A GEODESIC LANDMARK SHOOTING ALGORITHM FOR TEMPLATE MATCHING AND ITS APPLICATIONS

ROBERTO CAMASSA, DONGYANG KUANG, AND LONG LEE

Abstract. We present an efficient landmark shooting algorithm for template matching and its applications. The novelties of the algorithm include the use of a constant matrix to update the search direction of the geodesic shooting, instead of the traditional methods of forward-backward integration for updating the gradient or Newton's optimization, and the use of a non-smooth conic kernel for the particle system that accelerates the convergence of matching. To investigate the usage of the output quantities computed along the warping algorithm, such as the Hamiltonian metric and the momentum field, we introduce a multiscale decomposition method that separates the scales/components of the momentum and the Hamiltonian metric associated with the deformation. We numerically explore the potential of using the decomposed Hamiltonian metric and momentum vectors as input feature vectors into neural networks for clustering/classification analysis. The results of our numerical experiments are encouraging.

keywords: landmark, geodesic shooting, template matching, particle system, deformation, multiscale decomposition, Hamiltonian metric, neural networks.

1. Introduction

Researches in template deformation have been prosperous in the last two decades. Starting from Grenander's deformable template models [20], the study of Riemannian geometry of groups of diffeomorphisms and geodesics on manifolds have produced valuable theoretical results as well as actual algorithms in applications [3,14,26,30,41,42,44,55]. Template matching, or template deformation, is a common tool used in shape analysis. The many applications of shape analysis include image registration, pattern recognition, biomedical image analysis, morphometry, database retrieval, surveillance, biometrics, military target recognition and general computer vision [13, 32, 47, 50, 52].

The Euler-Poincaré (EP) equations, also called the Euler equations for diffeomorphisms, are of general interest as evolution equations on Riemannian manifolds endowed with Sobolev metrics [23,38]. Template matching can be formulated as finding the shortest or least expensive path of continuous deformation of one geometric object (reference template) into another one (target template). In this context, the time-dependent deformation process is the so-called geodesic evolution, and the derivation of the geodesic evolution equations falls into the Euler-Poincaré theory, which produces the EP equations [24, 26, 41]. The links between the EP equations and template deformation are well established [3,24,30,40,41,44], there are also some studies using of EP equations as numerical algorithms for template matching like in [2,39,41,42]. The paper [43] gives a detailed review of the history not limiting to EP equations.

Mathematically the EP equations describe geodesic motion on the diffeomorphism group, and are equivalent to the Euler-Lagrangian equations arising from minimizing an energy function defined for the deformation paths. A practical application for template matching is computational anatomy (CA) [9, 41]. For medical images in CA, instead of intensity on a mesh grid (pixel), geometry features of the medical images can be discretized into a set of the so-called *landmark points*. Therefore a template matching problem, in terms of landmark points,

Date: August 8, 2016.

becomes a landmark-matching problem. That is, given two collections of points $X_1, \cdots, X_N$ and $Y_1, \cdots, Y_N$, the matching problem is to find a time-dependent diffeomorphic path $\boldsymbol{\phi}(\cdot)$ that costs the minimum energy, such that $Y_k = \boldsymbol{\phi}(X_k)$, for $k = 1, \cdots N$ [24, 44]. To link the landmark-matching process to the EP equations, it is known that by casting the EP equations in the Lagrangian (or characteristics) form, i.e. following the characteristic variables, one can formulate EP equations as a finite-dimensional particle system of ordinary different equations (ODEs) [7]. This system of ODEs, referred to as the N-particle finite-dimensional dynamical system, or *N-particle system*, has two variables, the position variable, representing the positions of the particles, and the momentum variable, representing the momenta that drive the motion of the particles. Therefore for the full diffeomorphism group, a finite set of landmark points on the landmark space can be represented by the N-particles of the EP equations. The collections of the points $\{X_1, \cdots, X_N\}$ and $\{Y_1, \cdots, Y_N\}$ are the position variable of the N-particle system at two different times, respectively, and the landmark-matching problem is to find the *initial* momenta of the particles that drive the motion of the particles from one position $\{X_1, \cdots, X_N\}$ to another $\{Y_1, \cdots, Y_N\}$. This idea of finding the proper initial momenta to match the evolution of landmark points of two different templates turns the usual initial value problem of the N-particle system into a conventional boundary value problem of template matching. Shooting methods were introduced in the literature for finding the initial momenta (velocities), including methods of using Newton's optimization [2, 9, 39] and methods of forward-backward integration for updating the gradient [28, 29, 54]. For the review of shape deformation analysis from the optimal control viewpoint, we refer the reader to [1, 43].

In this paper, instead of the traditional methods of forward-backward integration for updating the gradient, or Newton's optimization, we introduce an efficient method to update the search direction of the geodesic shooting algorithm. Furthermore, we observe through a numerical example that the use of a non-smooth Green's function as the kernel of the N-particle system could accelerate the convergence of our shooting algorithm. We also illustrate that when combining with classifiers such as a multi-layer neural network, the Hamiltonian and momentum vector calculated from the proposed algorithm between two templates can serve as good features for clustering and classifying landmark templates.

2. Problem setting

The formulation by Grenander et al. [14, 20, 22] models the variations between two shapes or image templates by the action of Lie groups (diffeomorphisms) on manifolds. Template matching adopting this approach essentially considers an optimization problem with constraint:

$$\begin{aligned} &\min_{\boldsymbol{\phi}} \int_0^1 E(t) \\ &\text{where } I_1 = \boldsymbol{\phi}(\boldsymbol{u}; I_0, 1) \end{aligned} \tag{1}$$

In this setting, I_0 and I_1 are the point-distributions of the two shapes in a manifold M, where I_0 is called the reference template and I_1 is called the target template. $\int_0^1 E(t)$ is a functional introduced in the tangent bundle TM, usually defined as the energy required for carrying I_0 to I_1 along a certain trajectory in M [44]. The geodesic map $\boldsymbol{\phi}(\boldsymbol{u}; I_0, t)$ takes the initial value $\boldsymbol{u} \in T_{I_0}M$ as input carrying I_0 to another point in M along the geodesic at time t with respect to E. For exact image matching, the kinetic energy at time t is defined by

$$E(t) = \frac{1}{2}\|\boldsymbol{u}\|_{\mathcal{L}}^2 \tag{2}$$

where $\boldsymbol{u}$ defines the velocity field for the geodesic flow $\boldsymbol{\phi}$, $\mathcal{L}$ is an inertia operator of the form of $\mathcal{L} = (\boldsymbol{I} - \alpha^2\Delta)^\nu$, where $\boldsymbol{I}$ is the identity matrix, Δ is the Laplacian, α is constant depending

on the resolution level, and $\nu > 0$ is a constant depending on the metric used, i.e. the order of Sobolev norm. The norm in tangent space is introduced as:

$$\|\boldsymbol{u}\|_{\mathcal{L}}^2 =< \boldsymbol{u}, \boldsymbol{u} >_{\mathcal{L}} =< \mathcal{L}\boldsymbol{u}, \boldsymbol{u} >= \int_D \mathcal{L}\boldsymbol{u} \cdot \boldsymbol{u} \, d\boldsymbol{x}, \tag{3}$$

where D is a compact subset, and

$$\boldsymbol{u}(x,t) = \frac{\partial \boldsymbol{\phi}}{\partial t}\left(\boldsymbol{\phi}^{-1}(x,t), t\right). \tag{4}$$

Note that due to the complexity of geometric structure of the infinite dimensional manifolds, normally in practice only the finite dimensional case is considered [25,42]. The problem setting stated above is a generalization of the finite dimensional problem we consider in this paper.

Traditionally, for image matching (intensity assigned to a mesh grid), or landmark matching (point distribution for the geometrical features of images), finding the minimum-energy deformation path is a boundary value problem, for which the image templates I_0 and I_1 are treated as two end points of a cylinder domain $M \times [0,1]$. Under this constraint, the optimization process updates the current curve (flow) with the two fixed end points I_0 and I_1, by using methods such as the steep (gradient) descent, to obtain the minimizer of the given energy function in Eq. (1), The use of this diffeomorphisms (mappings) approach for analyzing images or shapes is mathematically sound and has been very successful [3,30,41,42,53], but a major limitation here is the high computational cost [47].

In the literature, numerical implementation for diffeomorphisms includes the strategy of using the shooting error to improve the original guess for initial velocity field by evolving the error backwards using the adjoint equation [54]. Inspired by the strategy, in this paper, we introduce a geodesic shooting algorithm. The algorithm treats the template matching problem as an initial value problem with an unknown initial condition. Given a guess of the initial velocity $\boldsymbol{u}$ (or momenta $\mathcal{L}\boldsymbol{u}$), a sequence of approximated target templates, $I_1^{(k)}$, $k = 0, 1, \cdots$ are generated by solving the N-particle system of the EP equations (also the Euler-Lagrange equation of Eq. (1) [44]). These approximated templates are used to obtain the corrections for updating a better guess. Through this iterative process, the initial velocities (or momenta) that carry the landmark points of the reference template to that of the target template can be established. The initial condition (momenta), when combined with particle system (11), can be treated as a representation for the deformation. Hence, information about the actually deformation can be analyzed through this momentum representation.

An algorithm, akin to the proposed algorithm in this paper, utilizing the N-particle system of the EP equations was introduced by McLachlan and Marsland for image registration [39]. For this algorithm, the initial conditions (momenta) that carry the landmarks from the reference template to target one are found by minimizing the Euclidean distance between the target template and the simulations. The minimization problem was solved by a sub-space trust region method based on the interior-reflective Newton method, provided by a MATLAB optimization package, and the metric of the N-particle system is the smooth Gaussian metric. Similar to the traditional minimization algorithms for template matching, this algorithm suffers from slow convergence and high computational cost, in particular for large numbers of landmarks [39].

The major difference between the proposed algorithm and the aforementioned ones is that instead of calculating the complicated gradient as searching direction or solving the adjoint equation in different problem settings, we assume a simple fixed format for updating the initial guess. In particular, we use a constant matrix to update the search direction. One advantage of this strategy is that with properly assumed conditions, parameters for this update can be estimated through some simple analysis.

In addition to using a constant matrix to update the search direction, we use a non-smooth kernel (the conical shape) for the N-particle system that reduces the interaction between particles and accelerates the convergence. Our results show that for certain planar landmark matching problems, the proposed algorithms converge relatively fast without the implementation of a fast-multipole method for large numbers of landmarks. Similar studies of variation of the kernel for fast computations was also seen in [28].

3. The EP equations and the Green's functions

By using index notation with Einstein convention on sums over repeated indexes for the (column) vectors, $\boldsymbol{m} \equiv \{m_i\}_{i=1}^n$ and $\boldsymbol{u} \equiv \left\{u^j\right\}_{j=1}^n$, the EP equations can be written as

$$\partial_t m_i + u^j \partial_j m_i + m_j \partial_i u^j + m_i \partial_j u^j = 0\,, \tag{5}$$

with $t \in \mathbb{R}^+$, $\boldsymbol{x}, \boldsymbol{u}$ and $\boldsymbol{m} \in \mathbb{R}^n$, and spatial partial derivatives are labeled by coordinate index. The velocity $\boldsymbol{u}$ and the momentum variable $\boldsymbol{m}$ are formally related by an elliptic operator $\mathcal{L}$

$$\boldsymbol{m} = \mathcal{L}\boldsymbol{u}\,. \tag{6}$$

Let $\mathcal{L}$ be a self-adjoint operator $\mathcal{L} \equiv \mathcal{L}^\nu$, where $\mathcal{L}^\nu$ is defined by

$$\mathcal{L}^\nu = (\boldsymbol{I} - \alpha^2 \nabla^2)^\nu, \tag{7}$$

parametrized by α^2 and power $\nu > 0$. Here $\boldsymbol{I}$ is the identity matrix. For any $\nu > 0$, including non-integer values, equation (6) is defined in the Fourier space

$$\begin{aligned} &\hat{\boldsymbol{u}} = (\hat{\mathcal{L}^\nu})^{-1}\hat{\boldsymbol{m}}, \quad \text{where } (\hat{\mathcal{L}^\nu})^{-1} = \frac{1}{(1+\alpha^2|\boldsymbol{k}|^2)^\nu}, \\ &|\boldsymbol{k}| = \sqrt{k_1^2 + k_2^2 \cdots + k_n^2}, \end{aligned} \tag{8}$$

where k_i is the i^{th} wavenumber. Since $\mathcal{L}^\nu$ is rotationally invariant and diagonal, then $\boldsymbol{G}(\boldsymbol{x}) = G_{\nu-n/2}(|\boldsymbol{x}|)\boldsymbol{I}$ for a scalar function $G_{\nu-n/2}$, with $|\boldsymbol{x}| = \sqrt{x_1^2 + x_2^2 + \cdot + x_n^2}$, where n is the dimension. The scalar Green function $G_{\nu-n/2}$ is then written as

$$G_{\nu-n/2}(|\boldsymbol{x}|) = \frac{2^{n/2-\nu}}{(2\pi\alpha)^{n/2}\alpha^\nu \Gamma(\nu)} |\boldsymbol{x}|^{\nu-n/2} K_{\nu-n/2}\left(\frac{|\boldsymbol{x}|}{\alpha}\right), \tag{9}$$

where $K_{\nu-n/2}$ is the modified Bessel function of the second kind of order $\nu - n/2$ and $\Gamma(\nu)$ is the usual notation for the Gamma function [44]. A notable special parametric choice is the two-dimensional Green function's for $\alpha = 1$ and $\nu = 3/2$, for which it takes the simple form

$$G_{1/2}(|\boldsymbol{x}|) = \frac{1}{2\pi} e^{-|\boldsymbol{x}|}\,. \tag{10}$$

We dub this Green's function as "conon," due to the spatially conical shape of the function. In general, for $n = 2$ (two-dimensional space), the regularity of the Green's function $2\pi G_{\nu-1}(r)$ is described as follows.

(1) For the range $1/4 < \nu \leq 1$ the Green's function $G_{\nu-1}(|\boldsymbol{x}|)$ is unbounded.
(2) For the range $1 < \nu < 3/2$ the function is bounded but non-differentiable at the peak, with the radial derivative suffering an infinite jump there (cusp).
(3) At $\nu = 3/2$, the jump in radial derivative becomes finite.
(4) For the range $3/2 < \nu \leq 2$ the derivative of the function is continuous, but with a infinite second derivative at the peak.
(5) Similar intervals can be defined for higher smoothness properties of the solution. In particular, for $2 < \nu < \infty$ the second derivative of the function is continuous.

