

A Riemannian Approach for Computing Geodesics in Elastic Shape Analysis

Yaqing You*, Wen Huang[†], Kyle A. Gallivan* and P.-A. Absil[†]

*Florida State University, Department of Mathematics 1017 Academic Way, Tallahassee, FL 32306-4510, US

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

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 geodesic between two shapes, iterative methods are used. 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. In this paper, we propose a Riemannian quasi-Newton method to compute a geodesic invariant to scaling, translation, rotation and reparameterization and show that it is more efficient than the coordinate-descent/path-straightening approach.

I. 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. Work on this topic can be traced back 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 given by two different landmarks. As a matter of fact, the represented points of the bottom figure in Figure 1 is chosen by using the elastic shape analysis. Specifically, unlike the landmarks approach, the elastic shape analysis also takes reparameterization of curves into account.

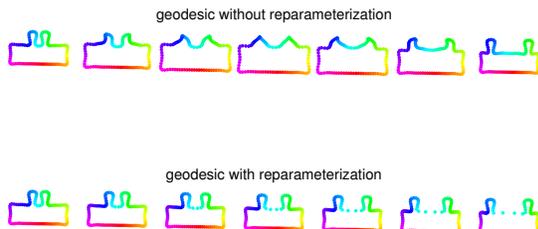


Fig. 1. Geodesics without and with reparameterization are given by the frameworks of landmark-based Kendall’s shape analysis [8], [2] and elastic shape analysis [13] respectively.

Many frameworks for elastic shape analysis have been proposed. Younes [15] first introduced a framework for general 2D curves. Younes et al. [16] studied on 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 functions (SRVFs), which include curves in Euclidean spaces of any dimension.

Elastic shape analysis draws more and more attention due to its superior theoretical results and effectiveness. The price for enhanced effectiveness can be an increase in computational cost in geodesic, mean, etc. In this paper, the SRVF framework defined in [13] is considered. The SRVF framework converts the complicated Riemannian metric into the standard \mathbb{L}^2 metric and preserves the isometry of the rotation and reparameterization group actions. This allows us to define the shape space in a relatively 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 of removing rotation and reparameterization. In this paper, we explore the same idea of [5] and improve the efficiency in the computation of removing the rotation and reparameterization without using approximation of closed curve distance. Also unlike the approach in [5], a geodesic is obtained.

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 points while the shooting method does not. In this paper, we focus on the path straightening method.

This paper is organized as follows. Section II presents the SRVF framework. Section III gives a previous approach that uses path straightening. Section IV defines the Riemannian approach and red experiments are presented in Section V.

II. SRVF FRAMEWORK

A shape or curve in \mathbb{R}^n is denoted by a parameterized function $\beta(t) : \mathbb{D} \rightarrow \mathbb{R}^n$, 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,

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

where $\|\cdot\|_2$ denotes 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 β 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 defined as

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

and for closed curves is defined as

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

where $\int_{\mathbb{S}^1} q(t) \|q(t)\|_2 dt = 0$ stresses the closure condition and the super script o and c denotes open or 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 $\{q \in \mathbb{L}^2(\mathbb{S}^1, \mathbb{R}^n) \mid \int_0^1 \|q(t)\|_2^2 dt = 1, \int_0^1 q(t) \|q(t)\|_2 dt = 0\}$ 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 rotation and reparameterization, we consider the rotation group

$$\text{SO}(n) = \{O \in \mathbb{R}^{n \times n} \mid O^T O = I_n, \det(O) = 1\}$$

and the reparameterization group

$$\Gamma = \{\gamma : \mathbb{D} \rightarrow \mathbb{D} \mid \gamma \text{ is orientation-preserving, smooth bijections.}\}$$

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

$$\begin{aligned} \text{SO}(n) \times l_n &\rightarrow l_n : (O, q) \rightarrow Oq, \\ l_n \times \Gamma &\rightarrow l_n : (q, \gamma) \rightarrow (q \circ \gamma) \sqrt{\dot{\gamma}} \end{aligned}$$

where \circ is the function composition operation. It has been shown in [13] 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

$$[q] = \left\{ O(q \circ \gamma) \sqrt{\dot{\gamma}} \mid (\gamma, O) \in \Gamma \times \text{SO}(n) \right\}$$

and the the shape space is defined as:

$$\mathfrak{L}_n = l_n / \Gamma \times \text{SO}(n) = \{\overline{[q]} \mid q \in l_n\},$$

where $\overline{[q]}$ denotes the closure of $[q]$ with respect to \mathbb{L}^2 . We mod out by the closures of the orbits instead of the orbits themselves because the space of diffeomorphisms is not closed under the \mathbb{L}^2 and Palais metrics [13].

