Riemannian gradient descent for spherical area-preserving mappings

Marco Sutti

Joint work with Mei-Heng Yueh

NCTS Postdoc Symposium

September 9, 2024

Overview

Paper: Riemannian gradient descent for spherical area-preserving mappings, M. Sutti and M.-H. Yueh, AIMS Math., Vol. 9(7), 19414–19445, 12 June 2024.

Main contributions:

(i) Combine tools from Riemannian optimization and computational geometry to propose a Riemannian gradient descent (RGD) method for computing spherical area-preserving mappings of topological spheres.

(ii) Numerical experiments on several mesh models demonstrate the accuracy and efficiency of the algorithm.

(iii) Competitiveness and efficiency of our algorithm over three state-of-the-art methods for computing area-preserving mappings.

This talk:

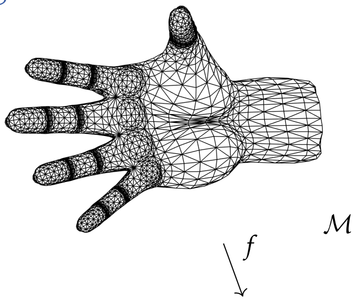
I. Simplicial surfaces and mappings, stretch and authalic energy.

II. Optimization on matrix manifolds, fundamental ideas and tools.

III. Numerical experiments.

I. Simplicial surfaces and mappings, authalic and stretch energies

Simplicial surfaces and mappings/1

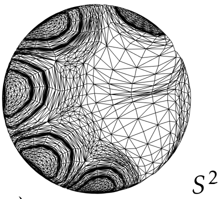


- A simplicial surface $\mathcal{M}$ is the underlying set of a simplicial 2-complex $\mathcal{K}(\mathcal{M}) = \mathcal{F}(\mathcal{M}) \cup \mathcal{E}(\mathcal{M}) \cup \mathcal{V}(\mathcal{M})$ composed of vertices

$$\mathcal{V}(\mathcal{M}) = \left\{ v_\ell = \left(v_\ell^1, v_\ell^2, v_\ell^2 \right)^\top \in \mathbb{R}^3 \right\}_{\ell=1}^{n},$$

oriented triangular faces

$$\mathcal{F}(\mathcal{M}) = \left\{ \tau_\ell = [v_{i_\ell}, v_{j_\ell}, v_{k_\ell}] \mid v_{i_\ell}, v_{j_\ell}, v_{k_\ell} \in \mathcal{V}(\mathcal{M}) \right\}_{\ell=1}^{m},$$



and undirected edges

$$\mathcal{E}(\mathcal{M}) = \left\{ [v_i, v_j] \mid [v_i, v_j, v_k] \in \mathcal{F}(\mathcal{M}) \text{ for some } v_k \in \mathcal{V}(\mathcal{M}) \right\}.$$

- A simplicial mapping $f : \mathcal{M} \to \mathbb{R}^3$ is a particular type of piecewise affine mapping with the restriction mapping $f|_\tau$ being affine, for every $\tau \in \mathcal{F}(\mathcal{M})$.

Simplicial surfaces and mappings/2

- We denote

$$\mathbf{f}_\ell := f(v_\ell) = \left(f_\ell^1, f_\ell^2, f_\ell^3\right)^\top,$$

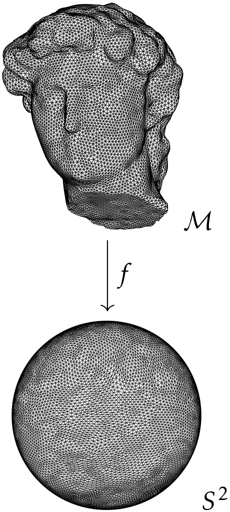
for every $v_\ell \in \mathcal{V}(\mathcal{M})$.

- The (image of the) mapping f can be represented as a matrix

$$\mathbf{f} = \begin{bmatrix} \mathbf{f}_1^\top \\ \vdots \\ \mathbf{f}_n^\top \end{bmatrix} = \begin{bmatrix} f_1^1 & f_1^2 & f_1^3 \\ \vdots & \vdots & \vdots \\ f_n^1 & f_n^2 & f_n^3 \end{bmatrix} =: \begin{bmatrix} \mathbf{f}^1 & \mathbf{f}^2 & \mathbf{f}^3 \end{bmatrix},$$

or a vector

$$\operatorname{vec}(\mathbf{f}) = \begin{bmatrix} \mathbf{f}^1 \\ \mathbf{f}^2 \\ \mathbf{f}^3 \end{bmatrix}.$$



- A simplicial mapping $f : \mathcal{M} \to \mathbb{R}^3$ is said to be area-preserving if $|f(\tau)| = |\tau|$ for every $\tau \in \mathcal{F}(\mathcal{M})$.

