# Non-Rigid Surface Correspondence

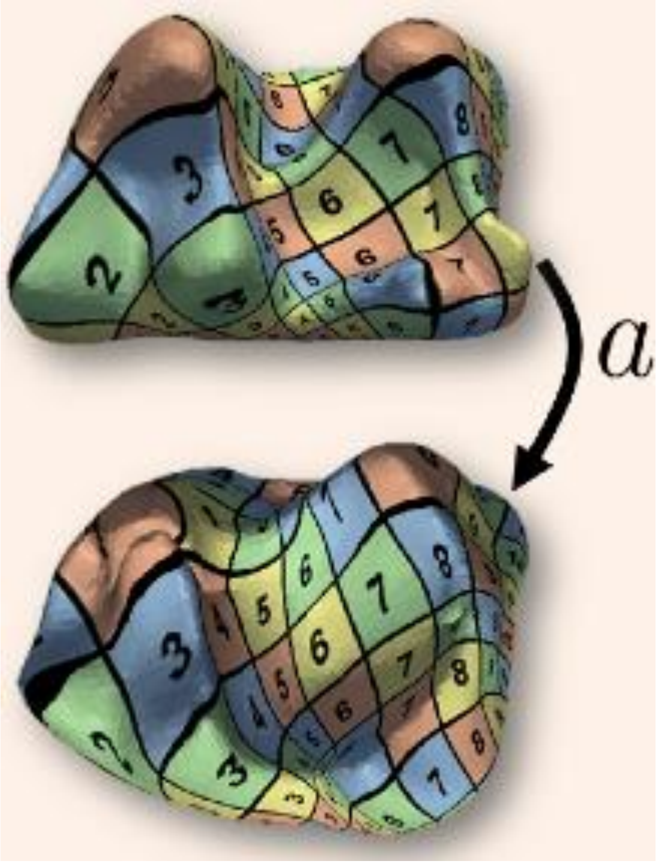
Thomas Funkhouser

Princeton University

# Goal

Find maps between surfaces

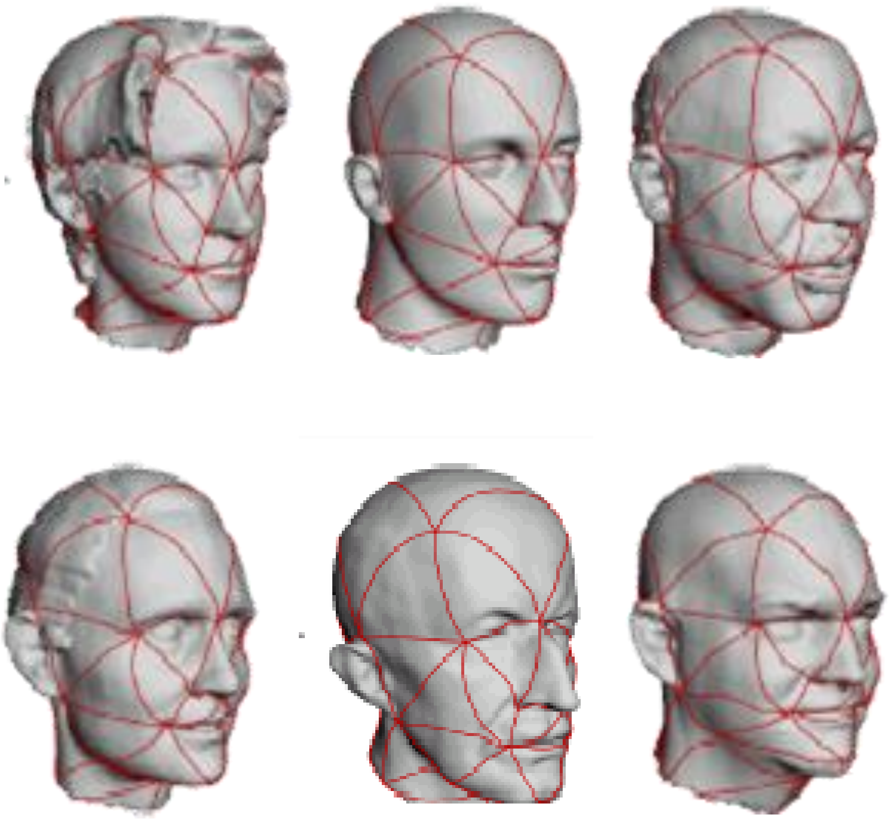
- Non-rigid
- Bijective
- Smooth
- Shape preserving
- Automatic
- Efficient computation
- Provide metric
- Semantic alignment



# Motivating Applications

Finding corresponding points on surfaces enables …

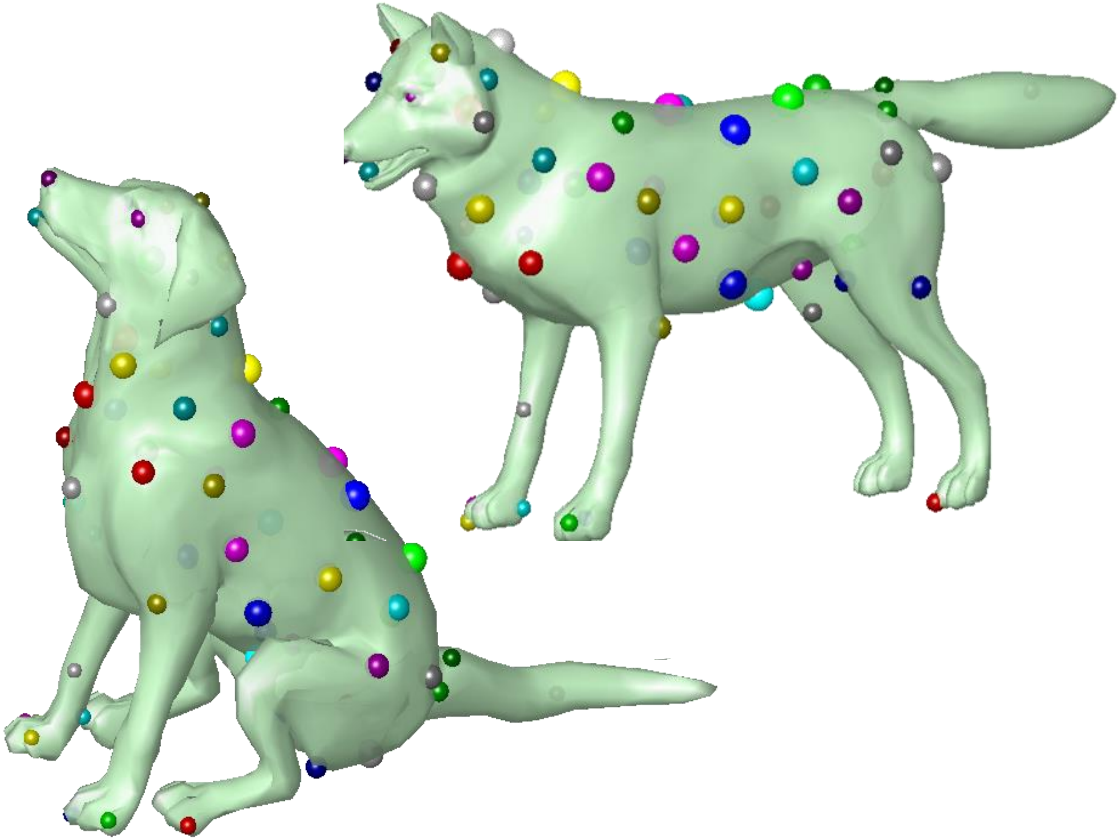
- Surface comparison
- Collection analysis
- Property transfer
- Morphing
- etc.



[Praun et al.]

