

# Gaussian diffeons for surface and image matching within a Lagrangian framework

LAURENT YOUNES

Lagrangian particle formulations of the large deformation diffeomorphic metric mapping algorithm (LDDMM) can be numerically challenging when the number of particles is large. In this paper, we introduce and discuss numerical schemes that can be used for surface and image matching and are based on representing the Eulerian velocity over a finite-dimensional basis that deforms over time. The method is described under the optimal control formalism, and optimality conditions are derived, together with the equations that are needed to implement gradient-based methods. Experimental results are shown, both with surfaces and images.\*

AMS 2000 SUBJECT CLASSIFICATIONS: 58D05, 49N90, 49Q10, 68E10.

KEYWORDS AND PHRASES: Shape analysis, optimal control, groups of diffeomorphisms, Subriemannian geometry.

## 1. Introduction

We introduce in this paper a new Galerkin-like approach for the numerical solution of the large-deformation diffeomorphic metric mapping (LDDMM) algorithm, focusing our discussion on the case of surfaces and images. The LDDMM algorithm, which has been introduced in [34] for point sets and in [44, 7] for images, has become widely used in the medical imaging literature and other applications [19, 45, 43, 4, 9, 46, 66, 13, 61, 48, 18, 51, 57, 1, 10, 62, 5, 49, 8, 47, 24, 56, 3, 23, 16].

LDDMM addresses the infinite-dimensional optimization problem

$$(1) \quad \min_{v \in L^2([0,1], V)} \int_0^1 \|v(t)\|_V^2 dt + \tilde{F}(\phi^v(1)),$$

in which  $V$  is a reproducing kernel Hilbert space (RKHS) of continuously differentiable vector fields on  $\mathbb{R}^d$  and  $\phi^v$  is the diffeomorphic flow associated

---

\*This work was partially supported by the ONR award N000140810606.

to  $v$ , such that

$$(2) \quad \phi^v(t, x) - x = \int_0^t v(s, \phi^v(s, x)) ds, \quad t \in [0, 1].$$

The cost function  $\tilde{F}$  represents a soft end-point condition on the diffeomorphism at time  $t = 1$ . For image matching, for example, one can take

$$(3) \quad \tilde{F}(\phi) = \|I_0 \circ \phi^{-1} - I_1\|_{L^2}^2$$

in order to register a target image  $I_1$  on a template  $I_0$ .

Such a formulation can be interpreted as an optimal control problem [13, 12, 52, 47] in which the diffeomorphism  $\phi$  is the state variable,  $v$  is the control and the flow equation (2) describes the state evolution. The optimal control framework allows for a simple analysis of the problem, straightforward reductions to smaller dimensional state spaces, easy generalizations. It moreover directly transcribes to discrete-time numerical algorithms.

Reduction occurs when diffeomorphisms are considered only through their actions on simpler objects, like shapes or images. This is the case, for example, when one uses a cost function like (3), which is right-invariant to any diffeomorphism  $\psi$  that leaves  $I_0$  invariant, i.e.,  $I_0 \circ \psi^{-1} = I_0$  implies  $\tilde{F}(\phi \circ \psi) = \tilde{F}(\phi)$ . In a more general setting, consider a shape space (say  $\mathcal{M}$ ) subject to the action of diffeomorphisms, which will be denoted  $(\phi, \mathbf{q}) \in \text{Diff} \times \mathcal{M} \rightarrow \phi \cdot \mathbf{q}$ , where  $\text{Diff}$  is a group of diffeomorphisms. Assume that the function  $\phi \mapsto \tilde{F}(\phi)$  depends only on the action of  $\phi$  on some element  $\mathbf{q}_0 \in \mathcal{M}$  so that  $\tilde{F}(\phi) = F(\phi \cdot \mathbf{q}_0)$ , with  $F : \mathcal{M} \rightarrow \mathbb{R}$ . In such cases, the original problem can most of the time be reduced to a simpler one, the most important case in practice being when the shape space is finite dimensional, because this includes many numerical implementations of landmark, curves and surface matching, and, in some situations, image matching. Assuming that each shape  $\mathbf{q}$  is discretized into a finite set of points  $\mathbf{q} = (x_1, \dots, x_N)$  with  $x_i \in \mathbb{R}^d$ , the reduced optimal control problem takes the form

$$(4) \quad \min_{\alpha} \sum_{k,l=1}^N \int_0^1 \alpha_k(t)^T K(x_k(t), x_l(t)) \alpha_l(t) dt + F(\mathbf{q}(1))$$

subject to  $\mathbf{q}(0) = \mathbf{q}_0$  and

$$(5) \quad \partial_t x_k = \sum_{l=1}^N K(x_k, x_l) \alpha_l.$$

Here  $K$  is the reproducing kernel of the RKHS  $V$ . It can be, in full generality, matrix valued [62], although, in most applications, including those discussed in this paper, one uses a scalar kernel, in which case,  $\alpha_k^T K(x_k, x_l) \alpha_l = K(x_k, x_l) \alpha_k^T \alpha_l$ . The control is  $\alpha = (\alpha_1, \dots, \alpha_N)$ , a time-dependent collection of  $d$ -dimensional vectors, and the state is  $\mathbf{q} = (x_1, \dots, x_N)$ .

If  $\mu$  is a measure on  $\mathbb{R}^d$ , and  $z : \text{Supp}(\mu) \rightarrow \mathbb{R}^d$  is  $\mu$ -measurable, (where  $\text{Supp}(\mu)$  is the support of  $\mu$ ), we will denote by  $z d\mu$  the linear form that associates to a vector field  $w : \mathbb{R}^d \rightarrow \mathbb{R}^d$  the number

$$(z d\mu | w) = \int_{\mathbb{R}^d} z(x)^T w(x) d\mu(x).$$

In (4) and (5), it will be convenient (and consistent with the underlying theory) to introduce the time-dependent vector measures

$$\pi = \sum_{k=1}^N \alpha_k d\delta_{x_k},$$

( $\delta_x$  being the Dirac measure at  $x$ ). With this notation, we can write

$$\sum_{l=1}^N K(x_k, x_l) \alpha_l = K \pi(x_k)$$

and, using properties of reproducing kernels,

$$\sum_{k,l=1}^N \alpha_k^T K(x_k, x_l) \alpha_l = (\pi | K \pi) = \|K \pi\|_V^2$$

with, for a vector measure  $z d\mu$ ,

$$K(z d\mu)(\cdot) = \int_{\mathbb{R}^d} K(\cdot, x) z(x) d\mu(x).$$

While the reduction from the initial infinite-dimensional problem is appreciable, the numerical cost associated to solving this problem can be significant when the number of points,  $N$ , is large. The state evolution equation has a complexity of  $N^2$  per time step, which can become crippling and impractical as soon as  $N$  reaches the few thousands.

In [63], we have introduced a general scheme in order to approximate this problem with a smaller dimensional one. Start with the simple idea

that consist in grouping variables  $\alpha_k$  in small clusters, and assuming that variables in the same cluster share the same value. This corresponds to partitioning  $\{1, \dots, N\}$  into subsets  $A_1, \dots, A_M$  (with  $M \ll N$ ) and replacing  $\pi$  by the approximation

$$\bar{\pi} = \sum_{k=1}^M \alpha_k d\nu_{A_k}$$

with  $\nu_{A_k} = \sum_{j \in A_k} \delta_{x_k}$ . This reduces the number of variables from  $N$  to  $M$ , but does not improve the computational complexity yet, since each step in (5) would still require  $O(N^2)$  computations. For this, we make a second approximation that replaces the sum of Dirac measures,  $\nu_{A_k}$ , by a single Gaussian measure with center  $c_k$  and covariance matrix  $S_k$ , that we denote, here and in the rest of the paper, by  $\mu_{c_k, S_k}$ . If  $\alpha \in \mathbb{R}^d$ ,  $c \in \mathbb{R}^d$  and  $S$  is a  $d$  by  $d$  matrix, the vector measure  $\alpha d\mu_{c, S}$  will be referred to as a Gaussian vector measure, or GVM. Recall that a Gaussian measure on  $\mathbb{R}^d$  is characterized by the expression of its Fourier transform

$$\hat{\mu}_{c, S}(u) = e^{-\iota u^T c - \frac{1}{2} u^T S u}$$

where  $\iota = \sqrt{-1}$ . If  $S$  is invertible,  $\mu_{(c, S)}$  has the usual density with respect to the Lebesgue measure

$$\frac{d\mu_{c, S}}{dx} = \frac{1}{(2\pi)^{d/2} \sqrt{\det S}} e^{-\frac{1}{2} (x-c)^T S^{-1} (x-c)}.$$

One of the main interests in using GVM's is that their convolutions with Gaussian kernels remain Gaussian with very simple transformation rules. More precisely, if

$$(6) \quad K(x, y) = \exp(-|x - y|^2 / 2\sigma^2)$$

then, for  $\alpha \in \mathbb{R}^d$ ,

$$(7) \quad K(\alpha d\mu_{c, S})(x) = \frac{\sigma^d \alpha}{\sqrt{\det(\sigma^2 \text{Id} + S)}} \exp\left(-\frac{1}{2} (x - c)^T (\sigma^2 \text{Id} + S)^{-1} (x - c)\right).$$

Moreover, there is a natural action of diffeomorphisms on Gaussian measures, namely

$$(8) \quad \phi \cdot \mu_{c, S} = \mu_{\phi(c, S)}$$

with, denoting  $A^T$  the transpose of a matrix  $A$ :

$$\phi \cdot (c, S) = (\phi(c), D\phi(c)SD\phi(c)^T),$$

which may be interpreted as a first order approximation of the push-forward of  $\mu_{c,S}$  by  $\phi$ . Taking

$$\hat{\pi} = \sum_{k=1}^M \alpha_k \mu_{c_k, S_k},$$

we have all required ingredients in order to formulate a new, smaller-dimensional, optimal control problem, over point sets, and more generally over diffeomorphisms.

