Riemannian Optimization for Elastic Shape Analysis

W. Huang[*], K. A. Gallivan[*], A. Srivastava[*] and P.-A. Absil[†]

[*]Florida State University [†]Catholic University of Louvain

MTNS 2014, Groningen, The Netherlands

Elastic Shape Analysis

- Elastic shape analysis invariants:
  - Rescaling
  - Translation
  - Rotation
  - Reparametrization
- Square Root Velocity Function framework used (Srivastava, Klassen, Joshi, and Jermyn [8]).
- extensive analysis and application of elastic shape
- much less work on understanding efficient and robust algorithms

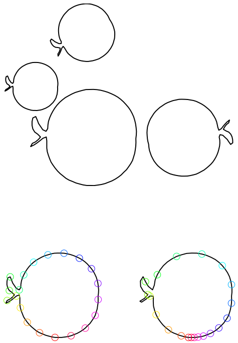


Figure : All are the same shape.

SRVF and Preshape Space

Preshape space, denoted $\mathfrak{l}_n$, removes translation and rescaling for $\mathbb{L}_2$.

- A shape is represented by a function $\beta : \mathbb{D} \to \mathbb{R}^2$, where $\mathbb{D}$ is $[0, 1]$ for open curves and unit circle $\mathbb{S}^1$ for closed curves.
- Square Root Velocity (SRV) function of the shape β is

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

- Preshape spaces (closure condition added for closed curves)

$$\mathfrak{l}_n^o = \{q : [0, 1] \to \mathbb{R}^n | \int_0^1 \|q(t)\|_2^2 dt = 1\}$$

$$\mathfrak{l}_n^c = \{q : \mathbb{S}^1 \to \mathbb{R}^n | \int_{\mathbb{S}^1} \|q(t)\|_2^2 dt = 1, \int_{\mathbb{S}^1} q(t)\|q(t)\|_2 dt = 0\}$$

Shape Space

Shape space removes rotation and reparameterization. Inherits metric from $\mathbb{L}_2$

$$\mathrm{SO}(n) = \{O \in \mathbb{R}^{n \times n} | O^T O = I_n, \det(O) = 1\}$$
$$\mathrm{SO}(n) \times \mathfrak{l}_n \to \mathfrak{l}_n : (O, q) \to Oq$$

$$\Gamma = \{\gamma : \mathbb{D} \to \mathbb{D} | \gamma \text{ is a diffeomorphism.}\}$$
$$\mathfrak{l}_n \times \Gamma \to \mathfrak{l}_n : (q, \gamma) \to (q \circ \gamma)\sqrt{\dot{\gamma}}$$

$$[q] = \{(O, (q, \gamma)) | O \in \mathrm{SO}(n), \gamma \in \Gamma\}$$

$$\mathfrak{L}_n = \mathfrak{l}_n / \mathrm{SO}(n) \times \Gamma = \{[q] | q \in \mathfrak{l}_n\}.$$

Best Rotation and Reparameterization

$$(O_*, \gamma_*) = \underset{(O,\gamma)\in \mathrm{SO}(n)\times\Gamma}{\operatorname{argmin}} \mathrm{dist}_{I_n}(q_1, O\sqrt{\dot{\gamma}}q_2 \circ \gamma).$$

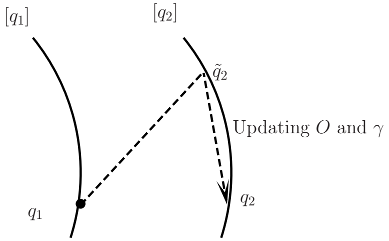


Figure : Align representation of $[q_2]$ with q_1.