Authalic energy

The authalic (or equiareal) energy for simplicial mappings $f : \mathcal{M} \to \mathbb{R}^3$ is

$$E_A(f) = E_S(f) - \mathcal{A}(f),$$

where $\mathcal{A}(f)$ is the image area, E_S is the stretch energy defined as

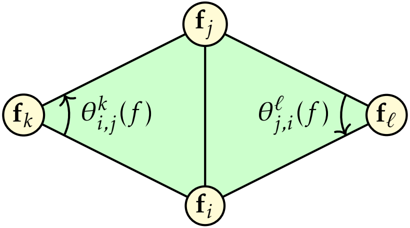
$$E_S(f) = \frac{1}{2}\operatorname{vec}(\mathbf{f})^\top (I_3 \otimes L_S(f)) \operatorname{vec}(\mathbf{f}),$$

where $L_S(f)$ is the weighted Laplacian matrix $L_S(f)$, defined by

$$[L_S(f)]_{i,j} = \begin{cases} -\sum_{[v_i,v_j,v_k]\in\mathcal{F}(\mathcal{M})}[\omega_S(f)]_{i,j,k} & \text{if } [v_i,v_j]\in\mathcal{E}(\mathcal{M}), \\ -\sum_{\ell\neq i}[L_S(f)]_{i,\ell} & \text{if } j = i, \\ 0 & \text{otherwise,} \end{cases}$$

in which $\omega_S(f)$ is the modified cotangent weight defined as

$$[\omega_S(f)]_{i,j,k} = \frac{\cot(\theta_{i,j}^k(f))\,|f([v_i,v_j,v_k])|}{2|[v_i,v_j,v_k]|}.$$



Stretch energy/1

- The stretch energy can be reformulated as [see Lemma 3.1, Yueh 2023]

$$E_S(f) = \sum_{\tau \in \mathcal{F}(\mathcal{M})} \frac{|f(\tau)|^2}{|\tau|}.$$

- (If the area-preserving simplicial mapping exists) then every minimizer of $E_S(f)$ is an area-preserving mapping and vice-versa [Theorem 3.3, Yueh 2023], i.e.,

$$f = \underset{|g(\mathcal{M})|=|\mathcal{M}|}{\operatorname{argmin}} E_S(g)$$

  if and only if $|f(\tau)| = |\tau|$ for every $\tau \in \mathcal{F}(\mathcal{M})$.

- It is also proved that $E_A(f) \geqslant 0$ and the equality holds if and only if f is area-preserving [Corollary 3.4, Yueh 2023].

Theoretical foundation of the stretch energy minimization for area-preserving simplicial mappings: [Yueh 2023]

Stretch energy/2

- Due to the optimization process, $\mathcal{A}(f)$ varies, hence we introduce a prefactor $|\mathcal{M}|/\mathcal{A}(f)$ and define the normalized stretch energy as

$$E(f) = \frac{|\mathcal{M}|}{\mathcal{A}(f)} E_S(f).$$

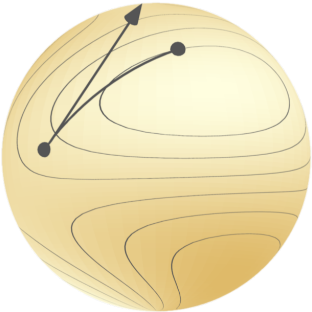
- To perform numerical optimization we need to compute the Euclidean gradient of $E(f)$. By applying the formula $\nabla E_S(f) = 2\,(I_3 \otimes L_S(f))\,\mathrm{vec}(\mathbf{f})$ from [Yueh 2023], the gradient of $E(f)$ can be formulated as

$$\begin{aligned}
\nabla E(f) &= \nabla\left(\frac{|\mathcal{M}|}{\mathcal{A}(f)} E_S(f)\right) \\
&= \frac{|\mathcal{M}|}{\mathcal{A}(f)} \nabla E_S(f) + E_S(f) \nabla \frac{|\mathcal{M}|}{\mathcal{A}(f)} \\
&= \frac{2|\mathcal{M}|}{\mathcal{A}(f)} (I_3 \otimes L_S(f))\,\mathrm{vec}(\mathbf{f}) - \frac{|\mathcal{M}| E_S(f)}{\mathcal{A}(f)^2} \nabla \mathcal{A}(f).
\end{aligned}$$

II. Riemannian optimization framework and geometry

Riemannian optimization/1

- The Riemannian optimization framework solves constrained optimization problems where the constraints have a geometric nature.
  - Exploit the underlying geometric structure of the problems. The optimization variables are constrained to a smooth manifold.