## 2. Optimal control with Gaussian diffeons

### 2.1. Formulation

*Gaussian diffeons* [63] are defined by

$$\begin{aligned} \gamma_i(\cdot; c, S) &= f(\cdot; c, S)e_i \\ &= \sigma^{-d} \sqrt{\det(\sigma^2 \text{Id} + S)} K(e_i d\mu_{c,S}) \end{aligned}$$

where  $e_i$  is the  $i$ th element of the canonical basis of  $\mathbb{R}^d$  (this explicitly defines the function  $f$  via (7)). Using a finite number of such functions to constrain the vector field  $v$  in equations (1) and (2), one defines a subriemannian optimal control system

$$\begin{cases} \partial_t \phi(t, x) = v(t, \phi(t, x)) \\ \partial_t c_k = v(t, c_k), \quad k = 1, \dots, M \\ \partial_t S_k = Dv(c_k)S_k + S_k Dv(c_k)^T, \quad k = 1, \dots, M \\ v(t, y) = \sum_{k=1}^M \sum_{i=1}^d \gamma_i(y; c_k, S_k) \alpha_{ki} = \sum_{k=1}^M f(y; c_k, S_k) \alpha_k \end{cases}$$

The evolution of  $(c_k, S_k)$  is the infinitesimal version of the action  $\phi \cdot (c, S)$  that was defined in equation (8). The coefficients  $\alpha = (\alpha_1, \dots, \alpha_M)$  can be interpreted as controls, to be optimized, given fixed initial conditions for  $\mathbf{c} = (c_1, \dots, c_M)$  and  $\mathbf{S} = (S_1, \dots, S_M)$ , in order to minimize the objective function in (1).

As suggested above, objects of interest (shapes or images) are advected by the diffeomorphic flow, and, as soon as the endpoint cost only depends on the deformed object, the optimal control problem can focus on object evolution only. Denote these objects by  $\mathbf{q}$ , with an interpretation that depends on the considered application (as discussed in the examples below). In all these examples, the time derivative  $\partial_t \mathbf{q}$  will depend linearly on  $v$ , and therefore take the form

$$\partial_t \mathbf{q} = A(\mathbf{q}, v) = \sum_{k=1}^M \zeta(\mathbf{q}, c_k, S_k) \alpha_k,$$

with  $\zeta(\mathbf{q}, c, S) \alpha = A(\mathbf{q}, f(\cdot; c, S) \alpha)$  for  $\alpha \in \mathbb{R}^d$ . (If  $\mathbf{q}$  is finite dimensional,  $\zeta$  takes values over  $\dim(\mathbf{q}) \times d$  matrices.)

Focusing on shape evolution, the optimal control problem that we solve minimizes

$$(9) \quad \int_0^1 \|v\|_V^2 dt + F(\mathbf{q}(1), \mathbf{c}(1), \mathbf{S}(1))$$

subject to constraints

$$(10) \quad \begin{cases} \partial_t \mathbf{q} = A(\mathbf{q}, v) \\ \partial_t c_k = v(t, c_k), \quad k = 1, \dots, M \\ \partial_t S_k = Dv(c_k) S_k + S_k Dv(c_k)^T, \quad k = 1, \dots, M \\ v(t, y) = \sum_{k=1}^M f(y; c_k, S_k) a_k. \end{cases}$$

Here  $F$  is an end-point cost that we allow to depend on variables  $c$  and  $S$ , in addition to the shape variable  $\mathbf{q}$ . Using (7) (see [63] for details), the norm of  $v$  can be rewritten as

$$\|v\|_V^2 = \sum_{k,l=1}^M g(c_k, S_k; c_l, S_l) \alpha_k^T \alpha_l$$

with

$$(11) \quad g(c, S; c', S') = \frac{\sqrt{\det(\sigma^2 \text{Id} + S) \det(\sigma^2 \text{Id} + S')}}{\sigma^d \sqrt{\det(\sigma^2 \text{Id} + S + S')}} \times \exp \left( -\frac{1}{2} (c' - c)^T (\sigma^2 \text{Id} + S + S')^{-1} (c' - c) \right).$$

Writing

$$Dv = \sum_{l=1}^M \alpha_l \nabla_1 f(\cdot; c_l, S_l)^T$$

(where  $\nabla_1$  refers to the gradient with respect to the first variable) and

$$Dv(c)S + SDv(c)^T = \sum_{l=1}^M (\alpha_l \nabla_1 f(c; c_l, S_l)^T S + S \nabla_1 f(c; c_l, S_l) \alpha_l^T),$$

we can rewrite the previous problem as minimizing

$$\sum_{k,l=1}^M \int_0^1 g(c_k, S_k; c_l, S_l) \alpha_k^T \alpha_l dt + F(\mathbf{q}(1), \mathbf{c}(1), \mathbf{S}(1))$$

subject to

$$\begin{cases} \partial_t \mathbf{q} = \sum_{l=1}^M \zeta(\mathbf{q}, c_l, S_l) \alpha_l \\ \partial_t c_k = \sum_{l=1}^M f(c_k; c_l, S_l) \alpha_l, \quad k = 1, \dots, M \\ \partial_t S_k = \sum_{l=1}^M (\alpha_l \nabla_1 f(c_k; c_l, S_l)^T S_k + S_k \nabla_1 f(c_k; c_l, S_l) \alpha_l^T), \quad k = 1, \dots, M \end{cases}$$

## 2.2. Maximum principle and gradient

We introduce some notation that will be helpful in writing the formulae that will come. Let  $f_{kl} = f(c_k; c_l, S_l)$ ,  $A_l = (\sigma^2 \text{Id} + S_l)^{-1}$ ,  $b_{kl} = A_l(c_k - c_l)$ ,  $g_{kl} = g(c_k, S_k; c_l, S_l)$ ,  $B_{kl} = (\sigma^2 \text{Id} + S_k + S_l)^{-1}$ ,  $\tilde{b}_{kl} = B_{kl}(c_k - c_l)$ .

We will also need the following variations with respect to  $\mathbf{S}$  and  $\mathbf{c}$ , that we write formally as

$$\begin{aligned} \delta_{\mathbf{c}} f_{kl} &= -(\delta c_k - \delta c_l)^T (f_{kl} b_{kl}), \\ \delta_{\mathbf{c}} b_{kl} &= A_l (\delta c_k - \delta c_l)^T, \\ \delta_{\mathbf{c}} g_{kl} &= -(\delta c_k - \delta c_l)^T (g_{kl} \tilde{b}_{kl}), \\ \delta_{\mathbf{S}} f_{kl} &= \frac{1}{2} \delta S_l \odot (f_{kl} b_{kl} b_{kl}^T), \\ \delta_{\mathbf{S}} (u^T b_{kl}) &= -\frac{1}{2} \delta S_l \odot (A_l u b_{kl}^T + b_{kl} (A_l u)^T), \quad u \in \mathbb{R}^d, \end{aligned}$$

$$\begin{aligned} \delta_S g_{kl} = & \frac{1}{2} \left( \delta S_k \odot (g_{kl}(\tilde{b}_{kl} \tilde{b}_{kl}^T - B_{kl} + A_k)) \right. \\ & \left. + \delta S_l \odot (g_{kl}(\tilde{b}_{kl} \tilde{b}_{kl}^T - B_{kl} + A_l)) \right), \end{aligned}$$

in which we have denoted  $A \odot B = \text{trace}(A^T B)$ . The derivatives in  $S$  are symmetrized, using  $\text{trace}(A^T S) = \text{trace}((A + A^T)S/2)$  if  $S$  is symmetric.

To apply Pontryagin's Maximum Principle [58], introduce co-state variables  $\mathbf{p}_q$ ,  $(p_{c,1}, \dots, p_{c,M})$ ,  $(p_{S,1}, \dots, p_{S,M})$  with control-dependent Hamiltonian:

$$\begin{aligned} H_\alpha(\mathbf{p}, \mathbf{q}) = & \sum_{l=1}^M \mathbf{p}_q^T \zeta(\mathbf{q}, c_l, S_l) a_l + \sum_{k,l=1}^M f_{kl}(p_{c,k}^T \alpha_l) \\ & - 2 \sum_{k,l=1}^M f_{kl}(\alpha_l^T p_{S,k} S_k b_{kl}) - \sum_{k,l=1}^M g_{kl}(\alpha_k^T \alpha_l). \end{aligned}$$

Here,  $\mathbf{p}_q$  has the same dimension as  $\mathbf{q}$ , each  $p_{c,i}$  is a  $d$ -dimensional vector and each  $p_{S,i}$  a  $d$  by  $d$  symmetric matrix.

The state evolution equations are

$$(12) \quad \begin{cases} \partial_t \mathbf{q} = \partial_{\mathbf{p}_q} H_\alpha = \sum_{l=1}^M \zeta(\mathbf{q}, c_l, S_l) a_l \\ \partial_t c_k = \partial_{p_{c,k}} H_\alpha = \sum_{l=1}^M f_{kl} \alpha_l \\ \partial_t S_k = \partial_{p_{S,k}} H_\alpha = - \sum_{l=1}^M f_{kl} (\alpha_l b_{kl}^T S_k + S_k b_{kl} \alpha_l^T) \end{cases}$$

Let the co-states obey the following equations.

$$(13) \quad \partial_t \mathbf{p}_q = -\partial_{\beta_k} H_\alpha = - \sum_{l=1}^M \partial_q (\mathbf{p}_q^T \zeta(\mathbf{q}, c_l, S_l) \alpha_l),$$

$$\begin{aligned} (14) \quad \partial_t p_{c,k} = & -\partial_{c_k} H_\alpha = -\partial_{c_k} (\mathbf{p}_q^T \zeta(\mathbf{q}, c_k, S_k) \alpha_k) \\ & + \sum_{l=1}^M f_{kl}(p_{c,k}^T \alpha_l) b_{kl} - \sum_{l=1}^M f_{lk}(p_{c,l}^T \alpha_k) b_{lk} \end{aligned}$$



$$\begin{aligned}
(15) \quad & -2 \sum_{l=1}^M f_{kl}(b_{kl}b_{kl}^T - A_l)S_k p_{S,k} \alpha_l \\
& + 2 \sum_{l=1}^M f_{lk}(b_{lk}b_{lk}^T - A_k)S_l p_{S,l} \alpha_k \\
& - 2 \sum_{l=1}^M g_{kl}(\alpha_k^T \alpha_l) \tilde{b}_{kl},
\end{aligned}$$

$$\begin{aligned}
(16) \quad \partial_t p_{S,k} = & -\partial_{S_k} H_{\alpha} = -\partial_{S_k} (\mathbf{p}_q^T \zeta(\mathbf{q}, c_k, S_k) a_k) \\
& - \frac{1}{2} \sum_{l=1}^M f_{lk}(p_{c,l}^T \alpha_k) b_{lk} b_{lk}^T + \sum_{l=1}^M f_{kl}(p_{S,k} \alpha_l b_{kl}^T + b_{kl} \alpha_l^T p_{S,k}) \\
& + \sum_{l=1}^M f_{lk}((\alpha_k^T p_{S,l} S_l b_{lk}) b_{lk} b_{lk}^T - A_k S_l p_{S,l} \alpha_k b_{lk}^T - b_{lk} \alpha_k^T p_{S,l} S_l A_k) \\
& + \sum_{l=1}^M g_{kl}(\alpha_k^T \alpha_l) (\tilde{b}_{kl} \tilde{b}_{kl}^T - B_{kl} + A_k).
\end{aligned}$$

The Maximum Principle states that an optimal control  $\alpha^*$  must satisfy

$$\alpha^*(t) = \operatorname{argmax}_{\alpha} H_{\alpha}(\mathbf{p}, \mathbf{q})$$

where  $\mathbf{p}, \mathbf{q}$  satisfy equations (12) to (16).

