present the mathematical framework, then the template estimation method and the Iterative Centroid method. We then present results of experiments on real and synthetic hippocampi.

## 2 Notation and mathematical setup

**Large Diffeomorphics Deformations** In the LDDMM framework, deformations maps  $\varphi : \mathbb{R}^3 \rightarrow \mathbb{R}^3$  are generated by integration of time-dependent vector fields  $v(x, t)$ , with  $x \in \mathbb{R}^3$  and  $t \in [0, 1]$ . If  $v(x, t)$  has enough regularity, the

transport equation: 
$$\begin{cases} \frac{d\phi}{dt}(x, t) = v(\phi_v(x, t), t) & \forall t \in [0, 1] \\ \phi_v(x, 0) = x & \forall x \in \mathbb{R}^3 \end{cases}$$
 has a unique solution

and one sets  $\varphi = \phi_v(\cdot, 1)$ , the diffeomorphism induced by  $v(x, t)$ . When considering vector fields  $(v(\cdot, t))_{t \in [0, 1]}$  in  $L^2([0, 1], V)$ , where  $V$  is a Reproducing Kernel Hilbert Space (R.K.H.S.) embedded in the space of  $C^1$  vector fields, the induced set of diffeomorphisms has a group structure with a right-invariant metric defined by the rule  $D^2(Id, \varphi) = \inf \{ \|v\|_V ; v \in V, \phi_v(\cdot, 1) = \varphi \}$ . Considering two surfaces  $S_1$  and  $S_2$ , the optimal matching between them is defined in an ideal setting as the map  $\varphi$  that minimizes  $D(Id, \varphi)$  under the constraint  $\varphi(S_1) = S_2$ . In practice such an exact matching is not feasible and one writes inexact unconstrained matching functionals. In both cases one can show that the vector fields  $v(x, t)$  which induce the optimal deformation map can be written via a convolution formula over the surface involving the reproducing kernel  $K_V$  of the Hilbert space  $V$ . In a discrete setting this writes  $v(x, t) = \sum_{i=1}^n K_V(x_i(t), n) \alpha_i(t)$ , with  $(x_i(t))_{i \in 1:n}$  vertices of the deformed mesh  $\varphi(S_1, t)$ , and  $\alpha_i$  momentum vectors.

**Currents** The use of currents was introduced in order to give a dissimilarity measure between surfaces which does not assume point correspondences between anatomical structures. Any differential  $m$ -form  $\omega$  of  $\mathbb{R}^d$  can be integrated over an oriented  $m$ -submanifold  $S$ . Hence one can define a linear form  $[S]$  over the space of  $m$ -forms via the rule  $[S](\omega) = \int_S \omega$ .  $[S]$  is called the  $m$ -current associated to  $S$ . Given an appropriate Hilbert structure  $W$  on the space of  $m$ -forms, the dual metric can be used to give a measure of the dissimilarity between submanifolds. When  $S$  is an oriented surface in  $\mathbb{R}^3$ , the dual norm

writes  $\| [S] \|_{W^*}^2 = \int_S \int_S \eta(x) * K_W(y, x) \eta(y) ds(y) ds(x)$ . The optimal match between two currents  $[S]$  and  $[T]$  is the diffeomorphism minimizing the functional  $J_{[S],[T]}(\varphi) = \gamma D^2(Id, \varphi) + \| [\varphi(S)] - [T] \|_{W^*}^2$ . In practice, faster convergence is obtained by using a multiscale approach in which the kernel  $K_W$  width is decreased a few number of times during the optimization scheme (each step provides an initial guess for the next one).

### 3 A template estimation for large database via LDDMM

#### 3.1 Why building a template ?

A central notion in computational anatomy is the generation of registration maps, mapping a large set of anatomical data to a common coordinate system to study intra-population variability and inter-population differences. In this paper, we are using the method introduced by Glaunes et al. [1] which estimates a template given a collection of unlabeled point sets and surfaces using a representation of points and surfaces as currents [7]. This method is posed as a minimum mean squared error estimation problem and uses a metric on the space of diffeomorphisms. Let  $S_i$  be  $N$  surfaces in  $\mathbb{R}^3$ . Let  $[S_i]$  be the corresponding current of  $S_i$ , or its approximation by a finite sum of vectorial Diracs. The minimum mean squared error problem for template estimation can be formulated, in terms of minimization with respect to the diffeomorphisms  $\varphi_i$  as:  $\{\hat{\varphi}_i\} = \operatorname{argmin} \sum_{i=1}^N \left\{ \left\| \frac{1}{N} \sum_{j=1}^N [\varphi_j(S_j)] - [\varphi_i(S_i)] \right\|_{W^*}^2 + D^2(Id, \varphi_i) \right\}$ . This method computes a template for the population described, in the  $W^*$  space, by a mean  $\frac{1}{N} \sum_{j=1}^N \varphi_j [S_j]$  of currents, which is a set of shapes from the population and represents a template for this population. Obtaining a template is useful, indeed it's possible to use statistical analysis of the deformations to characterize the subjects. We can run analysis on momentum vectors like PCA or estimate an approximation of pairwise distances between subjects using the estimated template and the method presented by Yang et al [8] to approximate these distances, in order to use methods like Isomap [9].