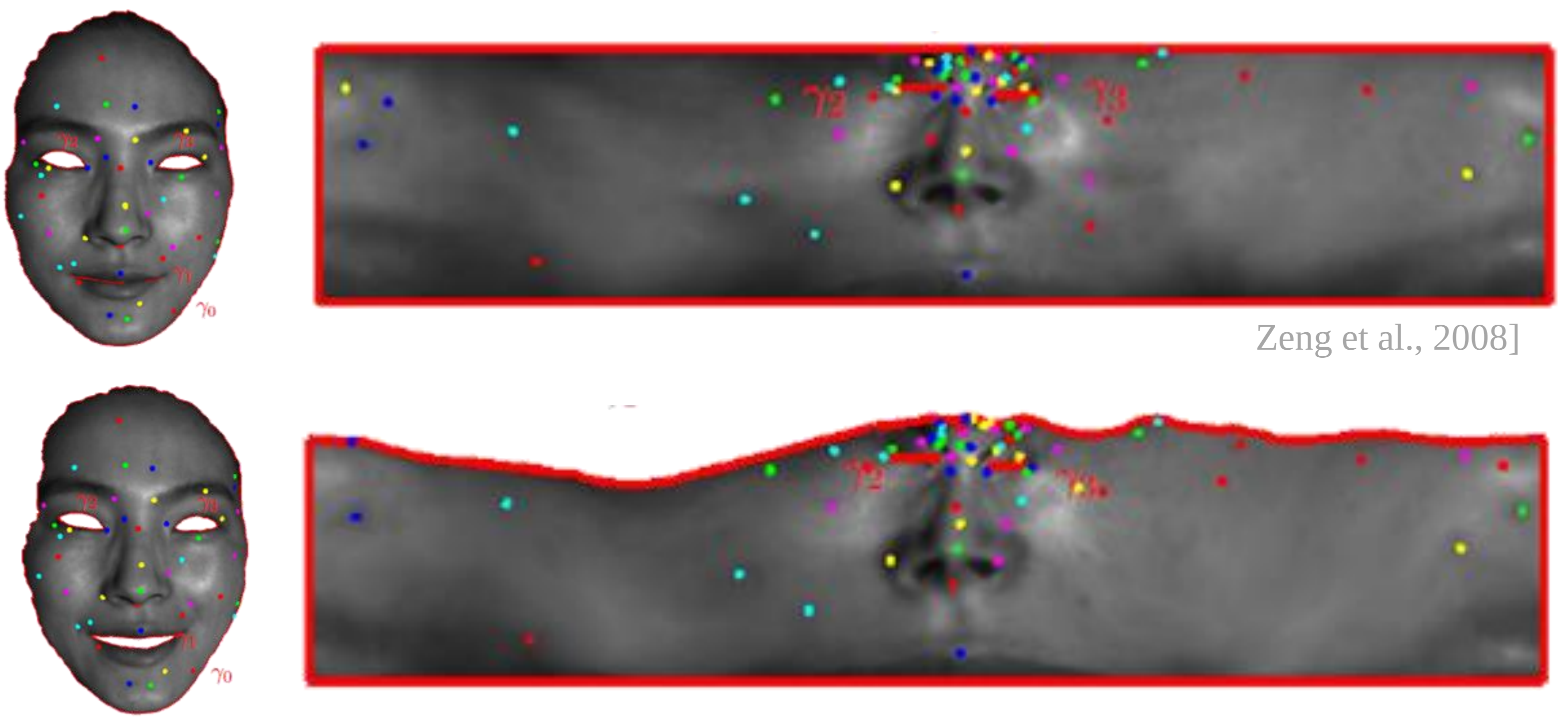
# Problem 1

Find a sparse set of feature correspondences



# Problem 2

Compute a dense map from
a sparse set of feature correspondences



Zeng et al., 2008]

Least Squares Conformal Map
(preserve angles as best as possible)

# Outline

Introduction

Some surface mapping algorithms

- Feature correspondence search
- High-dimensional embedding
- Möbius transformations
- Blended maps

Example Application

Conclusion

Future work

# Outline

Introduction

Some surface mapping algorithms

- Feature correspondence search
- High-dimensional embedding
- Möbius transformations
- Blended maps

Example Application

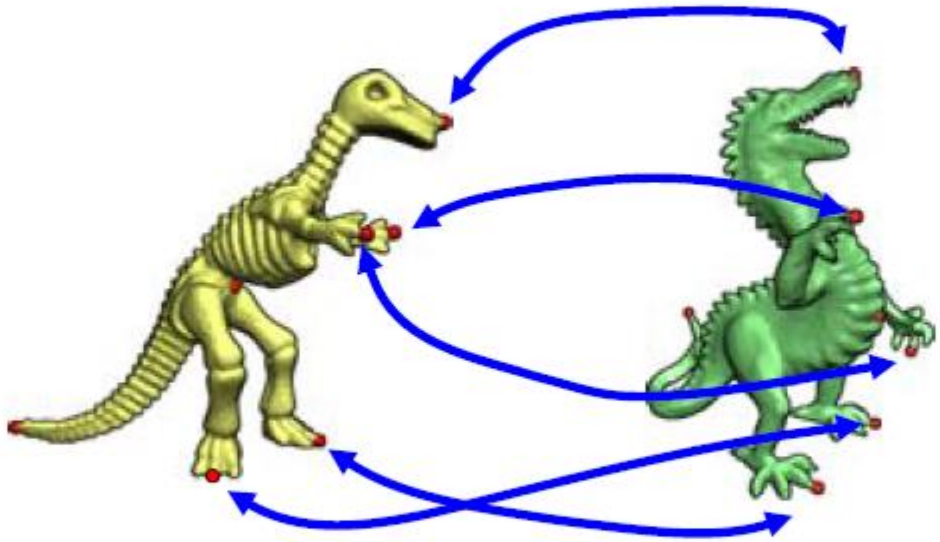
Conclusion

Future work

# Feature Correspondence Search

For each coarse set of feature correspondences:

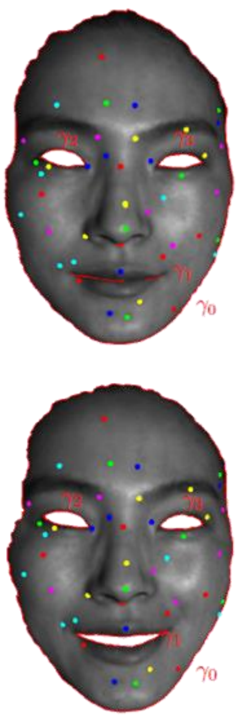
- Measure the deformation required to align them
  - … maybe by solving problem 2
- Remember the one with least deformation



# Feature Correspondence Search

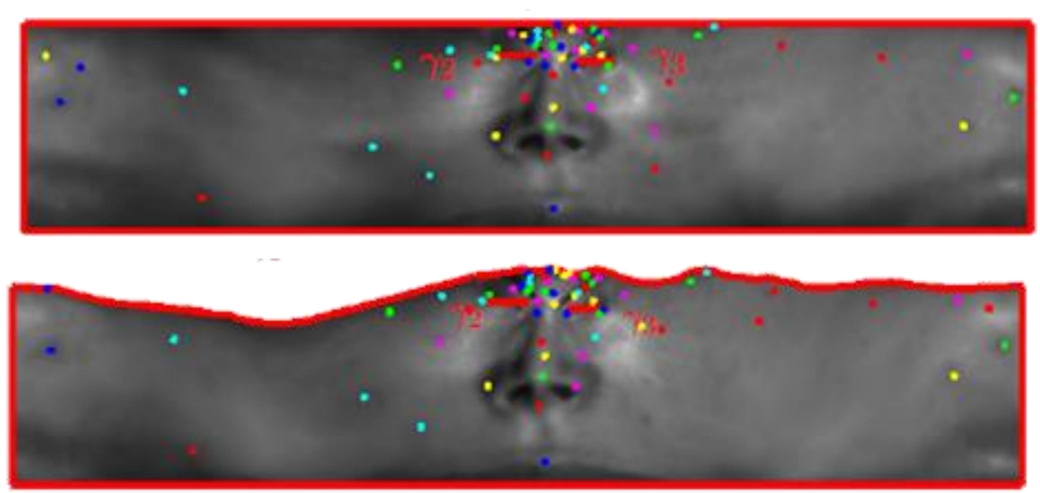
Measures of distortion:

- Differences in geodesic distances
- Differences in conformal factors (angles)
- etc.



Feature points

Branch and bound
Priority-driven search
etc.



Least squares conformal map
aligning corresponding feature points

[Zeng et al., 2008]

# Outline

Introduction

Some surface mapping algorithms

- Feature correspondence search
- High-dimensional embedding
- Möbius transformations
- Blended maps

