Karcher Mean in Elastic Shape Analysis*

Wen Huang[1]
wen.huang@uclouvain.be

Yaqing You[2]
yy12b@my.fsu.edu

Kyle A. Gallivan[2]
kgallivan@fsu.edu

P.-A. Absil[1]
absil@inma.ucl.ac.be

[1] ICTEAM Institute
Université catholique de Louvain
Avenue G. Lemaître 4, 1348
Louvain-la-Neuve, Belgium

[2] Department of Mathematics
Florida State University
1017 Academic Way, 32306
Tallahassee, FL, USA

Abstract

In the framework of elastic shape analysis, a shape is invariant to scaling, translation, rotation and reparameterization. Since this framework does not yield a closed form of geodesic between two shapes, iterative methods have been proposed. In particular, path straightening methods have been proposed and used for computing a geodesic that is invariant to curve scaling and translation. Path straightening can then be exploited within a coordinate-descent algorithm that computes the best rotation and reparameterization of the end point curves [13]. A Riemannian quasi-Newton method to compute a geodesic invariant to scaling, translation, rotation and reparameterization has been given in [15] and shown to be more efficient than the coordinate-descent/path-straightening approach. This paper extends [15] by showing that using the new approach to the geodesic when computing the Karcher mean yields a faster algorithm.

1 Introduction

Shape analysis of curves is important in various area such as computer vision, medical diagnostics, and bioinformatics. The basic idea is to obtain a boundary curve of an object in a 2D image or contours of a 3D object and analyse those curves to characterize the original object. The research on shape analysis is rich and various ideas have been proposed, e.g., point-based methods, domain-based shape representations and parameterized curve representations. One of the earliest can be traced to Kendall [8], in which the representation of a shape uses landmarks. However, the choices of landmarks is subjective and may significantly influence the analysis of the original objects. For example, Figure 1 shows the geodesics computed using two different approaches: Kendall's landmarks and elastic shape analysis. Specifically, unlike the landmarks approach, the elastic shape analysis also takes reparameterization of curves into account.

Many frameworks for elastic shape analysis have been proposed. Younes [16] first introduced this kind of framework for general 2D curves. Younes *et al*. [17] studied elastic analysis of closed curves using complex representations of 2D coordinates of curves. Srivastava *et al*. [13] further defined a novel mathematical framework called the square root velocity function (SRVF), which include curves in Euclidean spaces of any dimension.

Pages 2.1-2.11
https://dx.doi.org/10.5244/C.29.DIFFCV.2

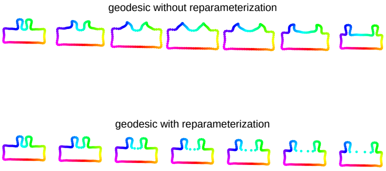
geodesic without reparameterization
geodesic with reparameterization

Figure 1: Geodesics without and with reparameterization are given by the frameworks of landmark-based Kendall's shape analysis [3, 8] and elastic shape analysis [13] respectively.

Elastic shape analysis is receiving increasing attention due to its superior theoretical results and effectiveness. The price for the improved effectiveness is the relative increase in expense in computing various objects, e.g., geodesic, mean. In this paper, the SRVF framework defined in [13] is considered. The advantage of SRVF framework is that it converts the complicate Riemannian metric into the standard $\mathbb{L}^2$ metric and preserve the isometry of rotation and reparameterization group action. This allows the shape space to be defined in a relative simple way. In [5], a closed form of distance approximation for closed curves, which is invariant to curve scaling and translation, is used and a Riemannian approach is proposed to improve the efficiency and effectiveness for further removing rotation and reparameterization. In [15], the idea of [5] is further explored when computing geodesics and the efficiency of removing the rotation and reparameterization is improved without using the distance approximation. In this paper, we use the approach in [15] to compute the Karcher mean of shapes and evaluate its performance.

This paper is organized as follows. Section 2 presents the SRVF framework of [13]. Section 3 reviews the approaches of computing the geodesic in [13] and [15]. Section 4 presents an algorithm for computing the Karcher mean and, finally, the performance of the algorithm is evaluated in Section 5.