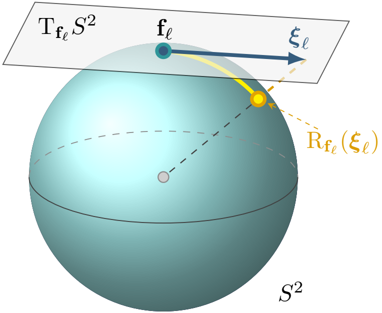


- In our setting: The problem is formulated on a power manifold of n unit spheres embedded in $\mathbb{R}^3$, and we use the RGD method for minimizing the cost function on this power manifold.
- Traditional optimization methods rely on the Euclidean space structure.
  - For instance, the steepest descent method for minimizing $g\colon \mathbb{R}^n \to \mathbb{R}$ updates $\mathbf{x}_k$ by moving in the direction $\mathbf{d}_k$ of the anti-gradient of g, by a step size α_k chosen according to an appropriate line-search rule.

Manifold optimization: [Edelman et al. 1998, Absil et al. 2008, Boumal 2023], …
The image above has been taken from the Manopt website: https://www.manopt.org/

Riemannian optimization/2

- A line-search method in the Riemannian framework determines at $\mathbf{x}_k$ on a manifold M a search direction $\boldsymbol{\xi}$ on $\mathrm{T}_{\mathbf{x}}M$.
- $\mathbf{x}_{k+1}$ is then determined by a line search along a curve $\alpha \mapsto \mathrm{R}_{\mathbf{x}}(\alpha\boldsymbol{\xi})$ where $\mathrm{R}_{\mathbf{x}} \colon \mathrm{T}_{\mathbf{x}}M \to M$ is the retraction mapping.
- Repeat for $\mathbf{x}_{k+1}$ taking the role of $\mathbf{x}_k$.



- Search directions can be the negative of the Riemannian gradient, leading to the Riemannian gradient descent method (RGD).
  - Other choices of search directions $\rightsquigarrow$ other methods, e.g., Riemannian trust-region method or Riemannian BFGS.

Riemannian trust-region method: [Absil/Baker/Gallivan 2007], Riemannian BFGS: [Ring/Wirth 2012]

Geometry of the unit sphere S^2

The unit sphere S^2 is a Riemannian submanifold of $\mathbb{R}^3$ defined as

$$S^2 = \{\mathbf{x} \in \mathbb{R}^3 : \mathbf{x}^\top \mathbf{x} = 1\}.$$

The Riemannian metric on the unit sphere is inherited from $\mathbb{R}^3$, i.e.,

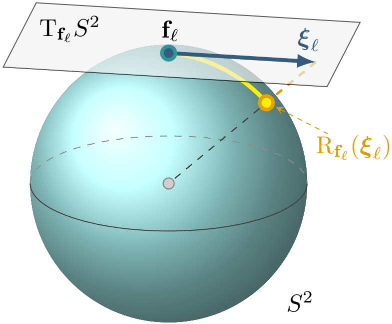
$$\langle \boldsymbol{\xi}, \boldsymbol{\eta} \rangle_{\mathbf{x}} = \boldsymbol{\xi}^\top \boldsymbol{\eta}, \quad \boldsymbol{\xi}, \boldsymbol{\eta} \in \mathrm{T}_{\mathbf{x}} S^2,$$

where $\mathrm{T}_{\mathbf{x}} S^2$ is the tangent space to S^2 at $\mathbf{x} \in S^2$, defined as the set of all vectors orthogonal to $\mathbf{x}$ in $\mathbb{R}^3$, i.e.,

$$\mathrm{T}_{\mathbf{x}} S^2 = \{\mathbf{z} \in \mathbb{R}^3 : \mathbf{x}^\top \mathbf{z} = 0\}.$$

The projector $\mathrm{P}_{\mathrm{T}_{\mathbf{x}} S^2} : \mathbb{R}^3 \to \mathrm{T}_{\mathbf{x}} S^2$ is defined by

$$\mathrm{P}_{\mathrm{T}_{\mathbf{x}} S^2}(\mathbf{z}) = (I_3 - \mathbf{x}\mathbf{x}^\top)\,\mathbf{z}.$$



In the following, points on the unit sphere are denoted by $\mathbf{f}_\ell$ (the vertices of the simplicial mapping f), and tangent vectors are represented by $\boldsymbol{\xi}_\ell$.

Geometry of the power manifold $\left(S^2\right)^n$

We aim to minimize the function $E(f) = E(\mathbf{f}_1, \ldots, \mathbf{f}_n)$, where each $\mathbf{f}_\ell$, $\ell = 1, \ldots, n$, lives on the same manifold S^2.