Technicality

- The orbit $[q]$ is not closed. (O_*, γ_*) may not exist.
- Closure of orbits can be characterized using a semigroup Γ_s.

$$\Gamma_s = \{\gamma_s : \mathbb{D} \to \mathbb{D} | \gamma_s \text{ is an absolutely continuous, non-decreasing and surjective function } \}$$

  and the group action is the same as that of Γ.
- An orbit with Γ_s is $\overline{[q]}$, the closure of $[q]$.
- Γ and $[q]$ are dense in Γ_s and $\overline{[q]}$ respectively.
- Minimization problem

$$\min_{O \in \mathrm{SO}(n), \gamma_s, \tilde{\gamma}_s \in \Gamma_s} \mathrm{dist}_{I_n}\left(\sqrt{\dot{\tilde{\gamma}}_s} q_1 \circ \tilde{\gamma}_s, O \sqrt{\dot{\gamma}_s} q_2 \circ \gamma_s\right).$$

- Approximation solution is considered using diffeomorphisms in Γ.

Cost Functions

- Minimization problem

$$\min_{O \in \mathrm{SO}(n), \gamma \in \Gamma} \mathrm{dist}_{\mathfrak{l}_n}(Oq_1, (q_2, \gamma)).$$

- Open curve

$$d_{l_n^o}(Oq_1, (q_2 \circ \gamma)\sqrt{\dot{\gamma}}) = \cos^{-1} \langle Oq_1, (q_2 \circ \gamma)\sqrt{\dot{\gamma}} \rangle_{\mathbb{L}^2}$$

$$H^o(O, \gamma(t)) = \int_0^1 \|Oq_1(t) - (q_2 \circ \gamma(t))\sqrt{\dot{\gamma}(t)}\|_2^2 dt$$

- Closed curve
  - Closed form of preshape space distance is unknown.
  - Extrinsic distance is used.

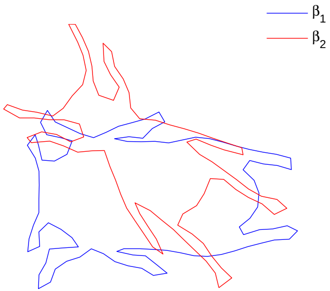
$$H^c(O, \gamma) = \int_{\mathbb{S}^1} \|Oq_1(t) - (q_2 \circ \gamma(t))\sqrt{\dot{\gamma}(t)}\|_2^2 dt$$

Coordinate Relaxation

Optimize rotation and reparameterization alternately.

- Open curves
  - Rotation: Procrustes problem solved using SVD
  - Reparameterization: Dynamic programming (DP) with slope constraints
- Closed curves
  - Choose a point on the closed curve and break it into an open curve
  - Apply coordinate relaxation method of open curves
  - Compare results for a sufficiently large number of break points

Two Shapes



Coordinate Relaxation Method

One iteration, denoted CR1, is used in [8].

- Complexity is $O(N^3)$, where N is the number of points in the curves.
- Note rotation and the correspondence of portions of the structures.
- Does iterating more improve results?

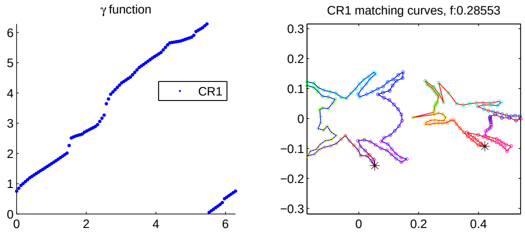
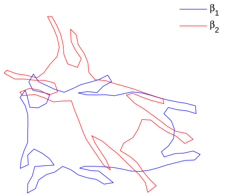


Figure : Results given by CR1

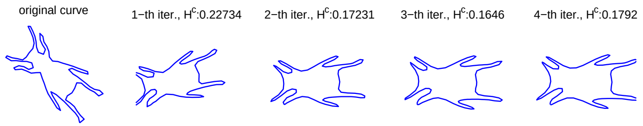
Representations and Implementation Difficulties

Representation Approach 1

- q_1 and q_2 are represented by points.
- Evaluation of (q_2, γ) over iterations on q
- $q_2^{(k+1)} = (q_2^{(k)}, \gamma^{(k)})$ computed on each iteration by evaluation of interpolating function of $q_2^{(k)}$.
- New interpolating function for $q_2^{(k+1)}$ $\rightarrow$ Shape of q_2 changes.



