HIGHER ORDER KERNELS AND LOCALLY AFFINE LDDMM REGISTRATION

STEFAN SOMMER*, MADS NIELSEN*,†, SUNE DARKNER* , AND XAVIER PENNEC‡

Abstract. To achieve sparse description that allows intuitive analysis, we aim to represent deformation with a basis containing *interpretable* elements, and we wish to use elements that have the description capacity to represent the deformation *compactly*. We accomplish this by introducing *higher order kernels* in the LDDMM registration framework. The kernels allow local description of affine transformations and subsequent compact description of non-translational movement and of the entire non-rigid deformation. This is obtained with a representation that contains directly interpretable information from both mathematical and modeling perspectives. We develop the mathematical construction behind the higher order kernels, we show the implications for *sparse* image registration and deformation description, and we provide examples of how the capacity of the kernels enables registration with a very low number of parameters. The capacity and interpretability of the kernels lead to natural modeling of articulated movement, and the kernels promise to be useful for quantifying ventricle expansion and progressing atrophy during Alzheimer's disease.

Key words. LDDMM, diffeomorphic registration, RHKS, kernels, momentum, computational anatomy

AMS subject classifications. 65D18, 65K10, 41A15

1. Introduction.

Atrophy occurs in the human brain among patients suffering from Alzheimer's disease, and the progressing atrophy can be detected by the expansion of the ventricles [16, 13]. We wish to describe the deformation of the brain caused by the progressing disease using as few parameters as possible and with a representation which allows intuitive analysis: we search for *sparse* representations with basis elements that have the *capacity* to describe deformation with few parameters while being directly *interpretable*.

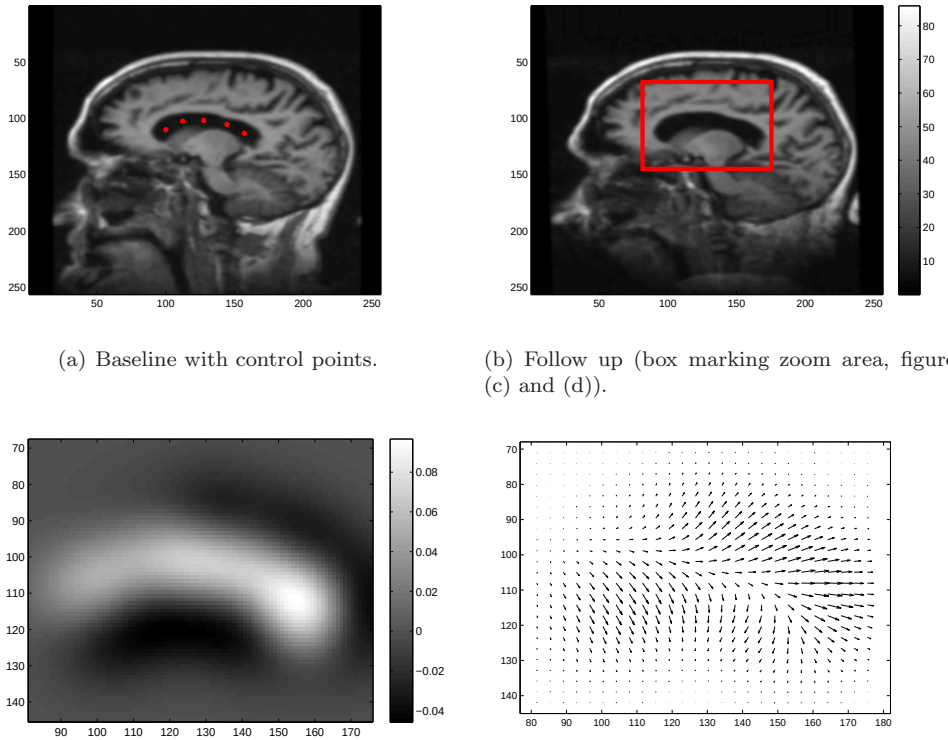
Image registration algorithms often represent translational movement in a dense sampling of the image domain. Such approaches fail to satisfy the above goals: low dimensional deformations such as expansion of the ventricles will not be represented sparsely; the registration algorithm must optimize a large number of parameters; and the expansion cannot easily be interpreted from the registration result.

In this paper, we introduce *higher order kernels* in the LDDMM registration framework to obtain a deformation representation promising *sparsity*, increased *capacity*, end *interpretability*. We show how higher order kernels allow local representation of affine transformations and that they increase the capacity of the representation at each point. We use the compact deformation description to register points and images using very few parameters, and we illustrate how the deformation coded by the kernels can be directly interpreted and that it represents information directly useful in applications: with low numbers of control points, we can detect the expanding ventricles of the patient shown in in Figure 1.1.

1.1. Background.

Among the many methods for non-rigid registration in medical imaging, the majority model the displacement of each spatial position by either a combination of suitable basis functions for the displacement itself or for the velocity of the voxels. The number of control points vary between one for each voxel [2, 15, 7] and

*Dept. of Computer Science, Univ. of Copenhagen, Denmark (sommer@diku.dk)
†BiomedIQ, Copenhagen, Denmark
‡Asclepios Project-Team, INRIA Sophia-Antipolis, France



(a) Baseline with control points.

(b) Follow up (box marking zoom area, figure (c) and (d)).

(c) log-Jacobian in ventricle area.

(d) Initial deformation field in ventricle area.

FIG. 1.1. *Progressing Alzheimer's disease cause atrophy and expansion of the ventricles. By placing five* deformation atoms *in the 2D MRI slices [17] and by using* higher order kernels*, we can register the expansion. (a) The position of the deformation atoms shown in the baseline scan; (b) the follow up scan; (c) the* log*-Jacobian determinant of the generated deformation in the ventricle area (red box in (b)); (d) the vector field at* $t = 0$ *of the generated deformation. The logarithm of the Jacobian determinant and the divergence at the deformation atoms are positive which is in line with the expected ventricle expansion, confer also Figure 5.5.*

fewer with larger basis functions [22, 5, 11]. For all methods, the infinite-dimensional space of deformations is approximated by the finite- but high-dimensional subspace spanned by the parametrization of the individual method. The approximation will be good if the underlying deformation is close to this subspace, and the representation will be compact, if few basis functions describe the deformation well. The choice of basis functions play a crucial role, and we will in the rest of the paper denote them *deformation atoms*. Two main observations constitute the motivation for the work presented in this paper:

Observation 1: Order of the Deformation Model. In the majority of registration methods, the deformation atoms model the local *translation* of each point. We wish a richer representation which is in particular able to model locally linear components in addition to local translations. The Polyaffine and Log-Euclidean Polyaffine [3, 1] frameworks pursue this by representing the velocity of a path of deformations locally by matrix logarithms. Ideas from the Polyaffine methods have recently been incorporated in e.g. the Demons algorithm [29] but, to the best of our knowledge, not

in the LDDMM registration framework. We wish to extend the set of deformation atoms used in LDDMM to allow representation of *first* and *higher order* structure and hence incorporate the benefits of the Polyaffine methods in the LDDMM framework.

Observation 2: Order of the Similarity Measure. When registering DT images, the reorientation is a function of the derivative of the warp; curve normals also contain directional information which is dependent on the warp derivative and airway trees contain directional information in the three structure which can be used for measuring similarity. These are examples of similarity measures containing *higher order information*. For the case of image registration, the warp derivative may also enter the equation either directly in the similarity measure [21, 19] or to allow use of more image information than provided by a sampling of the warp. Consider an image similarity measure on the form $U(\varphi) = \int_\Omega F(I_m(\varphi^{-1}(x)), I_f(x))dx$. A finite sampling of the domain Ω can approximate this with

$$\tilde{U}^0(\varphi) = \frac{1}{N}\sum_{i=1}^{N} F(I_m(\varphi^{-1}(x_i)), I_f(x_i)) \ .$$

Letting $\{p_1, \ldots, p_P\}$ be uniformly distributed points around 0, we can increase the amount of image information used in $\tilde{U}^0(\varphi)$ *without* additional sampling of the warp by using a first order approximation of φ^{-1}:

$$\tilde{U}^1(\varphi) = \frac{1}{NP}\sum_{i=1}^{N}\sum_{j=1}^{N} F(I_m(D\varphi^{-1}p_j + \varphi^{-1}(x_i)), I_f(p_j + x_i)) \ .$$

This can be considered an increase from *zero* to *first order* in the approximation of U. Besides including more image information than provided by the initial sampling of the warp, the increase in order allows capture of non-translational information - e.g. rotation and dilation - in the similarity measure. The approach can be seen as a specific case of similarity smoothing and more examples of smoothing in intensity based image registration can be found in [9][1].

We focus on deformation modeling with the Large Deformation Diffeomorphic Metric Mapping (LDDMM) registration framework which has the benefit of both providing good registrations and drawing strong theoretical links with Lie group theory and evolution equations in physical modeling [8, 32]. Most often, high-dimensional voxel-wise representations are used for LDDMM although recent interest in *compact* representations [11, 25] show that the number of parameters can be much reduced. These methods use interpolation of the velocity field by deformation atoms to represent translational movement but deformation by other parts of the affine group cannot be compactly represented.

The deformation atoms are in LDDMM called *kernels*. The kernels are centered at different spatial positions and parameters determine the contribution of each kernel. In this paper, we use the partial derivative reproducing property [33] to show that partial derivatives of kernels - *higher order kernels* - fit naturally in the LDDMM framework and constitute deformation atoms along with the original kernels. In particular, the higher order kernels have a singular momentum and the momentum stay singular when transported by the EPDiff evolution equations. We show how the higher order kernels allow modeling of locally affine deformations and hence extend

[1] An updated version of [9] is available at `http://diku.dk/english/staff/?id=383640&f=3&vis=medarbejder`.

the capacity of sparsely discretized LDDMM methods. In addition, they comprise the natural vehicle for incorporating first order similarity measures in the framework.

1.2. Related Work. A number of methods for non-rigid registration have been developed during the last decades including non-linear elastic methods [18], parametrizations using static velocity fields [2, 15], the demons algorithm [26, 29], and spline-based methods [22, 5]. For the particular case of LDDMM, the groundbreaking work appeared with the deformable template model by Grenander [14] and the flow approach by Christensen et al. [7] together with the theoretical contributions of Dupuis et al. and Trouvé [10, 27]. Algorithms for computing optimal diffeomorphisms have been developed in [4], and [28] uses the momentum representation for statistics and develops a momentum based algorithm for the landmark matching problem.

Locally affine deformations can be modeled using the Polyaffine and Log-Euclidean Polyaffine [3, 1] frameworks. The velocity of a path of deformations is here computed using matrix logarithms, and the resulting diffeomorphism flowed forward by integrating the velocity. Ideas from the Polyaffine methods have recently been incorporated in e.g. the Demons algorithm [29, 23]. In LDDMM, the deformation atoms, the kernels, represent translational movement and the non-translational part of affine transformations cannot directly be represented. We will show how higher order kernels constitute deformation atoms which allow representing the linear parts of affine transformations. From a mathematical points of view, this is possible due to the partial derivative reproducing property (Zhou [33]). The partial derivative reproducing property has been used in [6] to derive variations of flow equations for LDDMM DTI registration but higher order kernels are not used in the parametrization. Confer the monograph [32] for information on RKHSs and their role in LDDMM.

In order to reduce the dimensionality of the parametrization used in LDDMM, Durrleman et al. [11] introduced a control point formulation of the registration problem by choosing a finite set of control points and constraining the momentum to be concentrated as Dirac measures at the point trajectories. As we will see, higher order kernels make a finite control point formulation possible which is different in important aspects. Younes [31] in addition considers evolution in constrained subspaces.

Higher order kernels increase the capacity of the deformation parametrization, a goal which is also treated in sparse multi-scale methods such as the kernel bundle framework [25]. This method concerns the size of the kernel in contrast to the order which we deal with here. As we will discuss in the experiments section, the size of the kernel is important for higher order kernels as well, and higher order kernels and the kernel bundle method will likely complement each other nicely if applied together.

1.3. Content and Outline. We start the paper with an overview of LDDMM registration and the mathematical constructs behind the method. In the following section, we motivate the introduction of higher order kernels using zero- and first-order similarity measure approximations. We describe the derivative reproducing property, and show how it implies singular momentum for the kernels. The evolution of the momentum and velocity fields governed by the EPDiff evolution equations are then determined. To make actual registration possible, the next section describes the effect of varying the initial conditions and the backwards gradient transport before developing the actual matching algorithm. We give examples in the second last section, and we show how the higher order kernels are particularly useful when registering human brains with progressing atrophy. The paper ends with concluding remarks and outlook. The paper thus contributes by

(1) introducing higher order kernels in the LDDMM framework as the deformation atoms enabling locally affine transformations,
(2) showing how the order of the similarity measure approximation relates to higher order kernels,
(3) relating the derivative reproducing property to LDDMM and showing how it implies a singular momentum for the higher order kernels,
(4) deriving the EPDiff transport equations for higher order kernels,
(5) computing the forward variational equations and describing the backwards gradient transport,
(6) developing an algorithm allowing matching with the higher order kernels,
(7) and demonstrating the application of the kernels with registration examples.

2. LDDMM Registration, Kernels, and Evolution Equations. We here give a brief introduction to LDDMM registration. For further information, confer the monograph [32] with extensive information on the method.

In the LDDMM framework, registration is performed through the action of diffeomorphisms on geometric objects. This approach is very general and allows the framework to be applied to both landmarks, curves, surfaces, images, and tensors. In the case of images, the action of a diffeomorphism φ on the image $I : \Omega \to \mathbb{R}$ takes the form $\varphi.I = I \circ \varphi^{-1}$, and given a fixed image I_f and moving image I_m, the registration amounts to a search for φ such that $\varphi.I_m \sim I_f$. In exact matching, we wish $\varphi.I_m$ be exactly equal to I_f but, more frequently, we allow some amount of inexactness to account for noise in the images and allow for smoother diffeomorphisms. This is done by defining a similarity measure $U(\varphi) = U(\varphi.I_m, I_f)$ on images and a regularization measure E_1 to give a combined energy

$$E(\varphi) = E_1(\varphi) + \lambda U(\varphi.I_m, I_f) \ . \tag{2.1}$$

Here λ is a positive real representing the trade-off between regularity and goodness of fit. The similarity measure U is in the simplest form the L^2-error $\int_\Omega |\varphi.I_m(x) - I_f(x)|dx$ but more advanced measures can be used (e.g. [20, 30, 9]).

In order to define the regularization term E_1, we introduce some notations in the following: Let the domain Ω be a subset of $\mathbb{R}^d$ with $d = 2, 3$, and let V denote a Hilbert space of vector fields $v : \Omega \to \mathbb{R}^d$ such that V with associated norm $\|\cdot\|_V$ is included in $L^2(\Omega, \mathbb{R}^d)$ and admissible [32, Chap. 9], i.e. sufficiently smooth. Given a time-dependent vector field $t \mapsto v_t$ with

$$\int_0^1 \|v_t\|_V^2 \, dt < \infty \tag{2.2}$$

the associated differential equation $\partial_t \varphi_t = v_t \circ \varphi_t$ has with initial condition φ_s a diffeomorphism φ_{st}^v as unique solution at time t. The set G_V of diffeomorphisms built from V by such differential equations is a Lie group, and V is its tangent space at each point. The inner product on V associated to a norm $\|\cdot\|_V$ makes G_V a Riemannian manifold with right-invariant metric. Setting $\varphi_{00}^v = Id_\Omega$, the map $t \mapsto \varphi_{0t}^v$ is a path from Id_Ω to φ with energy given by (2.2) and generated by v_t. We will use this notation extensively in the following. A critical path for the energy (2.2) is a geodesic on G_V, and the regularization term E_1 is defined using the energy by

$$E_1(\varphi) = \min_{v_t \in V, \varphi_{01}^v = \varphi} \int_0^1 \|v_s\|_V^2 \, ds \ , \tag{2.3}$$

i.e. it measures the minimal energy of diffeomorphism paths from Id_Ω to φ. Since the energy is high for paths with great variation, the term penalizes highly varying paths, and a low value of $E_1(\varphi)$ thus implies that φ is regular.