2 SRVF Framework

A shape or curve in $\mathbb{R}^n$ is denoted by a parameterized function $\beta(t) : \mathbb{D} \to \mathbb{R}^n$, where $\mathbb{D}$ is $[0,1]$ for open curves and $\mathbb{D}$ is the unit circle $\mathbb{S}^1$ for closed curves. The representation of a shape starts from its square root velocity (SRV) function,

$$q(t) = \begin{cases} \frac{\dot{\beta}(t)}{\sqrt{||\dot{\beta}(t)||_2}}, & \text{if } ||\dot{\beta}(t)||_2 \neq 0; \\ 0, & \text{if } ||\dot{\beta}(t)||_2 = 0. \end{cases} \tag{2.1}$$

where $||\cdot||_2$ denote the 2-norm. The curve β can be recovered by $\beta(t) = \int_0^t q(s)||q(s)||_2 ds$ if $\beta(0)$ is 0. Note translation is removed since $\dot{\beta}$ is used. Further more, rescaling can be removed by restricting curves to be of unit length. Since the length of $\beta(t)$ is $\int_{\mathbb{D}} ||\dot{\beta}(t)||_2 dt = \int_{\mathbb{D}} ||q(t)||_2^2 dt$, the resulting space, called the preshape space l_n, for open curves is denoted as

$$l_n^o = \left\{ q \in \mathbb{L}^2([0,1], \mathbb{R}^n) | \int_0^1 ||q(t)||_2^2 dt = 1 \right\}, \tag{2.2}$$

and for closed curves is denoted as

$$l_n^c = \{q \in \mathbb{L}^2(\mathbb{S}^1, \mathbb{R}^n) | \int_{\mathbb{S}^1} ||q(t)||_2^2 dt = 1, \int_{\mathbb{S}^1} q(t)||q(t)||_2 dt = 0\} \tag{2.3}$$

where $\int_{\mathbb{S}}^1 q(t)||q(t)||_2 dt = 0$ stresses the closure condition and the super scripts ^{o} and ^{c} denote open and closed curves respectively. Statements without a superscript apply to both open and closed curves. A more intuitive way to denote the preshape space l_n^c is $\{q \in \mathbb{L}^2(\mathbb{S}^1, \mathbb{R}^n) | \int_0^1 ||q(t)||_2^2 dt = 1, \int_0^1 q(t)||q(t)||_2 dt = 0\}$ and the closure condition means the difference between $\beta(0)$ and $\beta(1)$ is zero since $\int_0^1 q(t)||q(t)||_2 dt = \beta(1) - \beta(0)$. It can be seen that l_n^c is a submanifold of l_n^o. The metrics of the spaces are endowed from $\mathbb{L}^2$, i.e., $\langle v_1, v_2 \rangle_{\mathbb{L}^2} = \int_0^1 v_1^T v_2 dt$ for $v_1, v_2 \in \mathbb{L}^2([0,1], \mathbb{R}^n)$.

In order to remove the rotation and reparameterization, we consider the rotation group

$$\mathrm{SO}(n) = \left\{ O \in \mathbb{R}^{n \times n} | O^T O = I_n, \det(O) = 1 \right\} \tag{2.4}$$

and the reparameterization group

$$\Gamma = \{\gamma : \mathbb{D} \to \mathbb{D} | \gamma \text{ is orientation-preserving, smooth bijections.}\}. \tag{2.5}$$

The actions of $\mathrm{SO}(n)$ and Γ on the SRV of a curve β are:

$$\mathrm{SO}(n) \times l_n \to l_n : (O, q) \to Oq, \qquad l_n \times \Gamma \to l_n : (q, \gamma) \to (q \circ \gamma)\sqrt{\dot{\gamma}} \tag{2.6}$$

and it is known that the two group actions are isometric with respect to the $\mathbb{L}^2$ metric. It follows that the orbit of the group actions is defined by

$$[q] = \left\{ O(q \circ \gamma)\sqrt{\dot{\gamma}} | (\gamma, O) \in \Gamma \times \mathrm{SO}(n) \right\} \tag{2.7}$$

and the shape space is defined as:

$$\mathfrak{L}_n = l_n / (\Gamma \times \mathrm{SO}(n)) = \{\overline{[q]} | q \in l_n\}, \tag{2.8}$$

where $\overline{[q]}$ denotes the closure of $[q]$ with respect to $\mathbb{L}^2$. The motivation of the closure can be found in, e.g., [13] or [5].