$\rightsquigarrow$ This leads us to consider the power manifold of n unit spheres

$$\left(S^2\right)^n = \underbrace{S^2 \times S^2 \times \cdots S^2}_{n \text{ times}},$$

with the metric of S^2 extended elementwise.

In the next slides, we present the tools from Riemannian geometry needed to generalize gradient descent to this manifold, namely:

- The projector onto the tangent space to $\left(S^2\right)^n$ is used to compute the Riemannian gradient.
- The projection onto $\left(S^2\right)^n$ turns points of $\mathbb{R}^{n\times 3}$ into points of $\left(S^2\right)^n$.
- The retraction turns an objective function defined on $\mathbb{R}^{n\times 3}$ into an objective function defined on the manifold $\left(S^2\right)^n$.

Projector onto the tangent space to $\left(S^2\right)^n$

Here, the points are denoted by $\mathbf{f}_\ell \in \mathbb{R}^3$, $\ell = 1, \ldots, n$, so we write

$$\mathrm{P}_{\mathrm{T}_{\mathbf{f}_\ell} S^2} = I_3 - \mathbf{f}_\ell \mathbf{f}_\ell^\top.$$

It clearly changes for every vertex $\mathbf{f}_\ell$. The projector from $\mathbb{R}^{n\times 3}$ onto the tangent space at $\mathbf{f}$ to the power manifold $\left(S^2\right)^n$ is a mapping

$$\mathrm{P}_{\mathrm{T}_{\mathbf{f}}\left(S^2\right)^n} \colon \mathbb{R}^{n\times 3} \to \mathrm{T}_{\mathbf{f}}\left(S^2\right)^n,$$

and can be represented by a block diagonal matrix of size $3n \times 3n$, i.e.,

$$\mathrm{P}_{\mathrm{T}_{\mathbf{f}}\left(S^2\right)^n} := \operatorname{blkdiag}\left(\mathrm{P}_{\mathrm{T}_{\mathbf{f}_1} S^2}, \mathrm{P}_{\mathrm{T}_{\mathbf{f}_2} S^2}, \ldots, \mathrm{P}_{\mathrm{T}_{\mathbf{f}_n} S^2}\right) = \begin{bmatrix} \mathrm{P}_{\mathrm{T}_{\mathbf{f}_1} S^2} & & & \\ & \mathrm{P}_{\mathrm{T}_{\mathbf{f}_2} S^2} & & \\ & & \ddots & \\ & & & \mathrm{P}_{\mathrm{T}_{\mathbf{f}_n} S^2} \end{bmatrix}.$$

Projection onto the power manifold $\left(S^2\right)^n$

The projection of a single vertex $\mathbf{f}_\ell$ from $\mathbb{R}^3$ to the unit sphere S^2 is given by the normalization

$$\widetilde{\mathbf{f}}_\ell = \frac{\mathbf{f}_\ell}{\|\mathbf{f}_\ell\|_2}.$$

Hence, the projection of the whole of $\mathbf{f}$ onto the power manifold $\left(S^2\right)^n$ is given by

$$\mathrm{P}_{\left(S^2\right)^n} \colon \mathbb{R}^{n\times 3} \to \left(S^2\right)^n,$$

defined by

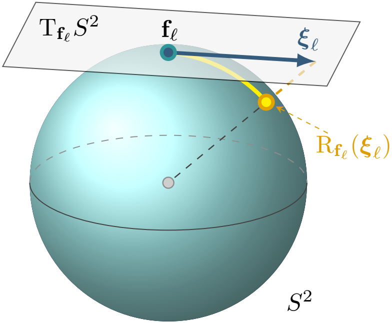
$$\mathbf{f} \mapsto \widetilde{\mathbf{f}} := \operatorname{diag}\left(\frac{1}{\|\mathbf{f}_1\|_2}, \frac{1}{\|\mathbf{f}_2\|_2}, \dots, \frac{1}{\|\mathbf{f}_n\|_2}\right)\begin{bmatrix}\mathbf{f}_1 & \mathbf{f}_2 & \cdots & \mathbf{f}_n\end{bmatrix}^\top.$$

This representative matrix is only shown for illustrative purposes; in the actual implementation, we use row-wise normalization of $\mathbf{f}$.