These equations, via the adjoint method, also provide a framework for the computation of the gradient of the objective function with respect to the control. Starting with current values of  $\alpha$ , the algorithm first integrates the state equations (12) forward to compute  $\mathbf{q}, \mathbf{c}$ , and  $\mathbf{S}$ , then sets

$$\mathbf{p}_q(1) = -\nabla_q F(\mathbf{q}(1), \mathbf{c}(1), \mathbf{S}(1)),$$

$$\mathbf{p}_c(1) = -\nabla_c F(\mathbf{q}(1), \mathbf{c}(1), \mathbf{S}(1))$$

and  $\mathbf{p}_S(1) = -\nabla_S F(\mathbf{q}(1), \mathbf{c}(1), \mathbf{S}(1))$  to integrate the co-state equations backward in time. The resulting gradient is then given by

$$(17) \quad \partial_{\alpha_k} E(t) = 2 \sum_{l=1}^M g_{kl} \alpha_l - \zeta(\mathbf{q}, c_k, S_k)^T \mathbf{p}_q - \sum_{l=1}^M f_{lk} p_{c,l} + 2 \sum_{l=1}^M f_{lk} p_{S,l} S_l b_{lk}.$$

### 3. Applications

#### 3.1. Point-set problems

We now consider specific applications, for which we complete the previous state and co-state evolution equations. The first and simplest example is when the shape variable is a collection of 3D points that move according to the diffeomorphic flow, so that  $\mathbf{q} = (x_1, \dots, x_N)$ ,  $x_j \in \mathbb{R}^d$  with  $\partial_t x_j = v(t, x_j)$  and  $\zeta(\mathbf{q}, c, S)$  is the vector with components

$$\zeta_j(\mathbf{q}, c, S) = f(x_j, c, S), \quad j = 1, \dots, N.$$

Denote  $\tilde{f}_{jl} = f(x_j, c_l, S_l)$  and  $\tilde{b}_{jl} = S_l^{-1}(x_j - c_l)$ . The first equation in (12) is

$$\partial_t x_j = \sum_{l=1}^M \tilde{f}_{jl} \alpha_l.$$

Equation (13) becomes

$$\partial_t \mathbf{p}_{q,j} = \sum_{l=1}^M \tilde{f}_{jl} (p_{q,j}^T \alpha_l) \tilde{b}_{jl}.$$

The first sum in (14) is

$$-\partial_{c_k} (p_q^T \zeta(\mathbf{q}, c_k, S_k) \alpha_k) = \sum_{j=1}^N \tilde{f}_{jl} (p_{q,j}^T \alpha_l) \tilde{b}_{jl}$$

and the first one in (16) is

$$-\partial_{S_k} (p_q^T \zeta(\mathbf{q}, c_k, S_k) \alpha_k) = -\frac{1}{2} \sum_{j=1}^N \tilde{f}_{jk} (p_{q,j}^T \alpha_k) \tilde{b}_{jk} \tilde{b}_{jk}^T.$$

Finally, the term  $\zeta(\mathbf{q}, c_k, S_k)^T \mathbf{p}_q$  in (17) is

$$\zeta(\mathbf{q}, c_k, S_k)^T \mathbf{p}_q = \sum_{j=1}^M \tilde{f}_{jk} p_{q,j}.$$

It is worth noticing, in this context, that one does not need to assume that points  $(x_1, \dots, x_N)$  in the shape representation are distinct from the

diffeon centers  $(c_1, \dots, c_M)$ . The two sets can overlap, or even coincide. It is easy to check, indeed, that if  $c_k = x_j$  at time  $t = 0$ , this equality is conserved at all times. The costate  $p_{c,k}$  that would be obtained after merging these two variables is simply the sum of the costates from the original formulation. In particular, the basic setting in equations (4) and (5) corresponds to the special case  $c \equiv x$  and  $S \equiv 0$ .

### 3.2. Surface matching

**3.2.1. Representation of triangulated surfaces** We now consider the situation in which the shapes that are compared are triangulated surfaces, for which we use the representation  $M = (\mathbf{q}, T)$  where  $\mathbf{q} = (x_1, \dots, x_N)$  is the set of vertices, and  $T$  is a set of oriented faces, where each face  $f$  is a consistently ordered triple  $f = (j_1, j_2, j_3)$  of integers in  $\{1, \dots, N\}$ . Diffeomorphisms act on such surfaces via  $\phi \cdot M = (\phi(\mathbf{q}), T)$ , so that the triangle structure remains unchanged and only the points are displaced.

Like in [53], we represent surfaces as geometric currents, or, equivalently, as vector measures. Starting with the continuous case, the vector measure  $\Gamma_M$  associated to an oriented surface  $M$  is equal to  $n_M d\sigma_M$  where  $\sigma_M$  is the area form on  $M$  and  $n_M$  is the unit normal. If  $M = (\mathbf{q}, T)$  is a triangulated surface, the associated vector measure is discretized as (see [53])

$$\Gamma_{(\mathbf{q}, T)} = \sum_{f \in T} n_f(\mathbf{q}) \delta_{c_f(\mathbf{q})}$$

where  $n_f(\mathbf{q}) = (x_{j_2} - x_{j_1}) \times (x_{j_3} - x_{j_1}) / 2$  is the area-weighted oriented normal and  $c_f(\mathbf{q}) = (x_{j_1} + x_{j_2} + x_{j_3}) / 3$  is the center of the face  $f = (j_1, j_2, j_3) \in T$ .

To compare such measures, the following norm can be used. Given a positive kernel  $\xi$ , define

$$\|z d\mu\|_\xi^2 = \int \int \xi(x, y) z(x)^T z(y) d\mu(x) d\mu(y) = (z d\mu \mid \xi(z d\mu))$$

which provides a norm on vector measures.

Given two surfaces  $M' = (\mathbf{q}', T')$  and  $M'' = (\mathbf{q}'', T'')$ , define the end-point cost function

$$(18) \quad F(\mathbf{q}) = \|\Gamma_{(\mathbf{q}, T')} - \Gamma_{(\mathbf{q}'', T'')}\|_\xi^2$$

measuring the proximity between the two triangulated surfaces. Using this cost function in (9)–(10) provides a surface-matching problem, which is a

special case of the framework considered in the previous section, with point sets. The LDDMM surface matching method that was introduced in [53] now corresponds to the further specialized case in which one assumes that  $c = x$  and  $S = 0$ . The gradient of  $F$  with respect to  $\mathbf{q}$  is computed in [53, 62] and we do not repeat it here.

**3.2.2. Computing initial diffeons** To complete the practical description of the surface matching method, it remains to specify how the initial ( $t = 0$ ) discretization  $(c_1, S_1), \dots, (c_M, S_M)$  is computed. Based on the interpretation given in the introduction, this discretization should group together points that are likely to share similar values of  $\alpha$  in (4) and (5).

This suggests that the centers,  $\mathbf{c}$ , should be chosen as centroids after clustering the vertices of the initial surface, say  $M_0$ . The topology used to perform this clustering task must, however, be chosen properly. When doing surface matching as described above, with a smooth kernel  $\xi$ , one can show that the optimal  $\alpha$  varies smoothly over the deforming surface [62], but that there can be sharp variations at points that are nearby in space but far away on the surface. This indicates that the clustering operation should adjust to the surface topology rather than to the Euclidean one. To avoid the computational cost associated to the evaluation of geodesic distances on the original surface, we use a spectral embedding approach and represent each point on the surface by the values of the first  $p$  eigenvectors of the Laplace-Beltrami operator (with  $p \gg 3$ ) and use standard clustering approaches, like K-means, in  $\mathbb{R}^p$ , to partition the original set of points into a small number of subsets. We can then associate to each such subset a centroid defined as the point in the original set that minimizes the sum of square distances (in  $\mathbb{R}^p$ ) to the points in the subset. Returning to the original surface, this provides us with centers  $(c_1, \dots, c_M)$  and regions in the triangulated surface that are associated to them. We then compute each matrix  $S_k$ , for  $k = 1, \dots, M$ , as the moment of inertia of the points in the associated region (each point weighted by the area of its own Voronoï region in the original triangulation).

We can use the  $c$ 's and  $S$ 's that are obtained this way directly to define the diffeon basis. We can also enhance these values by using them to initialize an additional optimization problem for the objective function

$$(19) \quad U(\mathbf{c}, \mathbf{S}) = \min_{\beta} \left\| \sum_{k=1}^M \beta_k d\mu_{c_k, S_k} - \Gamma_{(\mathbf{q}, T)} \right\|_{\xi}^2.$$

Letting  $\Theta = \sum_{k=1}^M \beta_k d\mu_{c_k, S_k}$ , and assuming that  $\xi$  is a Gaussian kernel with width  $\tau$ , we can as before use explicit expressions for Gaussian convo-

lutions to obtain a closed-form expression for (19). Note that one can write  $\Gamma_{(\mathbf{q},t)}$  as a sum of GVM's, namely

$$\Gamma_{(\mathbf{q},T)} = \sum_{j=1}^N n_j d\mu_{x_j,0}$$

since a zero variance Gaussian is a Dirac at the mean. Therefore, the only information we need is in the following formula,

$$\langle \beta d\mu_{c,S}, \beta' d\mu'_{c',S'} \rangle_{\xi} = h(c, S; c' S') \beta^T \beta'$$

with

$$(20) \quad h(c, S; c', S') = \frac{\tau^d \exp \left( -\frac{1}{2} (c' - c)^T (\tau^2 \text{Id} + S + S')^{-1} (c' - c) \right)}{\sqrt{\det(\text{Id} + S + S')}}.$$

(See (11). The factors  $\tau^{-d} \sqrt{\det(\sigma^2 \text{Id} + S)}$  and  $\tau^{-d} \sqrt{\det(\sigma^2 \text{Id} + S')}$ , which appeared in (11) are absent in (20) because they were included in the definition of the vector field bases, but not in the approximation of vector measures.)