Since $\mathfrak{L}_n$ is a quotient manifold of l_n and they have the same metric, a geodesic in $\mathfrak{L}_n$ can be represented by any geodesic in l_n that is perpendicular to any orbit that it intersects, and the distance between $\overline{[q_0]}$ and $\overline{[q_1]} \in \mathfrak{L}_n$ is given by

$$d_{\mathfrak{L}_n}(\overline{[q_0]}, \overline{[q_1]}) = \inf_{(\gamma, O) \in \Gamma \times \mathrm{SO}(n)} d_{l_n}(q_0, O(q_1 \circ \gamma)\sqrt{\dot{\gamma}}). \tag{2.9}$$

3 Computing a Geodesic

As shown later in Section 4, the gradient of the cost function of Karcher mean requires the computation of $\dot{\alpha}(1)$ given $\alpha(0)$ and $\alpha(1)$, where α is a geodesic. However, a closed form is unknown for shape space and, therefore, the geodesic and the velocity $\dot{\alpha}(1)$ must be computed numerically. In this section, we discuss the computation of the geodesic in shape space. Specifically, Section 3.1 gives an algorithm for computing the geodesic in preshape space and Section 3.2 presents two approaches to remove rotation and reparameterization to obtain a representation of geodesic in shape space.

3.1 Path-Straightening Method in Preshape Space l_n^c

The preshape space of open curves is a unit sphere and its geodesic is well-known, therefore, we focus on computing a geodesic of closed curves. Throughout this paper, the use of word "geodesic" means a path with a constant velocity.

Two commonly used methods of computing the geodesic in elastic shape space are the shooting method [10] and the path straightening method [11] [9] [7] [13]. The path straightening method has the advantage that all the iterates of paths connect the two end point curves while the shooting method does not. Here, we discuss path straightening method which has been used in [15].

Let $\mathscr{P}$ denote all the curves in l_n^c. Let the set of paths connecting two curves q_0, q_1 in l_n^c be

$$\mathscr{P}_{q_1,q_2} = \{\alpha : [0,1] \to l_n^c | \alpha(0) = q_0, \alpha(1) = q_1\} \tag{3.1}$$

We start off from an arbitrary path $\alpha(\tau)$ in $\mathscr{P}_{q_1,q_2}$, and iterate until reaching a critical point of the energy function

$$E : \mathscr{P}_{q_1,q_2} \to \mathbb{R} : \alpha \mapsto \frac{1}{2} \int_0^1 \langle \dot{\alpha}(\tau), \dot{\alpha}(\tau) \rangle \, d\tau. \tag{3.2}$$

It has been shown in [13, Lemma 4] that any critical of E is a geodesic of l_n^c.

A gradient method is proposed in [13], in which the search direction is along the negative gradient and a fixed step size is used.

Consider a path $\beta \in \mathscr{P}$ and a vector field $v \in \mathrm{T}_\beta \mathscr{P}$. The covariant derivative of v along β is the vector field obtained by projecting $\frac{dv}{d\tau}(\tau)$ onto the tangent space of $\mathrm{T}_{\beta(\tau)} l_n^c$ for all τ. A vector field $z \in \mathrm{T}_\beta \mathscr{P}$ is called a covariant integral of v along β if the covariant derivative of z is v, i.e., $\frac{Dz}{d\tau} = v$.

Let u denote the covariant integral of $\frac{d\alpha}{d\tau}$ with zero initial value at $\tau = 0$. The gradient of E is given by $w(\tau) = u(\tau) - \tau \tilde{u}(\tau)$, where $\tilde{u}$ is the vector field obtained by parallel translating $u(1)$ backwards along α, i.e., $\tilde{u}(1) = u(1)$ and $\frac{D\tilde{u}}{d\tau}(\tau) = 0$ for all $\tau \in [0,1]$.

Algorithm 1 outlines the path straightening method for computing geodesic in l_n^c of [13].

Algorithm 1 Path Straightening Method

```
Input: Two curves $\beta_0$ and $\beta_1$, and a step size $t > 0$
1: Compute the representations $q_0$ and $q_1$ in $l_n^c$.
2: Initialize a path $\alpha$ between $q_0$ and $q_1$ in $l_n^c$.
3: Compute the velocity vector field $\frac{d\alpha(\tau)}{d\tau}$ along the path $\alpha$.
4: Compute the covariant integral of $\frac{d\alpha(\tau)}{d\tau}$, denoted by $u$
5: Compute the backward parallel transport of the vector $u(1)$ along $\alpha$, denoted by $\tilde{u}$
6: Compute the full gradient vector field of the energy $E$ along the path $\alpha$, denoted by $w$,
   using $w(\tau) = u(\tau) - \tau\tilde{u}(\tau)$.
7: Update $\alpha$ along the vector field $tw$. If $\int_0^1 \|w(\tau)\|_{\mathbb{L}^2}^2 d\tau$ is small, then stop. Else, go to
   Step 3.
```

The initial path α between q_0 and q_1 is obtained by projecting the path α_o, the geodesic between q_0 and q_1 in l_n^o, onto the l_n^c (see details in [13, Item 1]). This usually offers good initial iterate and Algorithm 1 converges after only a few iterations to reach a tight stopping criterion, e.g., $\int_0^1 \|w(\tau)\|_{\mathbb{L}^2}^2 d\tau \leq 10^{-10}$.