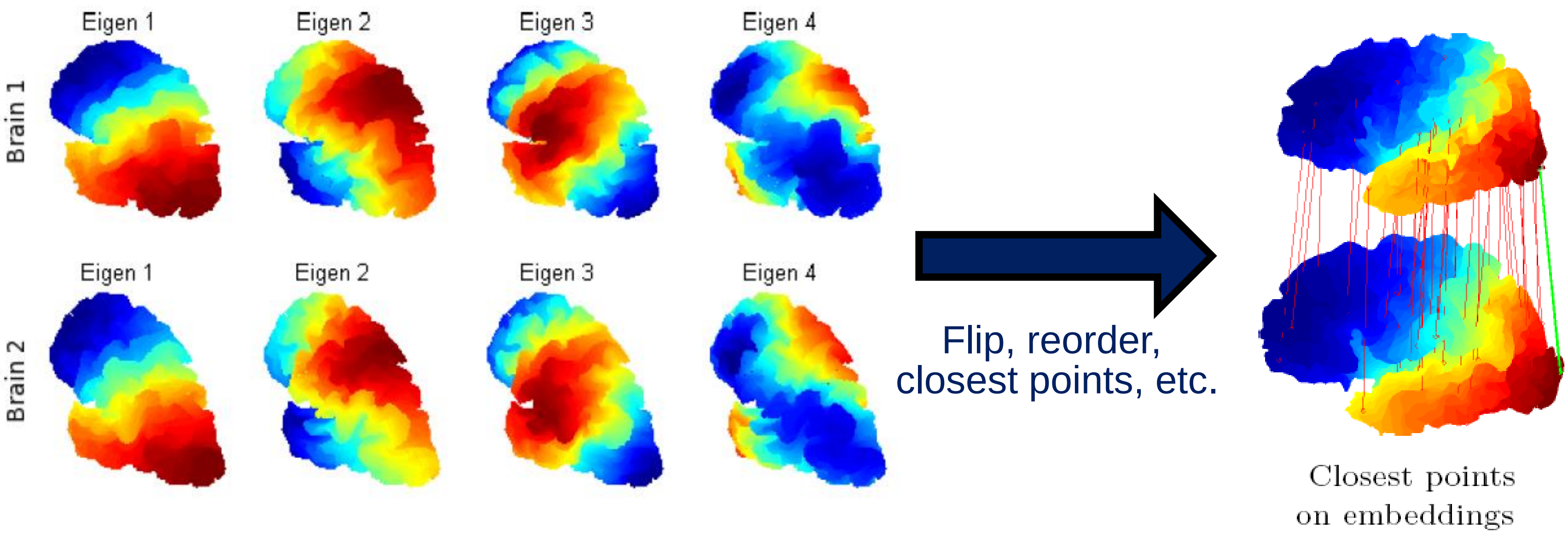
Example Application

Conclusion

Future work

# High-Dimensional Embedding

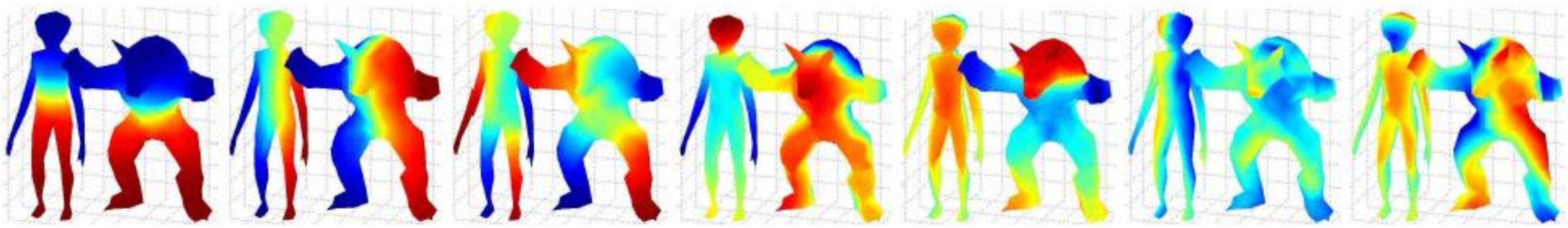
Find nearest neighbors after spectral embedding




Eigenfunctions of the Laplacian
[Lombaert et al. 2011]

# High-Dimensional Embedding

Find nearest neighbors after spectral embedding

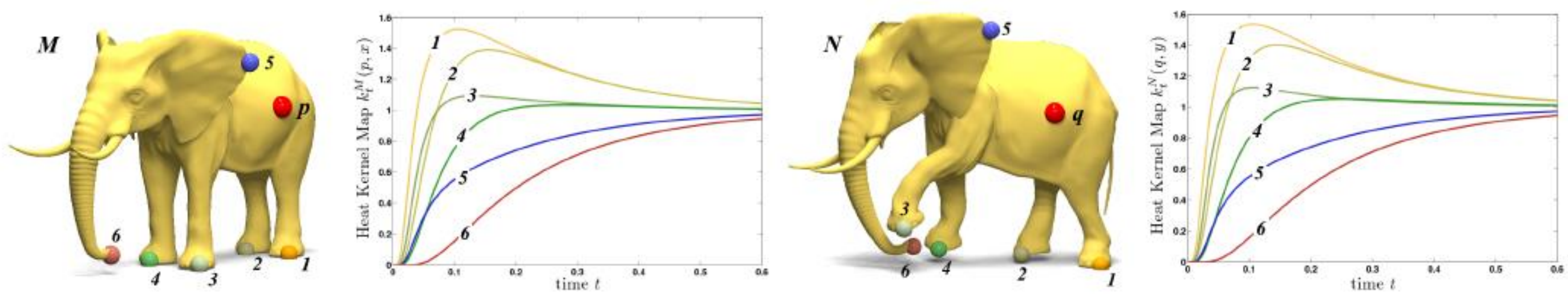


Eigenfunctions of the Laplacian

[Lombaert et al. 2011]

# High-Dimensional Embedding

Find nearest neighbors after heat kernel embedding implied by a single point correspondence




Heat Kernel Map

[Ovsjanikov et al. 2010]

# Outline

Introduction

Some surface mapping algorithms

- Feature correspondence search
- High-dimensional embedding
- Möbius transformations
- Blended maps