We therefore have

$$(21) \quad \begin{aligned} \|\Theta - \Gamma_{(\mathbf{q},T)}\|_{\xi}^2 &= \sum_{k,l=1}^M h(c_k, S_k; c_l, S_l) \beta_k^T \beta_l \\ &\quad - 2 \sum_{k=1}^M \sum_{j=1}^N h(c_k, S_k, x_j, 0) \beta_k^T n_j + \|\Gamma_{(\mathbf{q},T)}\|_{\xi}^2 \end{aligned}$$

(the last term being constant in the optimization process). The derivatives of this expression with respect to  $\mathbf{c}$  and  $\mathbf{S}$  can be obtained similarly to our previous derivation (see also the end of section 3.2.3). This approach, using Gaussian measures instead of Diracs, can be seen as a generalization of some of the methods that were introduced in [51, 16].

**3.2.3. Representing surfaces with Gaussian vector measures** The discussion in section 3.2.2 provides a method to approximate a surface vector measure  $\Gamma_{(\mathbf{q},T)}$  with a sum of GVM's of the form

$$\Theta_{(\beta, \tilde{\mathbf{c}}, \tilde{\mathbf{S}})} = \sum_{k=1}^N \beta_k d\mu_{\tilde{\mathbf{c}}_k, \tilde{\mathbf{S}}_k}.$$

Given this, replacing the template surface by this low-dimensional approximation is a natural idea (suggested by Joan Glaunès [17]). This requires using new shape variables  $\mathbf{q} = (\boldsymbol{\beta}, \tilde{\mathbf{c}}, \tilde{\mathbf{S}})$  instead of the vertices  $x$ . To derive the evolution equations that should drive such shape variables, we start with a few remarks on how diffeomorphic action transforms surface vector measures.

The action of a diffeomorphism  $\phi$  on a surface  $m$ ,  $\phi \cdot m = \phi(m)$ , induces an action on its vector measure via

$$\begin{aligned} (\phi \cdot \Gamma_m \mid w) &= (\Gamma_{\phi \cdot m} \mid w) \\ &= \int_{\phi(m)} w(x)^T n_{\phi(m)}(x) d\sigma_{\phi(m)}(x) \\ &= \int_m \det(D\phi)(D\phi^{-1}w(\phi(y)))^T n_m(\phi(y)) d\sigma_m(y) \\ &= (\Gamma_m \mid \det(D\phi) D\phi^{-1}w \circ \phi) \end{aligned}$$

The third identity in the computation above can be proved (for example) by expressing the integral in local charts, and we skip the argument. Note that, here and in the rest of the paper, the differential operator is always assumed to take precedence on matrix operations, so that  $D\phi^{-1}$  means  $(D\phi)^{-1}$ , while the derivative of the inverse map would be denoted  $D(\phi^{-1})$ .

We now apply this identity to the approximation  $\Theta_{(\boldsymbol{\beta}, \tilde{\mathbf{c}}, \tilde{\mathbf{S}})}$  to derive a diffeomorphic action on  $(\boldsymbol{\beta}, \tilde{\mathbf{c}}, \tilde{\mathbf{S}})$ . By linearity, it suffices to consider a single term and compute (assuming, to simplify, that  $S$  is invertible)

$$\begin{aligned} &(\beta d\mu_{c,S} \mid \det(D\phi) D\phi^{-1}w \circ \phi) \\ &= \int_{\mathbb{R}^d} \det(D\phi)(D\phi^{-T}(x)\beta)^T w \circ \phi(x) d\mu_{c,S}(x) \\ &= \int_{\mathbb{R}^d} (D(\phi^{-1})^T(y)\beta)^T w(y) \frac{\exp(-( \phi^{-1}(y) - c)^T S^{-1}(\phi^{-1}(y) - c)/2)}{(2\pi)^{d/2} \sqrt{\det S}} dx dy \end{aligned}$$

Letting  $c' = \phi(c)$ , we make the first-order approximation

$$\phi^{-1}(y) - c \simeq D(\phi^{-1})(c')(y - c') = D\phi^{-1}(c)(y - c')$$

so that, letting  $S' = D\phi(c)SD\phi^T(c)$ ,

$$(\beta d\mu_{c,S} \mid \det(D\phi) D\phi^{-1}w \circ \phi)$$

$$\begin{aligned}
&\simeq \int_{\mathbb{R}^d} (D(\phi^{-1})^T(y)\beta)^T w(y) \frac{\exp\left(-(y-c')^T S'^{-1}(y-c')/2\right)}{(2\pi)^{d/2} \sqrt{\det S'}} dx dy \\
&\simeq \int_{\mathbb{R}^d} \det D\phi(c) (D\phi^{-T}(c)\beta)^T w(y) \frac{\exp\left(-(y-c')^T S'^{-1}(y-c')/2\right)}{(2\pi)^{d/2} \sqrt{\det S'}} dx dy \\
&= \det D\phi(c) (D\phi^{-T}(c)\beta) d\mu_{c', S'}
\end{aligned}$$

in which we have made the additional approximations

$$D(\phi^{-1})^T(y)\beta \simeq D\phi^{-T}(c)\beta,$$

(the  $-T$  exponent referring to the inverse transpose) and

$$\det S' = (\det D\phi(c))^2 \det S \simeq (\det D\phi(\phi^{-1}(y)))^2 \det S.$$

The approximation can be extended to singular  $S$ 's using, for example, a limit argument. This defines the action

$$\phi \cdot (\beta, c, S) = (\det D\phi(c) D\phi^{-T}(c)\beta, \phi(c), D\phi(c) S D\phi(c)^T)$$

which is immediately extended to tuples  $\mathbf{q} = (\beta, \tilde{c}, \tilde{S})$ . Note that, as emphasized by the different notation, the  $c$  and  $S$  variables that are used in the surface approximation and those in the vector field decomposition do not have to coincide, since they represent objects of different nature. On the other hand, it is possible to have overlaps between the two sets  $(\mathbf{c}, \mathbf{S})$  and  $(\tilde{\mathbf{c}}, \tilde{\mathbf{S}})$  without creating inconsistencies since both sets transform identically under diffeomorphic action.

Using this action, the state evolution equation in terms of the vector field becomes

$$\begin{cases} \partial_t \tilde{c}_j = v(t, \tilde{c}_j) \\ \partial_t \tilde{S}_j = Dv(\tilde{c}_j) \tilde{S}_j + \tilde{S}_j Dv(\tilde{c}_j)^T \\ \partial_t \beta_j = -Dv(\tilde{c}_j)^T \beta_j + \nabla \cdot v(\tilde{c}_j) \beta_j \end{cases}$$

Let  $q_j = (\beta_j, \tilde{c}_j, \tilde{S}_j)$ . The function  $\zeta(\mathbf{q}, c, S)$  can now be decomposed in terms

$$(\zeta_\beta(\mathbf{q}_j, c, S), \zeta_c(q_j, c, S), \zeta_S(q_j, c, S)), \quad j = 1, \dots, N,$$

with

$$\begin{cases} \zeta_\beta(q_j; c, S) = -\nabla f(\tilde{c}_j; c, S) \beta_j^T + \beta_j \nabla f(\tilde{c}_j; c, S)^T \\ \zeta_c(q_j; c, S) = f(\tilde{c}_j; c, S) \text{Id}_{\mathbb{R}^3} \\ \zeta_S(q_j; c, S) \alpha = \alpha \nabla f(\tilde{c}_j; c, S)^T \tilde{S}_j + S_j \nabla f(\tilde{c}_j; c, S) \alpha^T \end{cases}$$

where  $\alpha$  in the last equation represents an arbitrary vector in  $\mathbb{R}^3$ . Letting  $\tilde{f}_{jk} = f(\tilde{c}_j; c_k, S_k)$ ,  $\tilde{b}_{jk} = S_k^{-1}(\tilde{c}_j - c_k)$ , one can express the  $\zeta$ -dependent terms in the state and co-state evolutions in a reasonably concise form. The previous system, for example, becomes

$$\begin{cases} \zeta_\beta(q_j; c_l, S_l)\alpha_l = \tilde{f}_{jl}(\beta_j^T \alpha_l \tilde{b}_{jl} - \tilde{b}_{jl}^T \alpha_l \beta_j) \\ \zeta_c(q_j; c_l, S_l)\alpha_l = \tilde{f}_{jl}\alpha_l \\ \zeta_S(q_j; c_l, S_l)\alpha_l = -\tilde{f}_{jl}(\alpha_l \tilde{b}_{jl}^T \tilde{S}_j + \tilde{S}_j \tilde{b}_{jl} \alpha_l^T) \end{cases}$$

Representing the co-state by variables  $p_{\beta,j}, p_{\tilde{c},j}, p_{\tilde{S},j}$  for  $j = 1, \dots, N$ , we have

$$\begin{aligned} \partial_t p_{\beta,j} &= - \sum_{l=1}^M \tilde{f}_{jl}((p_{\beta,j}^T \tilde{b}_{jl})\alpha_l - (\tilde{b}_{jl}^T \alpha_l)p_{\beta,j}), \\ \partial_t p_{\tilde{c},j} &= \sum_{l=1}^M \tilde{f}_{jl}((p_{\beta,j}^T \tilde{b}_{jl})(\beta_j^T \alpha_l) - (\tilde{b}_{jl}^T \alpha_l)p_{\beta,j}^T \beta_j) \tilde{b}_{jl} \\ &\quad - \sum_{l=1}^M \tilde{f}_{jl}(\beta_j^T \alpha_l A_l p_{\beta,j} - \beta_j^T p_{\beta,j} A_l \alpha_l) \\ &\quad + \sum_{l=1}^M \tilde{f}_{jl}(p_{\tilde{c},j}^T \alpha_l) \tilde{b}_{jl} - 2 \sum_{l=1}^M \tilde{f}_{jl}(\tilde{b}_{jl} \tilde{b}_{jl}^T - A_l) \tilde{S}_j p_{\tilde{S},j} \alpha_l \\ \partial_t p_{\tilde{S},j} &= \sum_{l=1}^M f_{kl}(p_{S,k} \alpha_l b_{kl}^T + b_{kl} \alpha_l^T p_{S,k}) \end{aligned}$$