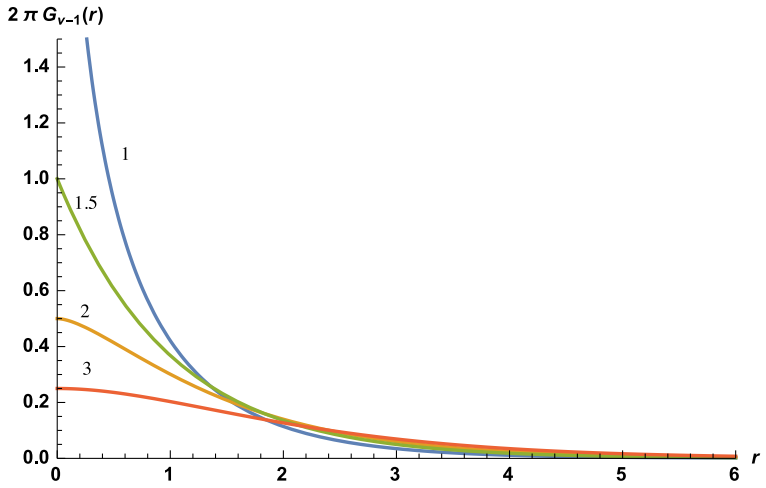


FIGURE 1. Plots of $2\pi G_{\nu-1}(r)$ for $\nu = 1, 1.5, 2$ and 3, where $G_{\nu-1}(r)$ is the two-dimensional Green's functions of the Yukawa operator $\mathcal{L}^\nu$. $\alpha = 1$ in the plots.

The function $2\pi G_{\nu-1}(r)$ for the critical values $\nu = 1, 1.5, 2$, and 3, respectively is plotted in Fig. 1.

Green's functions in the kernels of the N-particle system act like weight functions. From Figure 1, we see that for a small ν value, the Green's function decays faster than that with a large ν value. This indicates that for a non-smooth Green's function, a particle has strong influence on other particles only when those particles are close enough to this particle. On the other hand, for a smooth Green's function, the influence of a particle to other particles spreads further away. Based on our numerical experiments, the choice of the metric $\mathcal{H}^\nu$ (corresponding to the Green's function $G_{\nu-n/2}$) has an impact on the convergence rate of the geodesic shooting algorithms. In particular, we found that the metric $\mathcal{H}^{3/2}$ corresponding to the conical Green's function, $G_{1/2}$, provides the best convergence rate among the tested metrics.

Note 1. By the definition of diffeomorphism, a C^1 kernel is required for diffeomorphic properties of the transformation. The deformation using non-smooth kernels, such as $G_{1/2}$, does not satisfy the definition, since the deformation group associated with the kernel is in $C^0(\mathcal{R}^n, \mathcal{R}^m)$. Micheli et al. [35] studied the diffeomorphism of different kernels through a decomposition of the kernel into curl free and divergence free components, but the case $G_{1/2}$ or kernel with lower regularity is not included in their discussion. The existence and uniqueness of the variational problem or the geodesic with respect to the kernel of lower regularity, such as $G_{1/2}$, remains an open question and have not been thoroughly studied to our best knowledge. Furthermore, the $G_{1/2}$ kernel is smooth everywhere except at the origin where the first derivative has a finite jump. Although theoretical study is required to understand the analytical properties of the non-smooth kernels in template matching, we suspect the analysis of smooth kernels may still hold for this kernel in some finite time, especially when the number of particles is not large and the particles are not very close to each other. Moreover, properties of geodesics in landmark space can be complicated and unpredictable, as shown in [34], that the uniqueness of the geodesic is not guaranteed, even for smooth kernels.

We note that most of the previous work in the literature, including [39], used the smooth Gaussian kernel, G_∞, corresponding to the metric $\mathcal{H}^\infty$ in their algorithms. We also note that metrics with $\nu < 1.5$ are not suitable for our problems, since for those Green's functions, due

to the low regularities, some kind of mollification is needed for evaluating the kernels of the N-particle system. We investigate the convergence property of the metrics $\mathcal{H}^\nu$ in Section 7.

3.1. **The EP equations and the N-particle system.** Previously we have shown that the EP equation (5) is the Eulerian counterpart of the following Lagrangian formulation [7]:

$$\begin{aligned}\frac{d\boldsymbol{q}}{dt} &= \int_{\mathbb{R}^n} G_{\nu-n/2}\big(|\boldsymbol{q}(\boldsymbol{\xi},t)-\boldsymbol{q}(\boldsymbol{\eta},t)|\big)\,\boldsymbol{p}(\boldsymbol{\eta},t)\,dV_\eta\,,\\ \frac{d\boldsymbol{p}}{dt} &= -\int_{\mathbb{R}^n} G'_{\nu-n/2}\big(|\boldsymbol{q}(\boldsymbol{\xi},t)-\boldsymbol{q}(\boldsymbol{\eta},t)|\big)\frac{\boldsymbol{q}(\boldsymbol{\xi},t)-\boldsymbol{q}(\boldsymbol{\eta},t)}{|\boldsymbol{q}(\boldsymbol{\xi},t)-\boldsymbol{q}(\boldsymbol{\eta},t)|}\boldsymbol{p}(\boldsymbol{\xi},t)\cdot\boldsymbol{p}(\boldsymbol{\eta},t)\,dV_\eta\,,\end{aligned} \tag{11}$$

where $\boldsymbol{q}$ is the position variable and $\boldsymbol{p}$ is the momentum variable. In this form, the equations of motion are a canonical Hamiltonian system with respect to variational derivatives $\delta/\delta\boldsymbol{q}$ and $\delta/\delta\boldsymbol{p}$

$$\dot{\boldsymbol{q}}(\boldsymbol{\xi},t) = \frac{\delta\mathcal{H}}{\delta\boldsymbol{p}}\,,\qquad \dot{\boldsymbol{p}}(\boldsymbol{\xi},t) = -\frac{\delta\mathcal{H}}{\delta\boldsymbol{q}}\,, \tag{12}$$

of the Hamiltonian functional

$$\mathcal{H} \equiv \frac{1}{2}\int_{\mathbb{R}^n}\int_{\mathbb{R}^n} G_{\nu-n/2}\big(|\boldsymbol{q}(\boldsymbol{\xi},t)-\boldsymbol{q}(\boldsymbol{\eta},t)|\big)\,\boldsymbol{p}(\boldsymbol{\xi},t)\cdot\boldsymbol{p}(\boldsymbol{\eta},t)\,dV_\xi\,dV_\eta\,. \tag{13}$$

Discretizing the above integral-differential equations and absorbing the grid sizes dx and dy into the momentum variable p, yields the N-particle system

$$\begin{aligned}\frac{d\boldsymbol{q}_i}{dt} &= \sum_{j=1}^N G_{\nu-1}(|\boldsymbol{q}_i-\boldsymbol{q}_j|)\boldsymbol{p}_j,\\ \frac{d\boldsymbol{p}_i}{dt} &= -\sum_{\substack{j=1\\ j\neq i}}^N(\boldsymbol{p}_i\cdot\boldsymbol{p}_j)G'_{\nu-1}(|\boldsymbol{q}_i-\boldsymbol{q}_j|)\frac{\boldsymbol{q}_i-\boldsymbol{q}_j}{|\boldsymbol{q}_i-\boldsymbol{q}_j|},\end{aligned} \tag{14}$$

and the discrete Hamiltonian

$$\mathcal{H} = \frac{1}{2}\sum_{i=1}^N\sum_{j=1}^N(\boldsymbol{p}_i\cdot\boldsymbol{p}_j)G_{\nu-1}(\boldsymbol{q}_i-\boldsymbol{q}_j). \tag{15}$$

Note that the velocity $\boldsymbol{u}(x,t)$ can be reconstructed at any time t by

$$\boldsymbol{u}(\boldsymbol{x},t) = \sum_{j=1}^N G_{\nu-1}(|\boldsymbol{x}-\boldsymbol{q}_j(t)|)\boldsymbol{p}_j(t). \tag{16}$$

From Eqs. (2), (3), (15), and (16), we obtain

$$\begin{aligned}E &= \frac{1}{2}\int\mathcal{L}^\nu\sum_{i=1}^N G_{\nu-1}(\boldsymbol{x}-\boldsymbol{q}_i)\boldsymbol{p}_i\cdot\sum_{j=1}^N G_{\nu-1}(\boldsymbol{x}-\boldsymbol{q}_j)\boldsymbol{p}_j\;d\boldsymbol{x}\\ &= \frac{1}{2}\sum_{i=1}^N\sum_{j=1}^N\int\mathcal{L}^\nu G_{\nu-1}(\boldsymbol{x}-\boldsymbol{q}_i)\boldsymbol{p}_i\cdot G_{\nu-1}(\boldsymbol{x}-\boldsymbol{q}_j)\boldsymbol{p}_j\;d\boldsymbol{x}\\ &= \frac{1}{2}\sum_{i=1}^N\sum_{j=1}^N\int\boldsymbol{\delta}(\boldsymbol{x}-\boldsymbol{q}_i)\boldsymbol{p}_i\cdot G_{\nu-1}(\boldsymbol{x}-\boldsymbol{q}_j)\boldsymbol{p}_j d\boldsymbol{x}\\ &= \mathcal{H},\end{aligned} \tag{17}$$

where $\boldsymbol{\delta}$ is the Dirac delta function.

As mentioned in the last section, deformation associated with a low regularity kernel is hard to analyzed due to its complicated geometric nature. While from the particle system of view, there are work done with respect to the existence of solution for initial value problem of it as well as landmark dynamics like in [7, 8, 24, 34, 38]. In this setting, Hamiltonian metric could still be defined through the ODE system and will be conserved during deformation. It corresponds to the initial momentum vector that carries the flow from I_0 to I_1 and is used as a measurement that measures the deformation between I_0 and I_1 (or the logarithm of a conditional likelihood under the probability setting [14]) in the tangent plane. We remark that the second-order Runge method (also called the modified Euler method) is used as our time integrator for (14). While this explicit time integrator is not symplectic, numerically the Hamiltonian (and hence the initial momenta) is preserved quite well (within 0.1% error at the end of our time integration for most of our numerical experiments). The symplectic time integrators, such as the implicit mid-point method or the second-order Stömer-Verlet scheme are under our investigation for improving the convergence of the shooting algorithm introduced in the next section.

4. Algorithm for Landmark Shooting

We introduce a template matching algorithm in this section. We consider the exact landmark matching problem

$$I_1 = \boldsymbol{\phi}(\boldsymbol{u}; I_0, \Delta t), \tag{18}$$

where I_0 and I_1 are the shape or image templates represented by landmark points, and $\boldsymbol{\phi}$ is the geodesic flow determined by the corresponding Euler-Lagrangian equation. After the Euler-Lagrangian equation is recast in the Euler-Poincaré equation, the optimization problem becomes finding the initial condition for the initial value problem (14). In Eq. (18), $\boldsymbol{u}_0$ is the initial vector field we intend to find, and Δt is usually set to be 1 as a normalization. If the manifold $\boldsymbol{M}$ is geodesic complete, for given I_1, locally the existence and uniqueness of such $\boldsymbol{u}_0$ is guaranteed, and Eq. (18) is an one-to-one mapping [44]. The present algorithm is based on the idea of updating the velocity flow by directly using the difference between the landmarks of the target template and the solutions of the EP equations in each iteration. This process is reminiscent of the steep-decent method for the minimization problem.

The iterative process is terminated at the k^{th} iteration when $I_1^{(k)}$, calculated from the velocity $\boldsymbol{u}_0^{(k)}$, satisfies $||I_1 - I_1^{(k)}|| < \epsilon$ for some prescribed threshold ϵ. Here I_1 is a set of selected landmarks, $I_1^{(k)}$ consists of the landmarks $\boldsymbol{q}_i^{(k)}$ that are solutions of the particle system of the EP equations. It is worth noting that the new prediction of the initial velocity (momentum) is a linear combination of the previous one and a correction term from the observation quantity or pre-defined measurement. The geodesic landmark shooting algorithms are described in *Algorithm 1.*

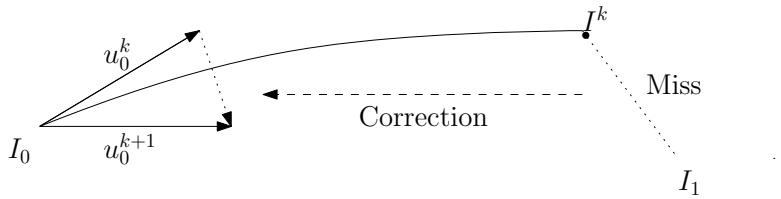


Figure 2. Schematic interpretation for *Algorithm 1.*

Note 2. (1) There are certain freedom for choices of placing the landmarks onto the image templates (i.e. selecting the position variable $\boldsymbol{q}$ of the particles in the particle system)

Algorithm 1 Geodesic shooting algorithm for template matching

Select corresponding landmarks (particles) between two shapes I_0 and I_1;

Initialize the vector field $\boldsymbol{u}_0^{(0)}$ associated with the landmark points of I_0 with an initial guess of momenta $\boldsymbol{p}^{(0)}$;

for $k = 0, 1, 2, \cdots$ until convergence **do**

1. Evolve the N-particle system of EP equations (14) to some fixed time ($t = 1$), using the momenta associated with $\boldsymbol{u}_0^{(k)}$ to obtain the approximate landmarks $\boldsymbol{q}_i^{(k)}$ for $I_1^{(k)}$, where $\boldsymbol{q}_i$ is the i^{th} landmark, $i = 1, \ldots, N$.
2. If the difference $||I_1 - I_1^{(k)}|| < \epsilon$ for some norm, then stop.
3. The vector field is dynamically updated by $\boldsymbol{u}_0^{(k+1)} = \boldsymbol{u}_0^{(k)} + M^{(k)} \cdot (I_1 - I_1^{(k)})$, where $M^{(k)}$ is an unknown matrix which we'll give a possible choice in Section 5 so that the algorithm converges at least locally.
4. Update the momenta of the particle system $\boldsymbol{p}_0^{(k+1)}$ from $\boldsymbol{u}_0^{(k+1)}$.

end for

depending on different applications. For instance, landmarks could be placed along the outline of a object, only at the points-of-interest or significance in the image, such as edges and corners, or we could use the discrepancy of images and put landmarks at places where the two images do not match, based on the objective function. The most "efficient" representation (i.e. a representation carries most of the information with the least number of landmarks.) varies, depending on different focuses of the actual applications, although it is known in the literature [10] that the presence of more landmarks on locations where large deformation occurs, improves the matching results considerably in the case of G_∞ kernel, whereas fewer than necessary particles on those locations results in a slower convergence or a failure of matching. In this paper, for all our experiments in shape matching, we place adequate numbers of landmarks on the contour curves, so that the curvatures of a contour are adequately represented, in the sense that each segment of the contour between two landmarks is roughly linear, while our algorithm produces numerically convergent results for prescribed convergence criteria.