2.1. Kernel and Momentum. As a consequence of the assumed admissibility of V, the evaluation functionals $\delta_x : v \mapsto v(x) \in \mathbb{R}^d$ is well-defined and continuous for any $x \in \Omega$. Thus, for any $a \in \mathbb{R}^d$ the map $a \otimes \delta_x : v \mapsto a^T v(x)$ belongs to the topological dual V^* consisting of the continuous linear maps of V. This in turn implies the existence of spatially dependent matrices $K : \Omega \times \Omega \to \mathbb{R}^{d\times d}$, the *kernel*, such that, for any constant vector $a \in \mathbb{R}^d$, the vector field $K(\cdot, x)a \in V$ represents $a \otimes \delta_x$ and $\langle K(\cdot, x)a, v\rangle_V = a \otimes \delta_x(v)$ for any $v \in V$, point $x \in \Omega$ and vector $a \in \mathbb{R}^d$. This latter property is denoted the reproducing property and gives V the structure of a reproducing kernel Hilbert space (RKHS). Tightly connected to the norm and kernels is the notion of *momentum* given by the linear momentum operator $L : V \to V^* \subset L^2(\Omega, \mathbb{R}^d)$ which satisfies

$$\langle Lv, w\rangle_{L^2(\Omega,\mathbb{R}^d)} = \int_\Omega \big(Lv(x)\big)^T w(x)dx = \langle v, w\rangle_V$$

for all $v, w \in V$. The momentum operator connects the inner product on V with the inner product in $L^2(\Omega, \mathbb{R}^d)$, and the image Lv of an element $v \in V$ is denoted the momentum of v. The momentum Lv might be singular and in fact $L\big(K(\cdot, y)a\big)(x)$ is the Dirac measure $\delta_y(x)a$. Considering K as the map $a \mapsto \int_\Omega K(\cdot, x)a(x)dx$, L can be viewed as the inverse of K. Confer [32] for a thorough introduction to reproducing kernels, especially with a view towards the LDDMM framework.

Instead of deriving the kernel from V, the opposite approach can be used: build V from a kernel, and hence impose the regularization in the framework from the kernel. With this approach, the kernel is often chosen to ensure rotational and translational invariance [32] and the scalar Gaussian kernel $K(x, y) = \exp(\frac{\|x-y\|^2}{\sigma^2})\mathrm{Id}_d$ is an often used choice. Confer [12] for details on the construction of V from Gaussian kernels.

2.2. Optimal Paths: The EPDiff Evolution Equations. The relation between norm and momentum lead to convenient equations for minimizers of the energy (2.1). In particular, the EPDiff equations for the evolution of the momentum a_t for optimal paths assert that if φ_t is a path minimizing $E_1(\varphi)$ with $\varphi_1 = \varphi$ minimizing $E(\varphi)$ and v_t is the derivative of φ_t then v_t satisfies the system

$$\begin{aligned} v_t &= \int_\Omega K(\cdot, x)a_t(x)dx \ , \\ \frac{d}{dt}a_t &= -Da_t v_t - a_t\nabla \cdot v_t - (Dv_t)^T a_t \end{aligned}$$

with Da_t and Dv_t denoting spatial differentiation of the momentum and velocity fields, respectively. The first equation connects the momentum a_t with the velocity v_t, and the second equation describes the time evolution of the momentum. In the most general form, the EPDiff equations describe the evolution of the momentum using the adjoint map. Following [32], we define $\mathrm{Ad}_\varphi v(x) = (D\varphi\, v) \circ \varphi^{-1}(x)$ for $v \in V$ and get a functional Ad^*_φ on the dual V^* of V by $(\mathrm{Ad}^*_\varphi \rho|v) = (\rho|\mathrm{Ad}_\varphi(v))$.[2] Define in addition $\mathrm{Ad}^T_\varphi v = K(\mathrm{Ad}^*_\varphi(Lv))$ which then satisfies $\left\langle \mathrm{Ad}^T_\varphi v, w \right\rangle = (\mathrm{Ad}^*_\varphi(Lv)|w)$, and let $\nabla_\varphi U$

[2] Here and in the following, we will use the notation $(p|v) := p(v)$ for evaluation of the functional $p \in V^*$ on the vector field $v \in V$.

denote the gradient of the similarity measure U with respect to the inner product on V so that $\langle \nabla_\varphi U, v\rangle_V = \partial_\epsilon U(\psi^v_{0\epsilon} \circ \varphi)$ for any variation $v \in V$ and diffeomorphism path $\psi^v_{0\epsilon}$ with derivative v. For optimal paths v_t, the EPDiff equations assert that $v_t = \mathrm{Ad}^T_{\varphi^v_{t1}} v_1$ with $v_1 = -\frac{1}{2}\nabla_{\varphi^v_{01}} U$ which leads to the conservation of momentum property for optimal paths. Conversely, the EPDiff equations reduce to simpler forms for certain objects. For landmarks $x_1, \ldots, x_N$, the momentum will be concentrated at point trajectories $x_{t,i} := \varphi_t(x_i)$ as Dirac measures $a_{t,i}\delta_{x_{t,i}}$ leading to the finite dimensional system of ODE's

$$\begin{aligned} v_t &= \sum_{l=1}^N K(\cdot, x_{t,l}))a_{t,l} \ , \quad \frac{d}{dt}\varphi_t(x_i) = v_t(x_{t,i}) \ , \\ \frac{d}{dt}a_{t,i} &= -\sum_{l=1}^N D_1 K(x_{t,i}, x_{t,l}) a^T_{t,i} a_{t,l} \ . \end{aligned} \tag{2.4}$$

3. Higher Order Kernels. We here introduce higher order kernels in the LDDMM registration framework. We start by motivating the construction by considering the approximation used when computing the similarity measure. We then link the kernels to the momentum using the derivative reproducing property, and derive the path energy. We consider locally affine transformations before deriving the EPDiff evolution equations for paths incorporating higher order kernels.

We will motivate the introduction of higher order kernels by considering a specific case of image registration: we take on the goal of using a control point formulation [11] when solving the registration problem (2.1) and hence aim for using a relatively sparse sampling of the velocity or momentum field. To achieve this, we will consider the coupling between the transported control points $\{\varphi^{-1}(x_1), \ldots, \varphi^{-1}(x_N)\}$ and the similarity measure in order to ensure the momentum stays singular and localized at the point trajectories while removing the need for warping the entire image at every iteration of the optimization process.

Considering a similarity measure $U(\varphi) = \int_\Omega F(I_m(\varphi^{-1}(x)), I_f(x))dx$ as discussed in the introduction, and a finite discretization $\tilde{U}^0(\varphi) = 1/N \sum_{i=1}^N F(\varphi.I_m(x_i), I_f(x_i))$ with a sparse set of control points $\{x_i\}$. While using $\tilde{U}^0(\varphi)$ to drive registration of the images will be very efficient in evaluating the warp in few points, it will suffer correspondingly from only using image information present in those points. Apart from not being robust under the presence of noise in the images, the discretization implies that local dilation or rotation around the points $\varphi^{-1}(x_i)$ cannot be detected: any variation $v \in V$ of φ keeping $\varphi^{-1}(x_i)$ constant for all $i = 1, \ldots, N$ will not change $\tilde{U}^0(\varphi)$. Formally, if $\psi_{0\epsilon}$ is a diffeomorphism path that is equal to φ at $t = 0$ and has derivative v at $t = 0$, i.e. $\partial_\epsilon \psi_{0\epsilon} = v$ and $\psi_{00} = \varphi$, then

$$\partial_\epsilon F(\psi_{0\epsilon}.I_m(x_i), I_f(x_i)) = \partial_1 F(\varphi.I_m(x_i), I_f(x_i)) \cdot \left(\nabla_{\varphi^{-1}(x_i)} I_m\right)^T v(\varphi^{-1}(x_i))$$

which vanishes if $v(\varphi^{-1}(x_i)) = 0$. Here $\partial_1 F$ denotes the derivative of $F : \mathbb{R}^2 \to \mathbb{R}$ with respect to the first variable.

A simple way to include more image in formation in the similarity measure is to convolve with a kernel K_s, and thus extend $\tilde{U}^0$ to

$$U^1(\varphi) = \frac{1}{N}\sum_{i=1}^N c_{K_s} \int_\Omega K_s(p + x_i, x_i) U(\varphi.I_m(p + x_i), I_f(p + x_i))dp$$

with c_{K_s} a normalization constant. If K_s is a box kernel, this amounts to a finer sampling of both the image and warp, and hence a finer discretization of the Riemann integral. The kernel K_s should not be confused with the RKHS kernel connected to the norm on V that is used when generating the V-gradient. A Gaussian kernel may be used for K_s [9], and more information on using smoothing kernels for intensity based image registration can be found in [9, 34].

The measure $U^1(\varphi)$ is problematic since a variation of φ would affect not only the point $\varphi^{-1}(x_i)$ but also $\varphi.I_m(p+x_i)$, and $U^1(\varphi)$ will therefore be dependent on $\varphi.I_m(p+x_i)$ for any p where $K_s(p,x_i)$ is non-zero. In this situation, the momentum is no longer concentrated in Dirac measures located at $\varphi_t^{-1}(x_i)$, and it will be necessary to increase the sampling of the warp. However, a first order expansion of φ^{-1} yields the approximation

$$\tilde{U}^1(\varphi) = \frac{1}{N}\sum_{i=1}^{N} c_{K_s} \int_\Omega K_s(p+x_i,x_i)U(I_m(D_{x_i}\varphi^{-1}p+\varphi^{-1}(x_i)), I_f(p+x_i))dp\ . \quad (3.1)$$

The measure $\tilde{U}^1(\varphi)$ is now again local depending only on $\varphi^{-1}(x_i)$ and the first order derivatives $D_{x_i}\varphi^{-1}$. It offers the stability provided by the convolution with K_s, and, importantly, variations v of φ keeping $\varphi^{-1}(x_i)$ constant but changing $D_{x_i}\varphi^{-1}$ do indeed affect the similarity measure. This implies that $\tilde{U}^1(\varphi)$ is able to catch rotations and dilations and drive the search for optimal φ accordingly. Please note the differences with the approach of Durrleman et al. [11]: when using $\tilde{U}^1(\varphi)$ as outlined here, the need for flowing the entire moving image forward is removed and the momentum field will stay singular *directly* thus removing the need for constraining the form of the velocity field.

This raises the question of how to represent variations of $D\varphi$ in the LDDMM framework. As we will see, higher order kernels appear as the natural choice of deformation atoms allowing singular momentum for variations of $D\varphi$ and hence keeping the benefits of the finite control point formulation.

3.1. Derivative Reproducing Property. Recall the reproducing property of the RKHS structure, i.e. $\langle K(\cdot,x)a, v\rangle_V = a \otimes \delta_x(v)$ for $v \in V$, $x \in \Omega$ and $a \in \mathbb{R}^d$. Zhou [33] shows that this property holds not only for the kernel but also for its partial derivatives. Letting $D_x^\alpha v$ denote the derivative of $v \in V$ at $x \in \Omega$ with respect to the multi-index α,

$$D_x^\alpha v = \frac{\partial^{|\alpha|}}{\partial_{x^1}^{\alpha_1}\dots\partial_{x^q}^{\alpha_q}} v(x)$$

and defining $(D_x^\alpha K a)(y) = D_x^\alpha(K(\cdot,y)a)$ for $a \in \mathbb{R}^d$, Zhou proves that $D_x^\alpha K a \in V$ and that the *partial derivative reproducing property*

$$\langle D_x^\alpha K a, v\rangle_V = a^T D_x^\alpha(v) \quad (3.2)$$

holds when the maps in V are sufficiently smooth for the derivatives to exist. In the following, we denote the matrices $D_x^\alpha K$ *higher order kernels*. Similarly, we denote the maps $a \otimes D_x^\alpha : V \to \mathbb{R}$ defined by $a \otimes D_x^\alpha(v) := a^T D_x^\alpha v$ *higher order Diracs*. It follows that

$$a \otimes D_x^\alpha = \left(v \mapsto \langle D_x^\alpha K a, v\rangle_V\right) \in V^*\ .$$

As a consequence of Zhous result, we can derive the momentum for the higher order kernels. Recall that the momentum map $L : V \to V^*$ satisfies $\langle Lv, w\rangle_{L^2} = \langle v, w\rangle_V$. With the higher order kernels,

$$\langle LD_x^\alpha Ka, v\rangle_{L_2} = \langle D_x^\alpha Ka, v\rangle_V = a \otimes D_x^\alpha(v) = \langle a \otimes D_x^\alpha, v\rangle_{L^2} \ .$$

Thus $LD_x^\alpha Ka = a \otimes D_x^\alpha$ or, shorter, $LD_x^\alpha K = D_x^\alpha$. That is, the higher order kernels and higher order Diracs corresponds just as the kernels and Diracs in the usual RKHS sense.

Consider a map on diffeomorphisms $U : G_V \to \mathbb{R}$ e.g. an image similarity measure dependent on φ. In a finite dimensional setting with N evaluation points x_i, U would decompose as $U(\varphi) = P \circ Q(\varphi)$ with $Q(\varphi) = (\varphi(x_1), \ldots, \varphi(x_N))$ and $P : \mathbb{R}^{dN} \to \mathbb{R}$. Introducing higher order kernels, we let $Q(\varphi) = (D_{x_1}^{\alpha_1}(\varphi), \ldots, D_{x_N}^{\alpha_J}(\varphi))$ with J multi-indices α_j, and decompose U as $U(\varphi) = P \circ Q(\varphi)$ with $P : \mathbb{R}^{dNJ} \to \mathbb{R}$. We allow α_j to be empty and hence incorporate the standard zero-order case. The partial derivative reproducing property now allows to compute the V-gradient of U as a sum of higher order kernels.

PROPOSITION 3.1. *Let $\nabla^{ij} P$ denote the gradient with respect to the variable indexed by $D_{x_i}^{\alpha_j}(\varphi)$ in the expression for Q. Then the gradient $\nabla_\varphi U \in V$ of U with respect to the inner product in V is given by $\nabla_\varphi U = \sum_{i=1}^N \sum_{j=1}^J D_{x_i}^{\alpha_j} K \nabla^{ij}_{Q(\varphi)} P$.*

Proof. The gradient $\nabla_\varphi U$ at φ is defined by $\langle \nabla_\varphi U, v\rangle = \partial_\epsilon U(\epsilon v + \varphi)$ for all variations $v \in V$. For such v, we get using (3.2) that

$$\begin{aligned}\partial_\epsilon U(\epsilon v + \varphi) &= \partial_\epsilon P \circ Q(\epsilon v + \varphi) = \partial_\epsilon P(D_{x_i}^{\alpha^j}(\epsilon v + \varphi)) = \partial_\epsilon P(\epsilon D_{x_i}^{\alpha^j} v + D_{x_i}^{\alpha^j}\varphi) \\ &= \sum_{i=1}^N \sum_{j=1}^J (\nabla^{ij}_{Q(\varphi)} P)^T D_{x_i}^{\alpha^j} v = \left\langle \sum_{i=1}^N \sum_{j=1}^J D_{x_i}^{\alpha^j} \nabla^{ij}_{Q(\varphi)} P, v \right\rangle_V \ .\end{aligned}$$