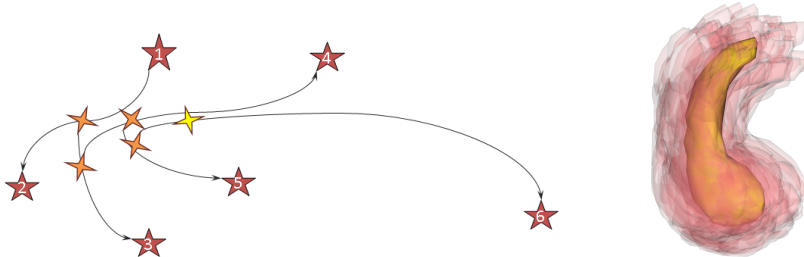
#### 3.2 Iterate Centroid for a template estimation

In order to reduce the time of the template estimation, we present here a way to give a better initialization for the template estimation method presented in [1].

**Method** The method shares some similarity with the method Riemannian 1-center approximation presented by Arnaudon et al. in 2013 [10], and is inspired of centroid computation in Euclidean space:  $\sum_{i=1}^n x_i/n = b_n$  satisfies  $b_1 = x_1$ ,  $b_{k+1} = k/(k+1) * b_k + 1/(k+1) * x_{k+1}$ . We have two possibles ways to run the method and we proceed as follow:

- $B_1 = S_1$
- For each  $i$  from 1 to  $N$ , with  $N$  the subject's number:

- $B_i$  is matched to  $S_{i+1}$  which results in a deformation map  $\phi_{v_i}(x, t)$ .
- To compute  $B_{i+1}$  we have two possible ways:
  1. set  $B_{i+1} = \phi_{v_i}(B_i, \frac{1}{i+1})$  which means we transport  $B_i$  along the geodesic and stop at time  $t = \frac{1}{i+1}$ .
  2.  $B_{i+1} = \frac{i}{i+1} * [\phi_{v_i}(B_i, \frac{1}{i+1})] + \frac{1}{i+1}[\phi_{u_i}(S_{i+1}, \frac{i}{i+1})]$ , where  $u_i(x, t) = -v_i(x, 1 - t)$ , i.e.  $\phi_{u_i}$  is the reverse flow map. At each iteration we add one mesh to  $B_{i+1}$ .
- At the end, we obtain a single mesh (way 1 we will discuss here), or a set of meshes (way 2) to represent the population’s centroid.



**Fig. 1.** Illustration of the method. Left image: red stars are population subjects, yellow star is the final Centroid, and orange stars are iterations of Centroid. Right image: Final Centroid with the hippocampus population from datas Data1 (red)

Of course the result of the Iterative Centroid (I.C.) does depend on the ordering of subjects. We will study this dependance in the experimental part. We will also study the effect on stopping the algorithm before it completes all iterations. Then we say that the I.C. is at  $x\%$  if iterations are going to step  $\frac{x}{100} * n$

## 4 Data

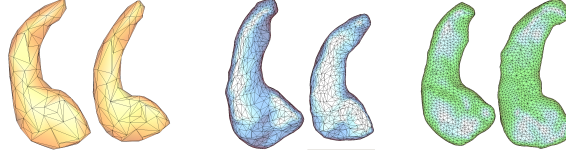
We evaluate the method on segmented 95 hippocampi on Magnetic Resonance Imaging from the European database IMAGEN with the software SACHA [11]. This database is constituted by young (14-16 years old) healthy subjects. This first database called RealData is composed of 95 meshes with around 2000 vertices (between 1716 and 2256) each one.

We also made two synthetic populations of hippocampi meshes, in order to have a larger and simpler dataset, having less noisy data than RealData.

Data1: We chose one subject  $S_0$  that we decimate (down to 135 vertices) and deform using geodesic shooting in random directions composed with small translations and rotations to build 500 different shapes. Data1 is composed of a population of 500 shapes with the same mesh (135 vertices per shape with the same connections between them), which provides a large database with simple meshes and mainly global deformations.