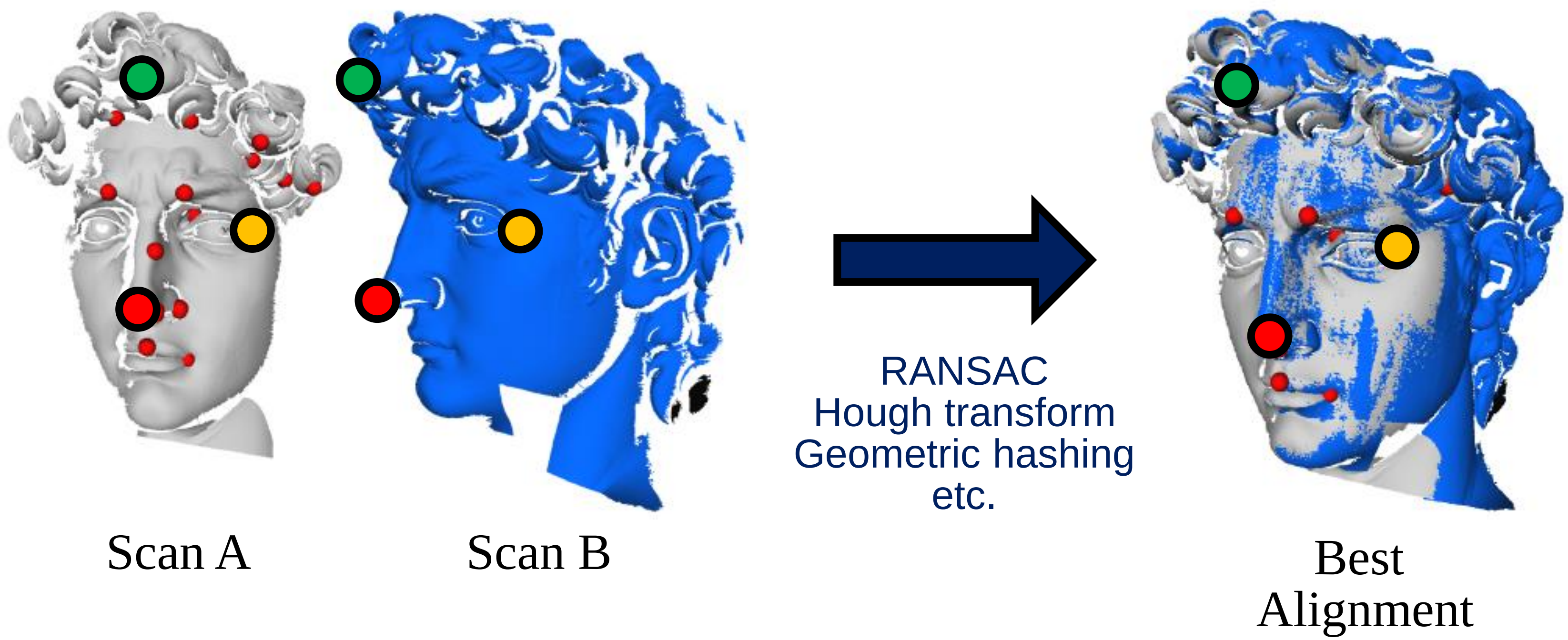
Example Application

Conclusion

Future work

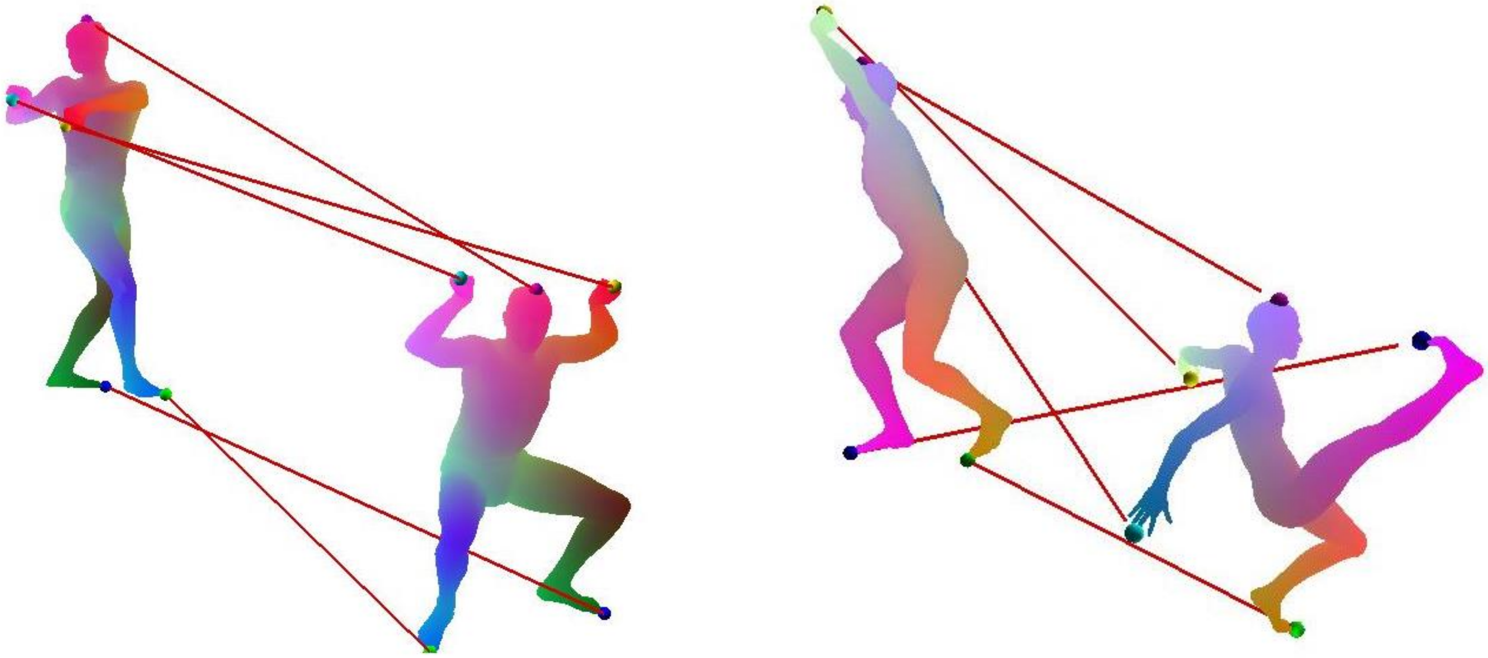
# Möbius Transformations

It would be nice to search a low-dimensional space of transformations to align non-rigid surfaces …



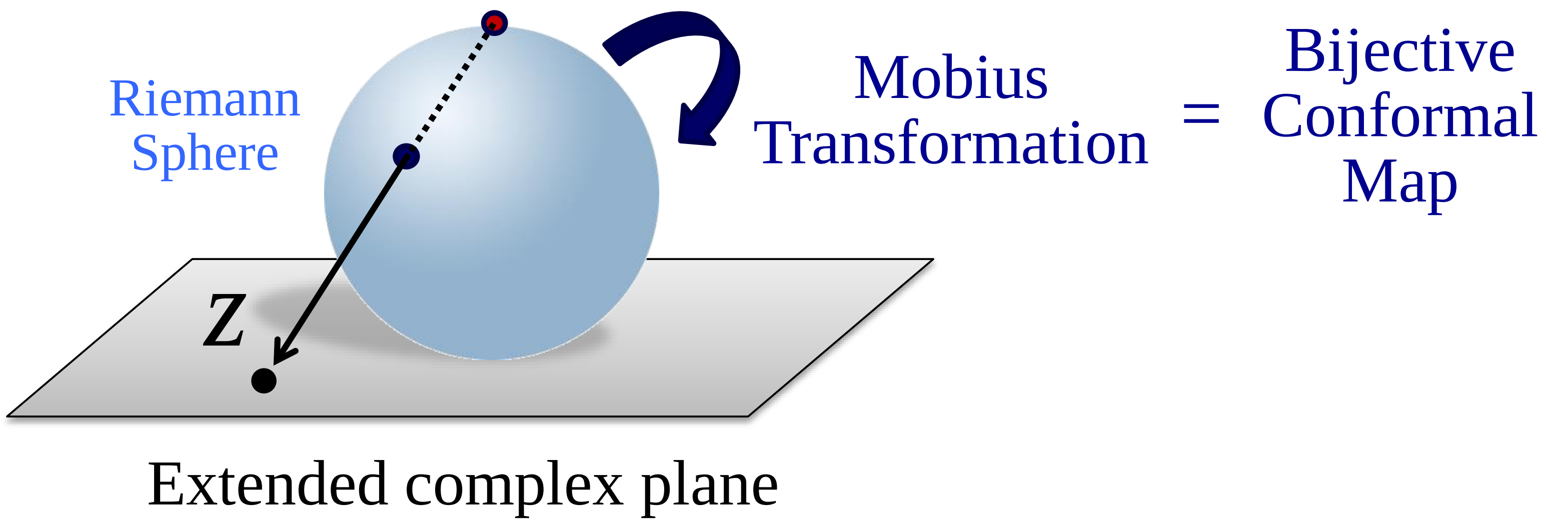
# Key Observation

The Möbius group provides a low-dimensional space to search efficiently for the “best” conformal map between genus zero surfaces



# Möbius Transformations I

Möbius transformations are a group of functions on the extended complex plane that represent bijective, conformal maps



# Möbius Transformations II

Möbius transformations are simple rational functions:

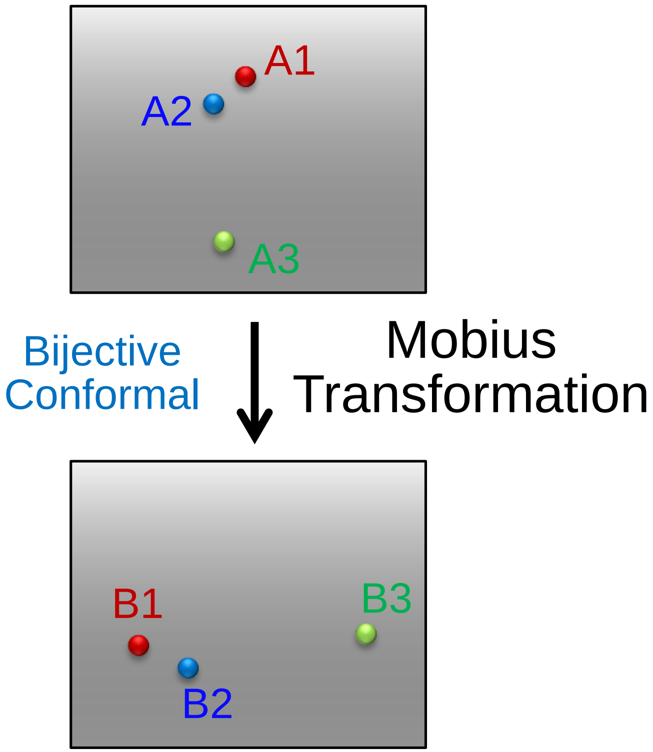
$$f(z) = \frac{az+b}{cz+d}, \quad ad - bc \neq 0, \; a,b,c,d \in C$$

They have only six degrees of freedom (they can be computed analytically from three point correspondences)


$$f(z_i) = y_i, i = 1,2,3$$

# Möbius Transformations III

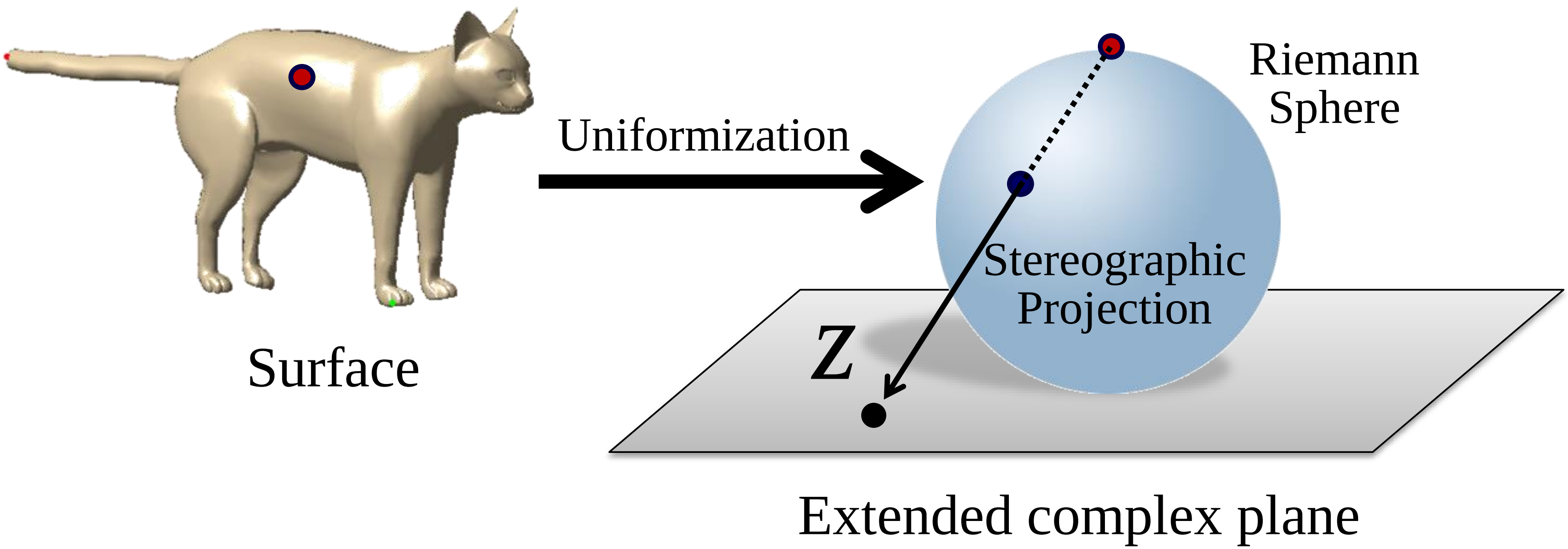
Therefore, any three point correspondences define a bijective, conformal map from the extended complex plane onto itself