Retraction

- The retraction of a tangent vector $\boldsymbol{\xi}_\ell$ from $\mathrm{T}_{\mathbf{f}_\ell} S^2$ to S^2 is a mapping $\mathrm{R}_{\mathbf{f}_\ell} \colon \mathrm{T}_{\mathbf{f}_\ell} S^2 \to S^2$, defined by

$$\mathrm{R}_{\mathbf{f}_\ell}(\boldsymbol{\xi}_\ell) = \frac{\mathbf{f}_\ell + \boldsymbol{\xi}_\ell}{\|\mathbf{f}_\ell + \boldsymbol{\xi}_\ell\|}.$$



- For the power manifold $\left(S^2\right)^n$, the retraction of all the tangent vectors $\boldsymbol{\xi}_\ell$, $\ell = 1, \ldots, n$, is a mapping $\mathrm{R}_{\mathbf{f}} \colon \mathrm{T}_{\mathbf{f}}\left(S^2\right)^n \to \left(S^2\right)^n$, defined by

$$\begin{bmatrix} \boldsymbol{\xi}_1 & \cdots & \boldsymbol{\xi}_n \end{bmatrix}^\top \mapsto \operatorname{diag}\left(\frac{1}{\|\mathbf{f}_1 + \boldsymbol{\xi}_1\|_2}, \ldots, \frac{1}{\|\mathbf{f}_n + \boldsymbol{\xi}_n\|_2}\right) \begin{bmatrix} \mathbf{f}_1 + \boldsymbol{\xi}_1 & \cdots & \mathbf{f}_n + \boldsymbol{\xi}_n \end{bmatrix}^\top.$$

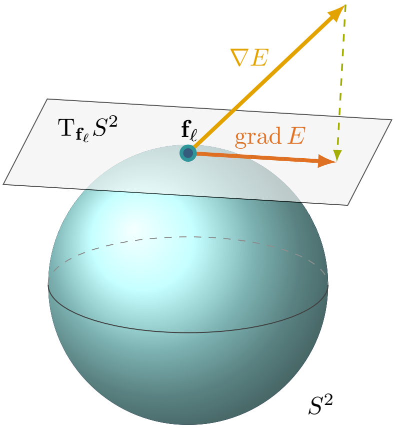
Constructing retractions: [Absil/Malick 2012]

Riemannian gradient descent method/1

- The Riemannian gradient of the objective function E is given by the projection onto $\mathrm{T}_{\mathbf{f}}\big(S^2\big)^n$ of the Euclidean gradient of E, namely,

$$\operatorname{grad} E(f) = \mathrm{P}_{\mathrm{T}_{\mathbf{f}}\left(S^2\right)^n}(\nabla E(f)).$$

- This is always the case for embedded submanifolds; see Prop. 3.6.1 in Absil et al., 2008.



Riemannian gradient descent method/2

Algorithm 1: The RGD method on $\left(S^2\right)^n$.

Given objective function E, power manifold $\left(S^2\right)^n$, initial iterate(*) $\mathbf{f}^{(0)} \in \left(S^2\right)^n$, projector $\mathrm{P}_{\mathrm{T}_{\mathbf{f}}\left(S^2\right)^n}$ from $\mathbb{R}^{n\times 3}$ to $\mathrm{T}_{\mathbf{f}}\left(S^2\right)^n$, retraction $\mathrm{R}_{\mathbf{f}}$ from $\mathrm{T}_{\mathbf{f}}\left(S^2\right)^n$ to $\left(S^2\right)^n$;

Result: Sequence of iterates $\{f^{(k)}\}$.

$k \leftarrow 0$;

while *$f^{(k)}$ does not sufficiently minimizes E* **do**

  Compute the Euclidean gradient of the objective function $\nabla E(f^{(k)})$;

  Compute the Riemannian gradient as $\operatorname{grad} E(f^{(k)}) = \mathrm{P}_{\mathrm{T}_{\mathbf{f}^{(k)}}\left(S^2\right)^n}\left(\nabla E(f^{(k)})\right)$;

  Choose the anti-gradient direction $\mathbf{d}^{(k)} = -\operatorname{grad} E(f^{(k)})$;

  Use a line-search procedure to compute a step size $\alpha_k > 0$ that satisfies the sufficient decrease condition;

  Set $\mathbf{f}^{(k+1)} = \mathrm{R}_{\mathbf{f}^{(k)}}(\alpha_k \mathbf{d}^{(k)})$;

  $k \leftarrow k + 1$;

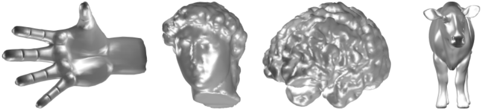
end while

(*) The initial mapping $\mathbf{f}^{(0)} \in \left(S^2\right)^n$ is computed via the fixed-point iteration (FPI) method of Yueh et al., 2019, until the first increase in energy is detected.