□

3.2. Momentum and Energy. As a result of Proposition 3.1, the momentum of the gradient of U is $L\nabla_\varphi U = \sum_{i=1}^N \sum_{j=1}^J \nabla^{ij}_{Q(\varphi)} P \otimes D_{x_i}^{\alpha_j}$. In general, if $v \in V$ is a sum of higher order kernels, the energy $\|v\|_V^2$ can be computed using (3.2) as a sum of the different order kernels evaluated at the points x_i. To keep the notation brief, we restrict to sums of zero- and first order kernels in the following. If $v(\cdot) = \sum_{i=1}^N \left(K(x_i, \cdot)a_i + \sum_{j=1}^d D^j K(x_i, \cdot) a_i^j\right)$, we get the energy

$$\begin{aligned}\|v\|_V^2 &= \left\langle \sum_{i=1}^N \left(K(x_i,\cdot)a_i + \sum_{j=1}^d D^j K(x_i,\cdot)a_i^j\right), \sum_{i=1}^N \left(K(x_i,\cdot)a_i + \sum_{j=1}^d D^j K(x_i,\cdot)a_i^j\right) \right\rangle_V \\ &= \sum_{i,l=1}^N \langle K(x_l,\cdot)a_l, K(x_i,\cdot)a_i\rangle_V + \sum_{i,l=1}^N \sum_{j,j'}^d \left\langle D^j K(x_l,\cdot)a_l^j, D^{j'} K(x_i,\cdot)a_i^{j'} \right\rangle_V \\ &\quad + 2\sum_{i,l=1}^N \sum_{j=1}^d \left\langle D^j K(x_l,\cdot)a_l^j, K(x_i,\cdot)a_i \right\rangle_V \\ &= \sum_{i,l=1}^N a_l^T K(x_l,x_i)a_i + \sum_{i,l=1}^N \sum_{j,j'}^d a_i^{j',T} D_2^{j'} D_1^j K(x_l,x_i) a_l^j + 2\sum_{i,l=1}^N \sum_{j=1}^d a_i^T D_1^j K(x_l,x_i) a_l^j\end{aligned} \tag{3.3}$$

with $D_i^j K(\cdot,\cdot)$ denoting differentiation with the respect to the ith variable, $i = 1, 2$, and jth coordinate, $j = 1, \ldots, d$. For scalar symmetric kernels such as Gaussians, this expression reduces to

$$\|v\|_V^2 = \sum_{i,l=1}^{N} a_l^T K(x_l, x_i) a_i + \sum_{i,l=1}^{N} \sum_{j,j'}^{d} \left(D_2 \nabla_1 K(x_l, x_i)\right)_j^{j'} a_i^{j',T} a_l^j + 2 \sum_{i,l=1}^{N} \sum_{j=1}^{d} (\nabla_1 K(x_l, x_i))^j a_i^T a_l^j \ .$$

3.3. Locally Affine Transformations. The Polyaffine and Log-Euclidean Polyaffine [3, 1] frameworks model locally affine transformations using matrix logarithms which has limited range. Though the higher order kernels can be seen as the LDDMM sibling of the Polyaffine methods, the methods differ in that diffeomorphism paths generated by higher order kernels, in particular, kernels of zero- and first order, can locally approximate all affine transformation with linear component having positive determinant. The approximation will depend only on how fast the kernel approaches zero towards infinity. The manifold structure of G_V provides this result immediately. Indeed, let $\varphi(x) = Ax + b$ be an affine transformation with $\det(A) > 0$. We define a path φ_t of finite energy such that $\varphi_1 \approx \varphi$ which shows that $\varphi_1 \in G_V$ and can be reached in the framework. The matrices of positive determinant is path connected so we can let ψ_t be a path from Id_d to A and define $\tilde{\psi}_t(x) = \psi_t x + bt$. Then with $\tilde{v}_t(x) = (\partial_t \psi_t)\tilde{\psi}_t^{-1}(x) + b$, we have $\partial_t \tilde{\psi}_t(x) = (\partial_t \psi_t)x + b = \tilde{v}_t \circ \tilde{\psi}_t(x)$ and

$$x + \int_0^1 \tilde{v}_t \circ \tilde{\psi}_t(x) dt = x + \int_0^1 (\partial_t \psi_t)x + b dt = \varphi(x) \ .$$

Now use that $(\partial_t \psi_t)\tilde{\psi}_t^{-1}(x) = (\partial_t \psi_t)(\psi_t)^{-1}(x - bt)$ and let the $M_t = (m_{1,t} \ldots m_{d,t})$ be the t-dependent matrix $(\partial_t \psi_t)(\psi_t)^{-1}$ so that the first term of $\tilde{v}_t(x)$ equals $M_t(x - bt)$. Then choose a radial kernel, e.g. a Gaussian K_σ, and define the approximation v_t of $\tilde{v}_t$ by

$$v_t(x) = \sum_{j=1}^{d} D^j_{\tilde{\psi}_t(0)} K_\sigma(x) m_{j,t} + K_\sigma(\tilde{\psi}_t(0), x) b \ . \tag{3.4}$$

The path φ_{01}^v generated by v_t then has finite energy, and

$$\varphi_{01}^v(x) = x + \int_0^1 v_t \circ \varphi_{0t}^v(x) dt \approx \varphi(x)$$

with the approximation depending only on the kernel scale σ. Note that the affine transformations with linear components having negative determinant can in a similar way be reached by starting the integrating at a diffeomorphism with negative Jacobian determinant.

In the experiments section, we will illustrate the locally affine transformations encoded by zero and first order kernels, and, therefore, it will be useful to introduce a notation for these kernels. We encode the translational part of either the momentum or velocity using the notation

$$\mathrm{Tsl}_x(b) = K_\sigma(x, \cdot) b$$

and the linear part by

$$\mathrm{Lin}_x(M) = \sum_{j=1}^{d} D_x^j K_\sigma(\cdot) m_j$$

with m_1, m_j being the columns of the matrix M. Equation (3.4) can then be written

$$v_t(x) = \mathrm{Lin}_{\tilde{\psi}_t(0)}(M_t) + \mathrm{Tsl}_{\tilde{\psi}_t(0)}(b) \ . \tag{3.5}$$

We emphasize that though we mainly focus on zero and first order kernels, the mathematical construction allows any order kernel permitted by the smoothness of the kernel at order zero.

3.4. EPDiff Equations. It is important to note that the higher order kernels offer a convenient representation for the gradients of maps U incorporating derivative information but since the kernels are members of V and their momentum in the dual V^*, the analytical of structure of LDDMM is not changed. In particular, the adjoint form of the EPDiff equations, i.e. that optimal paths v_t satisfy $v_t = \mathrm{Ad}^T_{\varphi^v_{t1}} v_1$ with $v_1 = -\frac{1}{2}\nabla_{\varphi^v_{01}} U$, is still valid. The momentum $\rho_1 = Lv_1$ is transported to the momentum ρ_t by $\mathrm{Ad}^*_{\varphi^v_{t1}} p_1$. Because

$$(\rho_t|w) = (\rho_1|\mathrm{Ad}_{\varphi^v_{t1}}(w)) = (\rho_1|(D\varphi^v_{t1}\, w) \circ (\varphi^v_{t1})^{-1}) \ ,$$

if ρ_1 is a sum of higher order kernels, ρ_t will be sum of higher order kernels for all t. However, since the time evolution of $(\rho_t|w)$ with the above relation involves derivatives of $D\varphi^v_{t1}$, this form is inconvenient for computing ρ_t. Instead, we make use of the Hamiltonian form of the EPDiff equations [32, P. 265]. Here, the momentum ρ_t is pulled back to ρ_0 but with a coordinate change of the evaluation vector field: the Hamiltonian form μ_t is defined by $\big(\mu_t\big|w\big) := \big(\rho_0\big|(D\varphi^v_{0t})^{-1}(y)w(y)\big)_y$ where the subscript stresses that $(D\varphi^v_{0t})^{-1}(y)w(y)$ is evaluated as a y-dependent vector field. Using this notation, the evolution equations become

$$\begin{aligned} \partial_t \varphi^v_{0t}(y) &= \sum_{k=1}^{d} \big(\mu_t\big|K^k(\varphi^v_{0t}(x), \varphi^v_{0t}(y))\big)_x e_k \\ \big(\partial_t \mu_t\big|w\big) &= -\sum_{k=1}^{d} \big(\mu_t\big|\big(\mu_t\big|D_2 K^k(\varphi^v_{0t}(x), \varphi^v_{0t}(y))w(y)\big)_x e_k\big)_y \ . \end{aligned} \tag{3.6}$$

For the case when $(\rho_0|w)$ does not involve derivatives of w, these equations form an ordinary differential equation describing the evolution of the path and momentum [32]. For the higher order case, we will need to incorporate additional information in the system.

Again we restrict to the zero- and first order case, and we hence work with initial momenta on the form

$$\rho_0 = \sum_{i=1}^{N} a_{0,i} \otimes \delta_{x_{0,i}} + \sum_{i=1}^{N}\sum_{j=1}^{d} a^j_{0,i} \otimes D^j \delta_{x_{0,i}} \tag{3.7}$$

with $x_{t,i}$ as usual denoting the point positions $\varphi^v_{0t}(x_i)$ at time t. Then

$$\begin{aligned}
\big(\mu_t\big|w\big) &= \big(\rho_0\big|D\varphi^v_{0t}(y)^{-1}w(y)\big)_y = \int_\Omega \Big(\sum_{i=1}^N a_{0,i}\otimes\delta_{x_{0,i}} + \sum_{i=1}^N\sum_{j=1}^d a^j_{0,i}\otimes D^j\delta_{x_{0,i}}\Big)D\varphi^v_{0t}(y)^{-1}w(y)dy \\
&= \sum_{i=1}^N \big(a_{0,i}\otimes\delta_{x_{0,i}}\big|D\varphi^v_{0t}(y)^{-1}w(y)\big)_y + \sum_{j=1}^d \big(a^j_{0,i}\otimes\delta_{x_{0,i}}\big|\big(D^jD\varphi^v_{0t}(y)^{-1})w(y)\big)_y \\
&\quad + \sum_{i=1}^N\sum_{j=1}^d \big(D\varphi^v_{0t}(x_{0,i})^{-1,T}a^j_{0,i}\otimes D^j\delta_{x_{0,i}}\big|w\big) \\
&= \sum_{i=1}^N \big(\big(D\varphi^v_{0t}(x_{0,i})^{-1,T}a_{0,i} + \sum_{j=1}^d \big(D^jD\varphi^v_{0t}(x_{0,i})^{-1}\big)^T a^j_{0,i}\big)\otimes\delta_{x_{0,i}}\big|w\big) \\
&\quad + \sum_{i=1}^N\sum_{j=1}^d \big(D\varphi^v_{0t}(x_{0,i})^{-1,T}a^j_{0,i}\otimes D^j\delta_{x_{0,i}}\big|w\big)
\end{aligned}$$

showing that $\mu_t = \sum_{i=1}^N \mu_{t,i}\otimes\delta_{x_{0,i}} + \sum_{i=1}^N\sum_{j=1}^d \mu^j_{t,i}\otimes D^j\delta_{x_{0,i}}$ with

$$\begin{aligned}
\mu_{t,i} &= D\varphi^v_{0t}(x_{0,i})^{-1,T}a_{0,i} + \sum_{j=1}^d \big(D^jD\varphi^v_{0t}(x_{0,i})^{-1}\big)^T a^j_{0,i} \\
\mu^j_{t,i} &= D\varphi^v_{0t}(x_{0,i})^{-1,T}a^j_{0,i}\ .
\end{aligned} \tag{3.8}$$

The momentum ρ_t can the be recovered as

$$\begin{aligned}
\big(\rho_t\big|w\big) &= \big(\mu_t\big|w\circ\varphi^v_{0t}\big) = \big(\sum_{i=1}^N \mu_{t,i}\otimes\delta_{x_{0,i}} + \sum_{i=1}^N\sum_{j=1}^d \mu^j_{t,i}\otimes D^j\delta_{x_{0,i}}\big)w\circ\varphi^v_{0t} \\
&= \sum_{i=1}^N \mu_{t,i}\otimes\delta_{x_{t,i}}w + \sum_{i=1}^N\sum_{j=1}^d \mu^{j,T}_{t,i}Dw(D^j\varphi^v_{0t})(x_{0,i}) \\
&= \sum_{i=1}^N \mu_{t,i}\otimes\delta_{x_{t,i}}w + \sum_{i=1}^N\sum_{j=1}^d \big(\sum_{k=1}^d (D^k\varphi^v_{0t})(x_{0,i})^j\mu^k_{t,i}\big)\otimes D^j\delta_{x_{t,i}}w
\end{aligned}$$

and hence the coefficients of the momentum $a_{t,i}$ and $a^j_{t,i}$ (confer (3.7)) are given by $a_{t,i} = \mu_{t,i}$ and $a^j_{t,i} = \sum_{k=1}^d (D^k\varphi^v_{0t})(x_{0,i})^j\mu^k_{t,i}$.

3.5. Time Evolution of the EPDiff Equations. Even though $\mu_{t,i}$ in (3.8) depend on the second order derivative of φ, we will show that the complete evolution in the zero- and first order case can be determined by solving for $\varphi^v_{0t}(x_{i,0})$, $D\varphi^v_{0t}(x_{i,0})$, and $\mu_{t,i}$. This will provide the computational representation we will use when implementing the systems. In order to simplify the notation, we will work mainly with scalar kernels so that $K^k_l(x,y) = K(x,y)$ if and only if $k = l$ and 0 otherwise.

Using (3.6), φ^v_{0t} evolves according to

$$\begin{aligned}
\partial_t\varphi^v_{0t}(y) &= \sum_{k=1}^d \int_\Omega \sum_{i=1}^N \big(\mu^T_{t,i}\otimes\delta_{x_{0,i}} + \sum_{j=1}^d \mu^j_{t,i}\otimes D^j\delta_{x_{0,i}}\big)K^k(\varphi^v_{0t}(x),\varphi^v_{0t}(y))dxe_k \\
&= \sum_{k=1}^d\sum_{i=1}^N \big(\mu^T_{t,i}K^k(\varphi^v_{0t}(x_{0,i}),\varphi^v_{0t}(y)) + \sum_{j=1}^d \mu^{j,T}_{t,i}D_1K^k(\varphi^v_{0t}(x_{0,i}),\varphi^v_{0t}(y))D^j\varphi^v_{0t}(x_{0,i})\big)e_k\ .
\end{aligned}$$

With scalar kernels, the trajectories $x_{t,i}$ are given by

$$\partial_t \varphi^v_{0t}(x_{0,n}) = \sum_{i=1}^{N} \left(K(\varphi^v_{0t}(x_{0,i}), \varphi^v_{0t}(x_{0,n}))\mu_{t,i} + \sum_{j=1}^{d} \nabla_1 K(\varphi^v_{0t}(x_{0,i}), \varphi^v_{0t}(x_{0,n}))^T D^j \varphi^v_{0t}(x_{0,i}) \mu^j_{t,i} \right) .$$

It is shown in [32] that the evolution of $D\varphi^v_{0t}(x_{i,0})$ is given by

$$\partial_t D\varphi^v_{0t}(y)a = \sum_{k=1}^{d} \left(\mu_t \middle| D_2 K^k(\varphi^v_{0t}(x), \varphi^v_{0t}(y)) D\varphi^v_{0t}(y)a \right)_x e_k .$$

Inserting the Hamiltonian form of the higher order momentum, each component (l,k) of the matrix $D\varphi^v_{0t}(y)$ thus evolves according to

$$\begin{aligned}
\partial_t D\varphi^v_{0t}(y)^l_k &= \left(\mu_t \middle| D_2 K^k(\varphi^v_{0t}(x), \varphi^v_{0t}(y)) D\varphi^v_{0t}(y)e_l \right)_x \\
&= \int_\Omega \sum_{i=1}^{N} \left(\mu_{t,i} \otimes \delta_{x_{0,i}} + \sum_{j=1}^{d} \mu^j_{t,i} \otimes D^j \delta_{x_{0,i}} \right) D_2 K^k(\varphi^v_{0t}(x), \varphi^v_{0t}(y)) D\varphi^v_{0t}(y)e_l dx \\
&= \sum_{i=1}^{N} \mu^T_{t,i} D_2 K^k(\varphi^v_{0t}(x_{0,i}), \varphi^v_{0t}(x_{0,n})) D\varphi^v_{0t}(x_{0,n})e_l \\
&\quad + \sum_{i=1}^{N} \sum_{j=1}^{d} \mu^{j,T}_{t,i} \Big(\sum_{m=1}^{d} \left(D^m_1 D_2 K^k(\varphi^v_{0t}(x_{0,i}), \varphi^v_{0t}(x_{0,n})) \right) \left(D^j \varphi^v_{0t}(x_{0,i}) \right)^m \Big) D\varphi^v_{0t}(x_{0,n})e_l .
\end{aligned}$$