(2) The algorithm introduced here assumes that the chosen landmarks for the templates are appropriately labeled such that each $\boldsymbol{q}_i \in I_0$ will be carried to $\boldsymbol{q}_i \in I_1$ by the flow when the matching process is complete. It is worth noting that there are algorithms developed for unlabeled landmark matching. We refer readers to the literature [18, 19] for more information.

(3) Our initial guess for the velocity field is normally $\boldsymbol{u}^{(0)} = 0$. This is a reasonable initialization to test the stability of our algorithm. That is, if the problem is viewed under a stochastic setting: I_0 and I_1 is two realizations of certain process, their difference will be a zero mean process, $E(I_0 - I_1) = 0$. Thus, zero is an unbiased estimation to begin with.

(4) Another way to find the velocity vector field is to "directly" solve the non-linear equation $I_1 - \boldsymbol{\phi}(\boldsymbol{u}; I_0, \Delta t) = 0$ for $\boldsymbol{u}$, by some iterative methods, such as the Newton iteration. Comparison in our numerical experiments shows that our choice of the optimizer is advantageous over the "direct" approach.

5. Local convergence theorem

In this section we present a local convergence theorem for our shooting algorithm. Based on this theorem, we select a matrix $M^{(k)}$ in *Algorithm 1* for our numerical experiments. For

fixed reference shape I_0 and time $\Delta t = 1$ needed to let I_0 flow along the geodesic, define $F : \mathrm{T}_p\mathrm{M} \mapsto M$, $F(\boldsymbol{u}) = \boldsymbol{\phi}(\boldsymbol{u}; I_0, 1)$ for fixed $p \in \mathrm{M}$. Let $\hat{F} : \mathbb{R}^d \mapsto \mathbb{R}^d$ be the coordinatization of F. That is, if φ is a smooth patch map from M to $\mathbb{R}^d$ and Ψ is a smooth patch map from T_pM to $\mathbb{R}^d$ (which usually takes the induced map: $D\varphi$). With these two patch maps, $\hat{F} = \varphi \circ F \circ \Psi^{-1}$. J_F could be calculated for every regular point in coordinates by $J_F = J_\varphi^{-1} J_{\hat{F}} J_\Psi$. In the rest of this section, $\hat{\cdot}$ represents the corresponding variable in coordinates.

Theorem 1. *Suppose that the residual in the approximation of first-order Taylor series expansion of $\hat{F}$ is uniformly bounded in the sense that*

$$\begin{aligned} \hat{F}(v) - \hat{F}(V) =& J_{\hat{F}} \cdot (v - V) + R(v - V), \\ & \text{where } ||R(v - V))|| \leq B||v - V||, \end{aligned} \tag{19}$$

for some constant $B \in \mathbb{R}$ in a region $\Omega := \{v, V \in \mathbb{R}^d : ||v - V|| < r\}$ with certain constant $r \in \mathbb{R}$. Here $R(\cdot)$ is the remainder of the Taylor series and $J_{\hat{F}}$ is the Jacobian matrix of $\hat{F}$. Further assume $J_{\hat{F}}$ is non-singular with $||(J_{\hat{F}}[\hat{\boldsymbol{u}}_0])^{-1}|| \leq C$ and $CB \neq 1$.

Then there exists a choice of matrix $M^{(k)}$ for the velocity update in coordinates under φ: $\hat{\boldsymbol{u}}_0^{(k+1)} = \hat{\boldsymbol{u}}_0^{(k)} + M^{(k)} \cdot (\hat{I}_1 - \hat{I}_1^{(k)})$, and a sufficiently large number N, for which the error $||\hat{\boldsymbol{e}}_k|| = ||\hat{\boldsymbol{u}}_0 - \hat{\boldsymbol{u}}_0^{(k)}|| \leq \epsilon$ for $k > N$ and $\epsilon \ll r$. Here $\hat{\boldsymbol{u}}_0$ is the true initial velocity for the reference template, and $\hat{\boldsymbol{u}}_0^{(k)}$ is the k^{th} iteration of the velocity vector found by Algorithm 1.

Proof: Let $\boldsymbol{I}$ be the identity matrix. Consider the $(k+1)^{th}$ update for the velocity field in coordinates: $\hat{\boldsymbol{u}}_0^{(k+1)} = \hat{\boldsymbol{u}}_0^{(k)} + M^{(k)} \cdot (\hat{I}_1 - \hat{I}_1^{(k)})$ in *Algorithm 1*, where $\hat{F}(\hat{\boldsymbol{u}}_0) = \hat{I}_1$, and $\hat{I}_1^{(k)}$ is the solution of the EP equations in the k^{th} iteration.

If condition (19) is satisfied, the bounded first-order Taylor expansion for $\hat{F}$ on the convex open domain $\{v, V \in \mathbb{R}^d : ||v - V|| < r\}$ for some constant $r \in \mathbb{R}$ at the k^{th} iteration yields the following calculations:

$$\hat{\boldsymbol{u}}_0^{(k+1)} - \hat{\boldsymbol{u}}_0^{(k)} = M^{(k)} \cdot \left(\hat{I}_1 - \hat{I}_1^{(k)}\right) \tag{20}$$

$$\Rightarrow \quad \left(\hat{\boldsymbol{u}}_0^{(k+1)} - \hat{\boldsymbol{u}}_0\right) - \left(\hat{\boldsymbol{u}}_0^{(k)} - \hat{\boldsymbol{u}}_0\right) = M^{(k)} \cdot \left(\hat{F}(\hat{\boldsymbol{u}}_0) - \hat{F}(\hat{\boldsymbol{u}}_0^{(k)})\right) \tag{21}$$

$$\Rightarrow \quad \hat{\boldsymbol{e}}_{k+1} - \hat{\boldsymbol{e}}_k = -M^{(k)} \cdot J_{\hat{F}}[\hat{\boldsymbol{u}}_0] \cdot \hat{\boldsymbol{e}}_k - M^{(k)} \cdot R(\hat{\boldsymbol{e}}_k) \tag{22}$$

$$\Rightarrow \quad \hat{\boldsymbol{e}}_{k+1} = \left(\boldsymbol{I} - M^{(k)} \cdot J_{\hat{F}}[\hat{\boldsymbol{u}}_0]\right) \cdot \hat{\boldsymbol{e}}_k - M^{(k)} \cdot R(\hat{\boldsymbol{e}}_k) \tag{23}$$

$$\Rightarrow \quad ||\hat{\boldsymbol{e}}_{k+1}|| \leq || \boldsymbol{I} - M^{(k)} \cdot J_{\hat{F}}[\hat{\boldsymbol{u}}_0] \, || \, ||\hat{\boldsymbol{e}}_k|| + B||M^{(k)}|| \, ||\hat{\boldsymbol{e}}_k|| \tag{24}$$

$$\Rightarrow \quad \frac{||\hat{\boldsymbol{e}}_{k+1}||}{||\hat{\boldsymbol{e}}_k||} \leq ||\boldsymbol{I} - M^{(k)} \cdot J_{\hat{F}}[\hat{\boldsymbol{u}}_0] \, || + B||M^{(k)}|| \tag{25}$$

Since $J_{\hat{F}}[\hat{\boldsymbol{u}}_0]$ is non-singular, $(J_{\hat{F}}[\hat{\boldsymbol{u}}_0])^{-1}$ exists. To simplify the analysis, we choose $M^{(k)} = h(J_{\hat{F}}[\hat{\boldsymbol{u}}_0])^{-1}$, for which we only need to prove that there exists such h so that the algorithm converges.

Eq. (25) now becomes

$$\frac{||\hat{e}_{k+1}||}{||\hat{e}_k||} \leq ||\boldsymbol{I} - h\boldsymbol{I}|| + hCB \leq |1 - h| + hCB := \beta. \tag{26}$$

If $0 < \beta < 1$ for all iterations, we have

$$||\hat{e}_{k+1}|| < \beta||\hat{e}_k|| < \cdots < \beta^{k+1}||\hat{e}_0|| \to 0, \text{ as } k \to \infty, \tag{27}$$

and hence the algorithm converges.

We now discuss the existence of such h.

Case 1: For $h \leq 1$, condition $0 < \beta < 1 \Leftrightarrow -1 < (CB-1)h < 0$.
This implies if $0 < CB < 1$, we must have $0 < h < -\frac{1}{CB-1}$ and $h \leq 1$. Therefore $0 < h \leq 1$.
Otherwise, if $CB > 1$, h satisfies $-\frac{1}{CB-1} < h < 0$.

Case 2: For $h > 1$, condition $0 < \beta < 1 \Leftrightarrow 1 < h(1+CB) < 2 \Leftrightarrow \frac{1}{1+CB} < h < \frac{2}{1+CB}$. In this case, we need $0 < CB < 1$ to have a possible choice for h, namely $1 < h < \frac{2}{1+CB}$.

After summarizing the above discussion, we see that if $0 < CB < 1$, we can pick $0 < h < \frac{2}{1+CB}$; on the other hand, if $CB > 1$, we can pick $-\frac{1}{CB-1} < h < 0$. In either case, there always exists a choice for h so that $0 < \beta < 1$. Hence the algorithm will converge under given conditions for the choice of $M^{(k)} = h(J_{\hat{F}}[\hat{\boldsymbol{u}}_0])^{-1}$. □

Note 3. See Section 11 for the derivation of an theoretic optimal $M^{(k)}$ from the view point of minimizing variance estimation.

In practice, it is too cumbersome and time consuming to numerically compute $(J_{\hat{F}}[\hat{\boldsymbol{u}}_0])^{-1}$. Instead, we let $M^{(k)}$ be a constant matrix in the form of $M^{(k)} = M = h\boldsymbol{I}$ with a certain choice of h for all k. By going through the similar analysis as that in the proof, we obtain a rough estimation of this choice of h. In this case,

$$\beta(h) = ||I - hJ_{\hat{F}}[\hat{\boldsymbol{u}}_0]|| + Bh. \tag{28}$$

Let λ_i be the eigenvalues of matrix $J_{\hat{F}}[\hat{\boldsymbol{u}}_0]$. The regularity of the matrix ensures $\lambda_i \neq 0$ for all i. By using a rough bound $||I - hJ_{\hat{F}}[\hat{\boldsymbol{u}}_0]|| \leq \max\left(\left|1 - h\lambda_i\right|\right)$ with the spectral radius as the matrix norm and assuming this maximum is achieved for $\lambda_e = a + bi$ in the finite dimensional case, we have the following results:

In order to obtain $0 < \beta < 1$, we can set $0 < \sqrt{(1-ah)^2 + (bh)^2} + Bh < 1$, provided $h < \min(\frac{2a}{a^2+b^2}, \frac{1}{B})$. Suppose $h > 0$, we shall solve the following inequality for h:

$$(1-ah)^2 + (bh)^2 < (1-Bh)^2. \tag{29}$$

Case 1: $(a^2 + b^2 - B^2) > 0$. Solving Eq. (29) yields $h < \frac{2(a-B)}{a^2+b^2-B^2}$, provided the additional condition $a > B$. In this case, we can pick $0 < h < \min(\frac{2(a-B)}{a^2+b^2-B^2}, \frac{1}{B}, \frac{2a}{a^2+b^2})$.

Case 2: $(a^2 + b^2 - B^2) < 0$. The condition indicates $B > |a|$. In this case, $h > \frac{2(a-B)}{a^2+b^2-B^2} > 0$, so we have to pick an h, so that $\frac{2(a-B)}{a^2+b^2-B^2} < h < \frac{1}{B}$. But since this requires $\frac{2(a-B)}{a^2+b^2-B^2} < \frac{1}{B} \Rightarrow (B-a)^2 + b^2 < 0$, there will be no possible choice for positive h.

Case 3: $(a^2 + b^2 - B^2) = 0$. This yields $B \geq |a|$, Eq. (29) will end up with $B < a$, which is also not possible.

From the above discussion, we see that for positive $h \in \left(0, \min(\frac{2(a-B)}{a^2+b^2-B^2}, \frac{1}{B}, \frac{2a}{a^2+b^2})\right)$ to exist, so that the constant matrix $M = h\boldsymbol{I}$ leads to the convergence of the shooting algorithm, at least one more restriction: $B < \mathrm{Re}(\lambda_e)$, should be further assumed for the problem.

The investigation of possible negative h is more tedious, we leave the detail in the Appendix. It is necessary to realize that since the above discussions are to find sufficient conditions/assumptions for possible value of h, certain h that are not included in above results may still make the algorithm converge. In the theorem, the restriction on B relies on the nature of actual deformation, which in turn depends on the geometric properties behind it. In landmark space $L(N)$, the geometry gets more complex as the number of landmark increase. This complexity causes difficulties in selecting h in applications, including the facts that the admissible range is usually short and require certain additional assumptions on the problem, and it is hard to select an optimal h without using trial-and-error in advance, as shown in our numerical experiments in Section 6.

Although solid theoretic results have not been obtained due to the aforementioned complexity, we have some empirical observations with respect to the choice of h that can be used as a guide for selecting h: When the number of landmarks sampled in a template is overwhelmingly large that causes the landmarks stay too close to each other, a practical choice of h may not be possible without resorting trial-and-error. On the other hand, if the template is not over sampled, i.e. within a small domain, the number of landmarks is small but enough to represent the feature of the template, a wide range of choice of h can be obtained. Moreover, one can calibrate the scalar parameter α^2 of the kernel if no convergence is observed for the choice of h in some cases, as illustrated in Section 7.

6. Numerical Experiments

In this section, we present several numerical experiments and study their numerical convergence properties to illustrate the advantages of *Algorithm 1*. In this section, unless specified otherwise, 64 landmarks ($N = 64$) are used to represent the outline of a given shape. We use the tolerance $\epsilon = 10^{-3}$ as our stopping criterion. The metric of the self-adjoint operator $\mathcal{L}$ is $\mathcal{H}^{3/2}$ with $\alpha = 1$ (i.e. $\mathcal{L}^{3/2} = (\boldsymbol{I} - \nabla^2)^{3/2}$), for which the Green's function is normalized and is written as $G_{1/2}(|\boldsymbol{x}|) = e^{-|\boldsymbol{x}|}$ ($\nu = \frac{3}{2}$). We use the Hamiltonian (15) to measure the distance between the reference template and the target one. We refer readers to the literature [41] for other invariant metrics and their properties. Finally, the numerical examples are computed by using Matlab(2011a) on Windows 7 Home Premium (SP1) platform with Intel® i5-2430 processor @ 2.40GHz. The deformation of the background grid is visualized by using a modified version of a visualizer [15] that deforms the embedded grid together with our computation.