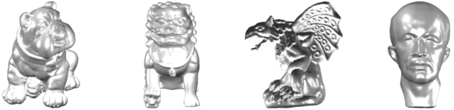
III. Numerical experiments

The benchmark triangular mesh models

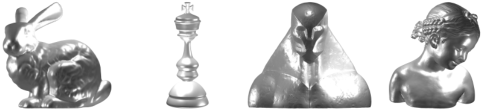
Model Name	Right Hand	David Head	Cortical Surface	Bull
# Faces	8,808	21,338	30,000	34,504
# Vertices	4,406	10,671	15,002	17,254



Model Name	Bulldog	Lion Statue	Gargoyle	Max Planck
# Faces	99,590	100,000	100,000	102,212
# Vertices	49,797	50,002	50,002	51,108

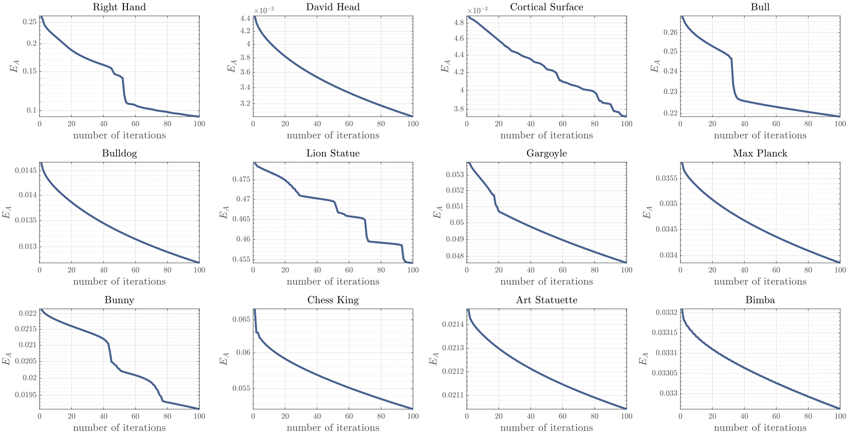


Model Name	Bunny	Chess King	Art Statuette	Bimba
# Faces	111,364	263,712	895,274	1,005,146
# Vertices	55,684	131,858	447,639	502,575



Resulting spherical mappings

Convergence behavior of RGD



Comparison with other methods/1

Comparison with the fixed-point iteration method for minimizing the authalic energy E_A of Yueh et al., 2019.

	Fixed point method [Yueh et al. 19]			Our RGD method		
Model Name	SD/Mean	$E_A(f)$	Time	SD/Mean	$E_A(f)$	Time
Right Hand	0.4598	2.92×10^{0}	1.35	0.1204	9.40×10^{-2}	4.07
David Head	0.0169	3.58×10^{-3}	4.30	0.0156	3.04×10^{-3}	9.16
Cortical Surface	0.0174	3.21×10^{-3}	5.62	0.0200	3.72×10^{-3}	16.01
Bull	0.1876	4.59×10^{-1}	6.90	0.1348	2.19×10^{-1}	18.89
Bulldog	0.1833	3.99×10^{-1}	22.22	0.0343	1.27×10^{-2}	61.93
Lion Statue	0.2064	5.28×10^{-1}	23.67	0.1894	4.54×10^{-1}	76.76
Gargoyle	4.1020	4.85×10^{2}	36.10	0.0646	4.76×10^{-2}	80.52
Max Planck	0.1844	1.67×10^{1}	25.99	0.0525	3.39×10^{-2}	75.60
Bunny	0.0394	3.96×10^{-2}	35.78	0.0390	1.91×10^{-2}	89.62
Chess King	1.0903	1.79×10^{1}	88.04	0.0647	5.23×10^{-2}	207.47
Art Statuette	0.0908	1.07×10^{-1}	342.95	0.0405	2.10×10^{-2}	654.57
Bimba Statue	0.0932	7.42×10^{-2}	305.00	0.0512	3.29×10^{-2}	775.36

Fixed-point iteration method for minimizing the authalic energy: [Yueh et al. 2019]

Comparison with other methods/2

Comparison with the adaptive area-preserving parameterization for genus-zero closed surfaces proposed by Choi et al., 2022.

	Choi et al., 2022			Our RGD method		
Model Name	SD/Mean	$E_A(f)$	Time	SD/Mean	$E_A(f)$	Time
Right Hand	18.3283	4.84×10^{3}	218.03	0.1204	9.40×10^{-2}	4.07
David Head	0.0426	2.27×10^{-2}	298.76	0.0156	3.04×10^{-3}	9.16
Cortical Surface	0.6320	1.14×10^{0}	420.20	0.0200	3.72×10^{-3}	16.01
Bull	8.5565	1.82×10^{3}	34.42	0.1348	2.19×10^{-1}	18.89
Bulldog	9.2379	1.22×10^{3}	183.94	0.0343	1.27×10^{-2}	61.93
Lion Statue	0.2626	8.96×10^{-1}	1498.91	0.1894	4.54×10^{-1}	76.76
Gargoyle	0.3558	1.30×10^{0}	1483.35	0.0646	4.76×10^{-2}	80.52
Max Planck	11.6875	1.49×10^{3}	195.39	0.0525	3.39×10^{-2}	75.60
Bunny	27.6014	8.94×10^{3}	157.87	0.0390	1.91×10^{-2}	89.62
Chess King	11.8300	1.65×10^{3}	608.55	0.0647	5.23×10^{-2}	207.47
Art Statuette	394.4414	9.93×10^{0}	2284.79	0.0405	2.10×10^{-2}	654.57
Bimba Statue	0.5110	2.01×10^{0}	16 773.34	0.0512	3.29×10^{-2}	775.36