3.2 Removing Orientations and Reparameterizations

3.2.1 Approach in [13]

In order to obtain a geodesic in shape space $\mathfrak{L}_n$, we must minimize the cost function $H(O,\gamma) = d_{l_n^c}(q_0, O(q_1\circ\gamma)\sqrt{\dot\gamma})$ over the product of manifolds $\mathrm{SO}(n)$ and Γ. The algorithm in [13] solves this optimization by alternately optimizing between $\mathrm{SO}(n)$ and Γ. This requires the computation of the gradient with respect to O and γ.

To this end, consider the cost function $d_{l_n^c}(q_0,\tilde q_1)$ with respect to $\tilde q_1$. Let α denote the geodesic between q_0 and $\tilde q_1$ in l_n^c. It is known from [13] that the gradient of $d_{l_n^c}(q_0,\tilde q_1)$ is $\eta = \dot\alpha(1)/\|\dot\alpha(1)\|$. It follows that the gradient with respect to O is

$$\mathrm{grad}_O H(O,\gamma) = P_O\left(\int_{\mathbb{D}} \eta\sqrt{\dot\gamma}(q_2\circ\gamma)^T ds\right) \tag{3.3}$$

where $P_O(M) = (M - OM^T O)/2$.

Since Γ is an infinite dimensional manifold, the gradient with respect to γ is approximated by

$$\mathrm{grad}_\gamma H(O,\gamma) \approx \sum_i^k b_i \,\mathrm{D}H(O,\gamma)[b_i], \tag{3.4}$$

where $\{b_i\}_{i=1}^\infty$ is a basis of the tangent space of γ, the tangent space of γ is $\mathrm{T}_\gamma\Gamma = \mathbb{L}^2(\mathbb{S}^1,\mathbb{R})$, k is the number of the elements of the basis used in the approximation and $\mathrm{D}H(O,\gamma)[b_i]$ denotes the directional derivative of $H(O,\gamma)$ along direction b_i. It can be shown that

$$\mathrm{D}H(O,\gamma)[b_i] = \left\langle \eta, O\left(\sqrt{\dot\gamma}\dot q_1(\gamma) b_i + \frac{1}{2\sqrt{\dot\gamma}}\dot b_i q_1(\gamma)\right)\right\rangle_{\mathbb{L}^2}. \tag{3.5}$$

The suggested basis $\{b_i\}$ is an orthonormal basis of the tangent space of γ under the Palais metric $\langle v_1, v_2\rangle_P = v_1(0)v_2(0) + \int_0^1 \dot v_1(\tau)\dot v_2(\tau) d\tau$, i.e.,

$$\{1, \frac{\sin(nt)}{n\pi}, \frac{\cos(nt)-1}{n\pi}, n = 1,2,\dots\}. \tag{3.6}$$

The algorithm of removing rotation and reparameterization is stated in Algorithm 2.

Algorithm 2 Removing rotation and reparameterization

```
Input: Two curves $\beta_0$ and $\beta_1$, and step sizes $t_1, t_2 > 0$
1: Set $\tilde\beta_1$ to be $\beta_1$, $O_0 = I$, $\gamma_0 = \gamma_{\mathrm{id}}$, and $k = 0$.
2: Compute the representations $q_0$ of $\beta_0$ and $q_1$ of $\tilde\beta_1$.
3: Compute the geodesic $\alpha$ between $q_0$ and $q_1$ in $l_n^c$ using Algorithm 1.
4: Update the rotation by $O_{k+1} = O_k \exp(t_1\,\mathrm{grad}_I H(I,\gamma_{\mathrm{id}}))$, where $\mathrm{grad}_I H(I,\gamma_{\mathrm{id}})$ is (3.3).
5: Update the reparameterization by $\gamma_{k+1} = \gamma_k \circ (\gamma_{\mathrm{id}} + t_2\,\mathrm{grad}_{\gamma_{\mathrm{id}}} H(I,\gamma_{\mathrm{id}}))$, where
   $\mathrm{grad}_{\gamma_{\mathrm{id}}} H(I,\gamma_{\mathrm{id}})$ is (3.4). Note that $t_2$ should be small enough such that $\gamma_{k+1}$ is non-
   decreasing.
6: Update $\tilde\beta_1 \leftarrow O_{k+1}\beta_1\circ\gamma_{k+1}$ and set $q_1$ to be the SRVF of $\tilde\beta_1$
7: If some stopping criterion is satisfied, then stop. Else, $k \leftarrow k+1$ and goto Step 2.
```

Note that in [13] the substitution of $l = \sqrt{\dot\gamma}$ is used in H. It follows that the cost function is defined on $\mathrm{SO}(n)$ and the first quadrant of the unit sphere $\mathcal{L} = \{l \in \mathbb{L}^2([0,1],\mathbb{R}) | \|l\|_{\mathbb{L}^2=1}\}$. It is pointed out that using the basis (3.6) essentially yields the same method in [13, Section 4.4] without the extra substitution step.