6.1. **Example 1.** We construct the deformation between a circle and a rotated-and-shifted ellipse by using the proposed *Algorithm 1* in this example. The reference template is a circle centered at the origin with radius 2, and the target template is an ellipse with $a = 4, b = 1$ rotated by 45° and shifted in positive x direction by 1 unit. This example illustrates the flexibility of *Algorithm 1* for handling deformations with translations and rotations. The reference template I_0 and the target template I_1 are described as follows.

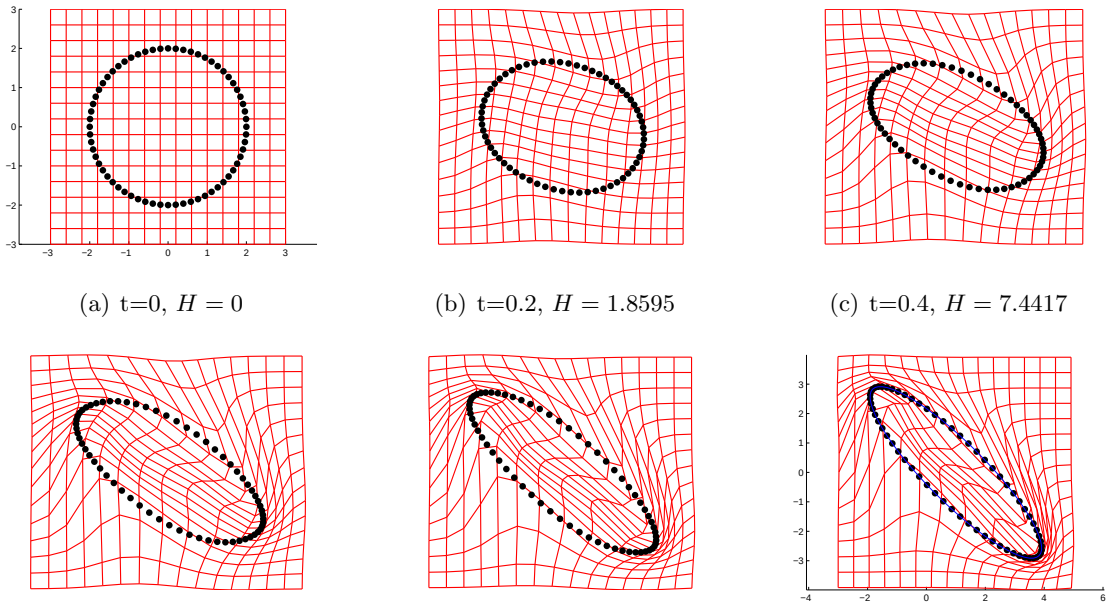
$$I_0 : x^2 + y^2 = 2^2, \tag{30}$$

$$I_1 : \begin{cases} x = & 4\cos(\frac{\pi}{4})\cos(\theta) + \sin(\frac{\pi}{4})\sin(\theta) + 1, \\ y = & \cos(\frac{\pi}{4})\sin(\theta) - 4\sin(\frac{\pi}{4})\cos(\theta). \end{cases} \tag{31}$$

The deformation from I_0 to I_1 goes through rotation, dilation and translation. The constant matrix $M = h\boldsymbol{I}$ with $h = 0.3$ is used for *Algorithm 1* in this example.

To demonstrate the efficiency of *Algorithm 1*, Table 1 compares *Algorithm 1* with the approach of solving the non-linear equation $I_1 - \boldsymbol{\phi}(\boldsymbol{u}; I_0, \Delta t) = 0$ for $\boldsymbol{u}$ directly by the MATLAB intrinsic function **fsolve**, a trust-region Newton algorithm, and a MATLAB implementation of the point LDDMM by Sommer [46] available at [15].

Table 1(a) shows the errors and elapsed CPU times by *Algorithm 1* for various N (the number of landmarks), Table 1(b) is the results by using **fsolve** when the prescribed stop criterion is achieved, and Table 1(c) records the results from the aforementioned LDDMM algorithm. Since the three algorithms have different stopping criteria, it is not possible to find a single stopping criterion for the comparison. Instead, we try to find some equivalent stopping criterion for each method, in hopes of a fair comparison. For example, we choose the stopping criterion 10^{-6} for **fsolve**, so that the algorithm generates the ℓ_2-norm error of the target landmark locations also less than 10^{-6}, the stopping criterion for *Algorithm 1*. We see that for this example when N is small, the elapsed CPU times of the *Algorithm 1* and **fsolve** are comparable, and are less than that of the point LDDMM method. However, when N



(a) t=0, $H = 0$ (b) t=0.2, $H = 1.8595$ (c) t=0.4, $H = 7.4417$

(d) t=0.6, $H = 16.7461$ (e) t=0.8, $H = 29.7594$ (f) t=1, H=46.5022

FIGURE 3. Deformation from a circle to a shifted and rotated ellipse. (a) is the reference template. (b)-(e) give the interpolative landmark shapes along the path of minimum energy found by *Algorithm 1*. The dots in (f) is the landmark distribution of the approximate target template, whereas the solid line is the actual target template.

increases, *Algorithm 1* and the point LDDMM method need less time to achieve their stopping criteria.

It is worth noting that the implementation of *Algorithm 1* is unoptimized while the point LDDMM method in (c) is optimized and run in parallel. It takes 197 iterations for *Algorithm 1* to reach the tolerance and complete the matching, while the point LDDMM method takes 282 iterations to complete. We remark that the ℓ_2-error of the target landmark locations the point LDDMM method in Table 1(c) is relatively large, because the stopping criterion for this method is not using the difference between the approximated and the exact target templates but the fluctuation of the objective functional value during the optimization, for which we also set to 10^{-6} in this example. It is shown in Table 1(a) that the computational cost for *Algorithm 1* is roughly $O(N^2)$.

6.2. **Example 2.** In the second example, we compute the deformation between a circle and a heart curve and illustrate some numerical properties of *Algorithm 1*. Similar to Example 1, the reference template is a circle defined by

$$I_0 : x^2 + y^2 = 2^2, \tag{32}$$

whereas the deformed target template is a 4^{th}-order heart curve defined as

$$I_1 : \begin{cases} x = & \frac{1}{5}(13\cos(\theta) - 5\cos(2\theta) - 2\cos(3\theta) - \cos(4\theta)) \\ y = & \frac{1}{5}(16\sin(\theta)^3). \end{cases} \tag{33}$$

The deformation involves the evolution of a convex region to a non-convex one, and the formation of curve singularities from a piece of smooth curve. This example demonstrates the

TABLE 1. Comparison between (a) *Algorithms 1*, (b) MATLAB **fsolve**, and (c) a point LDDMM method.

(a)

N	error(s)	Total time(s)
30	9.9032e-9	12.9654
40	4.5708e-10	28.5737
50	1.9258e-7	29.9143
60	8.8837e-8	43.8872

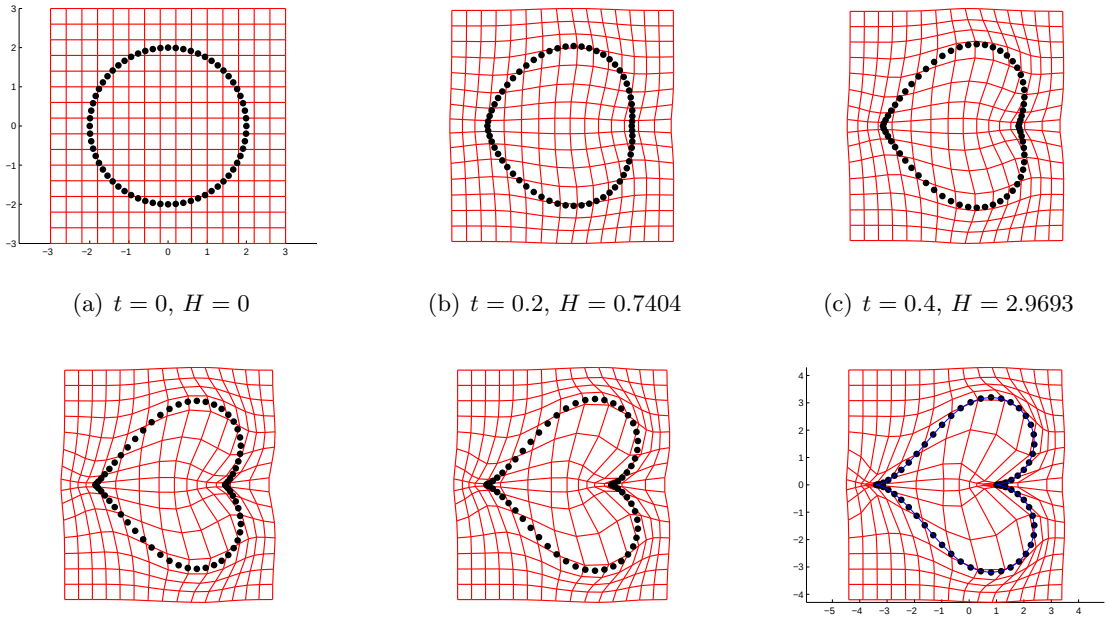
(b)

N	error(s)	Total time(s)
30	1.2006e-8	11.4370
40	5.1522e-10	26.2334
50	2.0985e-7	41.3252
60	1.1505e-7	77.0952

(c)

N	error(s)	Total time(s)
30	0.0046	28.8270
40	0.0045	34.3968
50	0.0052	43.7586
60	0.0042	54.7489

ability of the proposed algorithm to construct interpolative shapes between smooth (reference template) and sharp-edge (target template) planar curves.



(a) $t = 0$, $H = 0$ (b) $t = 0.2$, $H = 0.7404$ (c) $t = 0.4$, $H = 2.9693$

(d) $t = 0.6$, $H = 6.6782$ (e) $t = 0.8$, $H = 11.8702$ (f) $t = 1$, $H = 18.5728$

FIGURE 4. Deformation from a circle to a heart curve. (a) is the reference template. (b)-(e) are the interpolative shapes along the path of minimum energy found by the proposed algorithm. The dots in (f) is the landmark distribution of the approximated target template, whereas the solid line is the actual target template.

In real applications, the choice of the search length h and the number of landmarks N can be subtle for the performance of *Algorithm 1*. A optimal h should lead to the least number of

TABLE 2. The effects of different searching length h.

h	Total elapsed CPU time(sec)	# of iteration
0.2	7.5845	83
0.4	3.6137	40
0.6	2.3521	26
0.8	1.6251	18
1.0	3.4431	38
>1	NaN	NaN

iterations while maintaining convergence. As briefly discussed in Section 5, this optimal h is not easy to choose. Heuristically, one can use a subset of the whole template landmarks to test the performance of different candidates of h, and apply the optimal one to actual templates with more landmarks. To investigate optimal search length in this example, we fix the number of landmarks $N = 64$ in Example 2 and evaluate the required elapsed CPU times for various h, given a stopping criterion. Table 2 suggests that the algorithm converges faster for large h before the threshold, beyond which the algorithm begins to diverge. This is the typical behavior of steepest-descent type of methods. Choice of h is associated with actual problems as well as the metric used as discussed in last part of Section 5. In cases when one can not find available h for convergence, changing the form of metric should then be considered. Examples for this with different scaler parameters for metric $\mathcal{H}^\nu$ are demonstrated in the following section.

7. Convergence property related to the metric $\mathcal{H}^\nu$

The smooth Green's kernel associated with the metric $\mathcal{H}^\infty$, usually written as $G(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\frac{||\boldsymbol{x}-\boldsymbol{y}||^2}{\alpha^2})$, is commonly used for template matching algorithms in the literature. In this section, we numerically investigate the convergence property of *Algorithm 1*, associated with the search length h, the parameter α, and the metric $\mathcal{H}^\nu$. In particular, our study focuses on the comparison between the metrics of $\mathcal{H}^\infty$ (the smooth Gaussina kernel), $\mathcal{H}^2$ (the C^1 kernel), and $\mathcal{H}^{3/2}$ (the conic C^0 kernel). The smooth Gaussian kernel of $\mathcal{H}^\infty$ metric and the conic kernel of $\mathcal{H}^{3/2}$ metric have explicit forms of exponential functions that may result in certain computational advantages.

Using Example 2 as our test problem, Table 3 shows the iteration numbers for using various search lengths h and various parameters α (in the Yukawa operator) for the deformation. In the Table, (a) and (b) are the results by using the smooth Green's kernel of the metric $\mathcal{H}^\infty$, with the number of landmarks $N = 16$ and 32, respectively, (c) and (d) are the same calculations with the C^1 kernel of the $\mathcal{H}^2$ metric, and (e) and (f) are the result by using the conic kernel of $\mathcal{H}^{3/2}$ metric. The "$\times$" mark means either the iteration number exceeds 500 before the error reaches the stopping criterion, or the algorithm blows up before 500 iterations. From the table, we observe that using the conical metric $\mathcal{H}^{3/2}$ is advantageous for fast convergence in this example, especially for large N and/or large h. Moreover, if we look at Table 3(e) and (f), column by column, we observe that for the conical Green's kernel associated with the metric $\mathcal{H}^{3/2}$, the iterations are insensitive to the choice the parameter α^2 for a fixed search length h among the three tested kernels. Furthermore, the above calculations illustrate the potential advantage of the conic kernel of $\mathcal{H}^{3/2}$ when certain cusp structure appears in landmark templates, and the potential limitation of the smooth Gaussian kernel for *Algorithm 1* applied to deformations that require a large number of landmarks to specify details, such as the cusp structures in this example.

TABLE 3. Numerical convergence study. (a) $N = 16, \mathcal{H}^\infty$, (b) $N = 32, \mathcal{H}^\infty$, (c) $N = 16, \mathcal{H}^2$, (d) $N = 32, \mathcal{H}^2$, (e) $N = 16, \mathcal{H}^{3/2}$, (f) $N = 32, \mathcal{H}^{3/2}$. The tolerance is $\epsilon = 10^{-3}$.