With scalar kernels, the evolution at the trajectories is then

$$\begin{aligned}
\partial_t D\varphi^v_{0t}(x_{0,n})^l = \sum_{i=1}^{N} \Big(& \nabla_2 K(\varphi^v_{0t}(x_{0,i}), \varphi^v_{0t}(x_{0,n}))^T D^l \varphi^v_{0t}(x_{0,n}) \mu_{t,i} \\
& + \sum_{j=1}^{d} \left(D_1 \nabla_2 K(\varphi^v_{0t}(x_{0,i}), \varphi^v_{0t}(x_{0,n})) D^j \varphi^v_{0t}(x_{0,i}) \right)^T D^l \varphi^v_{0t}(x_{0,n}) \mu^j_{t,i} \Big) .
\end{aligned}$$

The complete derivation of the evolution of μ_t is notationally heavy and can be found in Appendix A. Combining this derivation with the expressions above, we arrive at the following result:

PROPOSITION 3.2. *The EPDiff equations in the scalar case with zero- and first order kernels are given in Hamiltonian form by the system*

$$\begin{aligned}
\partial_t \varphi^v_{0t}(x_{0,n}) &= \sum_{i=1}^{N} \left(K(x_{t,i}, x_{t,n})\mu_{t,i} + \sum_{j=1}^{d} \nabla_1 K(x_{t,i}, x_{t,n})^T D^j \varphi^v_{0t}(x_{0,i}) \mu^j_{t,i} \right) \\
\partial_t D\varphi^v_{0t}(x_{0,n})^l &= \sum_{i=1}^{N} \Big(\nabla_2 K(x_{t,i}, x_{t,n})^T D^l \varphi^v_{0t}(x_{0,n}) \mu_{t,i} \\
&\qquad + \sum_{j=1}^{d} \left(D_1 \nabla_2 K(x_{t,i}, x_{t,n}) D^j \varphi^v_{0t}(x_{0,i}) \right)^T D^l \varphi^v_{0t}(x_{0,n}) \mu^j_{t,i} \Big)
\end{aligned} \tag{3.9}$$

$$
\begin{aligned}
\partial_t \mu_{t,n} = &- \sum_{i=1}^{N} \Big((\mu_{t,n}^T \mu_{t,i}) \nabla_2 K(x_{t,i}, x_{t,n}) \\
&+ \sum_{j=1}^{d} \left(\mu_{t,n}^{j,T} \mu_{t,i} - \mu_{t,n}^T \mu_{t,i}^j\right) D_2 \nabla_2 K(x_{t,i}, x_{t,n}) D^j \varphi_{0t}^v(x_{0,n}) \\
&+ \sum_{j,j'=1}^{d} \left(\mu_{t,n}^{j',T} \mu_{t,i}^j\right) D_2 \big(D_1 \nabla_2 K(x_{t,i}, x_{t,n}) D^j \varphi_{0t}^v(x_{0,i})\big) D^{j'} \varphi_{0t}^v(x_{0,n}) \Big) \\
\mu_{t,i}^j = &\, D\varphi_{0t}^v(x_{0,i})^{-1,T} a_{0,i}^j \ .
\end{aligned}
$$

Note that both $x_{1,i} = \varphi_{01}^v(x_{0,i})$ and $D\varphi_{01}^v(x_{0,i})$ are provided by the equation, and hence can be used to evaluate a similarity measure such as $\tilde{U}^1$ which depend on these entities. As in the zero-order case, the entire evolution can be recovered by the initial conditions for the momentum.

4. Variations of the Initial Conditions. There exists various choices of optimization algorithms for LDDMM registration. Roughly, they can be divided into two groups based on whether they represent the initial momentum/velocity or the entire path φ_t. Here, we take the approach of incorporating higher order kernels with the shooting method of e.g. Vaillant et al. [28]. The algorithm will take a guess for initial momentum, integrate the EPDiff equations forward, compute the similarity measure gradient ∇U, and flow the gradient backwards to provide an improved guess. For this to work, we will need the variation of the EPDiff equations when varying the initial conditions. Following this, we discuss the backwards gradient transport and arrive at a full matching algorithm.

A variation $\delta\rho_0$ of the initial momentum will induce a variation of the system (3.9). By differentiating the system, we get the time evolution of the variation. To ease notation, we assume the scalar kernel has the form $K(x,y) = \gamma(|x-y|^2)$ and write $\gamma_{t,in} = K(x_{t,i}, x_{t,n})$. Variations of the kernel and kernel derivatives such as the entity $\delta\nabla_1 K(x_{t,i}, x_{t,n})$ below depend only on the variation of point trajectories $\delta x_{t,i}$. The full expressions for these parts are provided in Appendix B. The evolution of the derived system then takes the following form:

$$
\begin{aligned}
\partial_t \delta\varphi_{0t}^v(x_{0,n}) = &\sum_{i=1}^{N} \left(\delta K(x_{t,i}, x_{t,n}) \mu_{t,i} + \gamma_{t,in} \delta\mu_{t,i}\right) \\
&+ \sum_{i=1}^{N} \sum_{j=1}^{d} \big(\delta\nabla_1 K(x_{t,i}, x_{t,n})^T D^j \varphi_{0t}^v(x_{0,i}) \mu_{t,i}^j + \nabla_1 K(x_{t,i}, x_{t,n})^T \delta D^j \varphi_{0t}^v(x_{0,i}) \mu_{t,i}^j \\
&\quad + \nabla_1 K(x_{t,i}, x_{t,n})^T D^j \varphi_{0t}^v(x_{0,i}) \delta\mu_{t,i}^j\big)
\end{aligned}
\tag{4.1}
$$

$$
\begin{aligned}
\partial_t \delta D\varphi^v_{0t}(x_{0,n})^l = &\sum_{i=1}^N \big(\delta\nabla_2 K(x_{t,i},x_{t,n})^T D^l\varphi^v_{0t}(x_{0,n})\mu_{t,i} + \nabla_2 K(x_{t,i},x_{t,n})^T \delta D^l\varphi^v_{0t}(x_{0,n})\mu_{t,i} \\
&+ \nabla_2 K(x_{t,i},x_{t,n})^T D^l\varphi^v_{0t}(x_{0,n})\delta\mu_{t,i}\big) \\
&+ \sum_{i=1}^N\sum_{j=1}^d \Big(\big(\delta D_1\nabla_2 K(x_{t,i},x_{t,n})D^j\varphi^v_{0t}(x_{0,i})\big)^T D^l\varphi^v_{0t}(x_{0,n})\mu^j_{t,i} \\
&\quad + \big(D_1\nabla_2 K(x_{t,i},x_{t,n})\delta D^j\varphi^v_{0t}(x_{0,i})\big)^T D^l\varphi^v_{0t}(x_{0,n})\mu^j_{t,i} \\
&\quad + \big(D_1\nabla_2 K(x_{t,i},x_{t,n})D^j\varphi^v_{0t}(x_{0,i})\big)^T \delta D^l\varphi^v_{0t}(x_{0,n})\mu^j_{t,i} \\
&\quad + \big(D_1\nabla_2 K(x_{t,i},x_{t,n})D^j\varphi^v_{0t}(x_{0,i})\big)^T D^l\varphi^v_{0t}(x_{0,n})\delta\mu^j_{t,i}\Big) \\
\partial_t \delta\mu_{t,n} = &-\sum_{i=1}^N \big(\big(\delta\mu^T_{t,n}\mu_{t,i} + \mu^T_{t,n}\delta\mu_{t,i}\big)\nabla_2 K(x_{t,i},x_{t,n}) + \big(\mu^T_{t,n}\mu_{t,i}\big)\delta\nabla_2 K(x_{t,i},x_{t,n})\big) \\
&- \sum_{i=1}^N\sum_{j=1}^d \Big(\big(\delta\mu^{j,T}_{t,n}\mu_{t,i} + \mu^{j,T}_{t,n}\delta\mu_{t,i} - \delta\mu^T_{t,n}\mu^j_{t,i} - \mu^T_{t,n}\delta\mu^j_{t,i}\big)D_2\nabla_2 K(x_{t,i},x_{t,n})D^j\varphi^v_{0t}(x_{0,n}) \\
&\quad + \big(\mu^{j,T}_{t,n}\mu_{t,i} - \mu^T_{t,n}\mu^j_{t,i}\big)\delta D_2\nabla_2 K(x_{t,i},x_{t,n})D^j\varphi^v_{0t}(x_{0,n}) \\
&\quad + \big(\mu^{j,T}_{t,n}\mu_{t,i} - \mu^T_{t,n}\mu^j_{t,i}\big)D_2\nabla_2 K(x_{t,i},x_{t,n})\delta D^j\varphi^v_{0t}(x_{0,n})\big) \\
&- \sum_{i=1}^N\sum_{j,j'=1}^d \big(\big(\delta\mu^{j',T}_{t,n}\mu^j_{t,i} + \mu^{j',T}_{t,n}\delta\mu^j_{t,i}\big)D_2\big(D_1\nabla_2 K(x_{t,i},x_{t,n})D^j\varphi^v_{0t}(x_{0,i})\big)D^{j'}\varphi^v_{0t}(x_{0,n}) \\
&\quad + \big(\mu^{j',T}_{t,n}\mu^j_{t,i}\big)\delta D_2\big(D_1\nabla_2 K(x_{t,i},x_{t,n})D^j\varphi^v_{0t}(x_{0,i})\big)D^{j'}\varphi^v_{0t}(x_{0,n}) \\
&\quad + \big(\mu^{j',T}_{t,n}\mu^j_{t,i}\big)D_2\big(D_1\nabla_2 K(x_{t,i},x_{t,n})D^j\varphi^v_{0t}(x_{0,i})\big)\delta D^{j'}\varphi^v_{0t}(x_{0,n})\big)\ .
\end{aligned}
$$

The variation of $\mu^j_{t,i}$ is available as

$$
\delta\mu^j_{t,i} = -\big(D\varphi^v_{0t}(x_{0,i})^{-1}\delta D\varphi^v_{0t}(x_{0,i})D\varphi^v_{0t}(x_{0,i})^{-1}\big)^T a^j_{0,i} + D\varphi^v_{0t}(x_{0,i})^{-1,T}\delta a^j_{0,i}\ .
$$

However, when computing the backwards transport, we will need to remove the dependency on $\delta a^j_{0,i}$ which is only available for forward integration. Instead, by writing the evolution of $\mu^j_{t,i}$ in the form

$$
\begin{aligned}
\partial_t\mu^j_{t,i} &= \partial_t D\varphi^v_{0t}(x_{0,i})^{-1,T}a^j_{0,i} = -\big(D\varphi^v_{0t}(x_{0,i})^{-1}\partial_t D\varphi^v_{0t}(x_{0,i})D\varphi^v_{0t}(x_{0,i})^{-1}\big)^T a^j_{0,i} \\
&= -D\varphi^v_{0t}(x_{0,i})^{-1,T}\partial_t D\varphi^v_{0t}(x_{0,i})^T\mu^j_{t,i}\ ,
\end{aligned}
$$

we get the variation

$$
\begin{aligned}
\partial_t\delta\mu^j_{t,n} = &-\delta D\varphi^v_{0t}(x_{0,n})^{-1,T}\partial_t D\varphi^v_{0t}(x_{0,n})^T\mu^j_{t,n} - D\varphi^v_{0t}(x_{0,n})^{-1,T}\partial_t\delta D\varphi^v_{0t}(x_{0,n})^T\mu^j_{t,n} \\
&- D\varphi^v_{0t}(x_{0,n})^{-1,T}\partial_t D\varphi^v_{0t}(x_{0,n})^T\delta\mu^j_{t,n}\ .
\end{aligned}
$$

4.1. Backwards Transport. The correspondence between initial momentum ρ_0 and end diffeomorphism φ^v_{01} asserted by the EPDiff equations allows us to view the similarity measure $U(\varphi^v_{01})$ as a function of ρ_0. Let A denote the result of integrating the system for the variation of the initial conditions from $t = 0$ to $t = 1$ such that $w = A\delta\rho_0 \in V$ for a variation $\delta\rho_0$. We then get a corresponding variation δU in the

similarity measure. To compute the gradient of U as a function of ρ_0, we have

$$\delta U(\varphi_{01}^v) = \left\langle \nabla_{\varphi_{01}^v} U, w \right\rangle_V = \left\langle \nabla_{\varphi_{01}^v} U, A\delta\rho_0 \right\rangle_V = \left\langle A^T \nabla_{\varphi_{01}^v} U, \delta\rho_0 \right\rangle_{V^*} \ .$$

Thus, the V^*-gradient of $\nabla_{\rho_0} U$ is given by $A^T \nabla_{\varphi_{01}^v} U$. The gradient can equivalently be computed in momentum space at both endpoints of the diffeomorphism path using the map P defined in Proposition 3.1.

The complete system for the variation of the initial conditions is a linear ODE, and, therefore, there exists a time-dependent matrix M_t such that the ODE

$$\partial_t y_t = M_t y_t$$

has the variation as a solution y_t. It is shown in [32] that, in such cases, solving the backwards transpose system

$$\partial_t w_t = -M_t^T w_t \tag{4.2}$$

from $t = 1$ to $t = 0$ provides the value of $A^T w$. Therefore, we can obtain $\nabla_{\rho_0} U$ by solving the transpose system backwards. The components of M_t can be identified by writing the evolution equations for the variation in matrix form. This provides M_t^T and allows the backwards integration of the system 4.2. The components of the transpose matrix M_t are provided in Appendix C.

4.2. Algorithm. The registration problem (2.1) consists of both the similarity measure U and the minimal path energy E_1. For e.g. landmark based registration, $U(\varphi)$ is most often expressed in terms of φ directly whether as U is usually dependent on the inverse φ^{-1} for image registration. In the first case, the gradient $\nabla_\varphi U$ is known, and, given the initial momentum ρ_0, we can obtain the gradient $\nabla_{\rho_0} U$ for a gradient descent based optimisation procedure from the backwards transport equations (4.2) discussed above. For the energy part, it is a fundamental result property of critical paths in the LDDMM framework that the energy stays constant along the path. Thus, $\int_0^1 \|v_t\|_V^2 dt = \|v_0\|_V^2 = \left(\rho_0 \middle| K(\rho_0)\right)^2$ and we can easily compute the gradient from (3.3). Given this, the zero-order matching algorithm in the initial momentum is generalized to zero- and first order kernels in Algorithm 1.

Algorithm 1 Matching with Higher Order Kernels.

$\rho_0 \leftarrow$ initial guess
repeat
  Solve EPDiff equations forward
  Compute U and ∇P
  Solve backwards the transpose equations
  Compute the energy gradient $\nabla \|v_0\|^2$
  Update ρ_0 from $\nabla \|v_0\|^2 + \nabla_{\rho_0} U$
until convergence

Traditionally, the similarity measure $U(\varphi)$ is in image matching formulated using the inverse of φ, and this approach was taken when formulating the approximation (3.1). For this reason, at finite control point formulation is naturally expressed using a sampling $\{x_1, \ldots, x_N\}$ in the *target* image with the algorithm optimizing for the momentum ρ_1 at time $t = 1$. The evaluation points $\varphi^{-1}(x_i)$ are then generated by flowing *backwards* from $t = 1$ to $t = 0$, the gradient of $U(\varphi)$ can then be computed

in $\varphi^{-1}(x_i)$ and flowed *forwards* to update ρ_1. This corresponds to switching the role of the moving and target image combined with backwards integration of the flow equations. Algorithm 1 will accommodate this situation by just reversing the integration directions. The control points can be chosen either at e.g. anatomically important locations, at random, or on a regular grid. In the experiments, we will register expanding ventricles using control points placed in the ventricles.