3.2.2 Approach in [15]

The basic computational variants of the path straightening method in the shape space can be characterized as coordinate descent, alternating direction, or steepest descent with a fixed step size. It is well-known that such methods may from slow convergence, see e.g., [12]. In this paper, as in [15], a faster algorithm, a limited-memory version of Riemannian BFGS method (LRBFGS), which is introduced in [4] and shown to outperform many other start-of-the-art Riemannian algorithms for many large-scale problems, e.g., [5, 6, 12].

Since it is observed that Algorithm 2 dominates the computational time required when computing a geodesic, we only use LRBFGS to improve the performance of removing rotation and reparameterization.

For the closed curves, the reparameterization Γ can be characterized as

$$\Gamma^c = [0,1] \times \Gamma^o \tag{3.7}$$

and its action is therefore $l_n^c \times \Gamma^c \to l_n^c : ((q,m),\gamma)) \to (q(t+m \mod 1) \circ \gamma \mod 1))\sqrt{\dot{\gamma}}$, where Γ^o is the reparameterization group for open curves, i.e.,

$$\Gamma^o = \big\{\gamma : [0,1] \to [0,1] | \gamma \text{ is a diffeomorphism}\big\}. \tag{3.8}$$

Further setting $l = \sqrt{\dot{\gamma}}$, we obtain a cost function

$$f(O,m,l) = d_{l_n^c}(q_1, Olq_2(\int_0^t l^2(s)ds + m \mod 1)), \tag{3.9}$$

where $(O,m,l) \in \mathrm{SO}(n) \times \mathbb{R} \times \mathcal{L}$. We define the metric on the tangent space of $\mathrm{SO}(n) \times \mathbb{R} \times \mathcal{L}$ by $\langle (U_1,b_1,v_1),(U_2,b_2,v_2)\rangle = \mathrm{trace}(U_1^T U_2) + b_1 b_2 + \int_0^1 v_1 v_2 ds$. The Riemannian gradient of f with respect to this metric is given in Lemma 3.1 without proof.

Lemma 3.1 *The Riemannian gradient of $f(O,m,l)$ in (3.9) is*

$$\mathrm{grad}\, f(O,m,l) = (P_O(A), \int_0^1 y' ds, P_l(x - 2yl)), \tag{3.10}$$

where A denotes $\int_0^1 \eta l q_2^T (\int_0^t l^2(s)ds + m \mod 1)ds$, x denotes $\langle \eta, O(q_2(\int_0^t l^2(s)ds + m \mod 1))\rangle_2$, y' denotes $\langle \eta, O(lq_2'(\int_0^t l^2(s)ds + m \mod 1))\rangle_2$, η is the same as in (3.3) and $P_l(v) = v - l\frac{\langle v,l\rangle_{\mathbb{L}^2}}{\langle l,l\rangle_{\mathbb{L}^2}}$.

In order to apply the LRBFGS algorithm in [4, Algorithm 2], we also need a retraction and a vector transport. The chosen pair is the well-known exponential mapping and parallel translation for each component (see e.g., [1]). They are given here for completeness. The retraction is

$$R_{(O,m,l)}(A,a,v) = \Big(O\exp(O^T A), m + a, l\cos(\|v\|_{\mathbb{L}^2}) + \frac{v}{\|v\|_{\mathbb{L}^2}}\sin(\|v\|_{\mathbb{L}^2})\Big) \tag{3.11}$$

and the vector transport is

$$\mathcal{T}_{(A,a,v)}((B,b,w)) = \Big(O\exp(O^T A/2)O^T B\exp(O^T A/2), b, w - \frac{2\langle w, \tilde{l}\rangle_{\mathbb{L}^2}}{\|l + \tilde{l}\|_{\mathbb{L}^2}^2}(l + \tilde{l})\Big) \tag{3.12}$$

where $A, B \in \mathrm{T}_O\,\mathrm{SO}(n)$, $a, b \in \mathbb{R}$, $w, v \in \mathrm{T}_l\,\mathcal{L}$ and $\tilde{l} = l\cos(\|v\|_{\mathbb{L}^2}) + \frac{v}{\|v\|_{\mathbb{L}^2}}\sin(\|v\|_{\mathbb{L}^2})$.