The missing terms in the evolution of  $\mathbf{c}$  and  $\mathbf{S}$  are

$$\begin{aligned} -\partial_{c_k}(\mathbf{p}_q^T \zeta(\mathbf{q}, c_k, S_k)\alpha_k) &= - \sum_{j=1}^N \tilde{f}_{jk}((p_{\beta,j}^T \tilde{b}_{jk})(\beta_j^T \alpha_k) - (\tilde{b}_{jk}^T \alpha_k)p_{\beta,j}^T \beta_j) \tilde{b}_{jk} \\ &\quad + \sum_{j=1}^N \tilde{f}_{jk}(\beta_j^T \alpha_k A_k p_{\beta,j} - \beta_j^T p_{\beta,j} A_k \alpha_k) \\ &\quad - \sum_{j=1}^N \tilde{f}_{jk}(p_{\tilde{c},j}^T \alpha_k) \tilde{b}_{jk} \\ &\quad + 2 \sum_{j=1}^N \tilde{f}_{jk}(\tilde{b}_{jk} \tilde{b}_{jk}^T - A_k) \tilde{S}_j p_{\tilde{S},j} \alpha_k \end{aligned}$$



$$\begin{aligned}
& -\partial_{S_k}(\mathbf{p}_q^T \zeta(\mathbf{q}, c_k, S_k) \alpha_k) \\
& = -\frac{1}{2} \sum_{j=1}^N \tilde{f}_{jk}((p_{\beta,j}^T \tilde{b}_{jk})(\alpha_k^T \beta_j) - (\tilde{b}_{jk}^T \alpha_k)(p_{\beta,j}^T \beta_j)) \tilde{b}_{lk} \tilde{b}_{jk}^T \\
& + \frac{1}{2} \sum_{j=1}^N \tilde{f}_{jk}(\beta_j^T \alpha_k)(\tilde{b}_{jk} p_{\beta,j}^T A_k + A_k p_{\beta,j} \tilde{b}_{jk}^T) \\
& - \frac{1}{2} \sum_{j=1}^N \tilde{f}_{jk}(\beta_j^T p_{\beta,j})(\tilde{b}_{jk} \alpha_k^T A_k + A_k \alpha_k \tilde{b}_{jk}^T) - \frac{1}{2} \sum_{j=1}^N \tilde{f}_{jk}(p_{\tilde{c},j}^T \alpha_k) \tilde{b}_{jk} \tilde{b}_{jk}^T \\
& + \sum_{j=1}^N \tilde{f}_{jk}((\alpha_k^T p_{\tilde{S},j} \tilde{S}_j \tilde{b}_{jk}) \tilde{b}_{jk} \tilde{b}_{jk}^T - A_k \tilde{S}_j p_{\tilde{S},j} \alpha_k \tilde{b}_{jk}^T - \tilde{b}_{jk} \alpha_k^T p_{\tilde{S},j} \tilde{S}_j A_k).
\end{aligned}$$

The last painful item is the gradient of the endpoint cost function with respect to the state variables. Let  $\tilde{M} = (\tilde{\mathbf{q}}, \tilde{T})$  be the target surface,

$$F(\boldsymbol{\beta}, \tilde{\mathbf{c}}, \tilde{\mathbf{S}}) = \sum_{i,j=1}^N h(\tilde{c}_i, \tilde{S}_i; \tilde{c}_j, \tilde{S}_j)(\beta_i^T \beta_j) - 2 \sum_{i=1}^N \sum_{j=1}^K (\beta_i^T \tilde{n}_j) h(c_i, S_i; x_j, 0),$$

which is, up to a constant additive factor, equal to  $\|\Theta_{(\boldsymbol{\beta}, \tilde{\mathbf{c}}, \tilde{\mathbf{S}})} - \Gamma_{(\tilde{\mathbf{q}}, \tilde{T})}\|_{\xi}^2$ , consistently with (18). Then

$$\begin{aligned}
\partial_{\beta_i} F &= 2 \sum_{j=1}^N h_{ij} \beta_j - 2 \sum_{j=1}^K \tilde{h}_{ij} \tilde{n}_j, \\
\partial_{\tilde{c}_i} F &= -2 \sum_{j=1}^N h_{kl} (\beta_i^T \beta_j) \tilde{b}_{ij}' + 2 \sum_{j=1}^K \tilde{h}_{ij} (\beta_i^T \tilde{n}_j) \tilde{b}_{ij}''
\end{aligned}$$

and

$$\partial_{\tilde{S}_i} F = \sum_{j=1}^N h_{ij} (\beta_i^T \beta_j) (\tilde{b}_{ij}' \tilde{b}_{ij}'^T - B_{ij}') - \sum_{j=1}^K \tilde{h}_{ij} (\beta_i^T \tilde{n}_j) (\tilde{b}_{ij}'' \tilde{b}_{ij}''^T - A_i')$$

where  $h_{ij} = h(\tilde{c}_i, \tilde{S}_i; \tilde{c}_j, \tilde{S}_j)$ ,  $\tilde{h}_{ij} = h(\tilde{c}_i, \tilde{S}_i; x_j, 0)$ ,  $A_i' = (\tau^2 \text{Id} + \tilde{S}_i)^{-1}$ ,  $\tilde{b}_{ij}'' = A_i'(\tilde{c}_i - x_j)$ ,  $B_{ij}' = (\tau^2 \text{Id} + \tilde{S}_i + \tilde{S}_j)^{-1}$ ,  $\tilde{b}_{ij}' = B_{ij}'(c_i - c_j)$ .

**3.2.4. Experiments** We here present a few experimental results using the approaches that were described in the previous sections. The first experiment, in Figure 1, compares the registration of two hippocampus shapes

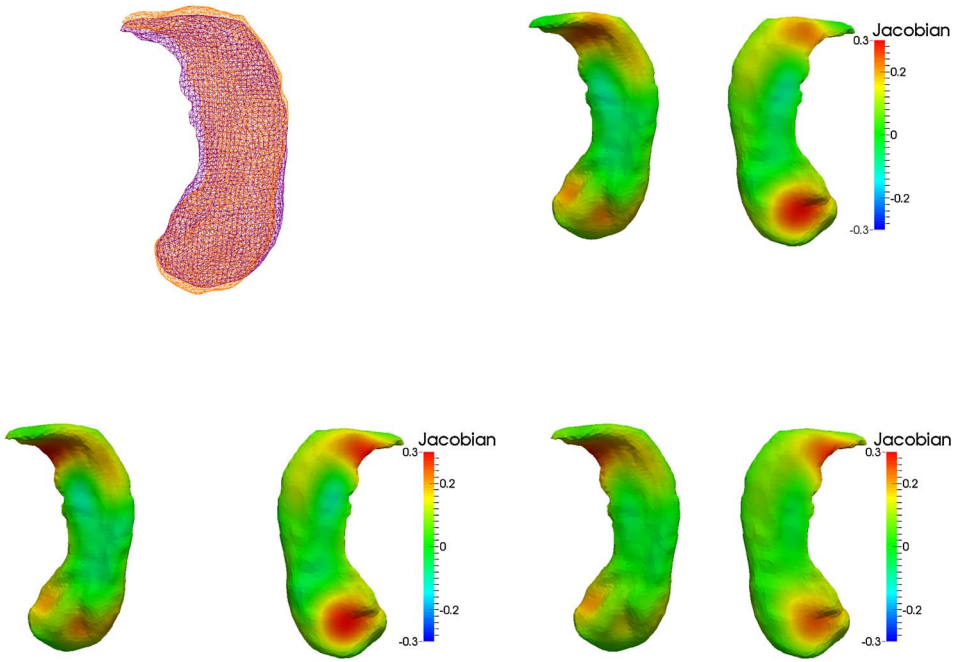


Figure 1: Comparing surface LDDMM and diffeon approximation for hippocampus shapes. Upper Left: template (red) and target (orange) shapes overlaid as wireframe; Upper Right: Deformed surface using the LDDMM-surface algorithm; Lower Left: Deformed surface using 120 diffeons (section 3.2.1); Lower Right: Deformed surface using 120 diffeons and 250 GVM's. Color maps reflect the jacobian determinant of the transformation, and each surface is visualized from two angles.

that were triangulated over approximately 2,000 vertices, first using the LDDMM surface matching method in [53], then the approach described in section 3.2.1 using 120 diffeons, then the one described in section 3.2.3, still using 120 diffeons to decompose the vector fields, but also approximating the template surface over 250 GVM's. The three methods return very similar results, as indicated by the jacobian determinant of the estimated transformation mapped over the deformed surface.

We next consider more challenging surface matching experiments mapping an ellipsoid onto two possible targets (called Target A and B), both involving substantial deformation (see Figure 2). Each surface is triangulated over 2,000 vertices. In both cases, the standard LDDMM surface matching algorithm and the one using 112 diffeons yield very similar results. Replacing

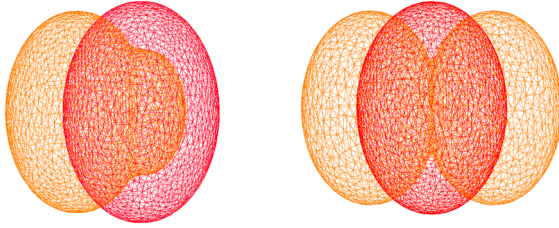


Figure 2: Surface matching experiments. For both, the template (red) is an ellipsoid. Two target surfaces (orange) are considered. Target A (left) is an ellipsoid with a bump protruding on the right. Target B (right) is formed with two ellipsoids glued together on each side of the template.

the surface template with an approximation using GVM's yields significant differences in the solution when only 250 measures are used, while the difference is milder, but still noticeable, when using 500 measures. If  $T$  is the number of time discretization steps,  $N$  the number of points in the template surface,  $\tilde{N}$  the number of points in the target surface,  $M$  the number of diffeons, and  $K$  the number of vector measures, the complexity of the original LDDMM time evolution is  $O(TN^2)$ ; the diffeon approximation reduces it to  $O(TMN)$  while using GVM's gives  $O(TKM)$ . The computation of the data attachment term is  $O(N(N + \tilde{N}))$  for standard LDDMM and its diffeon approximation, and is reduced to  $O(K(K + \tilde{N}))$  when using GVM's. There are of course multiplicative constants hidden within these orders of magnitude, and the gain in computing speed for diffeon-based methods is partially compensated by the additional cost induced by using more shape variables. In our experiments ( $N = \tilde{N} = 2,000$ ,  $M \simeq 100$  and  $K = 250$ ), diffeon-based methods were about three times faster than standard LDDMM. The ratio would obviously increase for larger values of  $N$ .

