A Riemannian Approach for Computing Geodesics in Elastic Shape Analysis

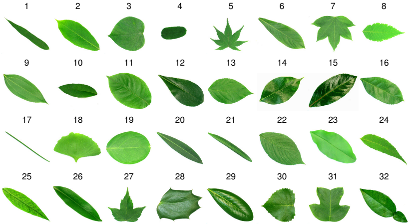
Yaqing You[1], Wen Huang[2], Kyle A. Gallivan[1] and P.-A. Absil[2]

[1]Florida State University

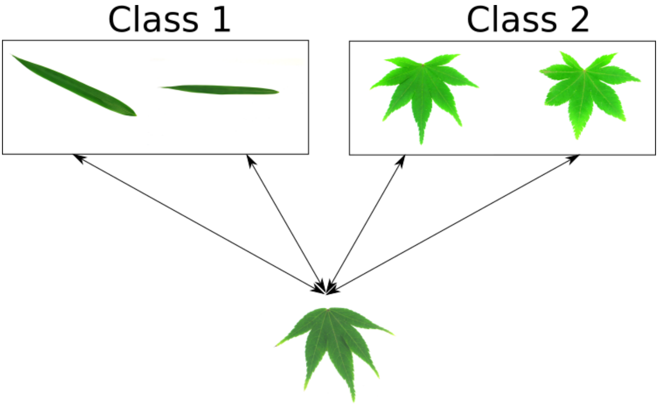
[2]Université Catholique de Louvain

December 15, 2015

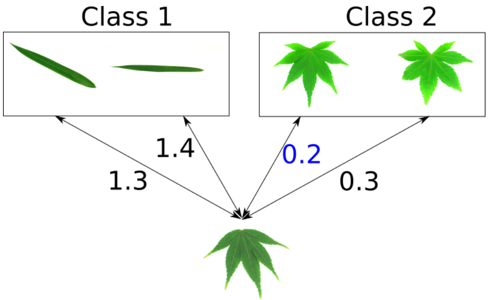
Possible Applications of Shape Analysis



A Simplified Shape Classification Problem

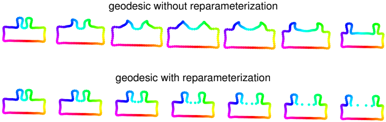


Outline: Shape Classification Based on Shape Distances



- A Classification method
- Requires a notion of distance between shapes
  → we use a well-known elastic shape distance
- An efficient method to compute shape distances
  - An algorithm that computes the distances
    → our contribution
  - Experiments on a bigger dataset

- In elastic shape analysis, a shape is invariant to
  - Translation
  - Scaling
  - Rotation
  - Reparameterization



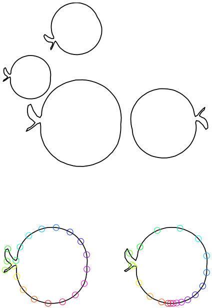


Figure: All are the same shape.

Geodesic and Distance

- Geodesic is path with constant velocity

- Distance is the length of the shortest geodesic

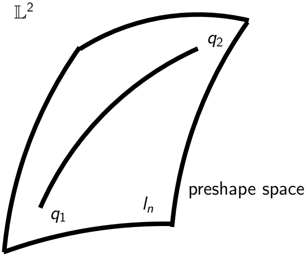
Representation of Shapes: Removing Translation and Rescaling

The square root velocity (SRV) framework given in [SKJJ11]:

- The SRVF is:

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

where $\|\cdot\|_2$ denotes the 2-norm.



- The preshape space l_n (that removes translation and rescaling) is

$$\left\{ q \in \mathbb{L}^2 | \int_{\mathbb{S}^1} ||q(t)|| dt = 1, \int_{\mathbb{S}^1} q(t)||q(t)|| dt = 0 \right\}.$$

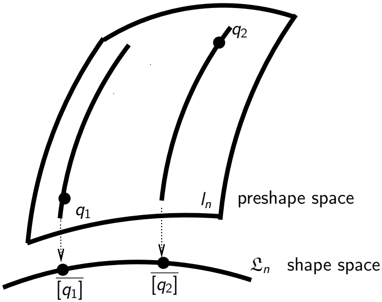
Rotation and Reparameterization Group

- The rotation group

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

- The reparameterization group

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



Shape Classification Based on Shape Distances:

Outline:

- Classification method: 1-Nearest Neighbor
- Requires a notion of distance between shapes
  $\rightarrow$ we use a well-known elastic shape distance
- An efficient method to compute shape distances
  - An algorithm that computes the distances
    $\rightarrow$ our contribution
  - Experiments on a bigger dataset

Geodesic Algorithm Description

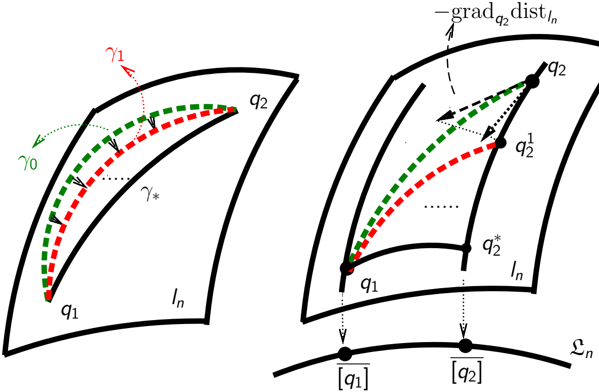


Figure: l_n: preshape space; $\mathfrak{L}_n$: shape space.