4 Karcher Mean

The Karcher mean of shapes $\overline{[q_i]}, i = 1, 2, \dots, N$ is defined to be the minimizer of the cost function

$$[q_*] = \arg\min_{[q] \in \mathcal{L}_n} \frac{1}{2N} \sum_{i=1}^{N} \mathrm{dist}^2_{\mathcal{L}_n}(\overline{[q]}, \overline{[q_i]}). \tag{4.1}$$

A representation of the gradient of (4.1) is given by $\frac{1}{N}\sum_{i=1}^{N} \dot{\alpha}_i(1)$, where $\alpha_i \subset l_n$ is a representation of the geodesic between $\overline{[q]}$ and $\overline{[q_i]}$ such that $\alpha_i(1) = q$ and $\alpha_i(0) \in \overline{[q_i]}$.

The Riemannian steepest descent algorithm [1] is used to compute the Karcher mean of shapes. The algorithm is sketched in Algorithm 3. The retraction is given by projection, i.e.,

$$q_+ = R_q(v) = P_{l_n}(q + v) \tag{4.2}$$

where P_{l_n} is the projection defined by [13, Item 1].

Algorithm 3 Karcher Mean

```
Input: Curves $\beta_i, i = 1, \dots, N$ and initial iterate $\beta^{(0)}$.
1: Compute the representations $q^{(0)}$ of $\beta^{(0)}$ and $q_i$ of $\beta_i$, $i = 1, \dots, N$ in $l_n$. Set $k = 0$.
2: Compute the geodesic $\alpha_i$ between $\overline{[q_i]}$ and $q^{(k)}$ such that $\alpha_i(1) = q^{(k)}$ and $\alpha_i(0) \in \overline{[q_i]}$ for
   all $i = 1, \dots, N$. The values of the cost function and its gradient are obtained during this
   computation.
3: Apply the backtracking line search algorithm [2, Algorithm A6.3.1] and find the step
   size $\lambda_k$ and the next iterate
       $q^{(k+1)} = R_{q^{(k)}}(-\lambda_k \zeta_k),$     (4.3)
   where $\zeta_k = \frac{1}{N}\sum_{i=1}^{N} \dot{\alpha}_i(1)$ is the gradient of (4.1).
4: If some stopping criterion is satisfied, then stop. Else, $k \leftarrow k + 1$ and goto Step 2.
```

5 Experiments

As shown in Step 2 of Algorithm 3, a geodesic is used for the cost function and gradient evaluation. In this section, we test the performance of Algorithm 3 when the two approaches in Sections 3.2.1 and 3.2.2 are used when computing geodesic.

The MPEG-7 dataset [14] is used in the experiments. It contains 70 clusters each of which has 20 shapes, i.e., 1400 shapes in total. The Matlab function BWBOUNDARIES is used to extract the boundary curves of the shapes and 100 uniformly-space points are chosen to represent each shape. A path in l_n^c is represented by 11 curves.

The tests are performed in Matlab R2014a on a 64 bit Ubuntu system with 3.6GHz CPU (Intel (R) Core (TM) i7-4790).

Algorithm 3 with the approaches in Sections 3.2.1 and 3.2.2 are denoted by MeanCD and MeanLRBFGS respectively. The stopping criterion of Algorithm 1 is $\int_0^1 \|w(\tau)\|^2_{\mathbb{L}^2} d\tau \leq 10^{-10}$. Both Algorithm 2 and LRBFGS run for 30 iterations. The step sizes t_1 and t_2 in Algorithm 2 are chosen to be 0.5 and 0.001 respectively. Algorithm 3 stops when the step size λ_k is less than 10^{-2}. Note that it is observed that $\lambda_k = 1$ usually is the step size found by the backtracking algorithm.

Table 1: Computational time, number of iterations and final cost function values of reported tests. t, iter and f denote computational time (second), number of iterations and final cost function values respectively. The subscript $-k$ indicates a scale of 10^{-k}.