!



Representations and Implementation Difficulties

- Representation Approach 2
  - q_1 is represented by points and q_2 is represented by an fixed interpolating curve.
- Difficulty: Lack of compuational associativity may not reduce the cost function in practice
- Cost function evaluated in DP uses points $(q_2, \gamma^{(k)})$, and evaluates $((q_2, \gamma^{(k)}), \tilde{\gamma}^{(k+1)})$.
- Next q iterate is obtained using $(q_2, \gamma^{(k)} \circ \tilde{\gamma}^{(k+1)}) = (q_2, \gamma^{(k+1)})$ since fixed interpolation function for q_2
- Cost function values for the two forms of applying $\gamma^{(k+1)}$ can differ

iteration (k)	1	2	3
H^c iterate form	0.390583	0.378312	0.390114
H^c in DP	0.285534	0.248016	0.241679

Table : Computed cost function values. Difference continues growing with k.

Riemannian Approach

- Optimizing H is a Riemannian optimization problem on $\mathrm{SO}(n) \times \Gamma$.
- Many Riemannian optimization algorithms have been systematically analyzed recently.
  - Riemannian trust-region Newton method (RTR-Newton) [2]
  - Riemannian Broyden family method including BFGS method and its limited-memory version (RBroyden family, RBFGS, LRBFGS) [7, 4, 6]
  - Riemannian trust-region symmetric rank-one update method and its limited-memory version (RTR-SR1, LRTR-SR1) [4, 5]
  - Riemannian Newton method (RNewton) [1]
- See W. Huang's thesis, Optimization algorithms on Riemannian manifolds with applications, FSU, Math Dept. [4] for details on analysis, applications and library design

Cost Function of Closed Curves

- Γ^c is represented by its covering space, i.e., $\tilde{\Gamma} \times \mathbb{R}$ where

$$\tilde{\Gamma} = \{\gamma : [0, 2\pi] \to [0, 2\pi] | \gamma \text{ is diffeomorphism}\}.$$

  and the $\tilde{\Gamma} \times \mathbb{R}$ group action on q is defined by

$$(q, (\gamma, m)) = (q(\gamma + m \mod 2\pi))\sqrt{\dot{\gamma}}, \quad (\gamma, m) \in \tilde{\Gamma} \times \mathbb{R}.$$

- The cost function on the Riemannian manifold $\mathrm{SO}(n) \times \mathbb{R} \times \tilde{\Gamma}$ is

$$H^c(O, m, \gamma) = \int_0^{2\pi} \|Oq_1(t) - (q_2(\gamma(t) + m \mod 2\pi))\sqrt{\dot{\gamma}(t)}\|_2^2 dt$$

  where $\gamma(0) = 0$, $\int_0^{2\pi} \dot{\gamma}(t)dt = 2\pi$, $\dot{\gamma} > 0$.

2-norm Sphere

- Optimization on the manifold $\tilde{\Gamma}$ directly has some difficulties, e.g., step limits due to limited domains of the exponential map $Exp_\gamma(v) = \gamma + v$
- $\tilde{\Gamma}$ can be replaced with the 2-norm sphere
- Replace the term $\sqrt{\dot{\gamma}(t)}$ in H^c by a function ℓ.
- $\ell \geq 0$ and $\ell \in \mathbb{S}_{\mathbb{L}_2}$, where $\mathbb{S}_{\mathbb{L}_2} = \{\ell \in C^0 | \int_0^{2\pi} \ell^2(t)dt = 2\pi\}$.
- A constrained optimization is obtained

$$\min_{O \in \mathrm{SO}(n), m \in \mathbb{R}, \ell \in \mathbb{S}_{\mathbb{L}_2}, \ell \geq 0} \int_0^{2\pi} \| O q_1(t) - q_2\left(\int_0^t \ell^2(s) ds + m \mod 2\pi\right) \ell(t) \|_2^2 dt.$$

4-norm Sphere

To avoid the constrained optimization, 4-norm sphere is used instead.