(a)

	h=0.2	0.4	0.6	0.8	1.0
α^2 =0.2	61	37	32	48	×
0.4	53	25	17	15	18
0.6	52	25	19	14	×
0.8	59	34	22	×	×
1.0	73	37	×	×	×

(b)

	h=0.2	0.4	0.6	0.8	1.0
α^2 =0.2	227	×	×	×	×
0.4	×	×	×	×	×
0.6	×	×	×	×	×
0.8	×	×	×	×	×
1.0	×	×	×	×	×

(c)

	h = 0.2	0.4	0.6	0.8	1.0
α^2 =0.2	42	19	12	10	12
0.4	44	22	14	11	17
0.6	47	23	15	13	27
0.8	50	24	16	15	45
1.0	52	25	16	18	95

(d)

	h = 0.2	0.4	0.6	0.8	1.0
α^2 =0.2	89	43	27	19	×
0.4	110	53	35	25	×
0.6	119	58	38	27	×
0.8	123	60	39	29	×
1.0	126	62	40	42	×

(e)

	h = 0.2	0.4	0.6	0.8	1.0
α^2 =0.2	39	18	12	9	8
0.4	40	18	11	8	9
0.6	40	18	11	9	11
0.8	40	18	11	9	12
1.0	40	18	11	10	13

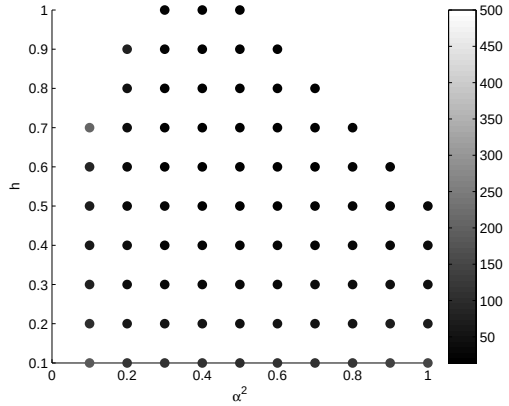
(f)

	h=0.2	0.4	0.6	0.8	1.0
α^2 =0.2	46	22	16	15	×
0.4	47	22	13	10	17
0.6	54	25	16	11	20
0.8	58	27	17	12	21
1.0	60	29	18	12	24

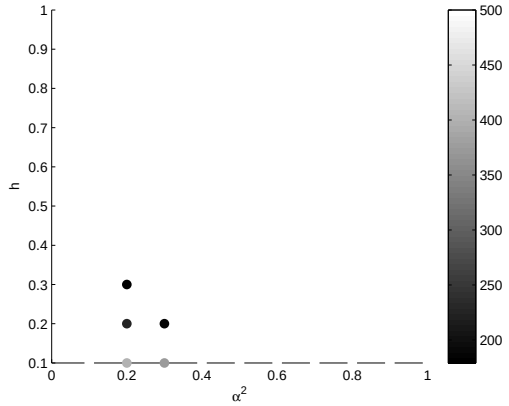
To better visualize the comparison, we place the convergence results in the $\alpha^2 \times h$ parameter space, using a grid size 0.1×0.1. A black dot indicates convergence, while a white dot suggests no convergence. The darker the dot is, the less iterations it takes to reach the tolerance. Figure 5 supports our previous conjecture that particles have a shorter interaction range with the non-smooth Green's kernel. It seems that the iteration number needed for convergence associated with the three test kernels can be roughly ranked as $\mathcal{H}^\infty > \mathcal{H}^2 > \mathcal{H}^{3/2}$. In terms of the choice of parameter, α^2, for the robustness of convergence, we also observe that $\mathcal{H}^{3/2}$ is superior than $\mathcal{H}^2$ and $\mathcal{H}^2$ is better than $\mathcal{H}^\infty$. We also remark that the above conclusions are purely empirical, obtained based on many numerical experiments that we have conducted. Our experiment suggests a kernel with lower regularity may have certain advantages in some specific situations, such as the conical kernel in this example. Theoretical studies of this non-smooth kernel for its unique properties in landmark template matching remains an open research question.

8. A procedure for multiscale decomposition of deformation

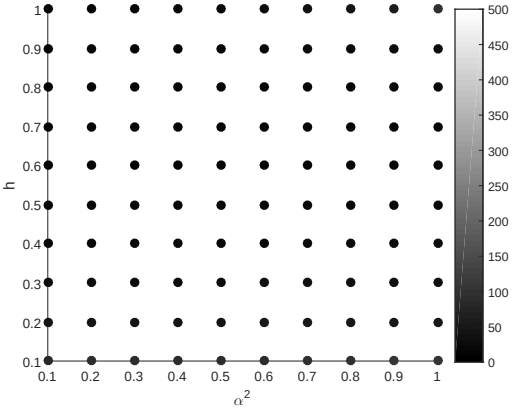
In the previous sections, we study a geodesic landmark shooting algorithm that interpolates deformations between two templates. In order to further explore the usage of this algorithm, in the following sections, we investigate the use of the algorithm in shape cluster/classifications. We study the multiscale decomposition of the Hamiltonian metric and the momentum variable obtained through the matching process, are used the decomposition as feature vectors in shape



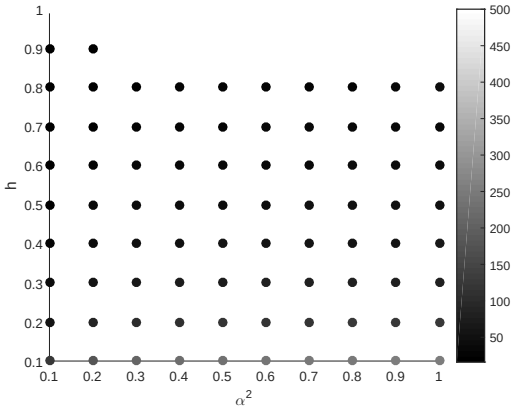
(a) $N = 16, \mathcal{H}^{\infty}$



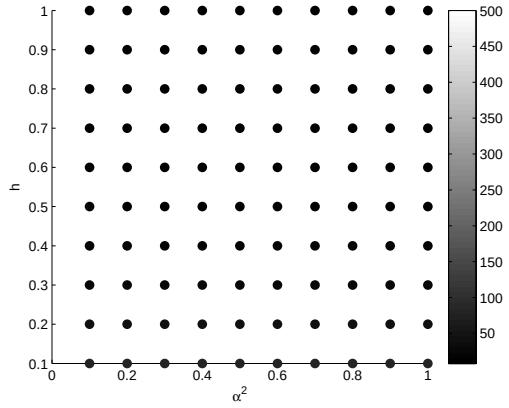
(b) $N = 32, \mathcal{H}^{\infty}$



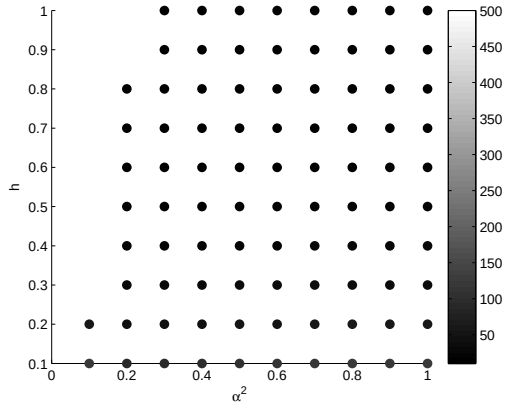
(c) $N = 16, \mathcal{H}^{2}$



(d) $N = 32, \mathcal{H}^{2}$



(e) $N = 16, \mathcal{H}^{3/2}$



(f) $N = 32, \mathcal{H}^{3/2}$

FIGURE 5. Numerical convergence study of the metrics $\mathcal{H}^{\infty}$, $\mathcal{H}^{2}$ and $\mathcal{H}^{3/2}$ in the $\alpha^2 \times h$ parameter space. The tolerance is $\epsilon = 10^{-3}$ in the maximum norm and the gray level represents the number of iterations.

cluster and classification. The aim of our numerical study is to show that combining our landmark shooting algorithm and the multiscale decomposition provides an alternative, when the conventional features used in the classical classification methods fail to provide satisfactory results.

The decomposition procedure that allows us to extract the components of deformations when combined with our geodesic shooting algorithm is described as follows. Recall that, under the discrete setting, the discrete Hamiltonian is computed by

$$\mathcal{H} = \frac{1}{2}\text{Trace}(\mathbf{P^T G P}), \tag{34}$$

where $\mathbf{P}$ is the initial momentum obtained from the shooting algorithm. Assume the eigenvalues and the corresponding eigenvectors of $\mathbf{G}$ are $\{\lambda_i\}$ and $\{\mathbf{v_i}\}$. Note that $\{\lambda_i\}$ are all positive since $\mathbf{G}$ is positive definite. We can rewrite the Hamiltonian:

$$\mathcal{H} = \frac{1}{2}\sum_{i=1}^{N} \lambda_i ||\mathbf{P}_i||^2, \tag{35}$$

where $\mathbf{P}_i$ is the projection of momentum $\mathbf{P}$ on $\mathbf{v}_i$ so that

$$\mathbf{P} = \sum_{i=1}^{N} \mathbf{P}_i = \sum_{i=1}^{N} \frac{< \mathbf{P}, \mathbf{v}_i >}{||\mathbf{v}_i||}\mathbf{v}_i. \tag{36}$$

Let $H_i = \frac{1}{2}\lambda_i ||\mathbf{P}_i||^2$ be the i^{th} component of Hamiltonian in the direction of $\mathbf{v}_i$. Without loss of generality, we assume Eq. (35) is written in descending order so that

$$\mathcal{H} = \sum_{i=1}^{N} H_i, \tag{37}$$

where $H_1 > H_2 > \cdots > H_N$. Since $\mathbf{P}$ can be treated as a representation of the deformation for a fixed reference template, Eq. (36) serves as a decomposition of the deformation. It is clear that quantities (H_i and $\boldsymbol{P}_i$) with smaller index contain information more globally related to the deformation. On the other hand, quantities with larger indexes are responsible for small calibration or local deformation. Hence, information that happens at different scales of the deformation can be investigated through the decomposed quantities with different indexes.

With the help of this decomposition, different scales of the deformation could be visualized separately. Finally, with a partial sum of $\mathbf{P}_i$, we can obtain an approximated deformation by evolving it using the particle system Eq. (11). We describe this procedure as follows.

(1) We apply the shooting algorithm, *Algorithm 1*, to I_0 and I_1 and obtain the momentum $\mathbf{P}$ that deforms I_0 to I_1. This $\mathbf{P}$ serves as a representation of the deformation under reference I_0.
(2) We decompose $\mathbf{P}$ and sort the result, according to their corresponding value of H_i with descending order, into the sum of $\mathbf{P}_i$. Each $\mathbf{P}_i$, when used as an initial condition in the particle system, represents a component of the total deformation with the Hamiltonian denoted by H_i and is used to visualize the corresponding deformation component. The smaller the index is, the more global information it should contain.
(3) An approximated deformation is constructed by taking a partial sum

$$\tilde{\mathbf{P}} = \sum_{i=1}^{k} \mathbf{P}_i, \ k < N,$$

and then this is used in the particle system (14) to evolve the reference template I_0. In the end of the evolution, we obtain an approximated target template $\tilde{\mathbf{Q}} = \mathbf{\Phi}(I_0, \tilde{\mathbf{P}}; 1)$.

We demonstrate the above procedure by the following example. Suppose we deform I_0: an ellipse ($a = 4, b = 2$) to I_1: a circle ($r = 3$). After the decomposition, the first three major components of the deformation are shown in Figure 6.

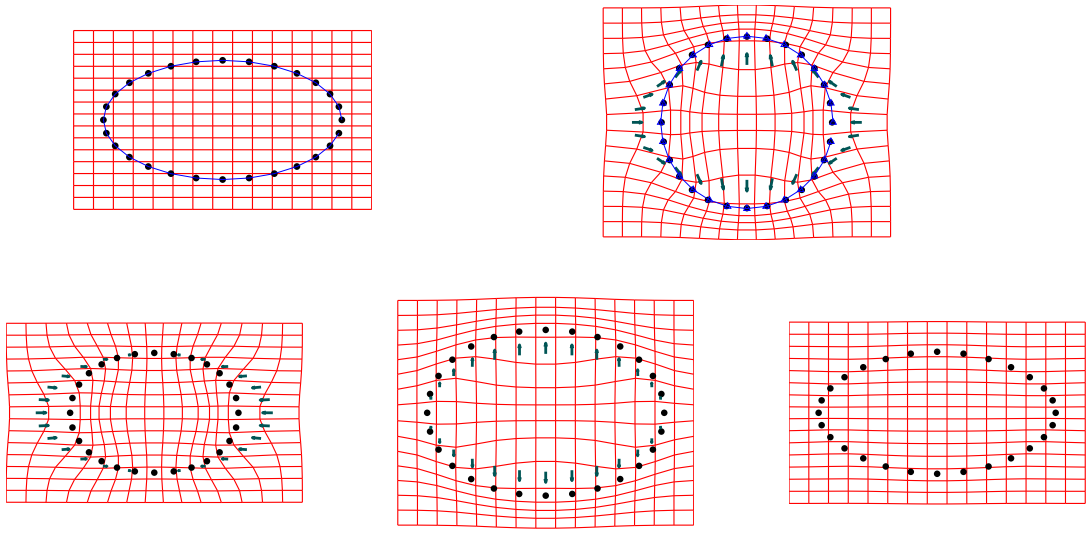


FIGURE 6. Deformation shown from the ellipse to the circle (first row), 28 landmark points are used. The first three major components are shown in the second row with those arrows indicate the associated momentum vectors. The total Hamiltonian is 5.0385, and the sum of the first three H_i is 5.0274.

The graphical description of these three major components agrees with our intuition about the deformation, i.e. the first major deformation is a compression in the x axis (55.16% of total $\mathcal{H}$), the second major deformation is a stretch in the y axis (44.38% of total $\mathcal{H}$), the third component barely deforms the reference (only 0.24% of total $\mathcal{H}$).

For a more complicated deformation, it would be more difficult to interpret each component of the deformation based on intuitions. Then this decomposition method is advantageous to shed light on how the different components of deformation combine together to form the whole one. In the next example, we show the first three major components of the deformation from the same ellipse to a heart shape curve in Figure 7. The three major deformations have H_i values $37.83\%, 28.20\%$ and 22.15% of the total Hamiltonian $\mathcal{H}$, respectively.

Figure 8 gives the full decomposition of $\mathcal{H}$ in examples shown in Figure 6 and 7. For the first example (Fig. 6), the first two components dominate the deformation, whereas for the second one (Fig. 7), the first three components appear to outstand others. However, when we use the first two components of the first example and the first three components of the second example to approximate the deformations, we discover that the former gives a very good approximation of the targeted circle, while the later does not recover the target heart shape entirely. This is because the heart shape curve has local structures that require more components to complete the deformation. The number of components included in an approximation determines the level of local structures that present in the approximation. The observation suggests that this decomposition procedure potentially can be used as a feature extraction tool for the task of clustering or classification.