### 3.3. Image matching

We now consider a framework in which two images  $I_0$  and  $I_1$  are given and the goal is to find a diffeomorphism  $\phi$  such that  $\phi \cdot I_0 = I_0 \circ \phi^{-1} \simeq I_1$ , as captured by the cost function

$$\tilde{F}(\phi) = \int_{\Omega} (I_0 \circ \phi^{-1}(y) - I_1(y))^2 dy = \int_{\Omega} (I_0(x) - I_1 \circ \phi(x))^2 \det D\phi(x) dx$$

Using the Lagrangian approach introduced in [32, 15], the discretization that we consider is based on a possibly non-uniform grid of points  $\mathbf{x} =$

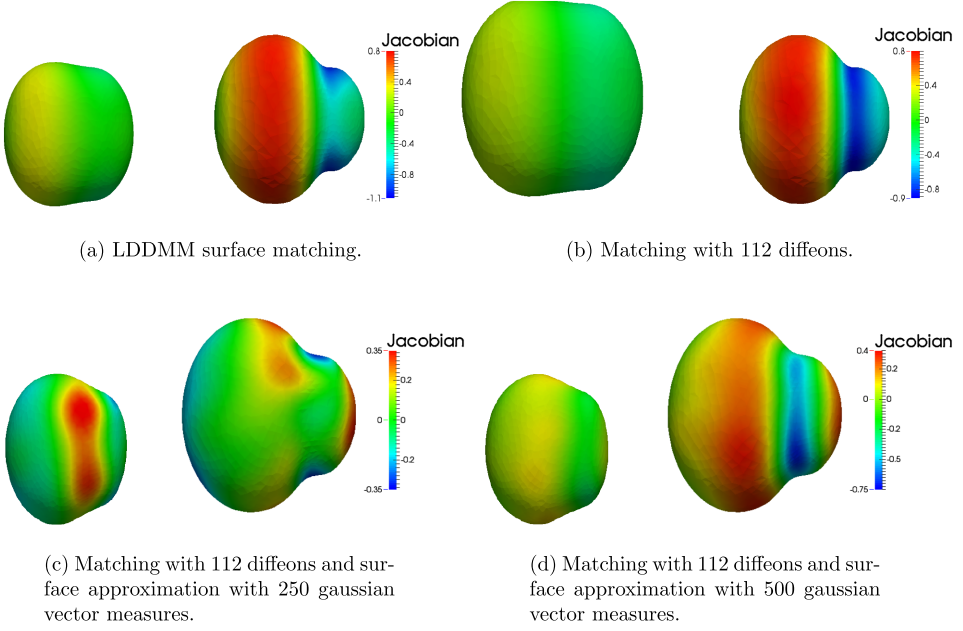


Figure 3: Matching results for Target A. The left surface in each row is the deformed template at mid-trajectory ( $t = 0.5$ ), and the right one is the final deformed template. Both surfaces are colored according to the jacobian determinant of the current transformation, each row being scaled according to a different color map. See main text for additional comments.

$(x_1, \dots, x_N)$ , with attached infinitesimal log-volumes  $\boldsymbol{\rho} = (\rho_1, \dots, \rho_N)$ , where  $x_j(t) = \phi(x_j(0), t)$  and  $\rho_j(t) = \log \det D\phi(x_j(0), t)$ . The previous integral is approximated by

$$F(\mathbf{x}, \boldsymbol{\rho}) = \sum_{j=1}^N (I_{0j} - I_1(x_j))^2 e^{\rho_j}$$

where  $I_{0j} = I_0(x_j(0))$ . Define the state variable  $\mathbf{q} = (\mathbf{x}, \boldsymbol{\rho})$ , evolving according to

$$\begin{cases} \partial_t x_j = v(t, x_j) \\ \partial_t \rho_j = \nabla \cdot v(t, x_j) \end{cases}$$

Using the same notation as in section 3.1, we can write

$$\begin{cases} \zeta_{j,x}(\mathbf{q}; c_l, S_l) \alpha_l = \tilde{f}_{jl} \alpha_l \\ \zeta_{j,\rho}(\mathbf{q}; c_l, S_l) \alpha_l = -\tilde{f}_{jl} \tilde{b}_{jl}^T \alpha_l \end{cases}$$

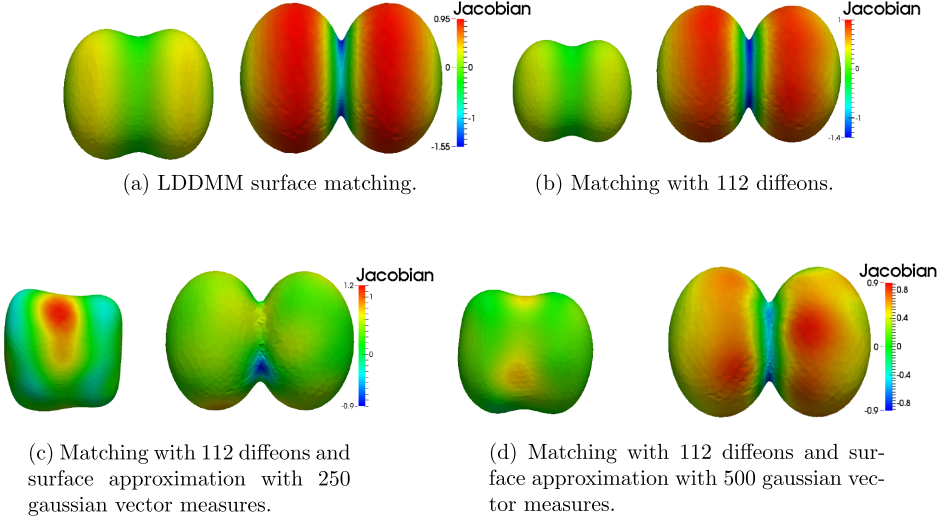


Figure 4: Matching results for Target B. See Figure 3 for description and main text for additional comments.

Co-state evolution becomes

$$\partial_t p_{x,j} = \sum_{l=1}^M \tilde{f}_{jl}(p_{x,j}^T \alpha_l) \tilde{b}_{jl} - \sum_{l=1}^M p_{\rho,j} \tilde{f}_{jl}((\tilde{b}_{jl}^T \alpha_l) \tilde{b}_{jl} - A_l \alpha_l).$$

$$\partial_t p_{\rho,j} = 0.$$

The first sum in (14) is

$$-\partial_{c_k}(\mathbf{p}_q^T \zeta(\mathbf{q}, c_k, S_k) \alpha_k) = -\sum_{j=1}^N \tilde{f}_{jk}(p_{x,k}^T \alpha_k) \tilde{b}_{jk} + \sum_{j=1}^N p_{\rho,j} \tilde{f}_{jk}((\tilde{b}_{jk}^T \alpha_k) \tilde{b}_{jk} - A_k \alpha_k)$$

and the first sum in (16) is

$$\begin{aligned} -\partial_{S_k}(\mathbf{p}_q^T \zeta(\mathbf{q}, c_k, S_k) \alpha_k) &= -\frac{1}{2} \sum_{j=1}^N \tilde{f}_{jk}(p_{x,j}^T \alpha_k) \tilde{b}_{jk} \tilde{b}_{jk}^T \\ &+ \frac{1}{2} \sum_{j=1}^N p_{\rho,j} \tilde{f}_{jk}(\tilde{b}_{jk}^T \alpha_k) \tilde{b}_{jk} \tilde{b}_{jk}^T - \frac{1}{2} \sum_{j=1}^N p_{\rho,j} \tilde{f}_{jk}(A_k \alpha_k \tilde{b}_{jk}^T + \tilde{b}_{jk} \alpha_k^T A_k) \end{aligned}$$

In the (very preliminary) 2D experiments that follow,  $x_j(0)$  is initialized to a subgrid of the discrete domain over which both  $I_0$  and  $I_1$  are defined,

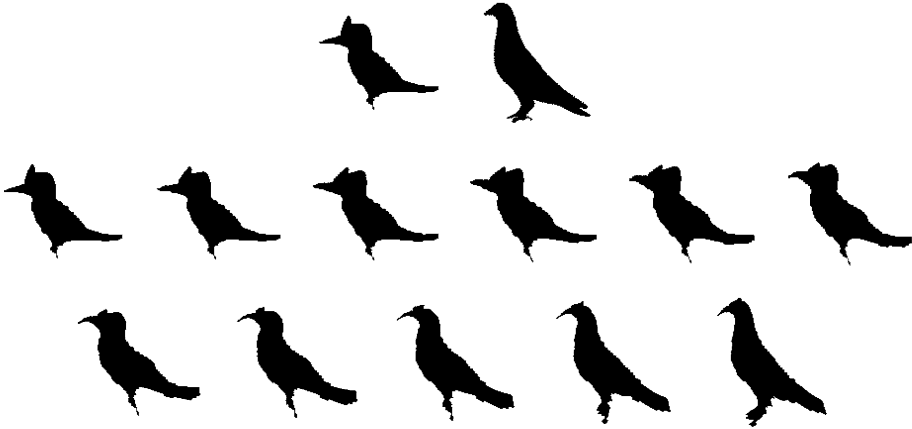


Figure 5: Evolution of the target image composed on the right by the time dependent diffeomorphisms. First row: Target and template images. Second row:  $t = 0$  to  $t = 0.5$ . Third row:  $t = 0.6$  to  $t = 1.0$ .

and  $e^{\rho_j}$  is taken equal to the volume of a subgrid cell. Initial diffeons  $\mathbf{c}(0)$  are centered on a sparser subgrid, and initial covariance matrices  $\mathbf{S}(0)$  are isotropic, with diagonal element equal to the radius of the grid size. To avoid dealing with boundary conditions, the image  $I_1$  is extended to the whole euclidean space, letting the value outside the initial image domain coincide with the value at the nearest point within the image domain. In terms of computing speed, using Lagrangian methods, even with a limited number of diffeons, remains less efficient than Eulerian implementations of image LDDMM, such as those described in [7, 56]. On the other hand, as noted in [15], Lagrangian implementations are more suitable when combining several shape modalities, like landmarks, surfaces and images.

The first rows of Figures 5 and 7 show the template and target images for two experiments: the first experiment maps two binary images (bird), and the second experiment registers two ultrasound heart images. In both cases, we present the backward image evolution from the target image to the template, which is explicitly computed by the algorithm (Figures 5 and 7) and the evolution of the diffeons (Figures 6 and 8) which provide an interesting glyph representation of the evolving diffeomorphism.