	MeanCD			MeanLRBFGS		
	t	iter	f	t	iter	f
Figure 2	2.94_2	27	5.03_{-2}	9.87_1	8	4.99_{-2}
Figure 4	7.05_2	26	3.93_{-2}	4.55_2	14	3.67_{-2}
Figure 5	1.46_3	19	1.18_{-1}	6.77_2	8	7.40_{-2}

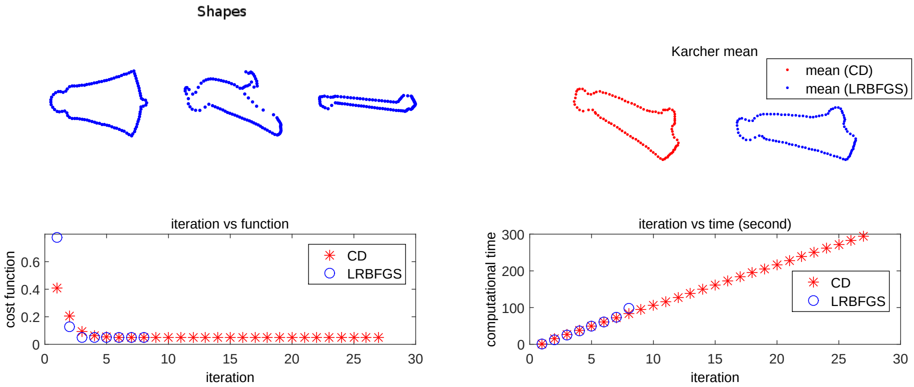

Figure 2: A representative test. The samples shapes, Karcher means by MeanCD and MeanLRBFGS, cost function values and computational time are given.

Three representative results are reported in Figures 2, 4 and 5. As shown in the right bottom figure in Figure 2 and Table 1, the computational time for each iteration are not too different for MeanCD and MeanLRBFGS. Both algorithms provide reasonable Karcher mean visually. In particular, Figure 3 shows the optimal rotation and reparameterization between mean and each sample shape. Therefore, MeanLRBFGS is faster than MeanCD and provide competitive results.

6 Conclusion and Future Work

In this paper, we consider computing the Karcher mean in the shape space of elastic curves. Two approaches for computing the geodesics required have been given in [13] and [15] and we compare their performance in computing the Karcher mean. It is shown that Algorithm 3 with the approach in [15] converges faster.

In the future, we will test the quality of the Karcher mean by MeanLRBFGS in the sense of superior clustering, classification and stochastic analysis. The algorithm will be included in the C++ Riemannian optimization library on http://www.math.fsu.edu/ROPTLIB.

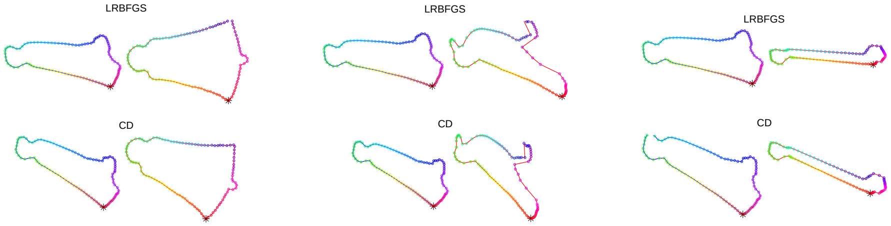

Figure 3: The rotation and reparameterization between samples and mean shapes. The colors of points on the two curves represent correspondence between two curves. The black stars represent the start/end points of the curves. The means in top and bottom rows are given by MeanLRBFGS and MeanCD respectively.

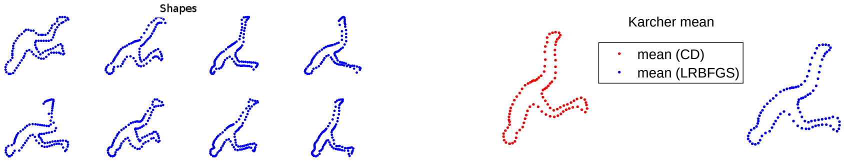

Figure 4: A representative test. The samples shapes, Karcher means by MeanCD and MeanLRBFGS are given.

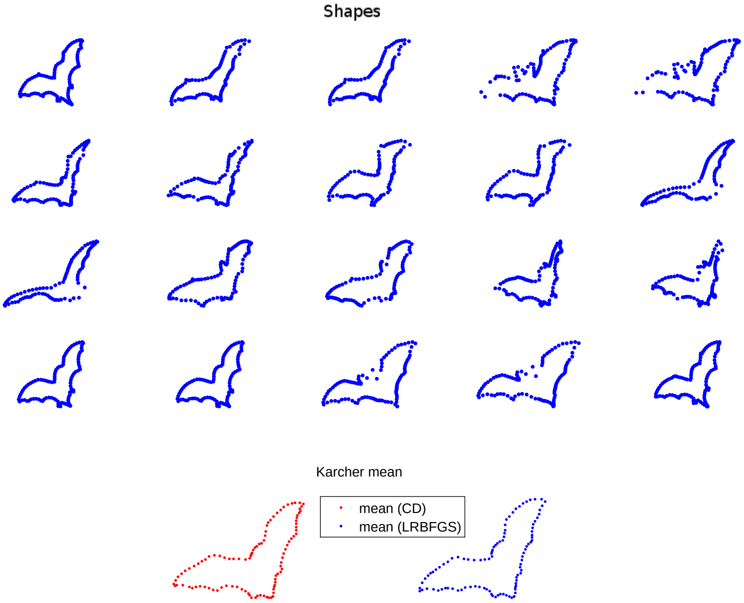

Figure 5: A representative test. The samples shapes, Karcher means by MeanCD and MeanLRBFGS are given.