It is worth pointing out this decomposition, though alike, is not quite a PCA approach. The method of PCA generally involves projecting observed data into eigenspaces of certain covariance matrix, while this decomposition does not involve calculations of covariance of

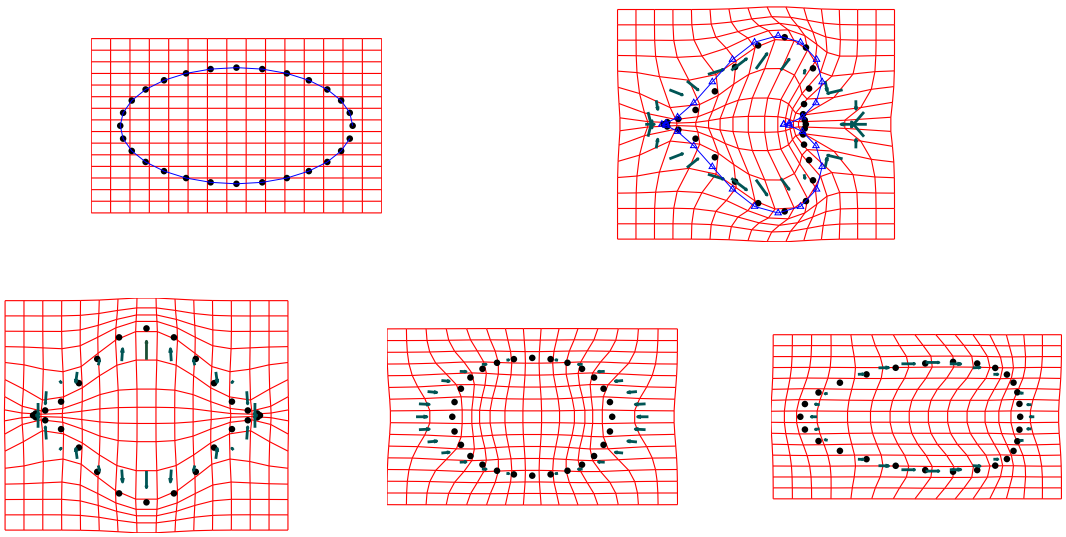


FIGURE 7. Deformation shown from the ellipse to the heart (first row), 28 landmark points are used. The first three major components are shown below. The arrows indicate the associated momentum vectors. The total Hamiltonian is 11.5270. The first three major deformation has a partial sum of Hamiltonian 10.1636.

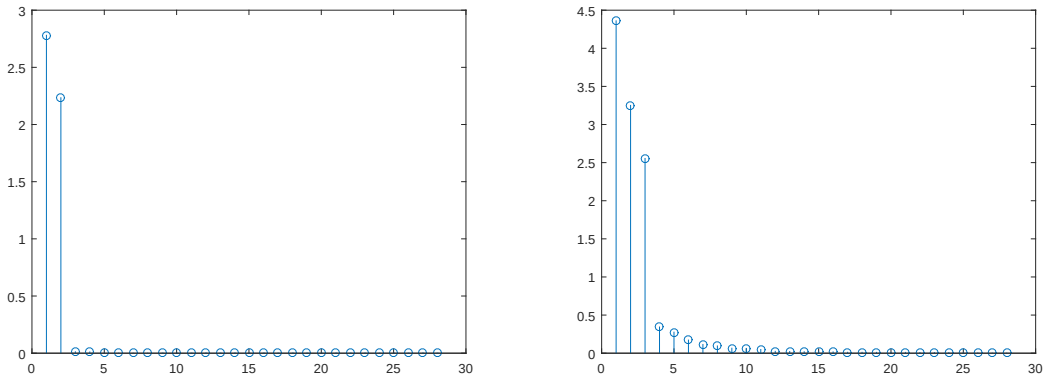


FIGURE 8. The decomposition of $\mathcal{H}$. The left panel is the decomposition of $\mathcal{H}$ when deforming the ellipse to the circle. The right panel shows the decomposition of $\mathcal{H}$ when deforming the same ellipse to the heart shape curve.

any kind but using a positive definite kernel associated with the Hamiltonian instead. We have shown that this decomposition help to decompose Hamiltonian $\mathcal{H}$ into different scale levels and its corresponding initial momentum components. If a landmark template has a small amount of outliers, we suspect the associated change of Hamiltonian only happens at a lower level, i.e the tailed terms of $\mathcal{H}_i$. Based on this idea, if we only choose the first few components for approximating the deformation, those unwanted outliers (outlying landmarks) could be eliminated from the dataset. Although theoretical analysis is required for any concrete conclusion, below we present an example to support the conjecture. In this example, we use the reference and target in Figure 6, but the target now has one outlier (top-left) or four outliers

(top-right). By the previous example, we have known that the deformation is dominated by the first two major components of the decomposition. We then pick the sum of first 2 major components as the input into the particle system to recover the "target" circle and the corresponding deformation. As we can see in the bottom row of Figure 9, the outlying landmarks are successfully removed in both cases.

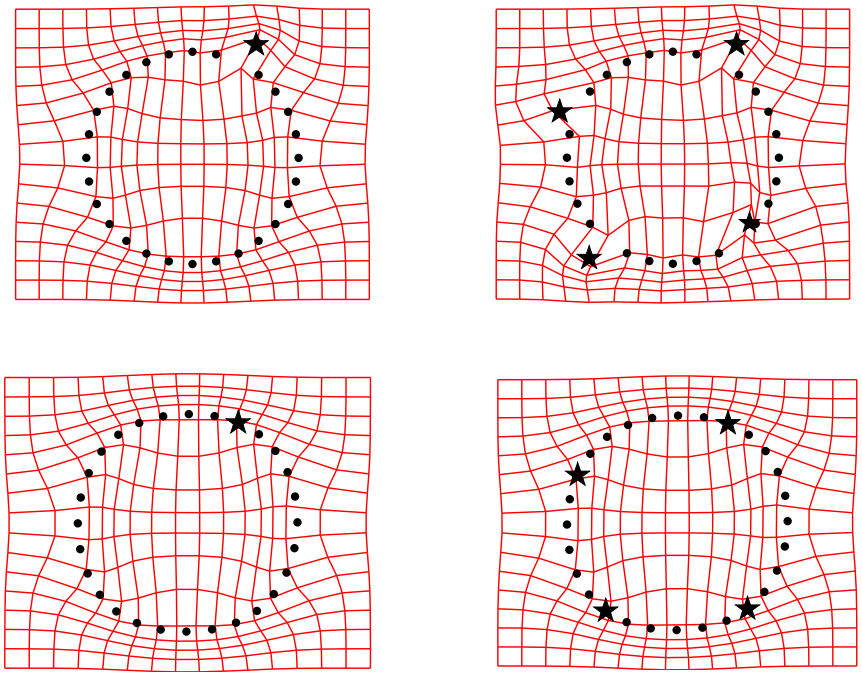


FIGURE 9. Deformations from ellipse to circle with outlying landmarks. The stars indicate outliers in each case (left column: one outlier; right column: four outliers). The first row shows the deformations of regular matching with all components. The second row shows the results of taking only the sum of first two major components of momenta from the regular matchings as the input into the particle system. In both cases, outliers are corrected.

9. Hamiltonian metric for clustering and classifying landmark templates

Since the Hamiltonian $\mathcal{H}$ is a metric, it could be used for the task of shape clustering or classification. In this section, we design an example to explore the potential of Hamiltonian metric in applications like shape clustering or classification.

We start with 11 shapes; each has 28 landmarks. The shapes are numbered and described as follows.

1: a heart shape curve; 2: a circle (r=3); 3: a circle(r=1); 4: an ellipse (a=2, b=4); 5: square(l=4) ; 6: a half circle (r=3) glued with a half ellipse (a=3,b=1); 7: an ellipse (a=4, b=2); 8: a hand ; 9: a key; 10: a different hand; 11: a different key.

A sample hand shape and a sample key shape are shown in Figure 10.

Shape # 2 is selected as the reference template from which other templates deform. We register shapes with respect to the reference template by *Algorithm 1.* The Hamiltonian $\mathcal{H}$ is

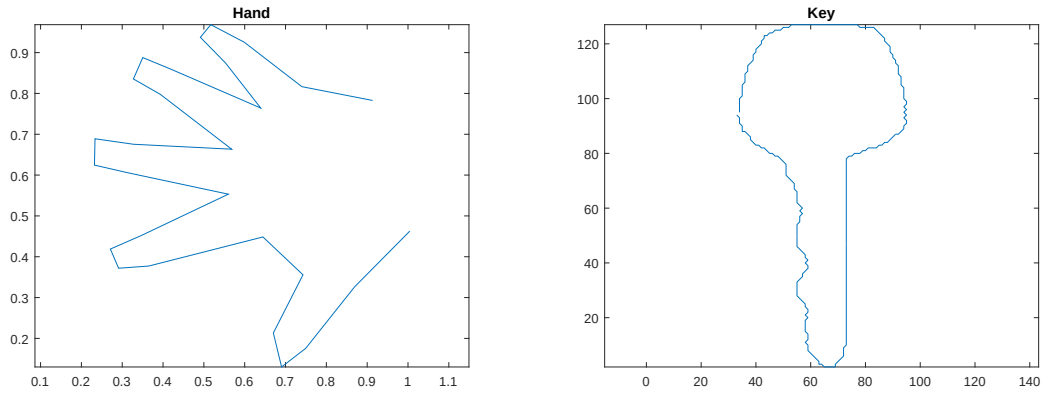


FIGURE 10. An example of the hand shape and the key shape used in the clustering experiment.

calculated for each registration. The reason for choosing shape # 2 as the reference template is because it is isotropic in all directions and we expect the comparison by using $\mathcal{H}$ for the registrations is a fair comparison.

A Procrustes fit [6] with respect to the reference template is applied to our shape class as a pre-process to mod all linear transformations (translation, orthogonal rotation, and scaling), so that $\mathcal{H}$ is a quantity related to the nonlinear deformation only. We denote the Hamiltonian between the reference template and shape # i $\mathcal{H}_i$.[1] We use $\mathcal{H}_i$ to grow a binary hierarchy cluster tree with different methods as shown in Fig. 11 for evaluating the performance of the Hamiltonian metric.

Fig. 11 shows the comparison of binary hierarchy cluster of test shapes by using *Algorithm 1* with $\mathcal{H}$ metric (left column) and Procrustes metric (right column). The horizontal axis marks the indexes for different shapes to be clustered, while the vertical axis marks the normalized distances. The lower two shapes (shape groups) are linked, the more alike these two shapes (shape groups) are. Three different methods (from top to bottom: average distance, nearest neighbor distance and weighted average distance) provided by Matlab's "dendrogram" function are computed. The reason of using three different methods is to illustrate that when the three methods all give similar results, it could suggest that the results are robust and they in principle reflect the intrinsic properties of the metric (or feature space) that is used for the clustering test. As we can see, in the left column of Fig. 11, the results obtained by using the $\mathcal{H}$ metric, the hand shape (shape #8 and #10 appears in one cluster) and key shape (shape #9 and #11 in another cluster) are all successfully distinguished. As a comparison, the right column of Fig. 11 that uses Procrustes distance as the metric tends to treat hand shape and key shape as one big cluster (shape #8 and #9 are clustered first, then #10 and #11 are added to this cluster). These differences between the two columns can be observed for the three different methods that are used to grow the hierarchy tree. This suggests that the results are robust, and they may truly reflect the better nature of the $\mathcal{H}$ metric for template clustering.

Note 4. A well-known limitation of landmark-based algorithms for clustering or classification (also a limitation for Procrustes-based methods) is that deformation depends on the landmark correspondence between reference and target templates. Therefore, for some applications, a landmark-free algorithm may be more appropriate [31].

[1] $\mathcal{H}_2 = 0$, since the reference is the same as the target.

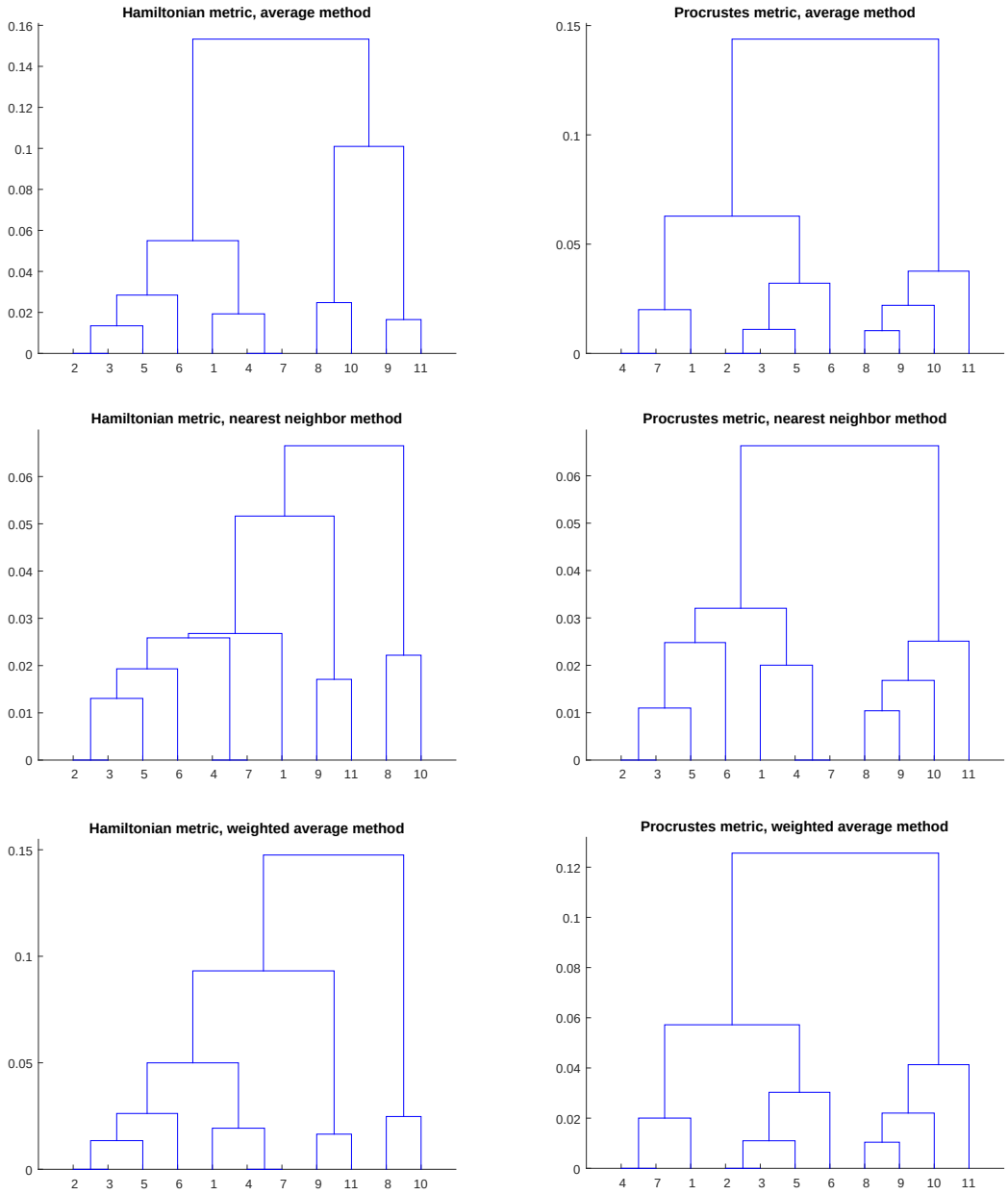


FIGURE 11. Binary hierarchy cluster of test shapes constructed by using *Algorithm 1* with $\mathcal{H}$ metric (left column) and Procrustes metric (right column). Clusters shown in the figure are robust under different methods, this suggests the above results may reflect the intrinsic properties of the $\mathcal{H}$ metric for template clustering.