Extended complex plane

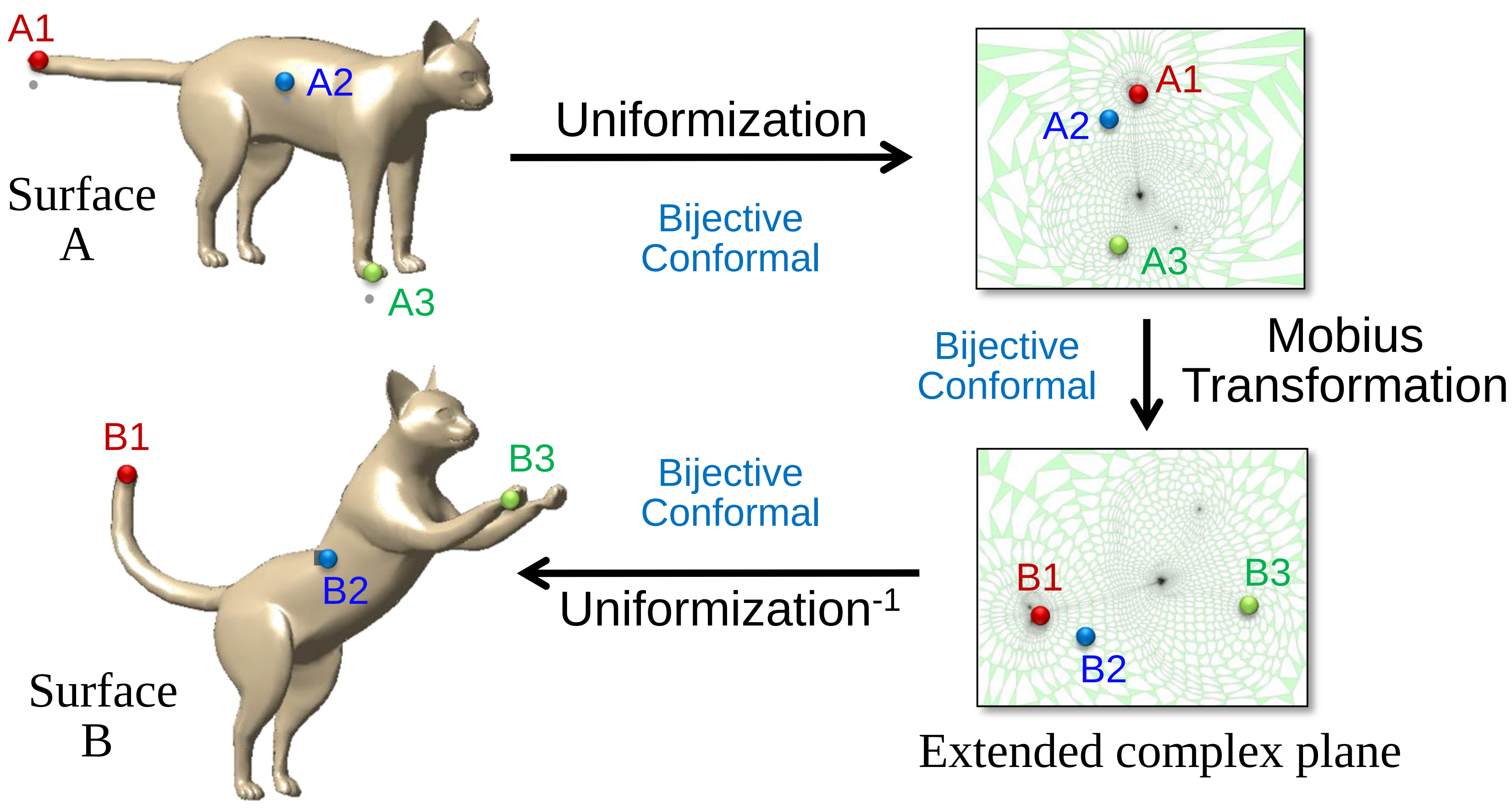
# Möbius Transformations IV

Since every genus zero surface can be mapped conformally onto the extended complex plane (Riemann sphere), …



# Möbius Transformations V

Any three point correspondences define a bijective, conformal map between genus zero surfaces



# Möbius Transformations VI

We can search for the "lowest distortion" bijective, conformal map between genus zero surfaces using algorithms that sample triplets of correspondences(e.g., RANSAC, Hough transform, etc.)

Polynomial-time algorithm
for non-rigid surface mapping

# Surface Mapping Algorithm

Example: RANSAC algorithm

- For i = 1 to ~$N^3$
  - Sample three points (A1,A2,A3) on surface A
  - Sample three points (B1,B2,B3) on surface B
  - Compute conformal map M: (A1,A2,A3)→(B1,B2,B3)
  - Remember M if distortion is smallest

# Surface Mapping Algorithm

Example: RANSAC algorithm

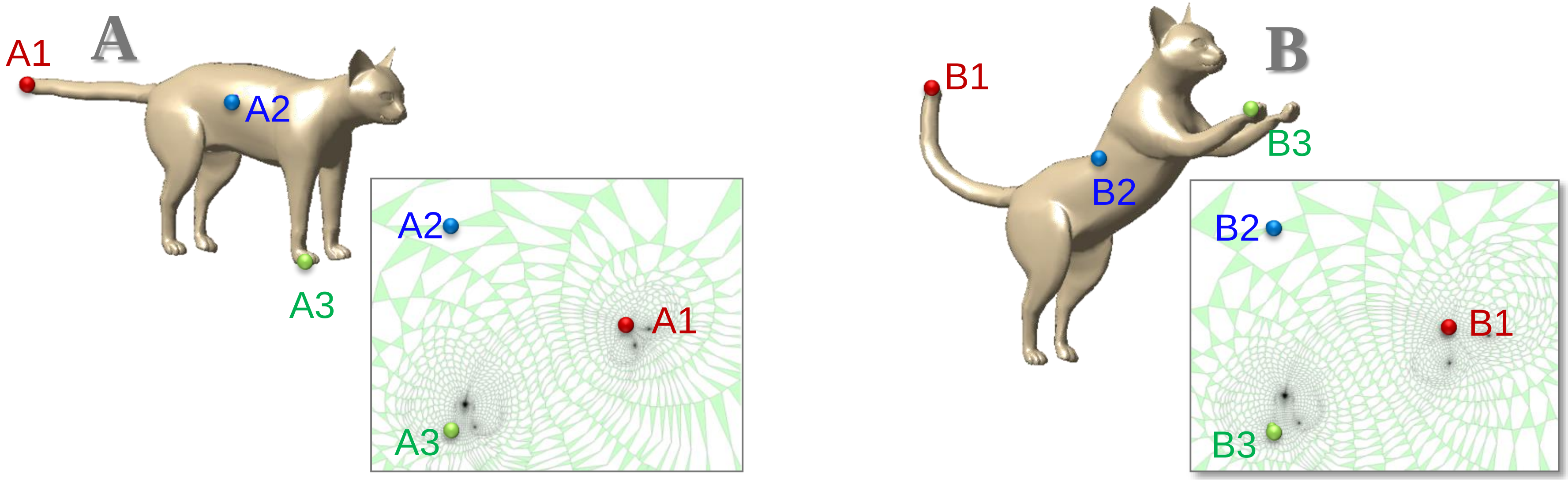
For i = 1 to $\sim N^3$

Sample three points (A1,A2,A3) on surface A

Sample three points (B1,B2,B3) on surface B

Compute conformal map M: (A1,A2,A3)→(B1,B2,B3)

Remember M if distortion is smallest




Measure distortion by relative change of area
(deviation from isometry)

# Surface Mapping Algorithm

Example: RANSAC algorithm

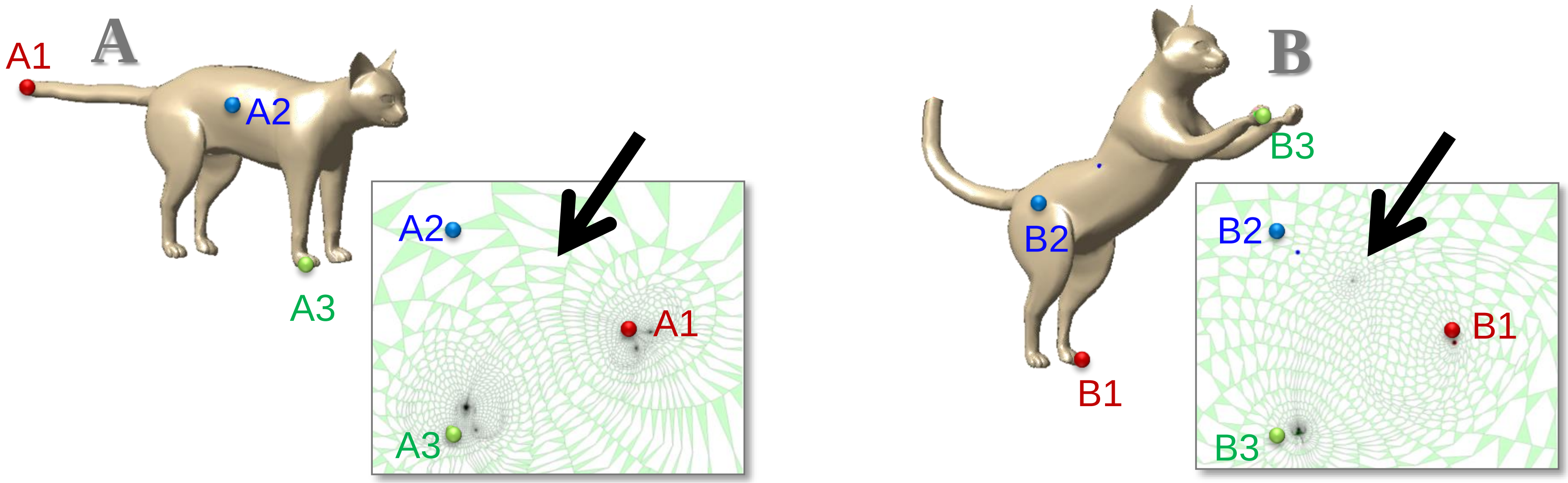
For i = 1 to ~$N^3$

Sample three points (A1,A2,A3) on surface A

Sample three points (B1,B2,B3) on surface B

Compute conformal map M: (A1,A2,A3)→(B1,B2,B3)