Left: Path-straightening method [SKJJ11] in preshape space of closed curves;
Right: Remove rotation and reparameterization with:

(i) coordinate descent [SKJJ11]

(ii): Riemannian method (our approach)

Removing Rotation and Reparameterization

- CD (Coordinate Descent) [SKJJ11]
  - Fixed stepsize
  - Slow convergence for small stepsize
  - May not converge for large stepsize

- Riemannian optimization methods:
  - RSD (Riemannian Steepest Descent)
    - Riemannian Armijo condition for stepsize choosing
    - Global convergence to a stationary point
    - Linear convergence rate
  - LRBFGS (Limited-memory version of Riemannian BFGS)
    - Riemannian Armijo condition for stepsize choosing
    - Global convergence to a stationary point
    - Faster convergence rate

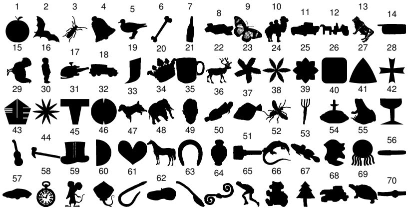
Shape Classification Based on Shape Distances

Outline:

- Classification method: 1-Nearest Neighbor
- Requires a notion of distance between shapes
  $\rightarrow$ we use a well-known elastic shape distance
- An efficient method to compute shape distances
  - An algorithm that computes the distances
    $\rightarrow$ our contribution
  - Experiments on a bigger dataset

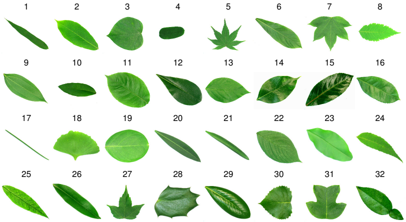
MPEG-7 dataset [Uni]

- 1400 binary images
- 70 clusters



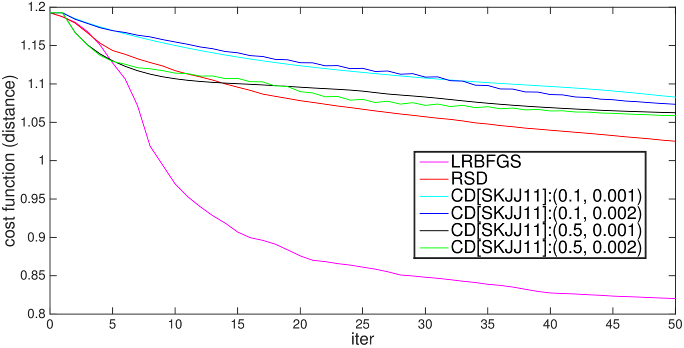
Flavia leaf dataset [WBX+07]

- 1907 images of leaves
- 32 species



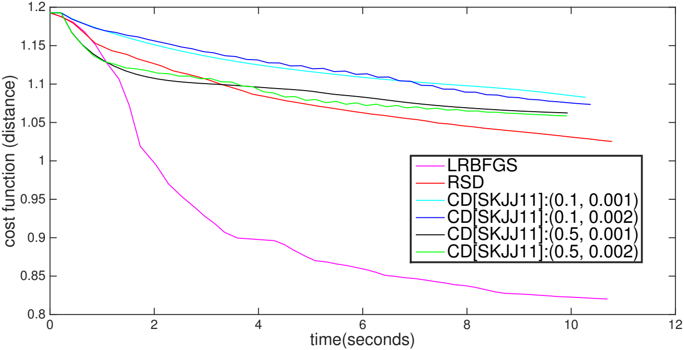
- Boundary curves: BWBOUNDARIES function in Matlab
- 100 points in $\mathbb{R}^2$ used for each boundary curve
- A path in preshape space is represented by 11 curves

Experiment I : Results



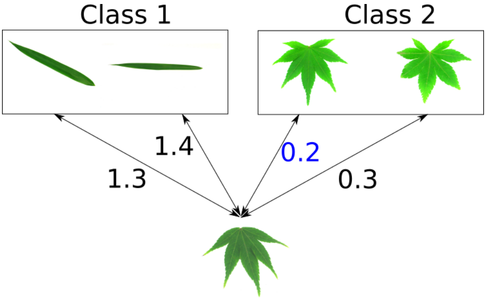
Comparisons of algorithms with 50 iterations. This figure shows the average relationship between the number of iterations and the cost function values.

Experiment I : Results



Comparisons of algorithms with 50 iterations. This figure shows the average relationship between the computational time and the cost function values.

Experiment II : One Nearest Neighbor Results



- The 1NN metric, μ, computes the percentage of points whose nearest neighbor are in the same class, i.e,

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

Experiment II : 1NN Results

- Flavia dataset is used
- Consider 10 species and 5 images from each species, i.e., 50 images in total
- All pairwise distances are considered
- Stopping criteria: relative change of the cost function (distance) smaller than 10^{-4}

Results:

	CD[SKJJ11]	RSD	LRBFGS
μ	0.88	0.90	0.90
Mean Compute Time(sec)	34.25	30.97	11.05

Summary

- A well-known elastic shape distance is used
- An existing path-straightening method in preshape space is used
- An efficient algorithm to compute geodesic and shape distances in shape space $\rightarrow$ our contribution
  - A Riemannian approach applied to the path-straightening in preshape space
  - A good initial guess for the O and γ is used
  - Better performance in terms of time and robustness
  - Comparable or even better 1NN results

References I

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.

Temple University.
Shape similarity research project.

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.
arXiv:0707.4289v1.