Data2: Using the mesh of  $S_0$ , but this time keeping 1001 vertices. We match this new mesh via LDDMM on each subject of the dataset RealData ( $n = 95$ ). Then

we have a new population with more local deformation than in Data1, which is closer to the anatomical truth constituted by the same mesh (so we have less noise than in RealData). Examples of subjects resulting from these dataset are seen on Fig. 2.



**Fig. 2.** Left to Right: two meshes from Data1, Data2 and RealData

## 5 Experiments Results

**Effect of subject ordering** For testing the method, we compared results in the space of currents, i.e. for comparing two surfaces  $S$  and  $T$ , we computed the norm  $\|[S] - [T]\|_{W^*}^2$ . For studying the influence of the ordering of subjects, we compared several centroids computed until the last iteration (Centroid at 100%) with different orderings. For each database 3 different centroids were computed. On each database, we computed 3 different I.C.  $C_1^z$ ,  $C_2^z$  and  $C_3^z$  with  $z \in \{data1, data2, realdata\}$ . We show, for each database, in Tab. 1 the distances between centroids, and distances between centroids and all the subjects of the database. For each database, means, variances, minimums and maximums are almost identical and the distances between centroids are very small compared to the population. In database Data2 and RealData, datasets are closer to cen-

**Table 1.** Comparison between  $C_i^z$  and the population, and distances between  $C_i^z$  and  $C_j^z$ , with  $i, j \in \{1, 3\}$  and  $z \in \{data1, data2, realdata\}$

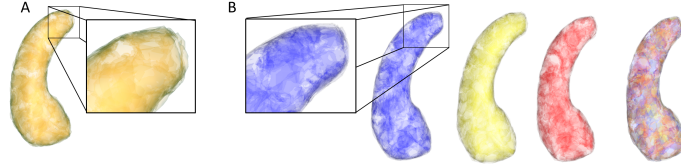
distance :	mean	std	max	min	to $C_1^{data1}$	to $C_2^{data1}$	to $C_3^{data1}$
$C_1^{data1}$	160.249	32.2907	250.2365	60.9317	0	0.5784	0.8294
$C_2^{data1}$	160.247	32.3882	250.2270	60.9540	0.5784	0	0.6417
$C_3^{data1}$	160.233	22.3800	250.2957	60.9423	0.8294	0.6417	0
distance :	mean	std	max	min	to $C_1^{data2}$	to $C_2^{data2}$	to $C_3^{data2}$
$C_1^{data2}$	55.2038	13.4862	103.7798	32.7979	0	1.0631	1.4788
$C_2^{data2}$	55.4303	13.4417	104.0364	32.9868	1.0631	0	1.4193
$C_3^{data2}$	55.3049	13.3439	103.5822	33.0183	1.4788	1.4193	0
distance :	mean	std	max	min	to $C_1^{realdata}$	to $C_2^{realdata}$	to $C_3^{realdata}$
$C_1^{realdata}$	57.7548	14.6558	108.5300	34.4756	0	3.6142	3.6658
$C_2^{realdata}$	57.7488	14.5384	108.1334	34.7786	3.6142	0	3.3797
$C_3^{realdata}$	57.7504	14.5733	108.2680	34.6085	3.6658	3.3797	0

troids, since in these populations there is less global diffeomorphic deformations between shapes. Data1 and Data2 are less noisy than RealData, so the centroids are closer to each others. In RealData, centroids have different meshes due to the different initializations.

**Effects of initialization and ordering on estimated template** We compute the distance in  $W^*$  between a template initialized with a standard initialization (the whole population) noted as  $T(t_0 = z)$  and a template initialized by a I.C. noted as  $T(t_0 = C_i^z)$ , and the distance between two templates initialized by two different Iterative Centroids. So we can note that the choice of the centroid does not influence the result of the template (see Tab. 2). For  $T(t_0 = z)$  distances are more important. As seen in Fig. 3A, the orange template converged, surfaces are close to each other but less than for templates from  $T(t_0 = C_i^z)$  (Fig. 3B).