References

[1] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, Princeton, NJ, 2008. ISBN 9780691132983. URL http://sites.uclouvain.be/absil/amsbook/.

[2] J. E. Dennis and R. B. Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations*. Springer, New Jersey, 1983. ISBN 0898713641.

[3] I. L. Dryden and K. V. Mardia. *Statistical shape analysis*. Wiley, 1998.

[4] W. Huang, K. A. Gallivan, and P.-A. Absil. A Broyden class of quasi-Newton methods for Riemannian optimization. *Accepted in SIAM Journal on Optimization*, 2014.

[5] Wen Huang, K. A. Gallivan, Anuj Srivastava, and P.-A. Absil. Riemannian optimization for registration of curves in elastic shape analysis. Technical Report UCL-INMA-2014.12, U.C.Louvain, December 2014.

[6] Wen Huang, K. A. Gallivan, and Xiangxiong Zhang. Solving phaselift by low rank riemannian optimization methods for complex semidefinite constraints. Technical Report UCL-INMA-2015.01, U.C.Louvain, January 2015.

[7] S. H. Joshi, E. Klassen, A. Srivastava, and I. Jermyn. An efficient representation for computing geodesics between n-dimensional elastic shapes. *In Proceedings of IEEE conference on Computer Vision and Pattern Recognitioin*, 2007.

[8] D. G. Kendall. Shape manifolds, procrustean metrics, and complex projective spaces. *Bulletin of the London Mathematical Society*, 16(2):81–121, March 1984. ISSN 0024-6093. doi: 10.1112/blms/16.2.81. URL http://blms.oxfordjournals.org/cgi/doi/10.1112/blms/16.2.81.

[9] E. Klassen and A. Srivastava. Geodesics between 3D closed curves using path-straightening. *Computer VisionqVECCV 2006*, pages 1–15, 2006. URL http://link.springer.com/chapter/10.1007/11744023_8.

[10] E. Klassen, A. Srivastava, W. Mio, and S. H. Joshi. Analysis of planar shapes using geodesic paths on shape spaces. *IEEE transactions on pattern analysis and machine intelligence*, 26(3):372–83, March 2004. ISSN 0162-8828. URL http://www.ncbi.nlm.nih.gov/pubmed/15376883. doi:10.1109/TPAMI.2004.1262333.

[11] J. Langer and D. A. Singer. Curve straightening and a minimax argument for closed elastic curves. *Topology*, 24(1):75–88, 1985. URL http://www.sciencedirect.com/science/article/pii/0040938385900

[12] J. Nocedal and S. J. Wright. *Numerical optimization*. Springer, second edition, 2006.

[13] A. Srivastava, E. Klassen, S. H. Joshi, and I. H. Jermyn. Shape analysis of elastic curves in Euclidean spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(7):1415–1428, September 2011. ISSN 1939-3539. URL http://www.ncbi.nlm.nih.gov/pubmed/20921581. doi:10.1109/TPAMI.2010.184.

[14] Temple University. Shape similarity research project. www.dabi.temple.edu/ shape/MPEG7/dataset.html.

[15] Y. You, W. Huang, K. A. Gallivan, and P.-A. Absil. A Riemannian Approach for Computing Geodesics in Elastic Shape Analysis. *In Proceedings of the 3rd IEEE Global Conference on Signal and Information Processing, Accepted*, 2015.

[16] L. Younes. Computable elastic distances between shapes. *SIAM Journal on Applied Mathematics*, 58(2):565–586, April 1998. ISSN 0036-1399. URL http://epubs.siam.org/doi/abs/10.1137/S0036139995287685. doi:10.1137/S0036139995287685.

[17] L. Younes, P. Michor, J. Shah, and D. Mumford. A metric on shape space with explicit geodesics. *Rendiconti Lincei - Matematica e Applicazioni*, 9(1):25–57, 2008. ISSN 1120-6330. URL http://www.ems-ph.org/doi/10.4171/RLM/506. doi:10.4171/RLM/506.