#### 4. Discussion

While this paper is focused on numerical improvements of the LDDMM algorithms, we strengthen the fact that these algorithms are one in many

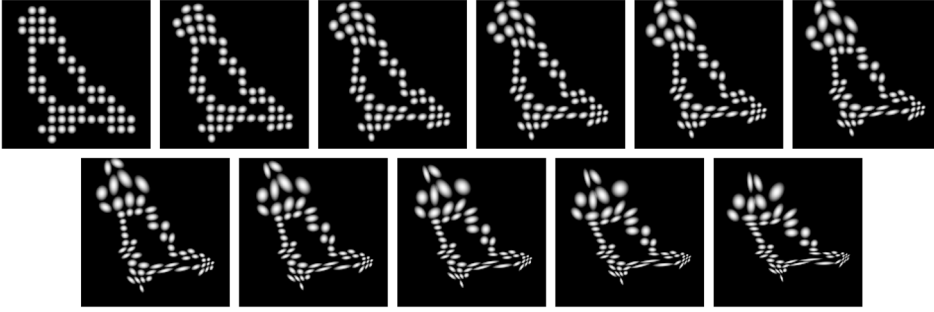


Figure 6: Diffeon evolution for the sequence in Figure 5. Diffeons are initially aligned with the template image, and evolve to align along the target shape. First row: Target and template images. Second row:  $t = 0$  to  $t = 0.5$ . Third row:  $t = 0.6$  to  $t = 1.0$ .

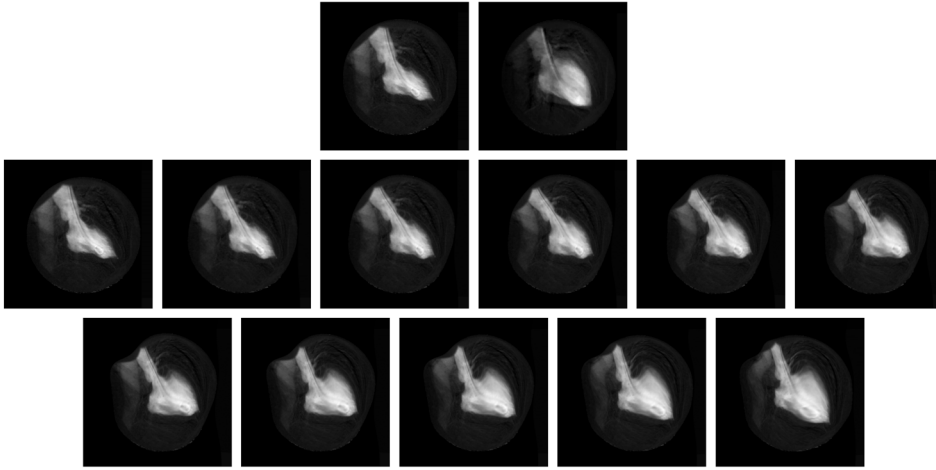


Figure 7: Evolution of the target image composed on the right by the time dependent diffeomorphisms. First row: Target and template images. Second row:  $t = 0$  to  $t = 0.5$ . Third row:  $t = 0.6$  to  $t = 1.0$ .

methods that have been proposed in the literature to perform registration tasks. These methods can be based on elastic matching energies [6, 14], while other rely, like LDDMM, on viscous fluid approaches [11, 50, 54, 2, 55]. For surfaces, several authors have developed approaches to find approximate conformal parametrizations with respect to the unit disc or sphere [28, 26, 31, 21, 33, 60, 29, 22, 30]. More recently, quasi-conformal parametrizations

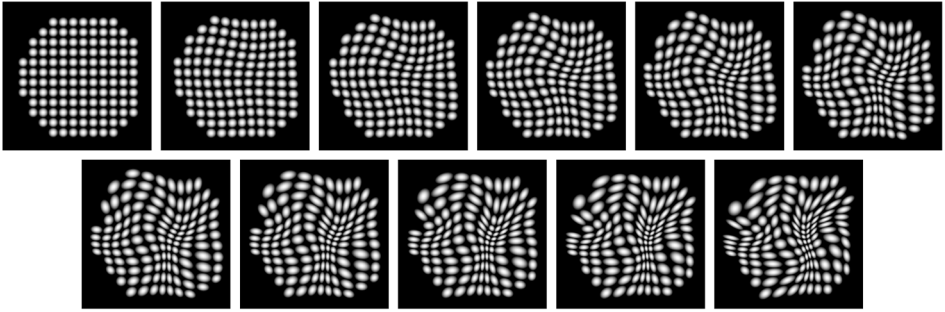


Figure 8: Diffeon evolution for the sequence in Figure 7. Diffeons are initially aligned with the template image, and evolve to align along the target shape. First row: Target and template images. Second row:  $t = 0$  to  $t = 0.5$ . Third row:  $t = 0.6$  to  $t = 1.0$ .

based on the minimization of the Beltrami coefficient have been designed [64, 39, 59]. Another class of non-rigid registration methods include those based on optimal mass transportation [25, 27, 37, 38], while [42, 40, 41] introduce comparison methods based on Gromov–Hausdorff or Gromov–Wasserstein distances. Computational methods based on integer programming and graph optimization have also been recently introduced [65, 67, 20]. It is difficult to be exhaustive when describing a still expanding field like image and surface registration, and the above list certainly is not. Comparison between methods is even more difficult, and require extensive work. One can find comparative studies in [35, 36], among others, for alternative registration algorithms.

In this paper, we have presented a new numerical framework for the computation of optimal paths in spaces of diffeomorphisms within the LDDMM setting, for surface and image matching. In this approach, which decomposes the Eulerian velocity over diffeon bases that deform during motion, as previously described in [63], we have introduced new methods to estimate diffeons for images and surfaces, and developed specific optimal control algorithms that optimize their trajectories.

Further work should include a reflection on how to insert the diffeon representation within a multi-scale setting. Working with Lagrangian variables, over non-uniform grids, does bring an extra computational burden in the image case, but it also brings more flexibility on point placement, allowing an implementation to dedicate more resources on regions that are likely to deform more, when this information is available, which is often the



case with medical images. This should lead to non-homogeneous refinement methods in a multi-resolution setting and will be developed in future work.

## References

- [1] Siamak Ardekani, Robert G. Weiss, Albert C. Lardo, Richard T. George, Joao A. C. Lima, Katherine C. Wu, Michael I. Miller, Raimond L. Winslow, and Laurent Younes. Cardiac motion analysis in ischemic and non-ischemic cardiomyopathy using parallel transport. In *Biomedical Imaging: From Nano to Macro, 2009. ISBI'09. IEEE International Symposium on*, pages 899–902. IEEE, 2009.
- [2] John Ashburner. A fast diffeomorphic image registration algorithm. *Neuroimage*, 38(1):95–113, 2007.
- [3] John Ashburner and Karl J. Friston. Diffeomorphic registration using geodesic shooting and gauss–newton optimisation. *NeuroImage*, 55(3):954–967, 2011.
- [4] Brian B. Avants, P. Thomas Schoenemann, and James C. Gee. Lagrangian frame diffeomorphic image registration: Morphometric comparison of human and chimpanzee cortex. *Medical image analysis*, 10(3):397–412, 2006.
- [5] Robert Azencott, Roland Glowinski, Jiwen He, Aarti Jajoo, Yipeng Li, Andrey Martynenko, Ronald H. W. Hoppe, Sagit Benzekry, and Stuart H. Little. Diffeomorphic matching and dynamic deformable surfaces in 3d medical imaging. *Comput. Methods Appl. Math.*, 10(3):235–274, 2010. [MR2770293](#)
- [6] Ruzena Bajcsy, Robert Lieberman, and Martin Reivich. A computerized system for the elastic matching of deformed radiographic images to idealized atlas images. *Journal of computer assisted tomography*, 7(4):618–625, 1983.
- [7] M. Faisal Beg, Michael I. Miller, Alain Trounev, and Laurent Younes. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *International journal of computer vision*, 61(2):139–157, 2005.
- [8] Martins Bruveris, François Gay-Balmaz, Darryl D. Holm, and Tudor S. Ratiu. The momentum map representation of images. *Journal of nonlinear science*, 21(1):115–150, 2011. [MR2775605](#)

- [9] Yan Cao, Michael I. Miller, Raimond L. Winslow, and Laurent Younes. Large deformation diffeomorphic metric mapping of vector fields. *Medical Imaging, IEEE Transactions on*, 24(9):1216–1230, 2005.
- [10] Can Ceritoglu, Kenichi Oishi, Xin Li, Ming-Chung Chou, Laurent Younes, Marilyn Albert, Constantine Lyketsos, Peter van Zijl, Michael I. Miller, and Susumu Mori. Multi-contrast large deformation diffeomorphic metric mapping for diffusion tensor imaging. *Neuroimage*, 47(2):618–627, 2009.
- [11] Gary E. Christensen, Richard D. Rabbitt, and Michael I. Miller. Deformable templates using large deformation kinematics. *Image Processing, IEEE Transactions on*, 5(10):1435–1447, 1996.
- [12] C. J. Cotter and D. D. Holm. Continuous and Discrete Clebsch Variational Principles. *Foundations of Computational Mathematics*, 9:221–242, 2009. [MR2496561](#)
- [13] Colin J. Cotter and Darryl D. Holm. Singular solutions, momentum maps and computational anatomy. *arXiv preprint nlin/0605020*, 2006.
- [14] Marc Droske and Martin Rumpf. A variational approach to nonrigid morphological image registration. *SIAM Journal on Applied Mathematics*, 64(2):668–687, 2004.
- [15] Jia Du, Laurent Younes, and Anqi Qiu. Whole brain diffeomorphic metric mapping via integration of sulcal and gyral curves, cortical surfaces, and images. *NeuroImage*, 56(1):162–173, 2011.
- [16] Stanley Durrleman, Stéphanie Allasonnière, and Sarang Joshi. Sparse adaptive parameterization of variability in image ensembles. *International Journal of Computer Vision*, 101(1):161–183, 2013. [MR3021058](#)
- [17] Joan Glaunès, 2013. Personal Communication.
- [18] Joan Glaunès, Anqi Qiu, Michael I. Miller, and Laurent Younes. Large deformation diffeomorphic metric curve mapping. *International journal of computer vision*, 80(3):317–336, 2008.
- [19] Joan Glaunès, Marc Vaillant, and Michael I. Miller. Landmark Matching via Large Deformation Diffeomorphisms on the Sphere. *Journal of Mathematical Imaging and Vision*, 20:179–200, 2004. [MR2049789](#)
- [20] Ben Glocker, Nikos Komodakis, Georgios Tziritas, Nassir Navab, and Nikos Paragios. Dense image registration through mrfs and efficient linear programming. *Medical image analysis*, 12(6):731–741, 2008.