Adaptive area-preserving parameterization for genus-zero closed surfaces: [Choi/Giri/Kumar 2022]

Comparison with other methods/3

Comparison with the spherical optimal transportation mapping proposed by Cui et al., 2019. The executable fails to output a mapping for eight mesh models among the twelve, which are not shown in the table.

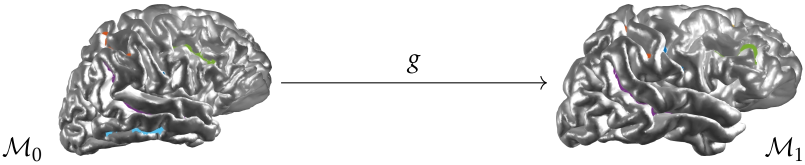
	Cui et al., 2019			Our RGD method		
Model Name	SD/Mean	$E_A(f)$	#Its.	SD/Mean	$E_A(f)$	Time
David Head	0.4189	2.25×10^0	27	0.0156	3.04×10^{-3}	9.16
Cortical Surface	0.5113	3.11×10^0	27	0.0200	3.72×10^{-3}	16.01
Bulldog	0.8665	1.00×10^1	33	0.0343	1.27×10^{-2}	61.93
Max Planck	0.5619	4.38×10^0	25	0.0525	3.39×10^{-2}	75.60

Spherical optimal transportation mapping: [Cui et al. 2019]

Surface registration/1

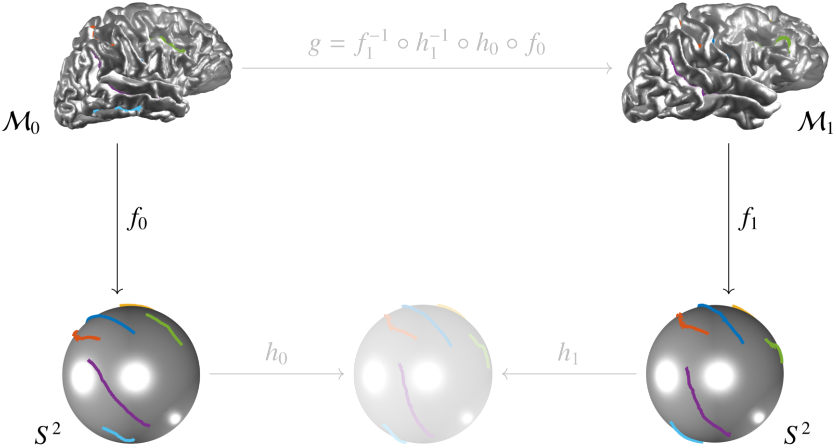
- A registration mapping between surfaces $\mathcal{M}_0$ and $\mathcal{M}_1$ is a bijective mapping $g\colon \mathcal{M}_0 \to \mathcal{M}_1$. An ideal registration mapping keeps important landmarks aligned while preserving specified geometry properties.
- Framework for the computation of landmark-aligned area-preserving parameterizations of genus-zero closed surfaces.
- Illustration with the landmark-aligned morphing process from one brain to another.

Problem statement: Given a set of landmark pairs $\{(p_i, q_i) \mid p_i \in \mathcal{M}_0,\ q_i \in \mathcal{M}_1\}_{i=1}^m$, our goal is to compute an area-preserving simplicial mapping $g\colon \mathcal{M}_0 \to \mathcal{M}_1$ that satisfies $g(p_i) \approx q_i$, for $i = 1, \ldots, m$.

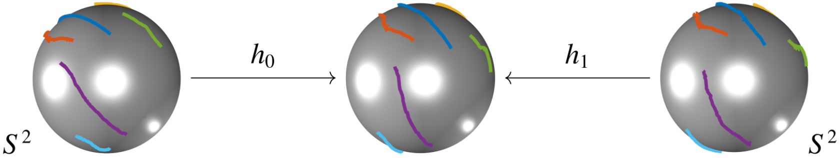


Surface registration/2

- First, we compute area-preserving parameterizations $f_0 \colon \mathcal{M}_0 \to S^2$ and $f_1 \colon \mathcal{M}_1 \to S^2$ of surfaces $\mathcal{M}_0$ and $\mathcal{M}_1$, respectively.