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

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

III. PATH STRAIGHTENING METHOD

A. 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. In this paper, we focus on computing a geodesic of closed curves. Throughout this paper, the use of word ‘‘geodesic’’ means a path with a constant velocity.

Let \mathcal{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

$$\mathcal{P}_{q_1, q_2} = \{\alpha : [0, 1] \rightarrow l_n^c \mid \alpha(0) = q_0, \alpha(1) = q_1\}$$

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

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

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 direction and a fixed step size is used.

Consider a path $\beta \in \mathcal{P}$ and a vector field $v \in T_{\beta} \mathcal{P}$. The covariant derivative of v along β is the vector field obtained by projecting $\frac{dv}{d\tau}(\tau)$ onto the tangent space of $T_{\beta(\tau)} l_n^c$ for all τ . A vector field $z \in T_{\beta} \mathcal{P}$ 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 β_0 and β_1 , and a step size $t > 0$

- 1: Compute the representations q_0 and q_1 in l_n^c .
 - 2: Initialize a path α 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 α .
 - 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 α , denoted by \tilde{u}
 - 6: Compute the full gradient vector field of the energy E along the path α , denoted by w , using $w(\tau) = u(\tau) - \tau \tilde{u}(\tau)$.
 - 7: Update α along the vector field tw . If $\int_0^1 \|w(\tau)\|_{\mathbb{L}^2}^2 d\tau$ is small, then stop. Else, goto 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., $\int_0^1 \|w(\tau)\|_{\mathbb{L}^2}^2 d\tau \leq 10^{-10}$.

B. Removing Orientations and Reparameterizations

In order to obtain a geodesic in shape space \mathcal{L}_n , we need to minimize the cost function $H(O, \gamma) = d_{l_n^c}(q_0, O(q_1 \circ \gamma)\sqrt{\gamma})$ over the product of manifolds $\text{SO}(n)$ and Γ . The algorithm in [13] solves this optimization by alternately optimizing between $\text{SO}(n)$ 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 can be shown 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

$$\text{grad}_O H(O, \gamma) = P_O \left(\int_{\mathbb{D}} \eta \sqrt{\gamma} (q_2 \circ \gamma)^T ds \right) \quad (\text{III.1})$$

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

Note that Γ is an infinite dimensional manifold. The gradient with respect to γ is approximated by

$$\text{grad}_\gamma H(O, \gamma) \approx \sum_i^k b_i D H(O, \gamma)[b_i], \quad (\text{III.2})$$

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

$$D H(O, \gamma)[b_i] = \left\langle \eta, O \left(\sqrt{\gamma} \dot{q}_1(\gamma) b_i + \frac{1}{2\sqrt{\gamma}} \dot{b}_i q_1(\gamma) \right) \right\rangle_{\mathbb{L}^2}.$$

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.,

$$\left\{ 1, \frac{\sin(nt)}{n\pi}, \frac{\cos(nt) - 1}{n\pi}, n = 1, 2, \dots \right\}. \quad (\text{III.3})$$