**Table 2.** Comparison between  $T(t_0 = C_i^z)$  and  $T(t_0 = z)$ , and between  $T(t_0 = C_i^z)$  and  $T(t_0 = C_i^z)$ , with  $z \in \{data1, data2, realdata\}$

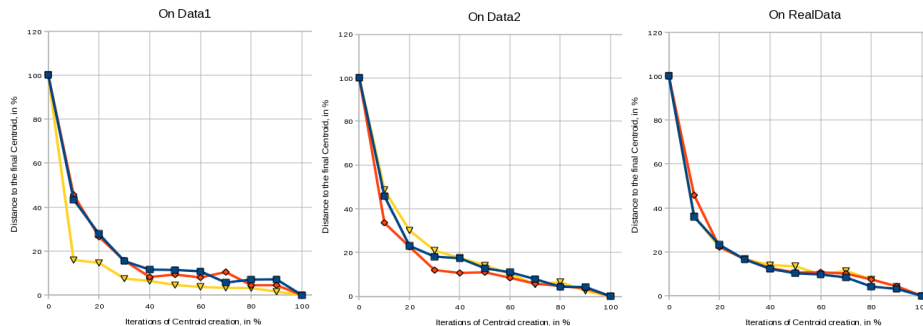
distance:	$T(t_0 = data1)$	$T(t_0 = C_1^{data1})$	$T(t_0 = C_2^{data1})$	$T(t_0 = C_3^{data1})$
$T(t_0 = C_1^{data1})$	41.0594	0	0.6669	1.1657
$T(t_0 = C_2^{data1})$	41.113	0.6669	0	1.1085
$T(t_0 = C_3^{data1})$	40.6269	1.1657	1.1085	0
distance:	$T(t_0 = data2)$	$T(t_0 = C_1^{data2})$	$T(t_0 = C_2^{data2})$	$T(t_0 = C_3^{data2})$
$T(t_0 = C_1^{data2})$	20.4625	0	0.5280	0.6660
$T(t_0 = C_2^{data2})$	20.1764	0.5280	0	0.8381
$T(t_0 = C_3^{data2})$	20.6301	0.6660	0.8381	0
distance:	$T(t_0 = realdata)$	$T(t_0 = C_1^{realdata})$	$T(t_0 = C_2^{realdata})$	$T(t_0 = C_3^{realdata})$
$T(t_0 = C_1^{realdata})$	27.4166	0	7.0284	6.2362
$T(t_0 = C_2^{realdata})$	26.6960	7.0284	0	1.8560
$T(t_0 = C_3^{realdata})$	26.5063	6.2362	1.8560	0



**Fig. 3.** A: the template  $T(t_0 = data2)$  with a zoom B: Blue:  $T(t_0 = C_1^{data2})$  with zoom, Yellow:  $T(t_0 = C_2^{data2})$  Red:  $T(t_0 = C_3^{data2})$ , and the last one is the superposition.

In order to check that the computed template is on the center of the population, we calculated the ratio of the norm of the means of initial momentum vectors from the template to each subjects by the mean of the norms of initial momentum vectors. For Data1, the ratio of  $T(t_0 = C_1^{data1}) = \mathbf{0.0062}$ , of  $T(t_0 = C_2^{data1}) = \mathbf{0.0056}$ , of  $T(t_0 = C_3^{data1}) = \mathbf{0.0059}$  and of  $T(t_0 = data1) = \mathbf{0.0212}$ . For Data2 we have, for the ratio,  $T(t_0 = data2) = \mathbf{0.0206}$ ,  $T(t_0 = C_1^{data2}) = \mathbf{0.0077}$ , for  $T(t_0 = C_2^{data2}) = \mathbf{0.0086}$ , and for  $T(t_0 = C_3^{data2})$  we obtain  $\mathbf{0.0060}$ . For RealData the ratio of  $T(t_0 = C_1^{realdata}) = \mathbf{0.0073}$ , of  $T(t_0 = C_2^{realdata}) = \mathbf{0.0060}$ , of  $T(t_0 = C_3^{realdata}) = \mathbf{0.0088}$  and of  $T(t_0 = realdata) = \mathbf{0.0094}$ . As an illustration, Fig. 3 shows the almost similarity of 3 templates of Data2 (95 surfaces per template) initialized by these 3 different centroids. We can observe that for all databases, the templates initialized by I.C. are more centered than templates with standard initialization. All templates are centered, a little more for  $T(t_0 = C_i^z)$ .

**Effect of the iteration number** We tested the influence of computing only a fraction of the Iterative Centroid, what brings each step of I.C. to the construction of the final centroid. Figure 4 shows that after processing 40% of the population, the I.C. covers more than 80% of the distance to the final centroid. We can ask if computing an I.C. at 40% could be enough to initialize a template estimation, but experiments show that it lowers significantly the quality of the resulting template, as well as it slows down convergence of the template estimation, as we will see next.



**Fig. 4.** Curves of distances for  $C_i^{data1}$ ,  $C_i^{data2}$  and  $C_i^{realdata}$ . The blue ( $C_1^z$ ) yellow ( $C_2^z$ ) and red ( $C_3^z$ ) curves show the distance between the I.C. at  $x\%$  and the final which correspond to 100%. each point represent 10%.