- 4-norm sphere
- Replace the term $\sqrt{\dot{\gamma}(t)}$ in H^c by a function ℓ^2.
- $\ell \in \mathbb{S}_{\mathbb{L}_4}$, where $\mathbb{S}_{\mathbb{L}_4} = \{\ell \in C^0 | \int_0^{2\pi} \ell^4(t)dt = 2\pi\}$.
- A unconstrained optimization is obtained

$$\min_{O \in \mathrm{SO}(n), m \in \mathbb{R}, \ell \in \mathbb{S}_{\mathbb{L}_4}} L(O, m, \ell)$$

where

$$L(O, m, \ell) = \int_0^{2\pi} \| O q_1(t) - q_2(\int_0^t l^4(s)ds + m \mod 2\pi)\ell^2(t)\|_2^2 dt.$$

Barrier Function

- A barrier function can be added to avoid the slope of γ being zero or going to ∞:

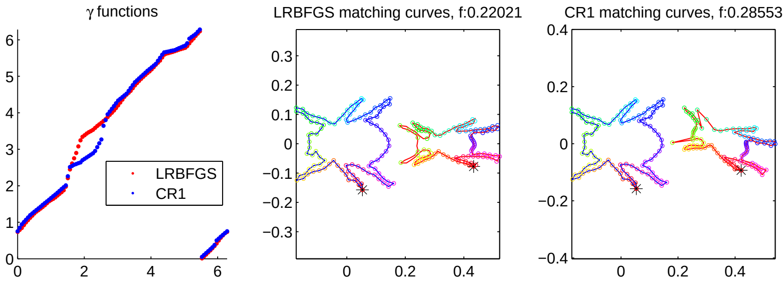
$$B(\gamma) = \int_0^{2\pi} (\dot{\gamma}(t) + \frac{1}{\dot{\gamma}(t)}) \sqrt{1 + \dot{\gamma}^2(t)} dt = \int_0^{2\pi} (\ell^4(t) + \frac{1}{\ell^4(t)}) \sqrt{1 + \ell^8(t)} dt$$

  which satisfies the symmetric property, i.e., $B(\gamma) = B(\gamma^{-1})$.
- The user can control the approach to a slope of 0 or ∞.

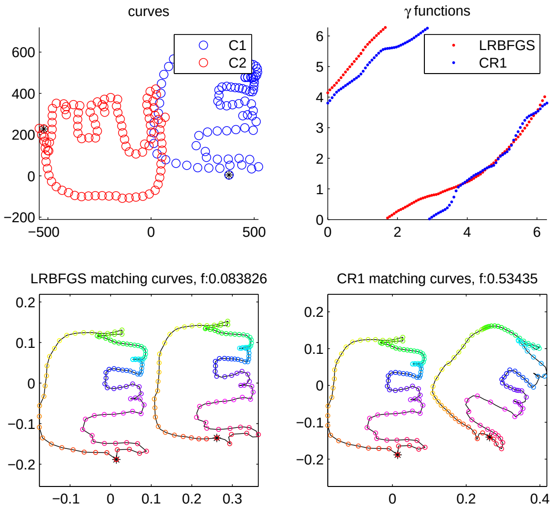
Riemmanian Algorithm

- q_1 is represented by points and q_2 is represented by an interpolating curve.
- Multiple values of m are used based on the variation of angle along the curve.
- Procrustes and DP on a coarse grid give initial ℓ_0 and O_0 for each m.
- Improvements
    - Keep the shape of q_2 constant
    - Avoid the problem with computational associativity of group action
    - Computational complexity reduces

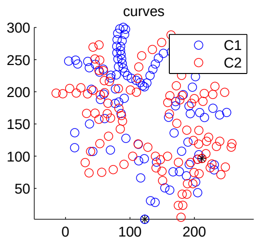
Example

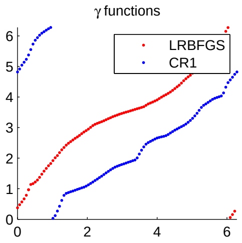

γ functions
LRBFGS
CR1
LRBFGS matching curves, f:0.22021
CR1 matching curves, f:0.28553
0
1
2
3
4
5
6
0
2
4
6
0.3
0.2
0.1
0
−0.1
−0.2
−0.3
0
0.2
0.4
0.4
0.2
0
−0.2
−0.4
0
0.2
0.4