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

Algorithm 2 Removing rotation and reparameterization

Input: Two curves β_0 and β_1 , and step sizes $t_1, t_2 > 0$

- 1: Set $\tilde{\beta}_1$ to be β_1 , $O_0 = I$, $\gamma_0 = \gamma_{\text{id}}$, and $k = 0$.
 - 2: Compute the representations q_0 of β_0 and q_1 of $\tilde{\beta}_1$.
 - 3: Compute the geodesic α between q_0 and q_1 in l_n^c using Algorithm 1.
 - 4: Update the rotation by $O_{k+1} = O_k \exp(t_1 \text{grad}_I H(I, \gamma_{\text{id}}))$, where $\text{grad}_I H(I, \gamma_{\text{id}})$ is (III.1).
 - 5: Update the reparameterization by $\gamma_{k+1} = \gamma_k \circ (\gamma_{\text{id}} + t_2 \text{grad}_{\gamma_{\text{id}}} H(I, \gamma_{\text{id}}))$, where $\text{grad}_{\gamma_{\text{id}}} H(I, \gamma_{\text{id}})$ is (III.2). Note that t_2 should be small enough such that γ_{k+1} is nondecreasing.
 - 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 $l = \sqrt{\gamma}$ is used in H . It follows that the cost function is defined on $\text{SO}(n)$ and the first quadrant of the unit sphere $\mathcal{L} = \{l \in \mathbb{L}^2([0, 1], \mathbb{R}) \mid \|l\|_{\mathbb{L}^2} = 1\}$. Note that using the basis (III.3) essentially yields the same method in [13, Section 4.4] without the extra substitution step.

IV. A RIEMANNIAN APPROACH

The path straightening method in the shape space can be characterized as a steepest descent method with a fixed step size. It is well-known that the steepest descent method may have slow convergence, see e.g., [12]. In this paper, we apply a faster algorithm, a limited-memory version of Riemannian BFGS (Broyden-Fletcher-Goldfarb-Shanno) method (LRBFGS), which is introduced in [4] and shown to outperform many other state-of-the-art Riemannian algorithms for many large-scale problems, e.g., [12], [5], [6].

Since it is observed that Algorithm 2 dominates the computational time in the sense that it needs a large number of iterations, we only use LRBFGS to improve the performance of removing rotation and reparameterization.

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

$$\Gamma^c = [0, 1] \times \Gamma^o$$

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

$$\Gamma^o = \{\gamma : [0, 1] \rightarrow [0, 1] \mid \gamma \text{ is a diffeomorphism}\}.$$

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

$$f(O, m, l) = d_{l_n^c}(q_1, O l q_2 \left(\int_0^t l^2(s) ds + m \bmod 1 \right)), \quad (\text{IV.1})$$

where $(O, m, l) \in \text{SO}(n) \times \mathbb{R} \times \mathcal{L}$. We define the metric on the tangent space of $\text{SO}(n) \times \mathbb{R} \times \mathcal{L}$ by $\langle (U_1, b_1, v_1), (U_2, b_2, v_2) \rangle = \text{trace}(U_1^T U_2) + b_1 b_2 + \int_0^1 v_1 v_2 ds$. The Riemannian gradient of f with respect to this metric is given in Lemma 4.1 without proof.

Lemma 4.1: The Riemannian gradient of $f(O, m, l)$ in (IV.1) is

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

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

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

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

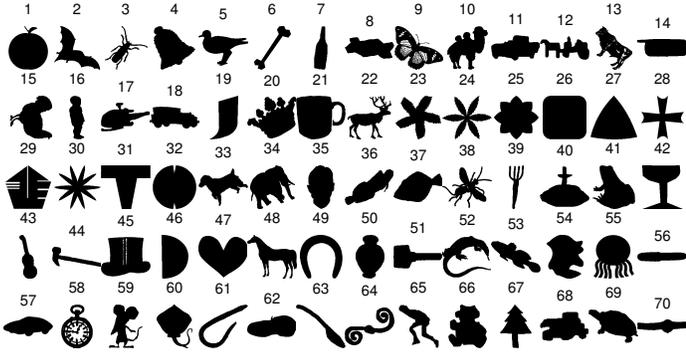


Fig. 2. Samples of curves from the MPEG-7 dataset. One sample per cluster is illustrated.

and the vector transport is

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

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

V. EXPERIMENTS

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. Figure 2 shows an example shape from each cluster. Matlab function `BWBOUNDARIES` is used to extract the boundary curves of the shapes and 100 uniformly-spaced points are chosen to represent each shape. A path in \mathcal{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).

We compare the performance of LRBFSGS and a Riemannian steepest descent (RSD) algorithm based on the framework in Section IV, as well as Algorithm 2 with multiple choices of step sizes t_1 and t_2 over many randomly chosen pairs of curves from the data set. The initial rotation and reparameterization for all algorithms are given by the approach in [5]. The average computational time and the average cost function values (distance) after each iteration are computed and the results are shown in Figure 3.