- [21] Xianfeng Gu, Yalin Wang, Tony F. Chan, Paul M. Thompson, and Shing-Tung Yau. Genus zero surface conformal mapping and its application to brain surface mapping. In *Information Processing in Medical Imaging*, pages 172–184. Springer, 2003.
- [22] Xianfeng Gu, Yalin Wang, Tony F. Chan, Paul M. Thompson, and Shing-Tung Yau. Genus zero surface conformal mapping and its application to brain surface mapping. *Medical Imaging, IEEE Transactions on*, 23(8):949–958, 2004.
- [23] Andreas Günther, Hans Lamecker, and Martin Weiser. Flexible shape matching with finite element based lddmm. *International Journal of Computer Vision*, pages 1–16, 2012.
- [24] Andreas Günther, Hans Lamecker, Martin Weiser, et al. Direct lddmm of discrete currents with adaptive finite elements. In *Proceedings of the Third International Workshop on Mathematical Foundations of Computational Anatomy-Geometrical and Statistical Methods for Modelling Biological Shape Variability*, pages 1–14, 2011.
- [25] Eldad Haber, Gallagher Pryor, John Melonakos, Allen Tannenbaum, et al. 3d nonrigid registration via optimal mass transport on the gpu. *Medical image analysis*, 13(6):931–940, 2009.
- [26] Steven Haker, Sigurd Angenent, Allen Tannenbaum, Ron Kikinis, Guillermo Sapiro, and Michael Halle. Conformal surface parameterization for texture mapping. *Visualization and Computer Graphics, IEEE Transactions on*, 6(2):181–189, 2000.
- [27] Steven Haker, Lei Zhu, Allen Tannenbaum, and Sigurd Angenent. Optimal mass transport for registration and warping. *International Journal of Computer Vision*, 60(3):225–240, 2004.
- [28] Monica K. Hurdal, Philip L. Bowers, Ken Stephenson, L. Sumners De Witt, Kelly Rehm, Kirt Schaper, and David A. Rottenberg. Quasi-conformally flat mapping the human cerebellum. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI’99*, pages 279–286. Springer, 1999.
- [29] Monica K. Hurdal and Ken Stephenson. Cortical cartography using the discrete conformal approach of circle packings. *NeuroImage*, 23:S119–S128, 2004.
- [30] Monica K. Hurdal and Ken Stephenson. Discrete conformal methods for cortical brain flattening. *Neuroimage*, 45(1):S86–S98, 2009.

- [31] Monica K. Hurdal, Ken Stephenson, Phil Bowers, De Witt Sumners, and David A. Rottenberg. Coordinate systems for conformal cerebellar flat maps. *NeuroImage*, 11(5):S467, 2000.
- [32] Aastha Jain. *Particle methods for diffeomorphic registration*. PhD thesis, Johns Hopkins University, 2010. [MR2995997](#)
- [33] Miao Jin, Yalin Wang, S.-T. Yau, and Xianfeng Gu. Optimal global conformal surface parameterization. In *Visualization, 2004. IEEE*, pages 267–274. IEEE, 2004.
- [34] Sarang C. Joshi and Michael I. Miller. Landmark matching via large deformation diffeomorphisms. *IEEE Transactions on Image Processing*, 9:1357–1370, 2000. [MR1808275](#)
- [35] Arno Klein, Jesper Andersson, Babak A. Ardekani, John Ashburner, Brian Avants, Ming-Chang Chiang, Gary E. Christensen, D. Louis Collins, James Gee, Pierre Hellier, et al. Evaluation of 14 nonlinear deformation algorithms applied to human brain mri registration. *Neuroimage*, 46(3):786–802, 2009.
- [36] Arno Klein, Satrajit S. Ghosh, Brian Avants, B. T. T. Yeo, Bruce Fischl, Babak Ardekani, James C. Gee, J. J. Mann, and Ramin V. Parsey. Evaluation of volume-based and surface-based brain image registration methods. *NeuroImage*, 51(1):214 – 220, 2010.
- [37] Yaron Lipman and Ingrid Daubechies. Surface comparison with mass transportation. *arXiv preprint arXiv:0912.3488*, 2009.
- [38] Yaron Lipman and Ingrid Daubechies. Conformal wasserstein distances: Comparing surfaces in polynomial time. *Advances in Mathematics*, 227(3):1047–1077, 2011. [MR2799600](#)
- [39] Lok Ming Lui, Tsz Wai Wong, Paul Thompson, Tony Chan, Xianfeng Gu, and Shing-Tung Yau. Shape-based diffeomorphic registration on hippocampal surfaces using beltrami holomorphic flow. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2010*, pages 323–330. Springer, 2010.
- [40] Facundo Mémoli. On the use of gromov-hausdorff distances for shape comparison. In *Eurographics symposium on point-based graphics*, pages 81–90. The Eurographics Association, 2007.
- [41] Facundo Mémoli. Gromov–wasserstein distances and the metric approach to object matching. *Foundations of Computational Mathematics*, 11(4):417–487, 2011. [MR2811584](#)

- [42] Facundo Mémoli and Guillermo Sapiro. A theoretical and computational framework for isometry invariant recognition of point cloud data. *Foundations of Computational Mathematics*, 5(3):313–347, 2005. [MR2168679](#)
- [43] Michael I. Miller, M. Faisal Beg, Can Ceritoglu, and Craig Stark. Increasing the power of functional maps of the medial temporal lobe by using large deformation diffeomorphic metric mapping. *Proceedings of the National Academy of Sciences of the United States of America*, 102(27):9685–9690, 2005.
- [44] Michael I. Miller, Alain Trouvé, and Laurent Younes. On the metrics and euler-lagrange equations of computational anatomy. *Annual review of biomedical engineering*, 4(1):375–405, 2002.
- [45] Michael I. Miller, Alain Trouvé, and Laurent Younes. The metric spaces, euler equations, and normal geodesic image motions of computational anatomy. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 2, pages II–635. IEEE, 2003.
- [46] Michael I. Miller, Alain Trouvé, and Laurent Younes. Geodesic shooting for computational anatomy. *Journal of mathematical imaging and vision*, 24(2):209–228, 2006. [MR2227097](#)
- [47] Marc Niethammer, Yang Huang, and François-Xavier Vialard. Geodesic regression for image time-series. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2011*, pages 655–662. Springer, 2011.
- [48] Anqi Qiu and Michael I. Miller. Cortical hemisphere registration via large deformation diffeomorphic metric curve mapping. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2007*, pages 186–193. Springer, 2007.
- [49] Laurent Risser, F. Vialard, Robin Wolz, Maria Murgasova, Darryl D. Holm, and Daniel Rueckert. Simultaneous multi-scale registration using large deformation diffeomorphic metric mapping. *Medical Imaging, IEEE Transactions on*, 30(10):1746–1759, 2011.
- [50] Jean-Philippe Thirion. Image matching as a diffusion process: an analogy with maxwell’s demons. *Medical image analysis*, 2(3):243–260, 1998.
- [51] Alain Trouvé, Stanley Durrleman, Xavier Pennec, and Nicholas Ayache. Sparse approximation of currents for statistics on curves and surfaces. In *Proceedings of MICCAI 2008*, 2008.

- [52] Alain Trouvé and François-Xavier Vialard. Shape splines and stochastic shape evolutions: a second order point of view. *Quart. Appl. Math.*, 70:219–251, 2012. [MR2953101](#)
- [53] Marc Vaillant and Joan Glaunes. Surface matching via currents. In *Information Processing in Medical Imaging*, pages 381–392, 2005.
- [54] Tom Vercauteren, Xavier Pennec, Aymeric Perchant, and Nicholas Ayache. Non-parametric diffeomorphic image registration with the demons algorithm. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2007*, pages 319–326. Springer, 2007.
- [55] Tom Vercauteren, Xavier Pennec, Aymeric Perchant, and Nicholas Ayache. Diffeomorphic demons: Efficient non-parametric image registration. *NeuroImage*, 45(1):S61–S72, 2009.
- [56] François-Xavier Vialard, Laurent Risser, Daniel Rueckert, and Colin J. Cotter. Diffeomorphic 3D image registration via geodesic shooting using an efficient adjoint calculation. *International Journal of Computer Vision*, pages 1–13, 2011.
- [57] Camille Vidal, Joshua Hewitt, Stephanie Davis, Laurent Younes, Sanjay Jain, and Bruno Jedynek. Template registration with missing parts: Application to the segmentation of m. tuberculosis infected lungs. In *Biomedical Imaging: From Nano to Macro, 2009. ISBI’09. IEEE International Symposium on*, pages 718–721. IEEE, 2009.
- [58] Richard Vinter. *Optimal Control*. Birkhäuser, 2000. [MR1756410](#)
- [59] Yalin Wang, Wei Dai, Xianfeng Gu, Tony F. Chan, Shing-Tung Yau, Arthur W. Toga, and Paul M. Thompson. Teichmüller shape space theory and its application to brain morphometry. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2009*, pages 133–140. Springer, 2009.
- [60] Yalin Wang, Xianfeng Gu, Tony F. Chan, Paul M. Thompson, and Shing-Tung Yau. Intrinsic brain surface conformal mapping using a variational method. In *Medical Imaging 2004*, pages 241–252. International Society for Optics and Photonics, 2004.
- [61] Laurent Younes. Jacobi fields in groups of diffeomorphisms and applications. *Quarterly of applied mathematics*, 65(1):113–134, 2007. [MR2313151](#)
- [62] Laurent Younes. *Shapes and diffeomorphisms*, volume 171. Springer, 2010. [MR2656312](#)

- [63] Laurent Younes. Constrained diffeomorphic shape evolution. *Foundations of Computational Mathematics*, 12(3):295–325, 2012. [MR2915564](#)
- [64] Wei Zeng and Xianfeng David Gu. Registration for 3d surfaces with large deformations using quasi-conformal curvature flow. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2457–2464. IEEE, 2011.
- [65] Yun Zeng, Chaohui Wang, Yang Wang, Xianfeng Gu, Dimitris Samaras, and Nikos Paragios. Dense non-rigid surface registration using high-order graph matching. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 382–389. IEEE, 2010.
- [66] Jiangyang Zhang, Linda J. Richards, Michael I. Miller, Paul Yarowsky, Peter van Zijl, and Susumu Mori. Characterization of mouse brain and its development using diffusion tensor imaging and computational techniques. In *Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE*, pages 2252–2255. IEEE, 2006.
- [67] Darko Zikic, Ben Glocker, Oliver Kutter, Martin Groher, Nikos Komodakis, Ali Kamen, Nikos Paragios, and Nassir Navab. Linear intensity-based image registration by markov random fields and discrete optimization. *Medical image analysis*, 14(4):550–562, 2010.

LAURENT YOUNES  
CENTER FOR IMAGING SCIENCE  
JOHNS HOPKINS UNIVERSITY  
3400 N. CHARLES ST. BALTIMORE  
MD 21218  
USA  
*E-mail address:* [laurent.younes@jhu.edu](mailto:laurent.younes@jhu.edu)

RECEIVED JULY 22, 2013