**Computation time** We present some results of computation time. We use a GPU implementation for the computation of kernel convolutions, which constitute the time-consuming part of such LDDMM algorithms. These computations were processed on a Nvidia Tesla C1060 card. On Data1, Computing a template with standard initialization takes 96 hours. Computing a template from I.C. takes in total 26.6 hours in average (1,8h for the I.C. and 24.8h for the template). Likewise, for Data2, computing an Iterative Centroid, and then use it as initialization for template estimation takes about  $(1.5 + 12) = 13.5$  hours against 20.6 hours without using I.C.. For RealData, computing an Iterative Centroid, and then use it as initialization for template estimation takes about  $(2.7 + 25.4) = 28.1$  hours against 99 hours without using I.C.. In average we save up 59.4% of computation time. 72.3% for Data1, 34.5% for Data2 and 71.6% for RealData.

Computing a  $T(t_0 = 40\%C_i^z)$  takes 23.6 hours for Data1, 10.0 hours for data2 and 40.2 for RealData. So in total, for a template  $T(t_0 = 40\%C_i^z)$  there is no saving time and results are not better or equivalent to  $T(t_0 = C_i^z)$ .

To increase the speed of the template estimation method, we can add at each end of iteration, a Matching Pursuit on currents as described in [12].

## 6 Discussion and conclusion

We proposed a new method for the initialization of template estimation algorithms. We showed that the Iterative Centroid method, regardless of the order-

ing of subjects, provides a centroid which is correctly centered in the population. Using this centroid as initialization for the template estimation method speeds up the convergence and can save up 72% of computation time.

This method can also be seen as a cheap alternative for template estimation, and could certainly be used to initialize others methods of template estimation, like the method presented by Durrleman in 2008 [6], which actually use an ellipsoid as initialization.

## Acknowledgments

The authors are grateful to Vincent Frouin, Jean-Baptiste Poline, Roberto Toro and Edouard Duschenay for providing a sample of subjects from the IMAGEN database and to Marie Chupin for use of the SACHA software.

## References

1. Glaunès, J., Joshi, S.: Template estimation from unlabeled point set data and surfaces for computational anatomy. In Pennec, X., Joshi, S., eds.: Proc. of the International Workshop on the Mathematical Foundations of Computational Anatomy (MFCA-2006). (1st of October 2006) 29–39
2. Grenander, U., Miller, M.I.: Computational anatomy: An emerging discipline. *Quarterly of applied mathematics* **56**(4) (1998) 617–694
3. Christensen, G.E., Rabbitt, R.D., Miller, M.I.: Deformable templates using large deformation kinematics. *Image Processing, IEEE Transactions on* **5**(10) (1996) 1435–1447
4. Beg, M.F., Miller, M.I., Trounev, A., Younes, L.: Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *International Journal of Computer Vision* **61**(2) (2005) 139–157
5. Ma, J., Miller, M.I., Trounev, A., Younes, L.: Bayesian template estimation in computational anatomy. *NeuroImage* **42**(1) (2008) 252–261
6. Durrleman, S., Pennec, X., Trounev, A., Ayache, N., et al.: A forward model to build unbiased atlases from curves and surfaces. In: 2nd Medical Image Computing and Computer Assisted Intervention Workshop on Mathematical Foundations of Computational Anatomy. (2008) 68–79
7. Vaillant, M., Glaunès, J.: Surface matching via currents. In: *Information Processing in Medical Imaging*, Springer (2005) 381–392
8. Yang, X.F., Goh, A., Qiu, A.: Approximations of the diffeomorphic metric and their applications in shape learning. In: *Information Processing in Medical Imaging: IPMI*. (2011) 257–270
9. Tenenbaum, J., Silva, V., Langford, J.: A global geometric framework for nonlinear dimensionality reduction. *Science* **290**(5500) (December 2000) 2319–2323
10. Arnaudon, M., Nielsen, F.: On approximating the riemannian 1-center. *Computational Geometry* **46**(1) (2013) 93–104
11. Chupin, M., Hammers, A., Liu, R.S.N., Colliot, O., Burdett, J., Bardinet, E., Duncan, J.S., Garnero, L., Lemieux, L.: Automatic segmentation of the hippocampus and the amygdala driven by hybrid constraints: Method and validation. *NeuroImage* **46**(3) (2009) 749–761
12. Durrleman, S., Pennec, X., Trounev, A., Ayache, N.: Statistical models of sets of curves and surfaces based on currents. *Medical Image Analysis* **13**(5) (2009) 793–808