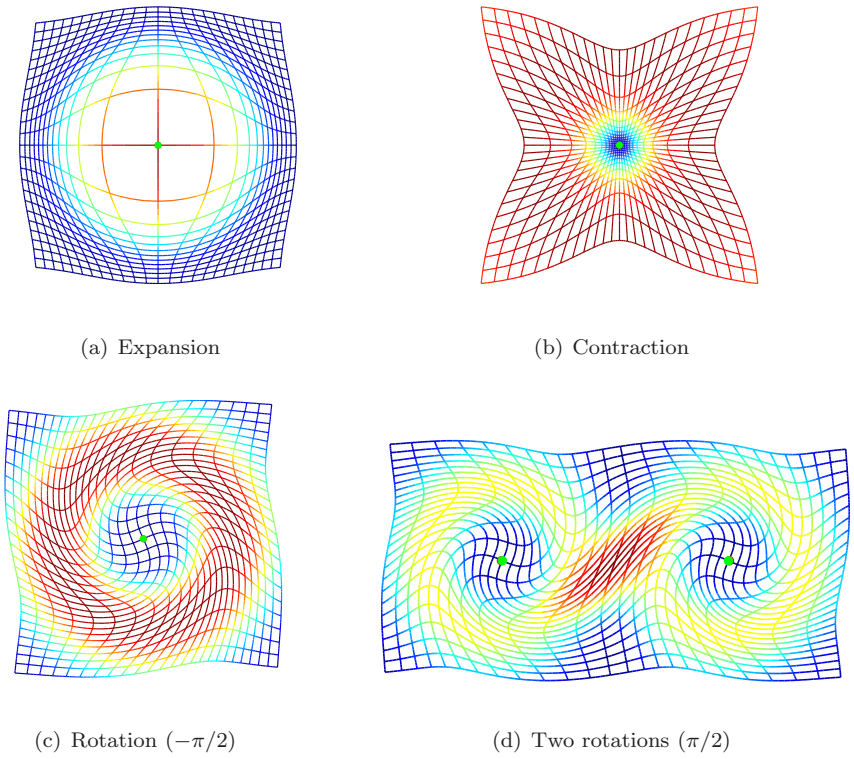
The integration of the ODEs can be performed with standard Runge-Kutta integrators such as Matlabs `ode45` procedure. With zero order kernels only and N points, the forward and backwards system consist of $2dN$ equations. With zero- and first order kernels, the forward system is extended to $N(2d+d^2)$ and the backwards system to $2N(d+d^2)$. For $d=3$, this implies an 2.5 time increase in the size of the system. In addition to this should be considered the extra floating point operations necessary for computing the somewhat more complicated evolution equations. This increase should, however, be viewed against the fact that the finite dimensional system contain orders of magnitude fewer control points, and the added capacity of deformation description included in the derivative information. In addition and in contrast to previous approaches, we transport the similarity gradient *only* at the control point trajectories, again an order of magnitude reduction of transported information. As we will see in the following section, the inclusion of higher order kernels provides information with very few control points.

5. Experiments. In order to demonstrate the efficiency and sparsity of representations using higher order kernels, we perform four sets of experiments. First, we provide four examples illustrating the type of deformations produced by zero- and first order kernels and the relation to the Polyaffine framework. We then use point based matching using first order information to show how complicated warps that would require many parameters with zero order deformation atoms can be generated with very compact representations using higher order kernels. When then underline the point that higher order kernels allow low-dimensional transformations to be registered using correspondingly low-dimensional representations: we show how synthetic test images generated by a low-dimensional transformation can be registered using only one deformation atom when representing using first order kernels and using the first order similarity measure approximation (3.1). We further emphasize this point by registering articulated movement using only one deformation atom per rigid part, and thus exemplify a natural representation that reduces the number of deformation atoms and the ambiguity in the placement of the atoms while also reducing the degrees of freedom in the representation. Finally, we illustrate how higher order kernels in a natural way allow registration of human brains with progressing atrophy. We describe the deformation field throughout the ventricles using few deformation atoms, and we thereby suggest a method for detecting anatomical change using few degrees of freedom. We start by briefly describing the similarity measures used throughout the experiments.

For the point examples below, we register moving points $x_1, \ldots, x_N$ against fixed points $y_1, \ldots, y_N$. In addition, we match first order information by specifying values of $D^j_{x_i}\varphi$. This is done compactly by providing matrices Y_i so that we seek $D_{x_i}\varphi = Y_i$ for all $i = 1, \ldots, N$. The similarity measure is simple sum of squares, i.e.

$$U(\varphi) = \sum_{i=1}^{N} \|\varphi(x_i) - y_i\|^2 + \|D_{x_i}\varphi - Y_i\|^2$$

using the matrix 2-norm. This amounts to fitting φ against a locally affine map with



(a) Expansion (b) Contraction

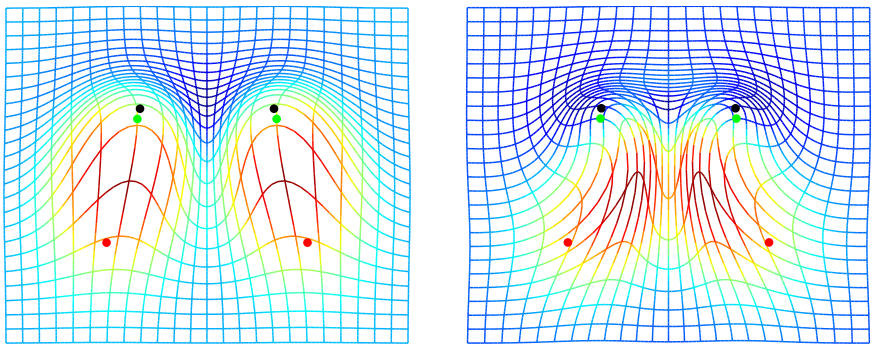
(c) Rotation $(-\pi/2)$ (d) Two rotations $(\pi/2)$

FIG. 5.1. *The effect of the generated deformation on an initially square grid for several initial first order momenta: Using the notation of Section 3.3, (a) expansion* $\rho_0 = \mathrm{Lin}_0(\mathrm{Id}_2)$*; (b) contraction* $\rho_0 = \mathrm{Lin}_0(-\mathrm{Id}_2)$*; (c) rotation* $\rho_0 = \mathrm{Lin}_0(\mathrm{Rot}(v))$*,* $v = -\pi/2$*; (d) two rotations* $v = \pi/2$*. The kernel is Gaussian with* $\sigma = 8$ *in grid units, and the grids are colored with the trace of Cauchy-Green strain tensor (log-scale). Notice the locality of the deformation caused by the finite scale of the kernel, and that the deformation stays diffeomorphic even when two rotations force conflicting movements.*

translational components y_i and linear components Y_i. For the image cases, we use L^1-similarity to build the first order approximation (3.1) with the smoothing kernel K_s being Gaussian of the same scale as the LDDMM kernel.

5.1. First Order Illustrations. To visually illustrate the deformation generated by higher order kernels, we show in Figure 5.1 the generated deformations on an initially square grid with four different first-order initial momenta. The deformation locally model the linear part of affine transformations and the the locality is determined by the Gaussian kernel that in the examples has scale $\sigma = 8$ in grid units. Notice for the rotations that the deformation stays diffeomorphic in the presence of conflicting forces. The similarity between the examples and the deformations generated in the Polyaffine framework [1] underlines the viewpoint that the registration using higher order kernels constitutes the LDDMM sibling of the Polyaffine framework.

5.2. First Order Point Registration. Figure 5.2 presents simple point based matching results with first order information. The lower points (red) are matched against the upper points (black) with match against expansion $D_\varphi(x_i) = 2\mathrm{Id}_2$ and



(a) Match with dilations (expansion) (b) Match with rotations ($-\pi/2$ and $\pi/2$)

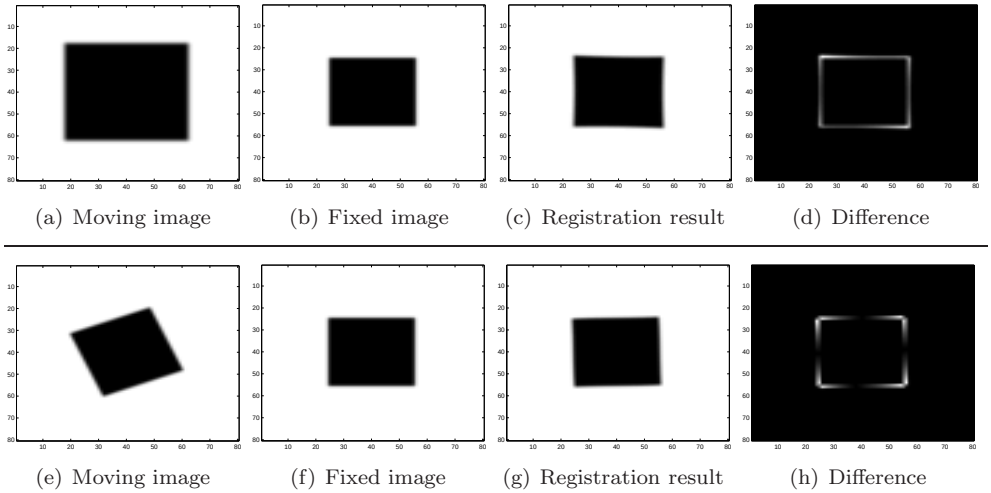
FIG. 5.2. *Two moving points (red) are matched against two fixed points (black) with results (green) and with match against (a) expansion* $D_\varphi(x_i) = 2\mathrm{Id}_2$, $i = 1, 2$*; and (b) rotation* $D_\varphi(x_i) = \mathrm{Rot}(v)$, $v = \mp\pi/2$, $i = 1, 2$. *The kernel is Gaussian with* $\sigma = 8$ *in grid units, and the grids are colored with the trace of Cauchy-Green strain tensor (log-scale).*

rotation $D_\varphi(x_i) = \mathrm{Rot}(v) = \begin{pmatrix} \cos(v), \sin(v) \\ -\sin(v), \cos(v) \end{pmatrix}$ for $v = \mp\pi/2$. The optimal diffeomorphisms exhibit the expected expanding and turning effect, respectively. We stress that the deformations are generated using only two deformations atoms with combined 12 parameters. Representing equivalent deformation using zero order kernels would require a significantly increased number of atoms and a correspond increase in the number of parameters.

5.3. Low Dimensional Image Registration. We now exemplify how higher order kernels allow low-dimensional transformations to be registered using correspondingly low-dimensional representations. We generate two test images by applying two linear transformations, an dilation and a rotation, to a binary image of a square, confer the moving images (a) and (e) in Figure 5.3. By placing one deformation atom in the center of each fixed image and by using the similarity measure approximation (3.1), we can successfully register the moving and fixed images. The result and difference plots are shown in Figure 5.3. The dimensionality of the linear transformations generating the moving images is equal to the number of parameters for the deformation atom. A registration using zero order kernels would need more than one deformation atom which would result in a number of parameters larger than the dimensionality. The scale of the Gaussian kernel used for the registration is 50 pixels.

5.4. Articulated Motion. The articulated motion of the finger[3] in Figure 5.4 (a) and (b) can be described by three locally linear transformations. With higher order kernels, we can place deformation atoms at the center of the bones in the moving and fixed images, and use the point positions together with the direction of the bones to drive a registration. This natural and low dimensional representation allows a fairly good match of the images resembling the use of the Polyaffine affine framework for articulated registration [23]. A similar registration using zero order kernels would

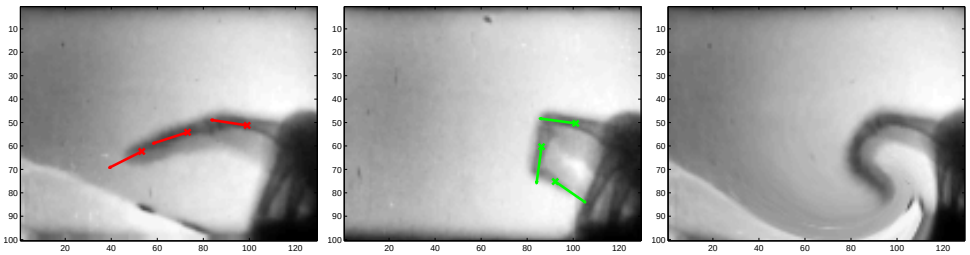
[3] X-ray frames from `http://www.archive.org/details/X-raystudiesofthejointmovements-wellcome`



(a) Moving image (b) Fixed image (c) Registration result (d) Difference

(e) Moving image (f) Fixed image (g) Registration result (h) Difference

FIG. 5.3. *With linear transformations, the dimensionality of the higher order representation matches the dimensionality of the transformation. A dilation (e) and rotation (d) is applied to the fixed binary images (b) and (f), respectively. The registration results (c) and (g) subtracted from the fixed images are shown in the difference pictures (d) and (h). The registration is performed with a single first order kernel in the center of the pictures, and the number of parameters for the registration thus matches the dimensionality of the linear representations. The slight differences between results and fixed images are caused by the first order approximation in* (3.1)*. Increasing the kernel size, adding more control points, or using second order kernels would imply less difference.*



(a) Moving (b) Fixed (c) Result, first order

FIG. 5.4. *Registering articulated movement using directional information of the bones: the landmarks and bone orientations (red points and arrows) in the moving image (a) are matched against the landmarks and bone orientations (green points and arrows) in the fixed image (b). The result using first order kernels (c) can be obtained with a low number of deformation atoms that can be consistently placed at the center of the bones. A corresponding zero order representation would use a higher number of atoms with a corresponding increase in the number of parameters.*

need two deformation atoms per bone and lacking a natural way to place such atoms, the positions would need to be optimized. With higher order kernels, the deformation atoms can be placed in a natural and consistent way, and the total number of free parameters is lower than a zero order representation using two atoms per bone.

5.5. Registering Atrophy. Atrophy occurs in the human brain among patients suffering from Alzheimer's disease, and the progressing atrophy can be detected by the expansion of the ventricles [16, 13]. Since first order kernels offer compact description

of expansion, this makes a parametrization of the registration based on higher order kernels suited for describing the expansion of the ventricles, and, in addition, the deformation represented by the kernels will be easily interpretable. In this experiment, we therefore suggest a registration method that using few degrees of freedom describes the expansion of the ventricles, and does so in a way that can be interpreted when doing further analysis of e.g. the volume change.

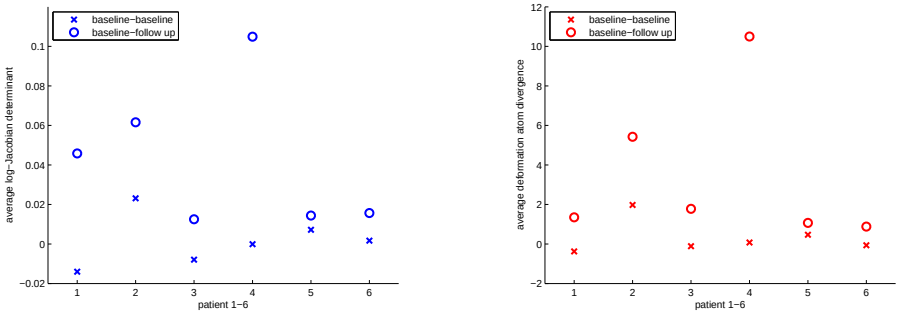
We will provide examples of 2D registration with the purpose of *illustrating* the use of the higher order kernels and suggest a method which can be applied in 3D. We do not aim at a quantitative evaluation but we plan to follow up on the experiment in future work with 3D registration of more subjects and explore the connection between first order initial momentum and actual ventricle expansion in greater detail.

We use the publicly available Oasis dataset[4] [17], and we select a small number of patients from which two baseline scans are acquired at the same day together with a later follow up scan. The patients are in various stages of dementia. We perform rigid registration [9] before selecting vertical 2D slices where the ventricles are clearly visible. The slice plane is the same for all three scans of each patient.

The expanding ventricles can be registered by placing deformation atoms in the form of higher order kernels in the center of the ventricles of the fixed image as shown in Figure 1.1. We manually place five deformation atoms in the ventricle area of each patient. It is important to note that though we localize the description of the deformation to the deformation atoms, the atoms control the deformation field throughout the ventricle area. Based on the size of the ventricles, we use Gaussian kernels with a scale of 15 voxels for the kernels, and we let the regularization weight in (2.1) be $\lambda = 16$. The effect of these choices is discussed below. Each deformation atom consists of a zero- and first order kernel, and, for each patient, we perform two registrations: we register the two baseline scans acquired at the same day, and we register one baseline scan against the follow up scan. Thus, the baseline-baseline registration should indicate no ventricle expansion, and we expect the baseline-follow up registration to indicate ventricle expansion. Figure 1.1 shows for one patient the placement of the control points in the baseline image, the follow up image, the log-Jacobian determinant in the ventricle area of the generated deformation, and the initial vector field driving the registration.