9.1. **Decomposed $\mathcal{H}$ and momentum as feature vectors for classification analysis.** Our shooting algorithm gives Hamiltonian of the deformation and momentum representation of the target template. In Section 8, the global quantity Hamiltonian are decomposed into a sum of components which can include more information for describing the deformation. We have demonstrated through previous example that Hamiltonian metric can serve as one good option in clustering different shapes. In this section, we will evaluate the merits of the

decomposed Hamiltonian metric $(H_1, \cdots, H_N)$, as well as that of the momentum variable in two-dimensional (2D) shape classification.

To evaluate the merits of using the decomposed $\mathcal{H}$ as the feature vectors in a standard neural network for classification analysis, we compare the result of this strategy against that of the standard PCA method. The standard PCA method uses the momentum variable or the landmark locations to produce the feature vectors. The reason of using the standard PCA approach for comparison is because the standard PCA method is perhaps one of the most classical statistical data analysis procedure and has been proven effective in many applications. Moreover, using PCA approaches on tangent spaces for data analysis is very common and often recommended in the literature [9, 44, 47, 52]. Some specially designed variations of the method have also demonstrated the advantages in topics related to template matching and registration [56, 57]. Since our focus is to evaluate the performance of different features, a relatively simple but robust approach like the standard PCA method is suitable for the comparison.

In the example below, we use the outlines of four different fighters annotated by landmarks for our tests in classification. The shapes of these fighter planes are shown in Figure 12. The data were analyzed in [51] and can be downloaded at [16]. We will pick the major components of decomposed Hamiltonian as input feature vectors into a simple classical neural network [12] for classification analysis. For the momentum and landmark location data, we will apply a standard PCA method on the data, and then use the same number of PCA components as that of the decomposed Hamiltonian metric as the feature vectors fed into the neural network.

The four fighters are *Mirage, F14 with closed wings, F14 with open wings and F15*. Each fighter category has 30 outline templates. The original number of landmarks for each shape template roughly ranges from 300 to 1600. In order to conduct our shooting and decomposition procedure effectively, we sampled 50 points for each template, and use the first 5 components (i.e. $H_1, H_2, \cdots, H_5$) of the computed Hamiltonian metric as the input feature vector for a 5-10-4 neural network. We remark that the number of components used in the neural network is empirical in this example. By using trial-and-error, we learned that using 10 or more components of the decomposed Hamiltonian as the input vectors gave almost the identical result as that of using as few as 5 components. Another justification is that the first 5 components of decomposed Hamiltonian obtained by matching a pairwise different shapes selected from the 4 categories are the major components of the Hamiltonian. A more systematic and theoretic method for choosing the number of components is an open question and is currently under investigation.

The network is trained with stochastic training protocol and 0.02 as the stopping criteria [12]. 20 independent test batches are formed in our numerical experiment for evaluation. Within each batch, 20 out of the 30 landmark templates for each fighter will be randomly selected to form the training set and the rest 10 are used as the test set[2]. 20 independent runs[3] are conducted within each batch. The first template in the training set is the default reference template for the preprocessing Procrustes fit and *Algorithm 1* that produces the decomposition components. The performance of three mentioned features are summarized in Table 4, 5 and 6 separately. Entries listed are averages across all runs and batches, i.e. 400 independent tests. To read the tables, for example, the first row of Table 4 indicates that among the 400 independent tests, 95.9% the model plane Mirage is recognized correctly, 4.025% it is recognized as an F14 (closed-wings), 0.025% it is recognized as an F14 (open-wings), and 0.05% it is recognized as an F15.

[2]This means the training set is of size 80 and the test set is of size 40 for each batch.

[3]Each batch will have an randomly generated independent training and test set for the neural network.

		Prediction			
		Mirage	F14(close)	F14(open)	F15
Actual	Mirage	95.90%	4.025%	0.025%	0.05%
	F14(close)	2.45%	97.00%	0%	0.55%
	F14(open)	0%	0.05%	97.175%	2.775%
	F15	0.725%	0.40%	6.275%	92.60%
Overall Correction Rate: 95.669%					

TABLE 4. Classification results of the 4 planes using decomposed $\mathcal{H}$ as feature vectors. Results are calculated by averaging all the 400 independent tests.

		Prediction			
		Mirage	F14(close)	F14(open)	F15
Actual	Mirage	89.100%	10.775%	0.075%	0.050%
	F14(close)	11.875%	87.375%	0.5%	0.25%
	F14(open)	0.350%	0.480%	98.075%	1.100%
	F15	0.950%	0.825%	3.25%	94.975%
Overall Correction Rate: 92.38%					

TABLE 5. Classification results of the 4 planes using standard PCA on residual vectors after Procrustes procedure of the original location data. Results are calculated by averaging all the 400 independent tests.

In the following discussion, we denote the decomposed $\mathcal{H}$ as "feature (a)", the components of the location data for the PCA as "feature (b)", and the components of the extracted momentum representation for the PCA as "feature (c)". Among the three tested features, feature (b) has the worst overall recognition rate. However, obtaining feature (b) is least expensive

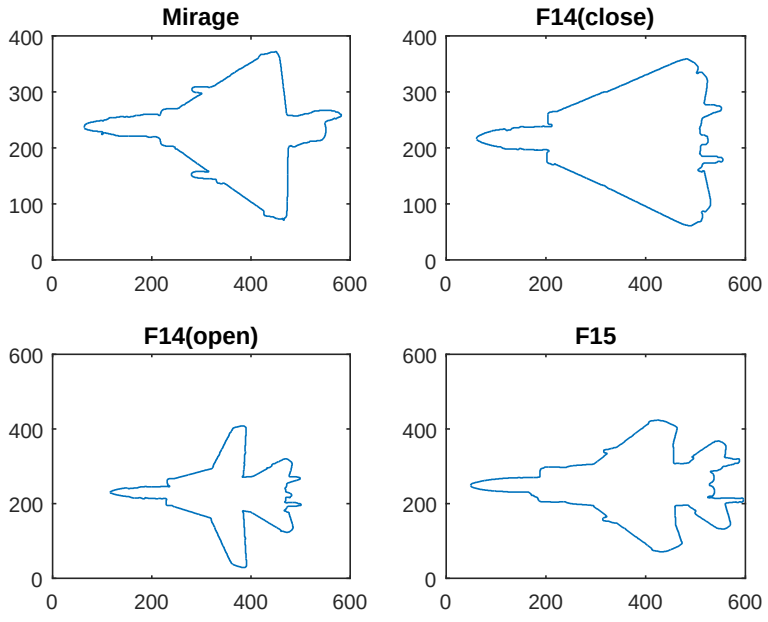


FIGURE 12. Sample shapes of the 4 fighters.

		Prediction			
		Mirage	F14(close)	F14(open)	F15
Actual	Mirage	98.25%	0%	1.225%	0.525%
	F14(close)	0.65%	99.35%	0%	0%
	F14(open)	0%	0%	100%	0%
	F15	0%	0%	0%	100%
Overall Correction Rate: 99.40%					

TABLE 6. Classification results of the 4 planes using standard PCA on extracted momentum vectors. Results are calculated by averaging all the 400 independent tests.

computationally, i.e. if M observations are collected as the training set with N sampling points per observation, the two major computational costs for obtaining feature (b) are computing a $2N \times 2N$ covariance matrix (a matrix multiplication of $2N \times M$ and $M \times 2N$ matrices), and computing the eigenvalue decomposition for the covariance matrix. On the other hand, feature (c) has the best recognition rate, but obtaining feature (c) is most expensive computationally, due to the costs of performing the matching procedure for each observation to obtain the momentum filed, forming a $2N \times 2N$ covariance matrix of the momentum field, and finding the corresponding eigenvalue decomposition for the covariance matrix. Feature (a) is almost in between Feature (b) and Feature (c), in the aspects of computational cost and recognition rate. The computational cost includes computing a matching procedure for each observation, and computing eigenvalue decomposition for an $N \times N$ kernel matrix, Eq. (35).

It is not to our surprise that feature (c) outperforms feature (a). This could be explained by the conjecture that feature (a), which contains the components of "energy" of the deformation, is considered as a global quantity, while feature (c), which is the quantity associated with each individual landmark, contains information of local details that in principle could help better classify the shapes.

Although we can see the advantages of using Hamiltonian metric or momentum representation in this example, we are cautious not to draw conclusion from this individual numerical event. We are fully aware that further numerical and/or analytic studies are required for understanding the mechanisms behind this numerical experiment before drawing any conclusion. A message brought from this example, however, is that for shape clustering or classification, using *Algorithm 1* to produce the Hamiltonian metric or extract the momentum information as the feature vectors could be an alternative when directly using the landmark location data does not provide a satisfactory result.

10. Conclusion and Future Work

We present an efficient iterative geodesic warping algorithm for template matching and its applications. This landmark-based algorithm find the initial momenta of the N-particle system of the Euler-Poincaré equations that deform corresponding landmarks from a reference to a target. Two new features introduced in the algorithm, distinct from the literature, are (1) the use of the non-smooth Green's function as the kernel of the N-particle system that accelerates the convergence and (2) the use of a constant matrix M to update the search direction, instead of the traditional forward-backward integration for updating the gradient or the Newton's optimization for finding the initial momenta that carry the geodesic deformation. A local convergence analysis for the proposed algorithm is also discussed in detail.

To investigate the usage of the output quantities computed along the warping algorithm, such as the Hamiltonian metric and the momentum field, we introduce a multiscale decomposition method that separates the scales/components of the momentum and the Hamiltonian metric associated with the deformation. We explore the potential of using the decomposed Hamiltonian metric and momentum vectors for clustering/classification analysis by conducting several numerical experiments. While the results of the numerical experiments are encouraging, further studies are required for understanding the mechanisms and drawing positive conclusion.

There are several directions for our future work. From the theoretical point of view, geometric property of landmark space associated with the non-kernel $G_{1/2}$ is an important problem. From the computation point of view, one topic is to extend the warping algorithm to solve three-dimensional template matching problems. Another is to investigate the search direction for the optimization problem beyond the current simple form that retains the efficiency but allows a larger domain of convergence. Finally we plan to apply the present algorithm to more practical applications, such as computational anatomy. A numerical experiment of applying the proposed algorithm to a small image database for identify brain structure change is currently in progress [21].

11. Appendix I

In this Appendix, we derive the expression of the optimal updating matrix M described in Section 5, but from the viewpoint of the estimation with least norm of covariance. The update of the velocity $\boldsymbol{u}$ is

$$\boldsymbol{u}_{i+1} = \boldsymbol{u}_i + M \cdot (I_1 - X_i) = \boldsymbol{u}_i + M\boldsymbol{Q}_i.$$

Assuming that an unbiased guess for each step is given by:

$$E[\boldsymbol{Q}_i] = 0, E(\boldsymbol{u}_i) = \boldsymbol{u}.$$

The covariance matrix can be calculated as

$$\begin{aligned} Cov(\boldsymbol{u}_{i+1}) &= E[(\boldsymbol{u}_i - \boldsymbol{u} + M\boldsymbol{Q}_i)][(\boldsymbol{u}_i - \boldsymbol{u} + M\boldsymbol{Q}_i)^T] && (38)\\ &= Cov(\boldsymbol{u}_i) + M \cdot Cov(\boldsymbol{Q}_i) \cdot M^T + M \cdot Cov(\boldsymbol{Q}_i, \boldsymbol{u}_i) + Cov(\boldsymbol{u}_i, \boldsymbol{Q}_i) \cdot M^T. && (39)\end{aligned}$$

Then, we can find M by calculating

$$\begin{aligned} \tfrac{\partial \operatorname{tr}(Cov(\boldsymbol{u}_{i+1}))}{\partial M} = 0; && (40)\\ \Rightarrow 2[M \cdot Cov(\boldsymbol{Q}_i) + Cov(\boldsymbol{Q}_i, \boldsymbol{u}_i)] = 0; && (41)\\ \Rightarrow M = -Cov(\boldsymbol{Q}_i, \boldsymbol{u}_i) \cdot Cov(\boldsymbol{Q}_i)^{-1}. && (42)\end{aligned}$$

$\boldsymbol{Q}_i$ can be calculated as

$$\boldsymbol{Q}_i = \phi(\boldsymbol{u}) - \phi(\boldsymbol{u}_i) = -\frac{\partial \phi}{\partial \boldsymbol{u}} \cdot (\boldsymbol{u}_i - \boldsymbol{u}) + O(|\boldsymbol{u}_i - \boldsymbol{u}|^2) \cdot 1_d.$$

If the second-order term is negligible compared to the first order term. We can then calculate

$$Cov(\boldsymbol{Q}_i) = E(\boldsymbol{Q}_i {\boldsymbol{Q}_i}^T) = \frac{\partial \phi}{\partial \boldsymbol{u}} \cdot Cov(\boldsymbol{u}_i) \cdot (\frac{\partial \phi}{\partial \boldsymbol{u}})^T, \quad (43)$$

and

$$Cov(\boldsymbol{Q}_i, \boldsymbol{u}_i) = E[(\boldsymbol{u} - \boldsymbol{u}_i)(\boldsymbol{Q}_i)^T] = -Cov(\boldsymbol{u}_i)(\frac{\partial \phi}{\partial \boldsymbol{u}})^T. \quad (44)$$

Plugging equations (43) and (44) into equation (42) yields the first order estimation of the optimal M

$$M = (\frac{\partial \phi}{\partial \boldsymbol{u}})^{-1},$$

which is consistent with the local convergent analysis in Section 5.