Remember M if distortion is smallest




Measure distortion by relative change of area
(deviation from isometry)

# Surface Mapping Algorithm

RANSAC algorithm properties:

- Non-rigid
- Bijective
- Smooth
- Shape preserving
- Automatic
- Efficient computation
- Provides metric
- Semantic alignment?

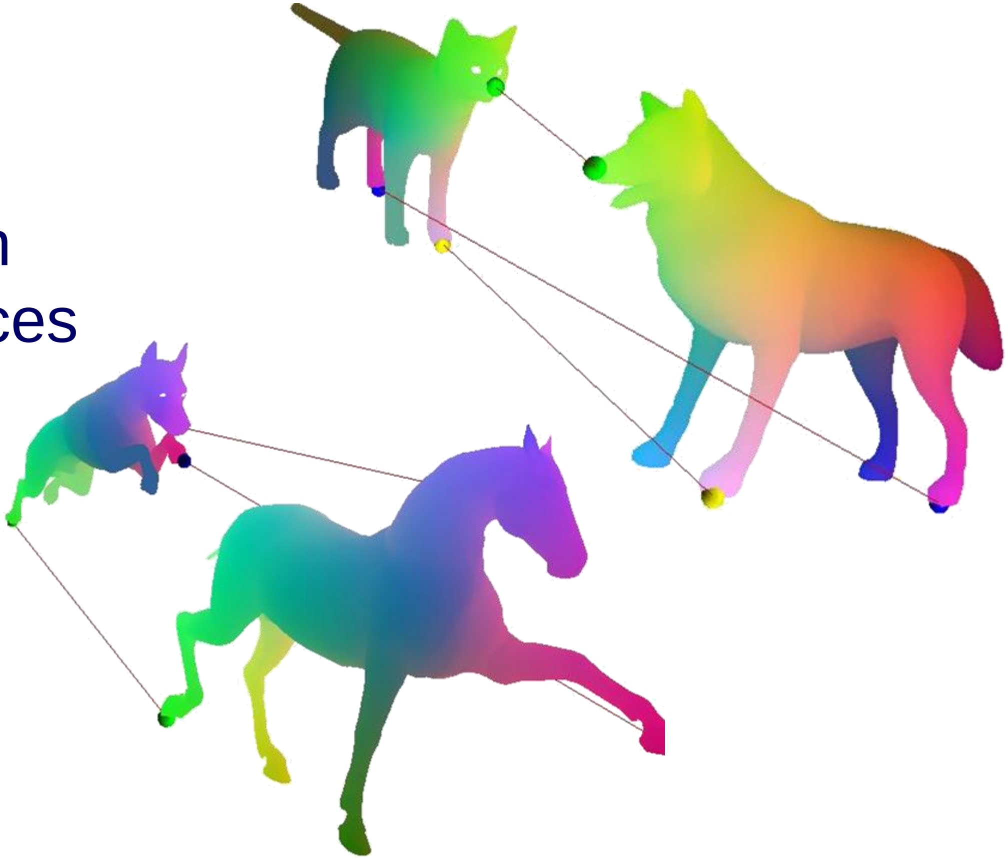
# Experimental Results

Data:

- 51 pairs of meshes representing animals from TOSCA and SHREC Watertight data sets

Methodology:

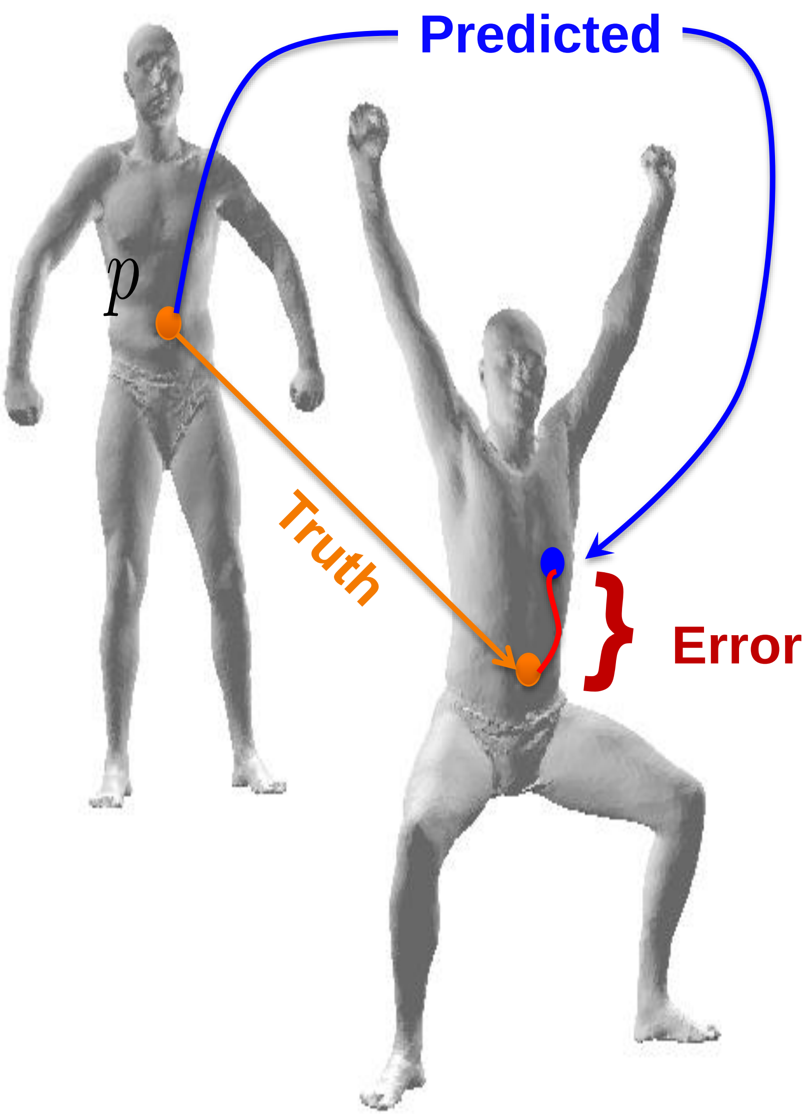
- Predict surface maps
- Compare to ground truth semantic correspondences