The use of first order kernels allows us to interpret the result of the registrations and to relate the results to possible expansion of the ventricles. The volume change is indicated by the Jacobian determinant of the generated deformation at the deformation atoms as well as by the divergence of the first order kernels. The latter is available directly from the registration parameters. We plot in Figure 5.5 the logarithm of the Jacobian determinant and the divergence for both the same day baseline-baseline registrations and for the baseline-follow up registrations. Patient $1-4$ are classified as demented, patient 5 and 6 as non-demented, and all patient have constant clinical dementia rating through the experiment. The time-span between baseline and follow up scan is 1.5-2 years with the exception of 3 years for patient four. As expected, the log-Jacobian is close to zero for the same day baseline-baseline scans but it increases with the baseline-follow up registrations of the demented patients. In addition, the correlation between the log-Jacobian and the divergence shows how the indicated volume change is available directly from the registration parameters. This result suggests the usefulness of the approach and points to future experiments to validate the method.

[4] `http://www.oasis-brains.org`



(a) The average log-Jacobian of the final deformation at the 5 deformation atoms for the baseline-baseline and baseline-follow up registrations

(b) The average divergence at the deformation atoms for the baseline-baseline and baseline-follow up registrations

FIG. 5.5. *Indicated volume change: (a) The average* log-*Jacobian determinant of the generated deformation at the 5 deformation atoms for six patients (1-4 demented, 5-6 non-demented); (b) divergence of the 5 higher order kernels representing the deformation. The divergence can be extracted directly from the parameters of the higher order kernels, and the correlation between the* log-*Jacobian and the divergence as seen by the similarity between (a) and (b) therefore shows the interpretability of the deformation atoms.*

We chose two important parameters above: the kernel scale and the regularization term. The choice of one scale for all patients works well if the ventricles to be registered are of approximately the same size at the baseline scans. If the ventricles vary in size, the scale can be chosen individually for each patient. Alternatively, a multi-scale approach could do this automatically which suggests combining the method with e.g. the kernel bundle framework [24]. Depending on the image forces, the regularization term in (2.1) will affect the amount of expansion captured in the registration. Because of the low number of control points, we can in practice set the contribution of the regularization term to zero without experiencing non-diffeomorphic results. It will be interesting in the future to estimate the actual volume expansion directly using the parameters of the deformation atoms with this less biased model.

6. Conclusion and Outlook. We have introduced higher order kernels in the LDDMM registration framework. The kernels allow *compact* representation of locally affine transformations by increasing the *capacity* of the deformation description. Coupled with similarity measures incorporating first order information, the higher order kernels improve the range of deformations reached by sparsely discretized LDDMM methods, and they allow direct capture of first order information such as expansion and contraction. In addition, the constitute deformation atoms for which the generated deformation is directly interpretable.

In the paper, we have shown how the partial derivative reproducing property implies singular momentum for the higher order kernels, and we used this to derive the EPDiff evolution equations. By computing the forward and backward variational equations, we are able to transport gradient information and derive a matching algorithm. We provide examples showing typical deformation coded by first order kernels and how images can be registered using a very few parameters, and we have applied the method to register human brains with progressing atrophy.

The experiments included here show only a first step in the application of higher

order kernels: the kernels may be applied to register entire images; merging the method with multi-scale approaches will increase the description capacity and may lead to further reduction in the dimensionality of the representation. Combined with efficient implementations, higher order kernels promise to provide a step forward in compact deformation description for image registration.

Appendix A. Time Evolution of μ_t. Inserting the Hamiltonian form of the momentum, we have

$$
\begin{aligned}
\big(\partial_t\mu_t\big|w\big) &= -\sum_{k=1}^{d}\big(\mu_t\big|\big(\mu_t\big|D_2K^k(\varphi_{0t}^v(x),\varphi_{0t}^v(y))w(y)\big)_x e_k\big)_y \\
&= -\sum_{k=1}^{d}\big(\mu_t\big|\int_\Omega\big(\sum_{i=1}^{N}\mu_{t,i}\otimes\delta_{x_{0,i}}+\sum_{i=1}^{N}\sum_{j=1}^{d}\mu_{t,i}^j\otimes D^j\delta_{x_{0,i}}\big)D_2K^k(\varphi_{0t}^v(x),\varphi_{0t}^v(y))w(y)dxe_k\big) \\
&= -\sum_{k=1}^{d}\big(\mu_t\big|\sum_{i=1}^{N}\mu_{t,i}^T D_2K^k(\varphi_{0t}^v(x_{0,i})),\varphi_{0t}^v(y))w(y)e_k\big) \\
&\quad -\sum_{k=1}^{d}\big(\mu_t\big|\sum_{i=1}^{N}\sum_{j=1}^{d}\mu_{t,i}^{j,T}\big(\sum_{m=1}^{d}D_1^mD_2K^k(\varphi_{0t}^v(x_{0,i}),\varphi_{0t}^v(y))(D^j\varphi_{0t}^v(x_{0,i}))^m\big)w(y)e_k\big) \\
&= -\sum_{k=1}^{d}\int_\Omega\big(\sum_{i=1}^{N}\mu_{t,i}\otimes\delta_{x_{0,i}}+\sum_{i=1}^{N}\sum_{j=1}^{d}\mu_{t,i}^j\otimes D^j\delta_{x_{0,i}}\big)\sum_{i=1}^{N}\mu_{t,i}^T D_2K^k(\varphi_{0t}^v(x_{0,i})),\varphi_{0t}^v(y))w(y)e_kdy \\
&\quad -\sum_{k=1}^{d}\int_\Omega\big(\sum_{i=1}^{N}\mu_{t,i}\otimes\delta_{x_{0,i}}+\sum_{i=1}^{N}\sum_{j=1}^{d}\mu_{t,i}^j\otimes D^j\delta_{x_{0,i}}\big) \\
&\qquad\qquad \sum_{i=1}^{N}\sum_{j=1}^{d}\mu_{t,i}^{j,T}\big(\sum_{m=1}^{d}D_1^mD_2K^k(\varphi_{0t}^v(x_{0,i}),\varphi_{0t}^v(y))(D^j\varphi_{0t}^v(x_{0,i}))^m\big)w(y)e_kdy \\
&= -\sum_{k=1}^{d}\sum_{i,n=1}^{N}\mu_{t,i}^T D_2K^k(\varphi_{0t}^v(x_{0,i})),\varphi_{0t}^v(x_{0,n}))w(x_{0,n})\mu_{t,n}^T e_k \\
&\quad -\sum_{k,j=1}^{d}\sum_{i,n=1}^{N}\mu_{t,i}^T\big(\sum_{m=1}^{d}D_2^mD_2K^k(\varphi_{0t}^v(x_{0,i}),\varphi_{0t}^v(x_{0,n}))(D^j\varphi_{0t}^v(x_{0,n}))^m\big)w(x_{0,n})\mu_{t,n}^{j,T}e_k \\
&\quad -\sum_{k,j=1}^{d}\sum_{i,n=1}^{N}\mu_{t,i}^T D_2K^k(\varphi_{0t}^v(x_{0,i})),\varphi_{0t}^v(x_{0,n}))D^jw(x_{0,n})\mu_{t,n}^{j,T}e_k \\
&\quad -\sum_{k,j=1}^{d}\sum_{i,n=1}^{N}\mu_{t,i}^{j,T}\big(\sum_{m=1}^{d}D_1^mD_2K^k(\varphi_{0t}^v(x_{0,i}),\varphi_{0t}^v(x_{0,n}))(D^j\varphi_{0t}^v(x_{0,i}))^m\big)w(x_{0,n})\mu_{t,n}^T e_k \\
&\quad -\sum_{k,j,j'=1}^{d}\sum_{i,n=1}^{N}\mu_{t,i}^{j,T}\big(\sum_{m,m'=1}^{d}D_2^{m'}D_1^mD_2K^k(\varphi_{0t}^v(x_{0,i}),\varphi_{0t}^v(x_{0,n}))(D^j\varphi_{0t}^v(x_{0,i}))^m(D^{j'}\varphi_{0t}^v(x_{0,n}))^{m'}\big) \\
&\qquad\qquad w(x_{0,n})\mu_{t,n}^{j',T}e_k \\
&\quad -\sum_{k,j,j'=1}^{d}\sum_{i,n=1}^{N}\mu_{t,i}^{j,T}\big(\sum_{m=1}^{d}D_1^mD_2K^k(\varphi_{0t}^v(x_{0,i}),\varphi_{0t}^v(x_{0,n}))(D^j\varphi_{0t}^v(x_{0,i}))^m\big)D^{j'}w(x_{0,n})\mu_{t,n}^{j',T}e_k\ .
\end{aligned}
$$

Thus

$$\begin{aligned}
\partial_t \mu^T_{t,n} = &- \sum_{k=1}^{d} (\mu_{t,n})^k \sum_{i=1}^{N} \mu^T_{t,i} D_2 K^k(\varphi^v_{0t}(x_{0,i})), \varphi^v_{0t}(x_{0,n})) \\
&- \sum_{k,j=1}^{d} (\mu^j_{t,n})^k \sum_{i=1}^{N} \mu^T_{t,i} \big(\sum_{m=1}^{d} D_2^m D_2 K^k(\varphi^v_{0t}(x_{0,i}), \varphi^v_{0t}(x_{0,n}))(D^j \varphi^v_{0t}(x_{0,n}))^m \big) \\
&- \sum_{k,j=1}^{d} (\mu_{t,n})^k \sum_{i=1}^{N} \mu^{j,T}_{t,i} \big(\sum_{m=1}^{d} D_1^m D_2 K^k(\varphi^v_{0t}(x_{0,i}), \varphi^v_{0t}(x_{0,n}))(D^j \varphi^v_{0t}(x_{0,i}))^m \big) \\
&- \sum_{k,j,j'=1}^{d} (\mu^{j'}_{t,n})^k \sum_{i=1}^{N} \mu^{j,T}_{t,i} \\
&\qquad \big(\sum_{m,m'=1}^{d} D_2^{m'} D_1^m D_2 K^k(\varphi^v_{0t}(x_{0,i}), \varphi^v_{0t}(x_{0,n}))(D^j \varphi^v_{0t}(x_{0,i}))^m (D^{j'} \varphi^v_{0t}(x_{0,n}))^{m'} \big) \ .
\end{aligned}$$

We write K^k_l (column-row) and get

$$\begin{aligned}
\partial_t \mu_{t,n} = &- \sum_{k,l=1}^{d} \sum_{i=1}^{N} \Big((\mu_{t,n})^k (\mu_{t,i})^l \nabla_2 K^k_l(\varphi^v_{0t}(x_{0,i})), \varphi^v_{0t}(x_{0,n})) \\
&+ \sum_{j=1}^{d} (\mu^j_{t,n})^k (\mu_{t,i})^l D_2 \nabla_2 K^k_l(\varphi^v_{0t}(x_{0,i}), \varphi^v_{0t}(x_{0,n})) D^j \varphi^v_{0t}(x_{0,n}) \\
&+ \sum_{j=1}^{d} (\mu_{t,n})^k (\mu^j_{t,i})^l D_1 \nabla_2 K^k_l(\varphi^v_{0t}(x_{0,i}), \varphi^v_{0t}(x_{0,n})) D^j \varphi^v_{0t}(x_{0,i}) \\
&+ \sum_{j,j'=1}^{d} (\mu^{j'}_{t,n})^k (\mu^j_{t,i})^l D_2 \big(D_1 \nabla_2 K^k_l(\varphi^v_{0t}(x_{0,i}), \varphi^v_{0t}(x_{0,n})) D^j \varphi^v_{0t}(x_{0,i}) \big) D^{j'} \varphi^v_{0t}(x_{0,n}) \Big)
\end{aligned}$$

.

For scalar kernels $K^k_l(x,y) = K(x,y)$ iff $k = l$, and hence

$$\begin{aligned}
\partial_t \mu_{t,n} = &- \sum_{i=1}^{N} \Big(\big(\mu^T_{t,n} \mu_{t,i} \big) \nabla_2 K(\varphi^v_{0t}(x_{0,i})), \varphi^v_{0t}(x_{0,n})) \\
&+ \sum_{j=1}^{d} \big(\mu^{j,T}_{t,n} \mu_{t,i} \big) D_2 \nabla_2 K(\varphi^v_{0t}(x_{0,i}), \varphi^v_{0t}(x_{0,n})) D^j \varphi^v_{0t}(x_{0,n}) \\
&+ \sum_{j=1}^{d} \big(\mu^T_{t,n} \mu^j_{t,i} \big) D_1 \nabla_2 K(\varphi^v_{0t}(x_{0,i}), \varphi^v_{0t}(x_{0,n})) D^j \varphi^v_{0t}(x_{0,i}) \\
&+ \sum_{j,j'=1}^{d} \big(\mu^{j',T}_{t,n} \mu^j_{t,i} \big) D_2 \big(D_1 \nabla_2 K(\varphi^v_{0t}(x_{0,i}), \varphi^v_{0t}(x_{0,n})) D^j \varphi^v_{0t}(x_{0,i}) \big) D^{j'} \varphi^v_{0t}(x_{0,n}) \Big) \ .
\end{aligned}$$

Using symmetry and rewriting,

$$\begin{aligned}\partial_t \mu_{t,n} = -\sum_{i=1}^{N} \Big(&\big(\mu_{t,n}^T \mu_{t,i}\big) \nabla_2 K(x_{t,i}, x_{t,n}) \\ &+ \sum_{j=1}^{d} \big(\mu_{t,n}^{j,T} \mu_{t,i} - \mu_{t,n}^T \mu_{t,i}^j\big) D_2 \nabla_2 K(x_{t,i}, x_{t,n}) D^j \varphi_{0t}^v(x_{0,n}) \\ &+ \sum_{j,j'=1}^{d} \big(\mu_{t,n}^{j',T} \mu_{t,i}^j\big) D_2 \big(D_1 \nabla_2 K(x_{t,i}, x_{t,n}) D^j \varphi_{0t}^v(x_{0,i})\big) D^{j'} \varphi_{0t}^v(x_{0,n})\Big) \ .\end{aligned}$$

Appendix B. Variation of the Kernel and Derivatives. With $K(x,y) = \gamma(|x-y|^2)$, we have the following expressions for the derivatives of the kernel:

$$\begin{aligned}
&\nabla_1 K = 2\dot\gamma(|x-y|^2)(x-y) \\
&\nabla_2 K = -2\dot\gamma(|x-y|^2)(x-y) \\
&D_1\nabla_1 K = 2\dot\gamma(|x-y|^2)\mathrm{Id}_d + 4\ddot\gamma(|x-y|^2)(x-y)(x-y)^T \\
&D_2\nabla_1 K = -2\dot\gamma(|x-y|^2)\mathrm{Id}_d - 4\ddot\gamma(|x-y|^2)(x-y)(x-y)^T \\
&D_1\nabla_2 K = -2\dot\gamma(|x-y|^2)\mathrm{Id}_d - 4\ddot\gamma(|x-y|^2)(x-y)(x-y)^T \\
&D_2\nabla_2 K = 2\dot\gamma(|x-y|^2)\mathrm{Id}_d + 4\ddot\gamma(|x-y|^2)(x-y)(x-y)^T \\
&D_2(D_1\nabla_2 K a) = -D_2(2\dot\gamma(|x-y|^2)a + 4\ddot\gamma(|x-y|^2)(x-y)^T a(x-y)) \\
&\qquad = +4\ddot\gamma(|x-y|^2)a(x-y)^T + 8\dddot\gamma(|x-y|^2)(x-y)^T a(x-y)(x-y)^T \\
&\qquad\quad + 4\ddot\gamma(|x-y|^2)(x-y)a^T + 4\ddot\gamma(|x-y|^2)(x-y)^T a\mathrm{Id}_d \\
&\qquad = 4\Big(\ddot\gamma(|x-y|^2)a(x-y)^T + \ddot\gamma(|x-y|^2)(x-y)a^T \\
&\qquad\quad + \ddot\gamma(|x-y|^2)(x-y)^T a\mathrm{Id}_d + 2\dddot\gamma(|x-y|^2)(x-y)^T a(x-y)(x-y)^T\Big) \ .
\end{aligned}$$