Since the step sizes, t_1 and t_2 are fixed, the choice of their values is important in Algorithm 2. If they are taken too small then the convergence would be slow. If taken too large then the cost function may not decrease. It is shown in Figure 2 that the average function values with $t_2 = 0.002$ oscillate. This is due to the fact that for that value of t_2 Algorithm 2 does not converge for some of the pairs of curves – the cost function value oscillates and does not decrease. Note, however, that all algorithms that converge for a particular pair of curves converge to the same critical point. RSD and LRBFSGS both use an efficient line search algorithm to determine a step size for each iteration that satisfies appropriate termination criteria

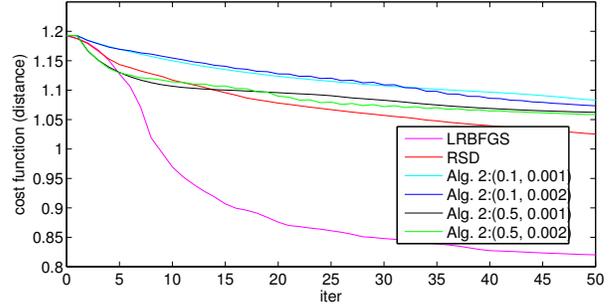
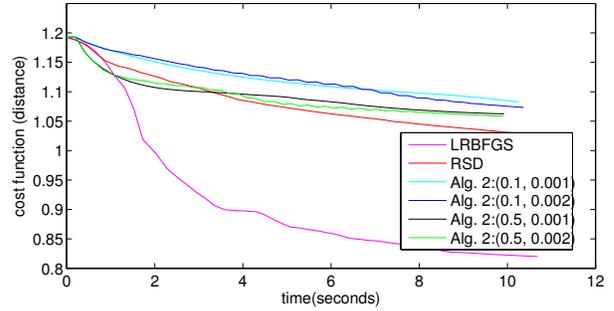


Fig. 3. Comparisons of algorithms with 50 iterations. The top graph shows the relationship between the computational time and the cost function values. The bottom graph shows the relationship between the number of iterations and the cost function values. The notation $\text{Alg.2:}(t_1, t_2)$ denotes the step sizes in Algorithm 2.

to guarantee convergence [3] and are therefore more robust than Algorithm 2.

RSD and Algorithm 2 have similar performance when the latter converges and both converge very slowly. LRBFSGS achieves a much smaller cost function value in less time and is clearly the best algorithm among the three when the effectiveness is taken into consideration.

VI. CONCLUSION AND FUTURE WORK

In this paper, we consider computing the geodesic in the shape space of elastic curves. A Riemannian manifold optimization approach is proposed as a replacement for the current state-of-the-art coordinate-descent/path-straightening approach in [13]. The Riemannian quasi-Newton method, LRBFSGS, is shown to be superior in both robustness and computational efficiency.

In the future, we will test the quality of the distance obtained by LRBFSGS in the sense of superior clustering, classification, and Karcher mean computations. The Riemannian approach will be included in the C++ Riemannian optimization library on <http://www.math.fsu.edu/ROPTLIB>.

ACKNOWLEDGMENT

This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme initiated by the Belgian Science Policy Office. This work was supported by grant FNRS PDR T.0173.13.

REFERENCES

- [1] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, Princeton, NJ, 2008.
- [2] I. L. Dryden and K. V. Mardia. *Statistical shape analysis*. Wiley, 1998.
- [3] W. Huang. *Optimization algorithms on Riemannian manifolds with applications*. PhD thesis, Florida State University, Department of Mathematics, 2013.
- [4] Wen Huang, K. A. Gallivan, and P.-A. Absil. A broyden class of quasi-newton methods for riemannian optimization. *SIAM Journal on Optimization*, 25(3):1660–1685, 2015.
- [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 Recognition*, 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.
- [9] E. Klassen and A. Srivastava. Geodesics between 3D closed curves using path-straightening. *Computer Vision/VECCV 2006*, pages 1–15, 2006.
- [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. 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.
- [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. doi:10.1109/TPAMI.2010.184.
- [14] Temple University. Shape similarity research project. www.dabi.temple.edu/shape/MPEG7/dataset.html.
- [15] L. Younes. Computable elastic distances between shapes. *SIAM Journal on Applied Mathematics*, 58(2):565–586, April 1998. doi:10.1137/S0036139995287685.
- [16] 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. doi:10.4171/RLM/506.