# Experimental Results

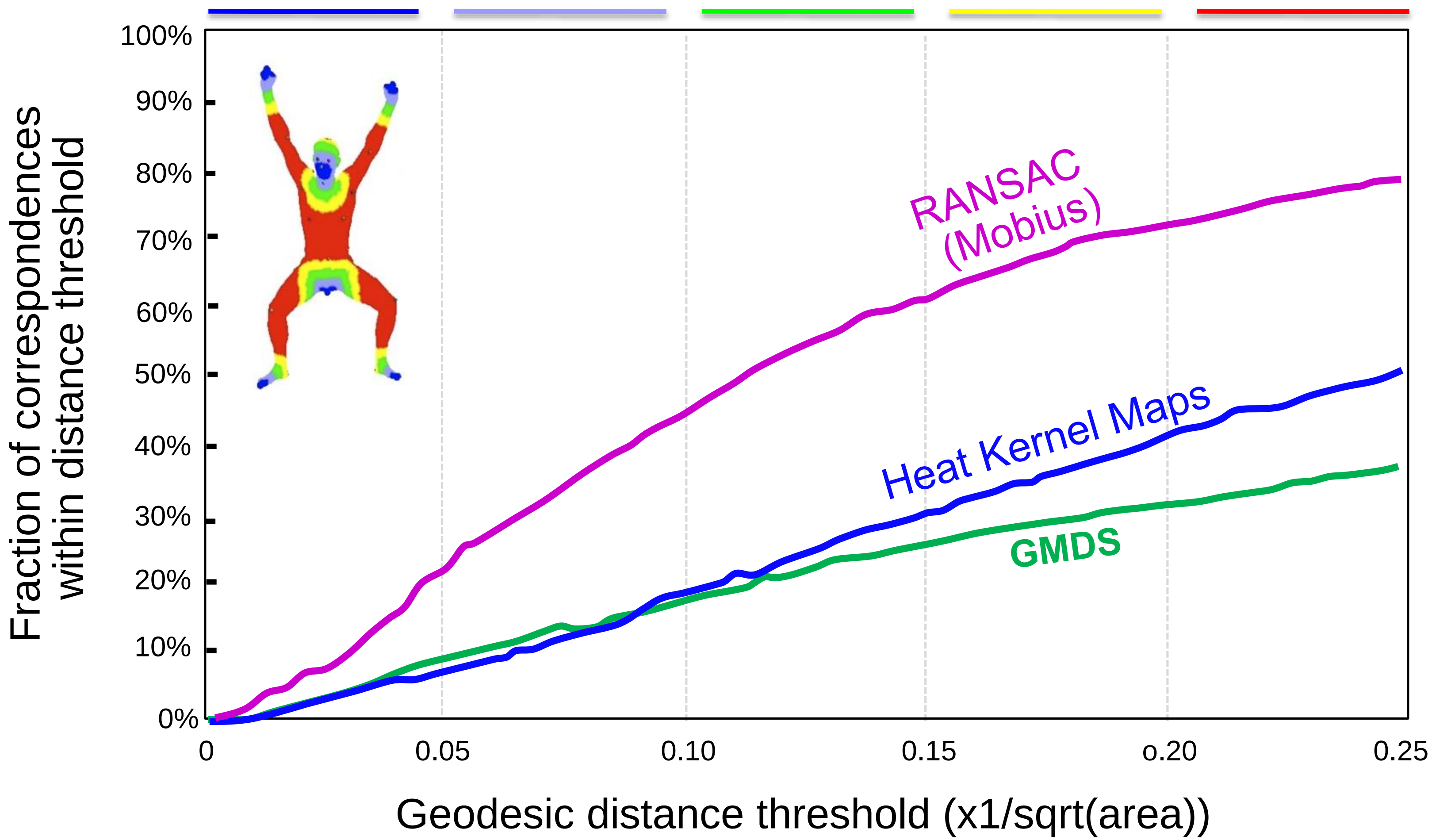
Evaluation:

1. For every point with a ground truth correspondence, measure geodesic distance between predicted correspondence and ground truth correspondence
2. Plot fraction of points within geodesic error threshold



# Experimental Results

Results:



# Outline

Introduction

Some surface mapping algorithms

- Feature correspondence search
- High-dimensional embedding
- Möbius transformations
- Blended maps

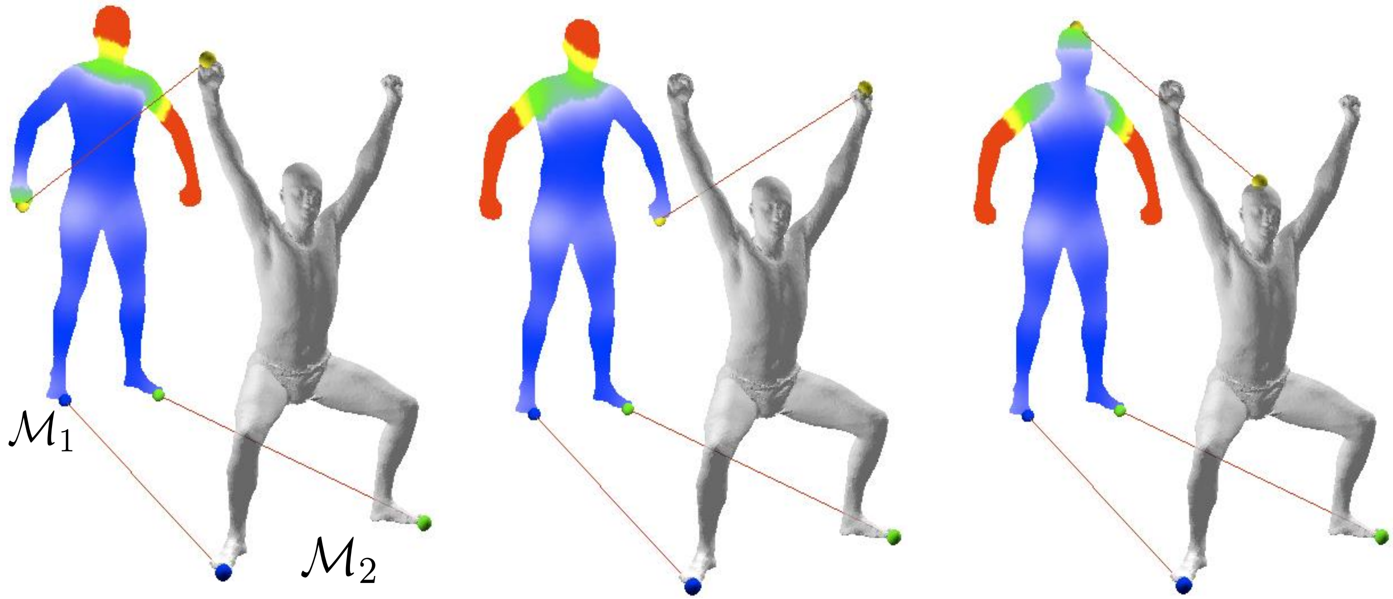
Example Application

Conclusion

Future work

# Blended Maps

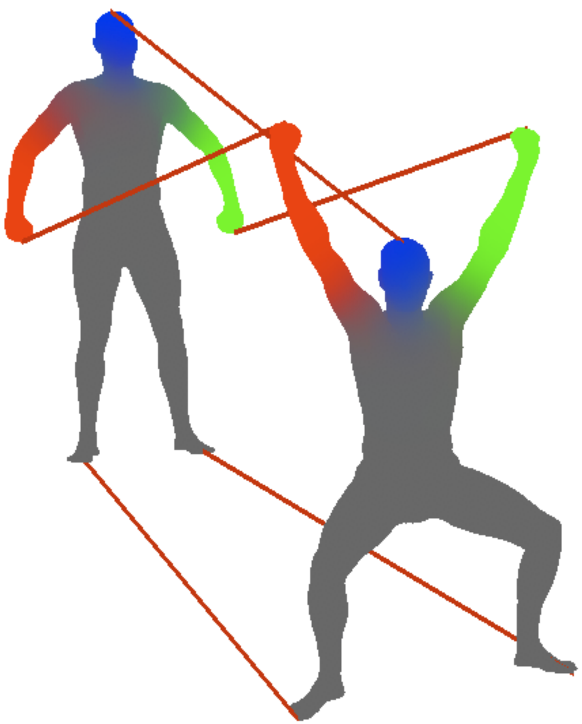
For significantly different surfaces, no single conformal map provides low distortion everywhere