Variations of these expressions the take the form

$$\begin{aligned}
&\delta K(x_{t,i}, x_{t,n}) = 2\dot\gamma_{t,in}(x_{t,i}-x_{t,n})^T(\delta x_{t,i} - \delta x_{t,n}) \\
&\delta\nabla_1 K(x_{t,i}, x_{t,n}) = 4\ddot\gamma_{t,in}(x_{t,i}-x_{t,n})^T(\delta x_{t,i} - \delta x_{t,n})(x_{t,i}-x_{t,n}) + 2\dot\gamma_{t,in}(\delta x_{t,i} - \delta x_{t,n}) \\
&\delta\nabla_2 K(x_{t,i}, x_{t,n}) = -\delta\nabla_1 K(x_{t,i}, x_{t,n}) \\
&\delta D_1\nabla_2 K(x_{t,i}, x_{t,n}) = -4\ddot\gamma_{t,in}(x_{t,i}-x_{t,n})^T(\delta x_{t,i} - \delta x_{t,n})\mathrm{Id}_d \\
&\qquad - 8\dddot\gamma_{t,in}(x_{t,i}-x_{t,n})^T(\delta x_{t,i} - \delta x_{t,n})(x_{t,i}-x_{t,n})(x_{t,i}-x_{t,n})^T \\
&\qquad - 4\ddot\gamma_{t,in}(\delta x_{t,i} - \delta x_{t,n})(x_{t,i}-x_{t,n})^T \\
&\qquad - 4\ddot\gamma_{t,in}(x_{t,i}-x_{t,n})(\delta x_{t,i} - \delta x_{t,n})^T \\
&\delta D_2\nabla_2 K(x_{t,i}, x_{t,n}) = -\delta D_1\nabla_2 K(x_{t,i}, x_{t,n})
\end{aligned}$$

$$
\begin{aligned}
\delta D_2\big(D_1\nabla_2 K(x_{t,i},x_{t,n})a\big) &= D_2\big(D_1\nabla_2 K(x_{t,i},x_{t,n})\delta a\big)\\
&+ 8\dddot{\gamma}_{t,in}(x_{t,i}-x_{t,n})^T(\delta x_{t,i}-\delta x_{t,n})a(x_{t,i}-x_{t,n})^T + 4\ddot{\gamma}_{t,in}a(\delta x_{t,i}-\delta x_{t,n})^T\\
&+ 8\dddot{\gamma}_{t,in}(x_{t,i}-x_{t,n})^T(\delta x_{t,i}-\delta x_{t,n})(x_{t,i}-x_{t,n})a^T + 4\ddot{\gamma}_{t,in}(\delta x_{t,i}-\delta x_{t,n})a^T\\
&+ 8\dddot{\gamma}_{t,in}(x_{t,i}-x_{t,n})^T(\delta x_{t,i}-\delta x_{t,n})(x_{t,i}-x_{t,n})^T a\mathrm{Id}_d + 4\ddot{\gamma}_{t,in}(\delta x_{t,i}-\delta x_{t,n})^T a\mathrm{Id}_d\\
&+ 16\ddddot{\gamma}_{t,in}(x_{t,i}-x_{t,n})^T(\delta x_{t,i}-\delta x_{t,n})(x_{t,i}-x_{t,n})^T a(x_{t,i}-x_{t,n})(x_{t,i}-x_{t,n})^T\\
&+ 8\dddot{\gamma}_{t,in}(\delta x_{t,i}-\delta x_{t,n})^T a(x_{t,i}-x_{t,n})(x_{t,i}-x_{t,n})^T\\
&+ 8\dddot{\gamma}_{t,in}(x_{t,i}-x_{t,n})^T a(\delta x_{t,i}-\delta x_{t,n})(x_{t,i}-x_{t,n})^T\\
&+ 8\dddot{\gamma}_{t,in}(x_{t,i}-x_{t,n})^T a(x_{t,i}-x_{t,n})(\delta x_{t,i}-\delta x_{t,n})^T\ .
\end{aligned}
$$

Appendix C. The Transpose Derivative System. We let M denote the time-dependent matrix governing the linear ODE (4.1) for the evolution of the variation of the initial conditions of the EPDiff equations, and we write M as a block matrix

$$
M = \begin{pmatrix} M^{\varphi\varphi} & M^{\varphi D\varphi} & M^{\varphi\mu} & M^{\varphi\mu^{j'}} \\ M^{D\varphi\varphi} & M^{D\varphi D\varphi} & M^{D\varphi\mu} & M^{\varphi\mu^{j'}} \\ M^{\mu\varphi} & M^{\mu D\varphi} & M^{\mu\mu} & M^{\varphi\mu^{j'}} \\ M^{\mu^j\varphi} & M^{\mu^j D\varphi} & M^{\mu^j\mu} & M^{\varphi^j\mu^{j'}} \end{pmatrix} = \begin{pmatrix} \big(a_{ni}^{\varphi\varphi}\big) & \Big(a_{ni}^{\varphi D^l\varphi}\Big) & \big(a_{ni}^{\varphi\mu}\big) & \Big(a_{ni}^{\varphi\mu^{j'}}\Big) \\ \Big(a_{ni}^{D\varphi^l\varphi}\Big) & \Big(a_{ni}^{D\varphi^l D\varphi^k}\Big) & \Big(a_{ni}^{D\varphi^l\mu}\Big) & \Big(a_{ni}^{D\varphi^l\mu^{j'}}\Big) \\ \big(a_{ni}^{\mu\varphi}\big) & \Big(a_{ni}^{\mu D\varphi^k}\Big) & \big(a_{ni}^{\mu\mu}\big) & \Big(a_{ni}^{\mu\mu^{j'}}\Big) \\ \Big(a_{ni}^{\mu^j\varphi}\Big) & \Big(a_{ni}^{\mu^j D\varphi^k}\Big) & \Big(a_{ni}^{\mu^j\mu}\Big) & \Big(a_{ni}^{\mu^j\mu^{j'}}\Big) \end{pmatrix}.
$$

In order to determine the transpose M^T, we isolate the components of the submatrices $M_{ni}^{\cdot\cdot}$ from the right-hand side of system (4.1). All components not listed below will be zero.

$$
\begin{aligned}
m_{ni}^{\varphi\varphi} &= 2\dot{\gamma}_{t,in}\mu_{t,i}(x_{t,i}-x_{t,n})^T\mathrm{Id}_d\\
&\quad + \sum_{j'=1}^{d}\mu_{t,i}^{j'}D^{j'}\varphi_{0t}^v(x_{0,i})^T\big(4\ddot{\gamma}_{t,in}(x_{t,i}-x_{t,n})(x_{t,i}-x_{t,n})^T + 2\dot{\gamma}_{t,in}\mathrm{Id}_d\big)\\
&\quad \text{iff } i=n:\\
&\quad \sum_{i'=1}^{n}-2\dot{\gamma}_{t,i'n}\mu_{t,i'}(x_{t,i'}-x_{t,n})^T\mathrm{Id}_d\\
&\quad - \sum_{i'=1}^{n}\sum_{j'=1}^{d}\mu_{t,i'}^{j'}D^{j'}\varphi_{0t}^v(x_{0,i'})^T\big(4\ddot{\gamma}_{t,i'n}(x_{t,i'}-x_{t,n})(x_{t,i'}-x_{t,n})^T + 2\dot{\gamma}_{t,i'n}\mathrm{Id}_d\big)\\
m_{ni}^{\varphi D\varphi^l} &= \mu_{t,i}^l\nabla_1 K(x_{t,i},x_{t,n})^T\\
m_{ni}^{\varphi\mu} &= \gamma_{t,in}\mathrm{Id}_d\\
m_{ni}^{\varphi\mu^j} &= \nabla_1 K(x_{t,i},x_{t,n})^T D^j\varphi_{0t}^v(x_{0,i})\mathrm{Id}_d
\end{aligned}
$$

$$
\begin{aligned}
m_{ni}^{D\varphi^l\varphi} &= -\mu_{t,i} D^l\varphi_{0t}^v(x_{0,n})^T\big(4\ddot{\gamma}_{t,in}(x_{t,i}-x_{t,n})(x_{t,i}-x_{t,n})^T + 2\dot{\gamma}_{t,in}\mathrm{Id}_d\big)\\
&\quad - \sum_{j'=1}^{N}\mu_{t,i}^{j'} D^l\varphi_{0t}^v(x_{0,n})^T\big(4\ddot{\gamma}_{t,in}D^{j'}\varphi_{0t}^v(x_{0,i})(x_{t,i}-x_{t,n})^T\\
&\qquad + 8\dddot{\gamma}_{t,in}(x_{t,i}-x_{t,n})(x_{t,i}-x_{t,n})^T D^{j'}\varphi_{0t}^v(x_{0,i})(x_{t,i}-x_{t,n})^T\\
&\qquad + 4\ddot{\gamma}_{t,in}(x_{t,i}-x_{t,n})^T D^{j'}\varphi_{0t}^v(x_{0,i})\mathrm{Id}_d + 4\ddot{\gamma}_{t,in}(x_{t,i}-x_{t,n})(D^{j'}\varphi_{0t}^v(x_{0,i}))^T\big)\\
&\quad \text{iff } i=n:\\
&\quad \sum_{i'=1}^{n}\mu_{t,i'} D^l\varphi_{0t}^v(x_{0,n})^T\big(4\ddot{\gamma}_{t,i'n}(x_{t,i'}-x_{t,n})(x_{t,i'}-x_{t,n})^T + 2\dot{\gamma}_{t,i'n}\mathrm{Id}_d\big)\\
&\quad + \sum_{i'=1}^{n}\sum_{j'=1}^{N}\mu_{t,i'}^{j'} D^l\varphi_{0t}^v(x_{0,n})^T\big(4\ddot{\gamma}_{t,i'n}D^{j'}\varphi_{0t}^v(x_{0,i'})(x_{t,i'}-x_{t,n})^T\\
&\qquad + 8\dddot{\gamma}_{t,i'n}(x_{t,i'}-x_{t,n})(x_{t,i'}-x_{t,n})^T D^{j'}\varphi_{0t}^v(x_{0,i'})(x_{t,i'}-x_{t,n})^T\\
&\qquad + 4\ddot{\gamma}_{t,i'n}(x_{t,i'}-x_{t,n})^T D^{j'}\varphi_{0t}^v(x_{0,i'})\mathrm{Id}_d + 4\ddot{\gamma}_{t,i'n}(x_{t,i'}-x_{t,n})(D^{j'}\varphi_{0t}^v(x_{0,i'}))^T\big)\\
m_{ni}^{D\varphi^l D\varphi^k} &= \mu_{t,i}^k D^l\varphi_{0t}^v(x_{0,n})^T D_1\nabla_2 K(x_{t,i},x_{t,n})\\
&\quad \text{iff } i=n,\ \text{iff } l=k:\\
&\quad \sum_{i'=1}^{n}\mu_{t,i'}\nabla_2 K(x_{t,i'},x_{t,n})^T + \sum_{i'=1}^{n}\sum_{j'=1}^{d}\mu_{t,i'}^{j'}\big(D_1\nabla_2 K(x_{t,i'},x_{t,n})D^{j'}\varphi_{0t}^v(x_{0,i'})\big)^T\mathrm{Id}_d\\
m_{ni}^{D\varphi^l\mu} &= \nabla_2 K(x_{t,i},x_{t,n})^T D^l\varphi_{0t}^v(x_{0,n})\mathrm{Id}_d\\
m_{ni}^{D\varphi^l\mu^j} &= \big(D_1\nabla_2 K(x_{t,i},x_{t,n})D^j\varphi_{0t}^v(x_{0,i})\big)^T D^l\varphi_{0t}^v(x_{0,n})\mathrm{Id}_d
\end{aligned}
$$

$$
\begin{aligned}
m_{ni}^{\mu\varphi} &= (\mu_{t,n}^T\mu_{t,i})\big(4\ddot{\gamma}_{t,in}(x_{t,i}-x_{t,n})(x_{t,i}-x_{t,n})^T + 2\dot{\gamma}_{t,in}\mathrm{Id}_d\big)\\
&\quad - \sum_{j'=1}^{d}\big(\mu_{t,n}^{j',T}\mu_{t,i} - \mu_{t,n}^T\mu_{t,i}^{j'}\big)\big(4\ddot{\gamma}_{t,in}D^{j'}\varphi_{0t}^v(x_{0,n})(x_{t,i}-x_{t,n})^T\\
&\qquad + 8\dddot{\gamma}_{t,in}(x_{t,i}-x_{t,n})(x_{t,i}-x_{t,n})^T D^{j'}\varphi_{0t}^v(x_{0,n})(x_{t,i}-x_{t,n})^T\\
&\qquad + 4\ddot{\gamma}_{t,in}(x_{t,i}-x_{t,n})^T D^{j'}\varphi_{0t}^v(x_{0,n})\mathrm{Id}_d + 4\ddot{\gamma}_{t,in}(x_{t,i}-x_{t,n})(D^{j'}\varphi_{0t}^v(x_{0,n}))^T\big)\\
&\quad - \sum_{j,j'=1}^{d}\big(\mu_{t,n}^{j',T}\mu_{t,i}^{j}\big)\big(8\dddot{\gamma}_{t,in}D^j\varphi_{0t}^v(x_{0,i})(x_{t,i}-x_{t,n})^T D^{j'}\varphi_{0t}^v(x_{0,n})(x_{t,i}-x_{t,n})^T\\
&\qquad + 8\ddot{\gamma}_{t,in}(x_{t,i}-x_{t,n})D^j\varphi_{0t}^v(x_{0,i})^T D^{j'}\varphi_{0t}^v(x_{0,n})(x_{t,i}-x_{t,n})^T\\
&\qquad + 4\ddot{\gamma}_{t,in}D^j\varphi_{0t}^v(x_{0,i})D^{j'}\varphi_{0t}^v(x_{0,n})^T + 4\ddot{\gamma}_{t,in}D^j\varphi_{0t}^v(x_{0,i})^T D^{j'}\varphi_{0t}^v(x_{0,n})\\
&\qquad + 8\ddot{\gamma}_{t,in}(x_{t,i}-x_{t,n})^T D^j\varphi_{0t}^v(x_{0,i})D^{j'}\varphi_{0t}^v(x_{0,n})(x_{t,i}-x_{t,n})^T\\
&\qquad + 4\ddot{\gamma}_{t,in}D^{j'}\varphi_{0t}^v(x_{0,n})D^j\varphi_{0t}^v(x_{0,i})^T\\
&\qquad + 16\ddot{\ddot{\gamma}}_{t,in}(x_{t,i}-x_{t,n})^T D^j\varphi_{0t}^v(x_{0,i})(x_{t,i}-x_{t,n})(x_{t,i}-x_{t,n})^T\\
&\qquad\qquad D^{j'}\varphi_{0t}^v(x_{0,n})(x_{t,i}-x_{t,n})^T\\
&\qquad + 8\dddot{\gamma}_{t,in}(x_{t,i}-x_{t,n})(x_{t,i}-x_{t,n})^T D^{j'}\varphi_{0t}^v(x_{0,n})D^j\varphi_{0t}^v(x_{0,i})^T\\
&\qquad + 8\dddot{\gamma}_{t,in}(x_{t,i}-x_{t,n})^T D^j\varphi_{0t}^v(x_{0,i})(x_{t,i}-x_{t,n})^T D^{j'}\varphi_{0t}^v(x_{0,n})\\
&\qquad + 8\dddot{\gamma}_{t,in}(x_{t,i}-x_{t,n})^T D^j\varphi_{0t}^v(x_{0,i})(x_{t,i}-x_{t,n})D^{j'}\varphi_{0t}^v(x_{0,n})^T\\
&\qquad)
\end{aligned}
$$