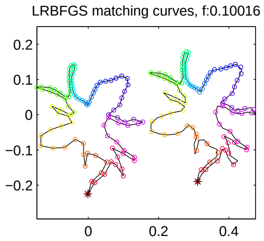
Known γ_T: rotation and γ off

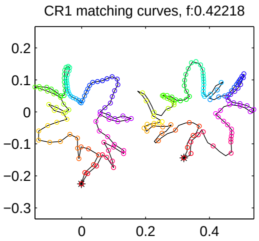


Known γ_T: rotation and γ off significantly









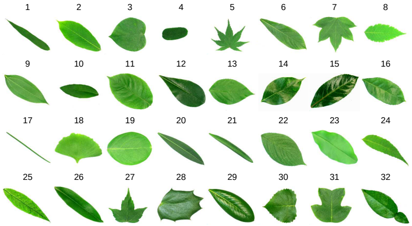
Data Sets

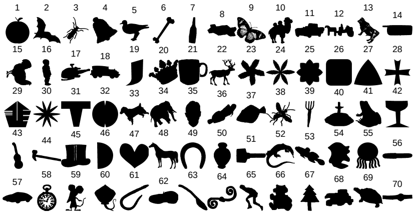
Flavia leaf dataset [10]

- 1907 images of leaves
- 32 species

MPEG-7 dataset [9]

- 1400 binary images
- 70 clusters





- Boundary curves: BWBOUNDARIES function in Matlab
- 100 points in $\mathbb{R}^2$ used for each boundary

Representative of Riemannian Algorithm

- Five Riemannian methods are tested.
- 1000 pairs of shape in each data set are used.
- Based on the following table, LRBFGS is chosen to be the representative one.

		RBFGS	LRBFGS	RTR-SR1	LRTR-SR1	RSD
Flavia dataset	L_{ave}	0.1727	0.1836	0.1772	0.1958	0.2079
	t_{ave}	0.4113	0.1525	0.4585	0.2052	0.2218
MPEG-7 dataset	L_{ave}	0.3639	0.3919	0.3735	0.4407	0.4798
	t_{ave}	1.2823	0.4370	1.3352	0.5572	0.7537

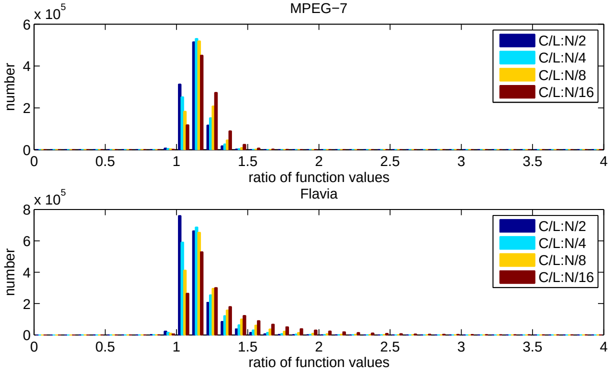
Table : Comparison of Riemannian Methods for representative sets from the Flavia and MPEG-7 datasets: average time per pair (t_{ave}) in seconds and average cost function per pair (L_{ave}).

Comparisons of LRBFGS and CR1

Test Environment and Tests Performed

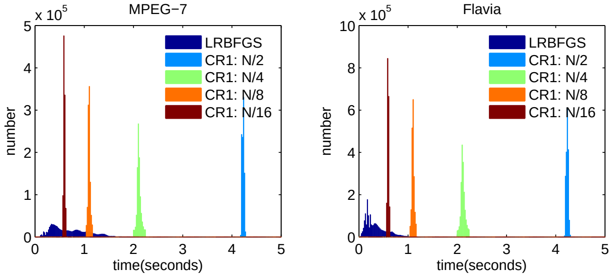
- Environment
  - All codes written in C and compiled with gcc
  - Performs on Florida State University HPC system using Quad-Core 2356 2.3 GHz Opterons [3]
- Experiments
  - Compute all pairwise distances in the Flavia and MPEG-7 respectively
  - For CR1 method, the results of the breaking points chosen to be every 2, 4, 8, 16 point are reported.