Surface registration/3



- The simplicial mapping $h\colon S^2 \to S^2$ that satisfies $h \circ f_0(p_i) = f_1(q_i)$, for $i = 1, \dots, m$, can be carried out by minimizing the registration energy

$$E_R(h) = E_S(h) + \sum_{i=1}^{m} \lambda_i \|h \circ f_0(p_i) - f_1(q_i)\|^2.$$

- Let $\mathbf{h}$ be the matrix representation of h. The gradient of E_R with respect to $\mathbf{h}$ can be formulated as

$$\nabla E_R(h) = 2\,(I_3 \otimes L_S(h))\,\mathrm{vec}(\mathbf{h}) + \mathrm{vec}(\mathbf{r}),$$

where $\mathbf{r}$ is the matrix of the same size as $\mathbf{h}$ given by

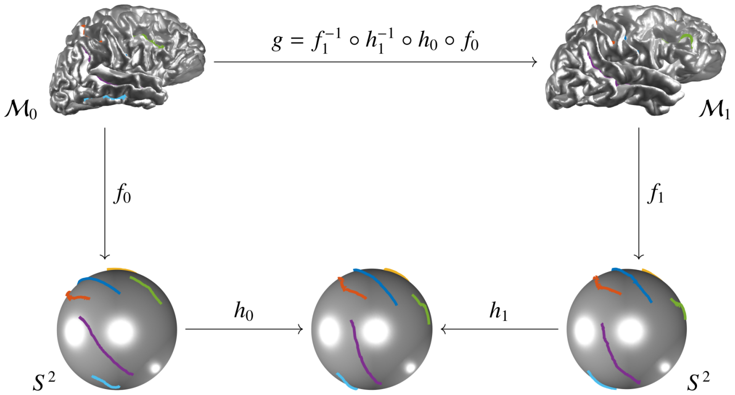
$$\mathbf{r}(i,:) = \begin{cases} 2\lambda_i\,(\mathbf{h}(i,:) - (f_1(q_i))^\top) & \text{if } p_i \text{ is a landmark,} \\ (0,0,0) & \text{otherwise.} \end{cases}$$

Surface registration/4

- In practice, we define the midpoints c_i of each landmark pairs on S^2 as

$$c_i = \frac{1}{2}(f_0(p_i) + f_1(q_i)),$$

for $i = 1, \ldots, m$, and compute h_0 and h_1 on S^2 that satisfy $h_0 \circ f_0(p_i) = c_i$ and $h_1 \circ f_1(q_i) = c_i$, respectively. The registration mapping $g\colon \mathcal{M}_0 \to \mathcal{M}_1$ is obtained by the composition $g = f_1^{-1} \circ h_1^{-1} \circ h_0 \circ f_0$.

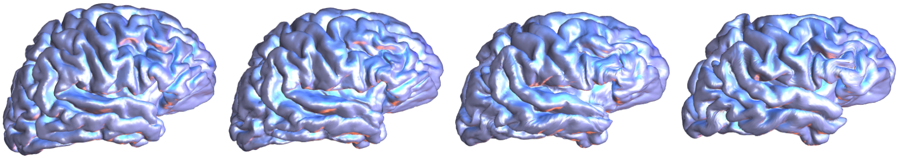


Brain morphing

- Brain morphing via the linear homotopy method.
- A landmark-aligned morphing process from $\mathcal{M}_0$ to $\mathcal{M}_1$ can be constructed by the linear homotopy $H\colon \mathcal{M}_0 \times [0,1] \to \mathbb{R}^3$ defined as

$$H(v,t) = (1-t)\,v + t\,g(v).$$

- We demonstrate the morphing process from one brain to another brain by four snapshots at four different values of t.



(a) $t = 0$ (b) $t = 0.5$ (c) $t = 2$ (f) $t = 15$

Conclusions

Summary:

- Combining the tools of Riemannian optimization and computational geometry, we introduced an RGD method for computing spherical area-preserving mappings of genus-zero closed surfaces.
- We conducted extensive numerical experiments on various mesh models to demonstrate the algorithm's stability and effectiveness.
- We applied our algorithm to the practical problem of landmark-aligned surface registration between two human brain models.

Outlook:

- Enhance the speed of convergence of the algorithm using appropriate Riemannian generalizations of the conjugate gradient method or the limited memory BFGS method.
- Target genus-one closed surfaces, e.g., the ring torus.

Thank you for your attention!