12. Appendix II

This Appendix investigates the possible negative choices of h.

Under the assumption of $h < 0$, on one hand, we will have

$$(a^2 + b^2 - B^2)h + 2(B - a) > 0 \tag{45}$$

from Eq. (29).

Case 1: $a^2 + b^2 - B^2 > 0$. We will have $\frac{2(a-B)}{a^2+b^2-B^2} < h < 0$ by requiring $a < B$.

Case 2: $a^2 + b^2 - B^2 < 0$. The solution would be $h < \min(0, \frac{2(a-B)}{a^2+b^2-B^2}) \Rightarrow h < 0$, since $B > |a|$ in this case.

Case 3: $a^2 + b^2 - B^2 = 0$. The inequality becomes $B > a$ which is always true under the condition, so we can use $h < 0$.

On the other hand, we shall also have

$$-Bh < \sqrt{(1 - ah)^2 + (bh)^2} \tag{46}$$

$$\Rightarrow (a^2 + b^2 - B^2)h^2 - 2ah + 1 > 0 \tag{47}$$

Case 1: $a^2 + b^2 - B^2 > 0$. In this case, there will always be a solution satisfying the above inequality.

If $B^2 - b^2 < 0$, we can pick $h < 0$, since the inequality can be written as $(b^2 - B^2)h^2 + (ah - 1)^2 > 0$, which is always true for any $h < 0$ under the chosen condition;

If $B^2 - b^2 \geq 0$, we can pick $h \in (-\infty, \frac{a-\sqrt{B^2-b^2}}{a^2+b^2-B^2}) \cup (\frac{a-\sqrt{B^2-b^2}}{a^2+b^2-B^2}, 0)$

Case 2: $a^2 + b^2 - B^2 < 0$. In this case, $\frac{a+\sqrt{B^2-b^2}}{a^2+b^2-B^2} < h < 0$

Case 3: $a^2 + b^2 - B^2 = 0$. This leads to condition $2ah < 1$.

Now we can summarize possible *negative* choices of h by taking the intersection for each case from the two inequalities.

- $a^2 + b^2 - B^2 > 0, a < B, B^2 < b^2$.
  We shall pick $\frac{2(a-B)}{a^2+b^2-B^2} < h < 0$
- $a^2 + b^2 - B^2 > 0, a < B, B^2 > b^2$.
  We shall pick $h \in (\frac{2(a-B)}{a^2+b^2-B^2}, 0) \cap \left((-\infty, \frac{a-\sqrt{B^2-b^2}}{a^2+b^2-B^2}) \cup (\frac{a-\sqrt{B^2-b^2}}{a^2+b^2-B^2}, 0)\right)$
- $a^2 + b^2 - B^2 < 0$.
  We shall pick $\frac{a+\sqrt{B^2-b^2}}{a^2+b^2-B^2} < h < 0$
- $a^2 + b^2 - B^2 = 0$.
  We shall pick $h < 0$ so that $2ah < 1$.

References

[1] Arguillére, S., Trélat, E., Trouvé, and Younes, L. Shape deformation analysis from the optimal control viewpoint. *Journal de mathematiques pures et appliquees*, **104**(1), 139–178, 2013.

[2] Ashburner, J. and Friston, K. J.. Diffeomorphic registration using geodesic shooting and Gauss-Newton optimisation. *NeuroImage*, **55**, 954–967, 2011.

[3] Beg, M. F., Miller M. I., Trouvé, A., and Younes, L.. Computing large deformation metric mappings via geodesics flows of diffeomorphisms. *Int. J. Comp. Vis.*, **61**(2), 139-157, 2005.

[4] Beg, M. F., Khan, A.. Computing an average anatomical using LDDMM and geodesic shooting. *IEEE*, ISBI, 1116-1119, 2006.

[5] Boostein, F. L.. Principal Warps: Thin-Plate Splines and the Decomposition of Deformations. *IEEE transactions on pattern analysis and machine intelligence.* **11**(4), 567–585, 1989.

[6] Boostein, F. L.. Landmark methods for forms without landmarks: morphometrics of group differences in outline shape. *Medical Image Analysis*, **1**(3), 225-243, 1996.

[7] Camassa, R., Kuang, D., and Lee, L.. Solitary waves and N-particle algorithms for a class of Euler Poincaré equations. Studies in Applied Mathematics. doi: 10.1111/sapm.12132.

[8] Chertock, A., Du Toit, P., and Marsden, J. E.. Integration of the EPDIFF equation by particle methods, *ESAIM:M2AN*, **46**, 515-534, 2012.

[9] Cotter, C. J. and Holm D. D.. *Singular solutions, momentum maps and computational anatomy.* 1st MICCAI Workshop on Mathematical Foundations of Computational Anatomy: Geometrical, Statistical and Registration Methods for Modeling Biological Shape Variability (2006) 18-28.

[10] Cotter, C. J., Clark A., and Peiró, J.. A reparameterisation based approach to geodesic constrained solvers for curve matching. *International Journal of Computer Vision*, **99**(1), 103–121, 2012.

[11] Du, J., Dryden, I. L., and Huang, X.. Size and shape analysis of error-prone shape data. *Journal of the American Statistical Association*, DOI: 10.1080/01621459.2014.908779, 2014.

[12] Duda, R. O., Hart, P. E., and Stork D. G.. *Pattern Classification*, 2nd, John Wiley & Sons, 2001.

[13] Dryden, I. L. and Mardia, K. V.. *Statistical Shape Analysis.* John Wiley & Sons, 1998.

[14] Dupuis, P., Grenander, U., Miller, M. I.. Variational problems on flows of diffeomorphisms for image matching. *Q. Appl. Math.*, **56**, 587-600, 1998.

[15] Dr. Stefan Sommer's lddmm implementation in MATLAB: https://github.com/nefan/segframe.git

[16] Fighters outline data.
http://visionlab.uta.edu/shape_data.htm

[17] Fletcher, P. T., Venkatasubramanian, S., Joshi, S. Robust statistics on Riemannian manifolds via the geometric median. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 2008, pp. 1-8, doi:10.1109/CVPR.2008.4587747.

[18] Glaunes, J. and Joshi, S. *Template estimation from unlabeled point set data and surfaces for computational anatomy.* Proceedings of the International Workshop on the Mathematical Foundations of Computational Anatomy (MFCA-2006).

[19] Glaunes, J., Trouvé A. ,and, Younes, L. *Diffeomorphic matching 10 of distributions a new approach for unlabelled point-sets and submanifolds matching.* Proceedings of CVPR'04.

[20] Grenander U. *General pattern theory.* Oxford University Press, Clarendon, UK, 1993.

[21] Huzurbazar, S., Kuang, D., and Lee, L.. A landmark-based algorithm for automatic pattern recognition and abnormality detection. http://arxiv.org/pdf/1602.05572v1.pdf, preprint, 2016

[22] Grenander, U. and Miler, M. I.. Computational anatomy: An emerging discipline. *Q. Appl. Math.*, **LVI**(4), 617-694, 1998.

[23] Holm, D. D. and Staley, M. F.. Wave structure and nonlinear balances in a family of evolutionary PDEs. *SIAM J. Applied Dynamical System*, **2**(3), 323-380, 2003.

[24] Holm, D. D, Ratnanather, J. T., Trouvé, A., and Younes, L.. Soliton dynamics in computational anatomy. *Neuroimage*, **23**, S170-S178, 2004.

[25] Holm, D. D., Schmah, T., and Stoica, C.. *Geometric Mechanics and Symmetry: From Finite to Infinite Dimensions.* . Oxford University Press, USA, 2009.

[26] Holm, D. D., Trouvé, A., and Younes, L.. The Euler-Poincaré Theory of Metamorphosis. *Q. Appl. Math.*, LXVII(4), 661–685, 2009.

[27] Izenman, A. J.. Recent developments in nonparametric density estimation. *Journal of the American Statistical Association*, **86**(413), 205-224, 1991.

[28] Jain, A. and Younes, L.. A kernel class allowing for fast computations in shape spaces induced by diffeomorphisms. *J. Comput. Appl. Math.*, **245**, 162–181, 2013.

[29] Fishbaugh, J., Prastawa, M., Gerig G., and Durrieman, S.. Geodesic shape regression in the framework of currents. *Inf Process Med Imaging*, **23**, 718–729, 2013.

[30] Joshi, S. and Miller, M. I.. Landmark matching via large deformation diffeomorphisms. *IEEE Trans. Image Processing*, **9**, 1357-1370, 2000.

[31] Laga, H. Kurtek, S. Srivastava, A., and Miklavcic, S. J.. Landmark free statistical analysis of the shape of plant leaves. *J. Theoret. Biol.*, **363**, 41–52, 2014.

[32] LeLe, S. and Richtsmeier, J. T.. *An invariant approach to statistical analysis of shapes.* CRC Press, 2001.

[33] Liu, R. and Yang, L.. Kernel estimation of multivariate cumulative distribution function. *Journal of Nonparametric Statistics*, **20**(8), 661-677, 2008.

[34] Micheli, M., Michor, P. W., and Mumford, D.. Sectional Curvature in Terms of the Cometric, with Applications to the Riemannian Manifolds of Landmarks. SIAM J. IMAGING SCIENCES , **5**(1), 394–433, 2012.

[35] Micheli, M., Glaunś, J. A.. Matrix-valued Kernels for Shape Deformation Analysis.. *Geometry Imaging and Computing*, **1**(1), 57-139, 2014.

[36] Marsland S. and Twining, C.. Constructing diffeomorphic representations for the groupwise analysis of non-rigid registrations of medical images. *IEEE Transactions on Medical Imaging*, **23**(8), 1006-1020, 2004.

[37] Mathworks Documentation.
http://www.mathworks.com/help/optim/ug/equation-solving-algorithms.html#f51887
[38] McLachlan, R. I. and Marsland S.. N-particle dynamics of the Euler equations for planar diffeomorphisms. *Dyn. Sys.*, **22**(3), 269-290, 2007.
[39] McLachlan, M. I. and Marsland , S.. *Discrete Mechanics and Optimal Control for Image Registration.* Anziam Journal,, **48**, C1-C16, 2007.
[40] Miller, M. I and Younes, L.. Group actions, homeomorphisms, and matching: A general framework. *International Journal of Computer Vision*, **41**(1/2), 61-84, 2001.
[41] Miller, M. I., Trouvé, A., and Younes, L.. On metrics and the Euler-Lagrange equations of computational anatomy. *Annual Reviews in Biomedical Engineering*, **4**, 375-405, 2002.
[42] Miller, M. I., Trouvé, A., and Younes, L.. Geodesic shooting for computational anatomy. *J Math Imaging*, **24**, 209-228, 2006.
[43] Miller, M. I., Trouvé, A., and Younes, L.. Hamiltonian systems and optimal control in computational anatomy: 100 years since D'Arcy Thompson. *Annu. Rev. Biomed. Eng.*, **17**, 447-509, 2015.
[44] Mumford, D. and Desolneux, A.. *Pattern Theory: The Stochastic Analysis of Real World Signals.* A K Peters, Lid, Natick, MA, 2010.
[45] Sain, S. R.. Multivariate locally adaptive density estimation. *Computational Statistics & Data Analysis* , **39**, 165-186, 2002.
[46] Sommer, Stefan Horst; Nielsen, Mads; Lauze, Francois Bernard; Pennec, Xavier. A Multi-Scale Kernel Bundle for LDDMM : Towards Sparse Deformation Description Across Space and Scales. *Lecture Notes in Computer Science*, Vol. **6801**, 624-35, 2011.
[47] Srivastava, A., Joshi, S. H., Mio, W., and Liu, X.. Statistical Shape Analysis: Clustering, Learning, and Testing. *IEEE Transactions on pattern analysis and machine intelligence*, **27**(4), 590–602, 2005.
[48] Trouvé, A.. An infinite dimensional group approach for physics based model. Technical report, 1995.
[49] Trouvé, A. and Younes, L. Metamorphoses through Lie group action. *Found. Comput. Math.*, (2005), 173-198
[50] Marsland, S. and Twining C.. Constructing diffeomorphic representations for the groupwise analysis of non-rigid registrations of medical images. *IEEE Transactions on Medical Imaging*, **23**(8), (2004) 1006-1020.
[51] Thakoor, N., Gao, J., and Jung, S. *Hidden Markov Model-Based Weighted Likelihood Discriminant for 2-D Shape Classification* IEEE Transactions on Image Processing. Volume 16, Issue 11, Nov. 2007, Page(s): 2707 - 2719.
[52] Vaillant M., Miller, M. I., Younes L., and Trouvé A.. Statistics on diffeomorphisms via tangent space representations. *NeuroImage*, **23**, (2004) S161-S169.
[53] Vialard, F.-X., Risser, L., Rueckert, D., and Holm, D. D.. Diffeomorphic atlas estimation using geodesic shooting on volumetric images *Annals of the BMVA*, **2012**(5), 1-12, 2012.
[54] Younes, L.. *Shapes and Diffeomorphisms.* Appl. Math. Sci., vol. 171, Springer-Verlag, Berlin, 2010.
[55] Younes L., Arrate F., and Miller M. I.. *Evolutions equations in computational anatomy.* NeuroImage 45(2009) S40-S50.
[56] Zhang, Miaomiao and Fletcher, P Thomas. *Bayesian principal geodesic analysis in diffeomorphic image registration.* International Conference on Medical Image Computing and Computer-Assisted Intervention, 2014.
[57] Zhang, Chao and Heeren, Behrend and Rumpf, Martin and Smith, William AP. *Shell PCA: statistical shape modelling in shell space.* Proceedings of the IEEE International Conference on Computer Vision, 2015.

Department of Mathematics, University of North Carolina, Chapel Hill, 27599, USA
E-mail address: camassa@amath.unc.edu, TEL:919-962-8476

Department of Mathematics, University of Wyoming, Laramie, WY 82071-3036, USA
E-mail address: dkuang@uwyo.edu, TEL:307-766-4221

Department of Mathematics, University of Wyoming, Laramie, WY 82071-3036, USA
E-mail address: llee@uwyo.edu, TEL:307-766-4368