$$
\begin{aligned}
&\text{iff } i = n: \\
&- \sum_{i'=1}^{N} (\mu_{t,n}^T \mu_{t,i'}) \big(4\ddot{\gamma}_{t,i'n}(x_{t,i'} - x_{t,n})(x_{t,i'} - x_{t,n})^T + 2\dot{\gamma}_{t,i'n}\mathrm{Id}_d\big) \\
&+ \sum_{i'=1}^{N} \sum_{j'=1}^{d} \big(\mu_{t,n}^{j',T} \mu_{t,i'} - \mu_{t,n}^T \mu_{t,i'}^{j'}\big) \big(4\ddot{\gamma}_{t,i'n} D^{j'} \varphi_{0t}^v(x_{0,n})(x_{t,i'} - x_{t,n})^T \\
&\quad + 8\dddot{\gamma}_{t,i'n}(x_{t,i'} - x_{t,n})(x_{t,i'} - x_{t,n})^T D^{j'} \varphi_{0t}^v(x_{0,n})(x_{t,i'} - x_{t,n})^T \\
&\quad + 4\ddot{\gamma}_{t,i'n}(x_{t,i'} - x_{t,n})^T D^{j'} \varphi_{0t}^v(x_{0,n})\mathrm{Id}_d \\
&\quad + 4\ddot{\gamma}_{t,i'n}(x_{t,i'} - x_{t,n})(D^{j'} \varphi_{0t}^v(x_{0,n}))^T\big) \\
&+ \sum_{i'=1}^{N} \sum_{j,j'=1}^{d} \big(\mu_{t,n}^{j',T} \mu_{t,i'}^{j}\big)\big(\\
&\quad 8\dddot{\gamma}_{t,i'n} D^j \varphi_{0t}^v(x_{0,i'})(x_{t,i'} - x_{t,n})^T D^{j'} \varphi_{0t}^v(x_{0,n})(x_{t,i'} - x_{t,n})^T \\
&\quad + 4\ddot{\gamma}_{t,i'n} D^j \varphi_{0t}^v(x_{0,i'}) D^{j'} \varphi_{0t}^v(x_{0,n})^T \\
&\quad + 8\dddot{\gamma}_{t,i'n}(x_{t,i'} - x_{t,n}) D^j \varphi_{0t}^v(x_{0,i'})^T D^{j'} \varphi_{0t}^v(x_{0,n})(x_{t,i'} - x_{t,n})^T \\
&\quad + 4\ddot{\gamma}_{t,i'n} D^j \varphi_{0t}^v(x_{0,i'})^T D^{j'} \varphi_{0t}^v(x_{0,n}) \\
&\quad + 8\dddot{\gamma}_{t,i'n}(x_{t,i'} - x_{t,n})^T D^j \varphi_{0t}^v(x_{0,i'}) D^{j'} \varphi_{0t}^v(x_{0,n})(x_{t,i'} - x_{t,n})^T \\
&\quad + 4\ddot{\gamma}_{t,i'n} D^{j'} \varphi_{0t}^v(x_{0,n}) D^j \varphi_{0t}^v(x_{0,i'})^T \\
&\quad + 16\dddot{\gamma}_{t,i'n}(x_{t,i'} - x_{t,n})^T D^j \varphi_{0t}^v(x_{0,i'})(x_{t,i'} - x_{t,n})(x_{t,i'} - x_{t,n})^T \\
&\qquad\qquad D^{j'} \varphi_{0t}^v(x_{0,n})(x_{t,i'} - x_{t,n})^T \\
&\quad + 8\dddot{\gamma}_{t,i'n}(x_{t,i'} - x_{t,n})(x_{t,i'} - x_{t,n})^T D^{j'} \varphi_{0t}^v(x_{0,n}) D^j \varphi_{0t}^v(x_{0,i'})^T \\
&\quad + 8\dddot{\gamma}_{t,i'n}(x_{t,i'} - x_{t,n})^T D^j \varphi_{0t}^v(x_{0,i'})(x_{t,i'} - x_{t,n})^T D^{j'} \varphi_{0t}^v(x_{0,n}) \\
&\quad + 8\dddot{\gamma}_{t,i'n}(x_{t,i'} - x_{t,n})^T D^j \varphi_{0t}^v(x_{0,i'})(x_{t,i'} - x_{t,n}) D^{j'} \varphi_{0t}^v(x_{0,n})^T \\
&\quad)
\end{aligned}
$$

$$
\begin{aligned}
m_{ni}^{\mu D\varphi^l} = &- \sum_{j'=1}^{d} 4\big(\mu_{t,n}^{j',T} \mu_{t,i}^{l}\big) \big(\ddot{\gamma}_{t,in}(x_{i,t} - x_{n,t})^T D^{j'} \varphi_{0t}^v(x_{0,n})\mathrm{Id}_d \\
&\quad + \ddot{\gamma}_{t,in}(x_{i,t} - x_{n,t}) D^{j'} \varphi_{0t}^v(x_{0,n})^T \\
&\quad + \ddot{\gamma}_{t,in} D^{j'} \varphi_{0t}^v(x_{0,n})(x_{i,t} - x_{n,t})^T \\
&\quad + 2\dddot{\gamma}_{t,in}(x_{i,t} - x_{n,t})(x_{i,t} - x_{n,t})^T D^{j'} \varphi_{0t}^v(x_{0,n})(x_{i,t} - x_{n,t})^T\big)
\end{aligned}
$$

$$
\begin{aligned}
&\text{iff } i = n: \\
&- \sum_{i'=1}^{N} \big(\mu_{t,n}^{l,T} \mu_{t,i'} - \mu_{t,n}^T \mu_{t,i'}^{l}\big) D_2 \nabla_2 K(x_{t,i'}, x_{t,n}) \\
&- \sum_{i'=1}^{N} \sum_{j'=1}^{d} \big(\mu_{t,n}^{l,T} \mu_{t,i'}^{j'}\big) D_2 \big(D_1 \nabla_2 K(x_{t,i'}, x_{t,n}) D^{j'} \varphi_{0t}^v(x_{0,i'})\big)
\end{aligned}
$$

$$m_{ni}^{\mu\mu} = -\nabla_2 K(x_{t,i}, x_{t,n})\mu_{t,n}^T - \sum_{j'=1}^{d} D_2\nabla_2 K(x_{t,i}, x_{t,n}) D^{j'}\varphi_{0t}^v(x_{0,n})\mu_{t,n}^{j',T}$$

$$\text{iff } i = n:$$

$$-\sum_{i'=1}^{N} \nabla_2 K(x_{t,i'}, x_{t,n})\mu_{t,i'}^T + \sum_{i'=1}^{N}\sum_{j'=1}^{d} D_2\nabla_2 K(x_{t,i'}, x_{t,n}) D^{j'}\varphi_{0t}^v(x_{0,n})\mu_{t,i'}^{j',T}$$

$$m_{ni}^{\mu\mu^j} = +D_2\nabla_2 K(x_{t,i}, x_{t,n}) D^j\varphi_{0t}^v(x_{0,n})\mu_{t,n}^T$$

$$-\sum_{j'=1}^{d} D_2\big(D_1\nabla_2 K(x_{t,i}, x_{t,n}) D^j\varphi_{0t}^v(x_{0,i})\big) D^{j'}\varphi_{0t}^v(x_{0,n})\mu_{t,n}^{j',T}$$

$$\text{iff } i = n:$$

$$-\sum_{i'=1}^{N} D_2\nabla_2 K(x_{t,i'}, x_{t,n}) D^j\varphi_{0t}^v(x_{0,n})\mu_{t,i'}^T$$

$$-\sum_{i'=1}^{N}\sum_{j'=1}^{d} D_2\big(D_1\nabla_2 K(x_{t,i'}, x_{t,n}) D^{j'}\varphi_{0t}^v(x_{0,i'})\big) D^j\varphi_{0t}^v(x_{0,n})\mu_{t,i'}^{j',T}$$

$$m_{ni}^{\mu^j\varphi} = -\sum_{j'=1}^{d} D\varphi_{0t}^v(x_{0,i})^{-1,T} e_{j'}\mu_{t,n}^{j,T} m_{ni}^{D\varphi^{j'}\varphi}$$

$$m_{ni}^{\mu^j D\varphi^l} = -\sum_{j'=1}^{d} D\varphi_{0t}^v(x_{0,i})^{-1,T} e_{j'}\mu_{t,n}^{j,T} m_{ni}^{D\varphi^{j'} D\varphi^l}$$

$$\text{iff } i = n:$$

$$D\varphi_{0t}^v(x_{0,n})^{-1,T} e_l\big(D\varphi_{0t}^v(x_{0,n})^{-1,T}\partial_t D\varphi_{0t}^v(x_{0,n})^T\mu_{t,n}^j\big)^T$$

$$m_{ni}^{\mu^j\mu} = -\sum_{j'=1}^{d} D\varphi_{0t}^v(x_{0,i})^{-1,T} e_{j'}\mu_{t,n}^{j,T} m_{ni}^{D\varphi^{j'}\mu}$$

$$m_{ni}^{\mu^j\mu^{j'}} = -\sum_{j''=1}^{d} D\varphi_{0t}^v(x_{0,i})^{-1,T} e_{j''}\mu_{t,n}^{j,T} m_{ni}^{D\varphi^{j''}\mu^{j'}}$$

$$\text{iff } i = n,\ j = j':$$

$$-\,D\varphi_{0t}^v(x_{0,n})^{-1,T}\partial_t D\varphi_{0t}^v(x_{0,n})^T\ .$$

As described in Section 4, the gradient at $t = 0$ can then be obtained by solving the system

$$y_t = M_t^T y_t$$

backwards in time, confer also [32, p. 281].

REFERENCES

[1] Vincent Arsigny, Olivier Commowick, Nicholas Ayache, and Xavier Pennec, *A fast and Log-Euclidean polyaffine framework for locally linear registration*, J. Math. Imaging Vis., 33 (2009), pp. 222–238.

[2] Vincent Arsigny, Olivier Commowick, Xavier Pennec, and Nicholas Ayache, *A Log-Euclidean framework for statistics on diffeomorphisms*, in MICCAI 2006, 2006, pp. 924–931.

[3] Vincent Arsigny, Xavier Pennec, and Nicholas Ayache, *Polyrigid and polyaffine transformations: A novel geometrical tool to deal with non-rigid deformations – application to the registration of histological slices*, Medical Image Analysis, 9 (2005), pp. 507–523.

[4] M. Faisal Beg, Michael I. Miller, Alain Trouvé, and Laurent Younes, *Computing large deformation metric mappings via geodesic flows of diffeomorphisms*, IJCV, 61 (2005), pp. 139–157.

[5] F. L. Bookstein, *Linear methods for nonlinear maps: Procrustes fits, thin-plate splines, and the biometric analysis of shape variability*, in Brain warping, Academic Press, 1999, pp. 157–181.

[6] Yan Cao, M. I Miller, R. L Winslow, and L. Younes, *Large deformation diffeomorphic metric mapping of vector fields*, IEEE Transactions on Medical Imaging, 24 (2005), pp. 1216–1230.

[7] GE Christensen, RD Rabbitt, and MI Miller, *Deformable templates using large deformation kinematics*, Image Processing, IEEE Transactions on, 5 (2002).

[8] Colin J Cotter and Darryl D Holm, *Singular solutions, momentum maps and computational anatomy*, nlin/0605020, (2006).

[9] Sune Darkner and Jon Sporring, *Generalized partial volume: An inferior density estimator to parzen windows for normalized mutual information*, in Information Processing in Medical Imaging, Gábor Székely and Horst K. Hahn, eds., vol. 6801, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 436–447.

[10] Paul Dupuis, Ulf Grenander, and Michael I Miller, *Variational problems on flows of diffeomorphisms for image matching*, (1998).

[11] Stanley Durrleman, Marcel Prastawa, Guido Gerig, and Sarang Joshi, *Optimal data-driven sparse parameterization of diffeomorphisms for population analysis*, Information Processing in Medical Imaging: Proceedings of the ... Conference, 22 (2011), pp. 123–134. PMID: 21761651.

[12] Gregory E. Fasshauer and Qi Ye, *Reproducing kernels of generalized sobolev spaces via a green function approach with distributional operators*, Numerische Mathematik, (2011).

[13] N. C. Fox, E. K. Warrington, P. A. Freeborough, P. Hartikainen, A. M. Kennedy, J. M. Stevens, and M. N. Rossor, *Presymptomatic hippocampal atrophy in alzheimer's disease*, Brain, 119 (1996), pp. 2001 –2007.

[14] Ulf Grenander, *General Pattern Theory: A Mathematical Study of Regular Structures*, Oxford University Press, USA, Feb. 1994.

[15] Monica Hernandez, Matias Bossa, and Salvador Olmos, *Registration of anatomical images using paths of diffeomorphisms parameterized with stationary vector field flows*, International Journal of Computer Vision, 85 (2009), pp. 291–306.

[16] Clifford R. Jack, Ronald C. Petersen, Yue Cheng Xu, Stephen C. Waring, Peter C. O'Brien, Eric G. Tangalos, Glenn E. Smith, Robert J. Ivnik, and Emre Kokmen, *Medial temporal atrophy on MRI in normal aging and very mild alzheimer's disease*, Neurology, 49 (1997), pp. 786 –794.

[17] Daniel S Marcus, Anthony F Fotenos, John G Csernansky, John C Morris, and Randy L Buckner, *Open access series of imaging studies: longitudinal MRI data in nondemented and demented older adults*, Journal of Cognitive Neuroscience, 22 (2010), pp. 2677–2684. PMID: 19929323.

[18] X. Pennec, R. Stefanescu, V. Arsigny, P. Fillard, and N. Ayache, *Riemannian elasticity: A statistical regularization framework for non-linear registration*, in MICCAI 2005, 2005, pp. 943–950.

[19] J. P.W Pluim, J. B.A Maintz, and M. A Viergever, *Image registration by maximization of combined mutual information and gradient information*, IEEE Transactions on Medical Imaging, 19 (2000), pp. 809–814.

[20] Alexis Roche, Grégoire Malandain, Xavier Pennec, and Nicholas Ayache, *The correlation ratio as a new similarity measure for multimodal image registration*, in Proceedings of the First International Conference on Medical Image Computing and Computer-Assisted Intervention, MICCAI '98, Springer-Verlag, 1998, p. 1115–1124. ACM ID: 709612.

[21] Alexis Roche, Xavier Pennec, Grégoire Malandain, and Nicholas Ayache, *Rigid registration of 3D ultrasound with MR images: a new approach combining intensity and gradient information*, IEEE Transactions on Medical Imaging, 20 (2001), pp. 1038—1049.

[22] D Rueckert, L I Sonoda, C Hayes, D L Hill, M O Leach, and D J Hawkes, *Nonrigid registration using free-form deformations: application to breast MR images*, IEEE Transactions on Medical Imaging, 18 (1999), pp. 712–721. PMID: 10534053.

[23] Christof Seiler, Xavier Pennec, and Mauricio Reyes, *Geometry-Aware multiscale image registration via OBBTree-Based polyaffine Log-Demons*, in Medical Image Computing and

Computer-Assisted Intervention - MICCAI, Toronto, Canada, 2011.

[24] Stefan Sommer, Mads Nielsen, Francois Lauze, and Xavier Pennec, *A Multi-Scale kernel bundle for LDDMM: towards sparse deformation description across space and scales*, in IPMI 2011, Springer, 2011.

[25] Stefan Sommer, M. Nielsen, and X. Pennec, *Sparsity and scale: Compact representations of deformation for diffeomorphic registration*, in MMBIA at WACV 2012, 2012.

[26] J.-P. Thirion, *Image matching as a diffusion process: an analogy with maxwell's demons*, Medical Image Analysis, 2 (1998), pp. 243–260.

[27] Alain Trouvé, *An infinite dimensional group approach for physics based models in patterns recognition*, 1995.

[28] M. Vaillant, M.I. Miller, L. Younes, and A. Trouvé, *Statistics on diffeomorphisms via tangent space representations*, NeuroImage, 23 (2004), pp. S161–S169.

[29] Tom Vercauteren, Xavier Pennec, Aymeric Perchant, and Nicholas Ayache, *Diffeomorphic demons: efficient non-parametric image registration*, NeuroImage, 45 (2009), pp. 61–72.

[30] William M Wells, III, Paul Viola, and Ron Kikinis, *Multi-Modal volume registration by maximization of mutual information*, (1996).

[31] Younes, *Constrained diffeomorphic shape evolution*, submitted to foundations Comp Math, (2011).

[32] Laurent Younes, *Shapes and Diffeomorphisms*, Springer, 2010.

[33] Ding-Xuan Zhou, *Derivative reproducing properties for kernel methods in learning theory*, Journal of Computational and Applied Mathematics, 220 (2008), pp. 456–463.

[34] Xiahai Zhuang, S. Arridge, D. J Hawkes, and S. Ourselin, *A nonrigid registration framework using spatially encoded mutual information and Free-Form deformations*, IEEE Transactions on Medical Imaging, 30 (2011), pp. 1819–1828.