Cost Function Ratios



- Percent of Flavia pairs reduced 99.2%, 99.4%, 99.6% and 99.8% for $N/i, i = 2, 4, 8, 16$
- Percent of MPEG-7 pairs reduced 98.5%, 99.0%, 99.3% and 99.6% for $N/i, i = 2, 4, 8, 16$

Computational Time Ratios



- LRBFGS computation time adjusts with based on the complexity of shape based on number of *m* points.
- CR1 is essentially constant due to simple choice of number of break points.
- LRBFGS generically faster even with same number of initial points.

One Nearest Neighbor Results

- The quality of the extrinsic distance computations is assessed by the one nearest neighbor (1NN) metric
- The 1NN metric, μ, computes the percentage of points whose nearest neighbor are in the same cluster, i.e.,

$$\mu = \frac{1}{n}\sum_{i=1}^{n} C(i), \quad C(i) = \begin{cases} 1 & \text{if point } i \text{ and its nearest neighbor} \\ & \text{are in the same cluster;} \\ 0 & \text{otherwise.} \end{cases}$$

One Nearest Neighbor Results

		LRBFGS	CR1			
			$N/16$	$N/8$	$N/4$	$N/2$
Flavia	ave. time (sec.)	0.37201	0.59379	1.1026	2.1203	4.2404
	1NN of 32 species	87%	76%	79%	81%	85%
MPEG-7	ave. time (sec.)	0.74442	0.59272	1.1006	2.1164	4.2327
	1NN of 70 clusters	98%	92%	95%	96%	97%

Table : The average computation time and 1NN of LRBFGS and CR1 with break points chosen to be every 2, 4, 8 and 16 points.

Conclusion and Future Work

- Conclusion
  - CR with multiple iterations unreliable; composition unreliable
  - CR1 may not be able to find an accurate solution
  - Riemannian approach is faster, better results, and more robust for more complicated shapes than CR1
- Future work
  - Intrinsic optimization for closed curves
  - Analysis of effects of discretization on accuracy
  - Test the influence of the accuracy of distance in other shape analyses, e.g., geodesic, means
  - Combination with more robust global reparameterization optimization of Klassen et al.

References I

[1] P.-A. Absil, R. Mahony, and R. Sepulchre.
Optimization algorithms on matrix manifolds.
Princeton University Press, Princeton, NJ, 2008.

[2] C. G. Baker.
Riemannian manifold trust-region methods with applications to eigenproblems.
PhD thesis, Florida State University, 2008.

[3] Florida State University Research Computing Center.
FSU high performance computing system.

[4] W. Huang.
Optimization algorithms on Riemannian manifolds with applications.

PhD thesis, Florida State University, 2013.

[5] W. Huang, P.-A. Absil, and K. A. Gallivan.
A Riemannian symmetric rank-one trust-region method.
Mathematical Programming, 2014.

References II

[6] W. Huang, K. A. Gallivan, and P.-A. Absil.
A Broyden class of quasi-Newton methods for Riemannian optimization.
Submitted for publication, 2014.

[7] W. Ring and B. Wirth.
Optimization methods on Riemannian manifolds and their application to shape space.
SIAM Journal on Optimization, 22(2):596–627, January 2012.

[8] 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.

[9] Temple University.
Shape similarity research project.

References III

[10] S. G. Wu, F. S. Bao, E. Y. Xu, Y.-X. Wang, Y.-F. Chang, and Q.-L. Xiang.
A leaf recognition algorithm for plant classification using probabilistic neural network.
2007 IEEE International Symposium on Signal Processing and Information Technology, pages 11–16, 2007.