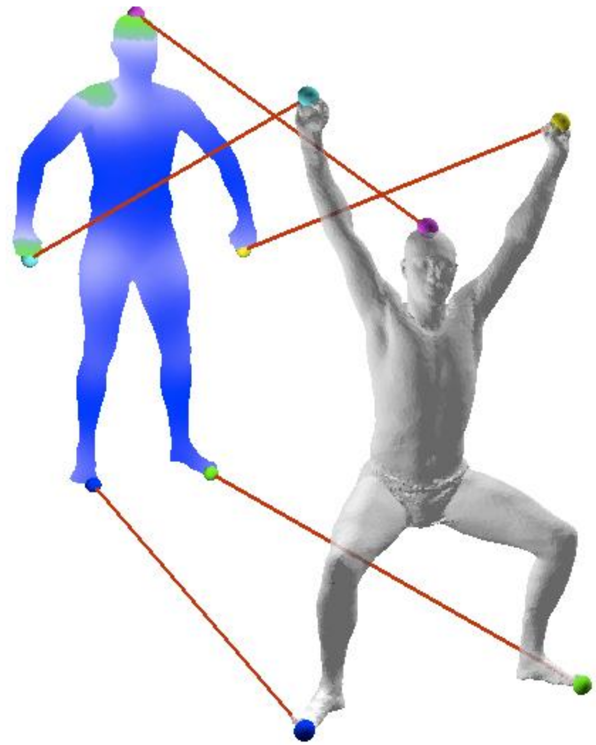

Distortion of $m_1$ Distortion of $m_2$ Distortion of $m_3$

# Blended Maps

Idea: blend conformal maps with smooth weights

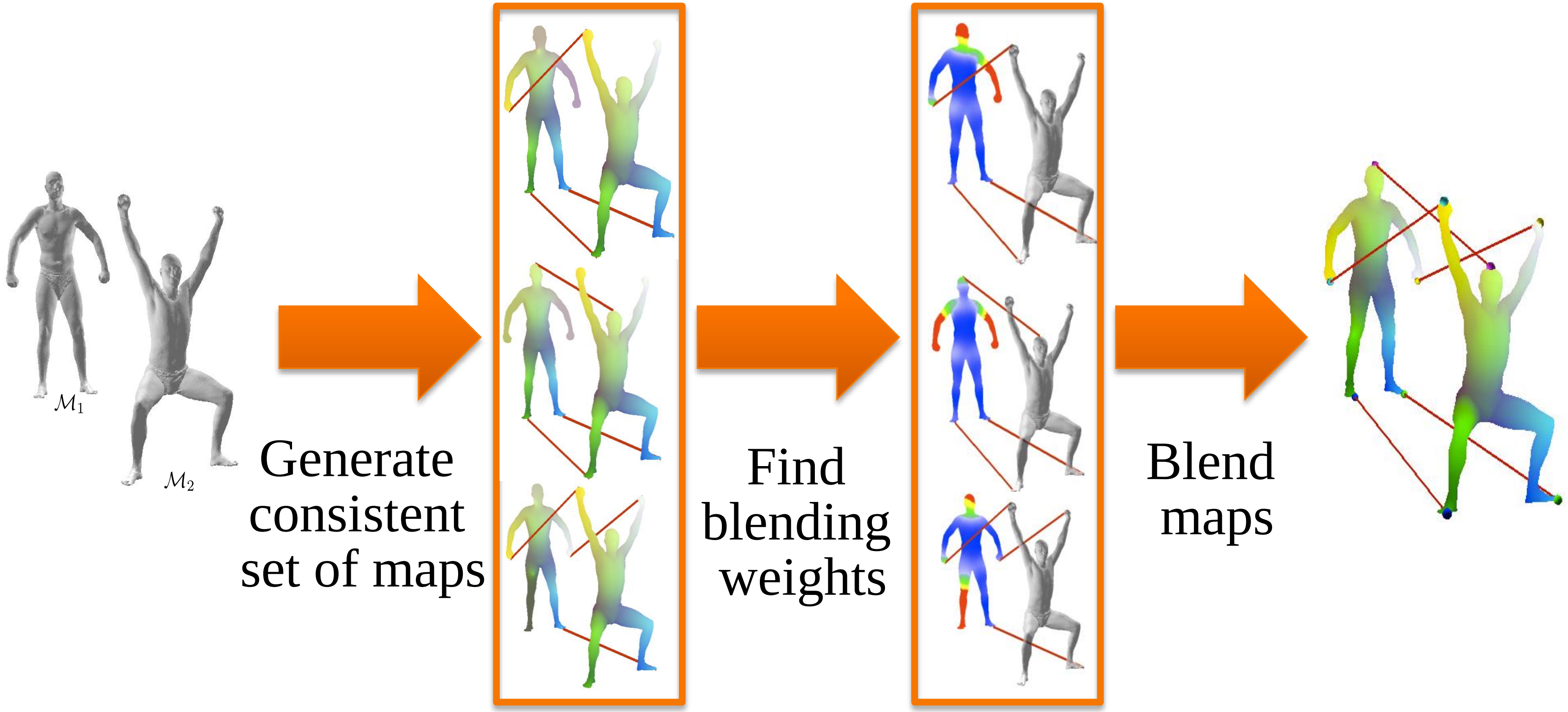


Blending Weights
for $m_1$, $m_2$, and $m_3$



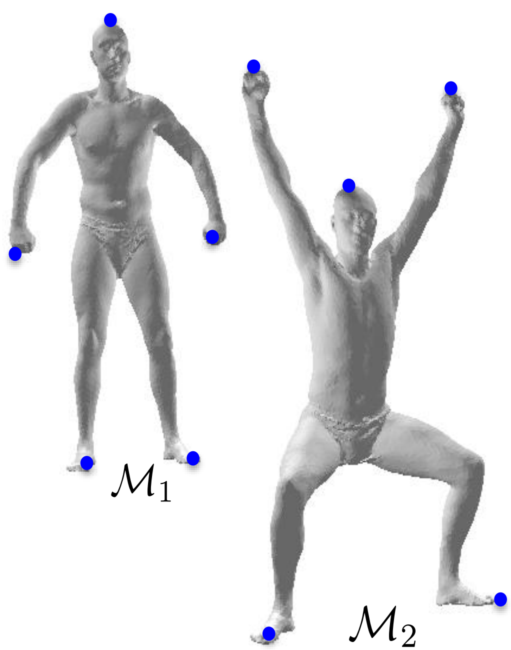
Distortion of the
Blended Map

# Computing Blended Maps

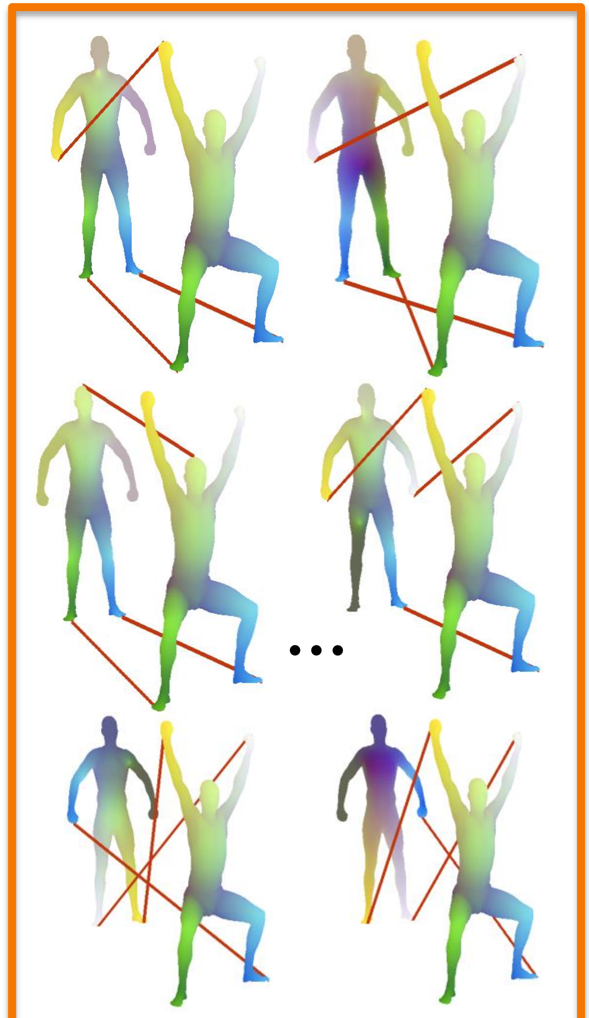


# Computing Blended Maps

1. Generate candidate maps by enumerating triplets of feature correspondences

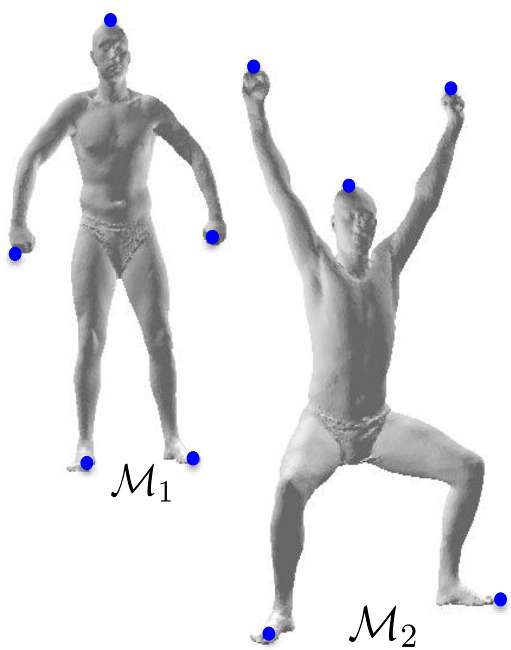



Set of candidate maps

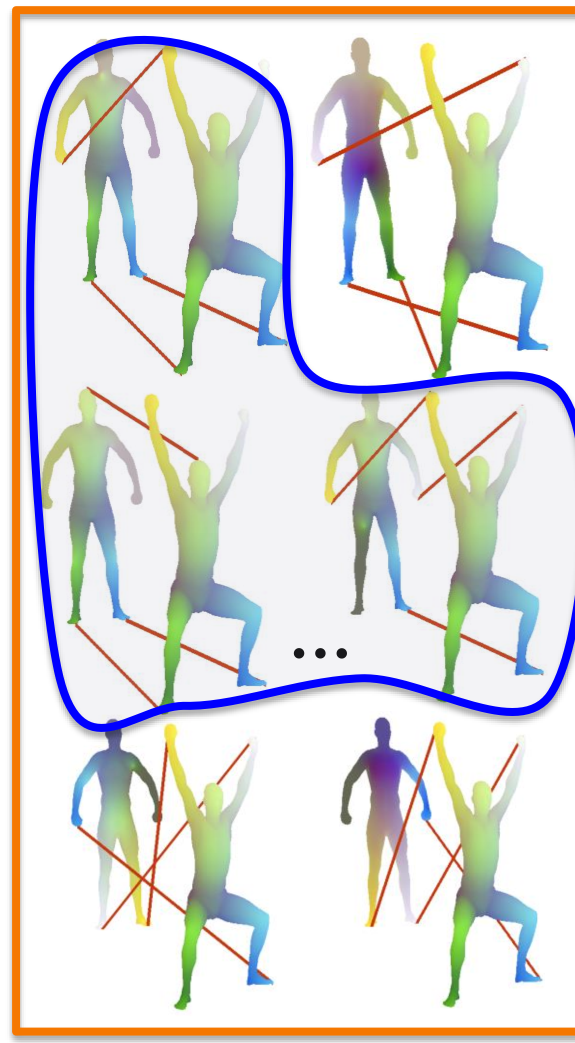


# Computing Blended Maps

2. Select consistent set of low-distortion candidate maps



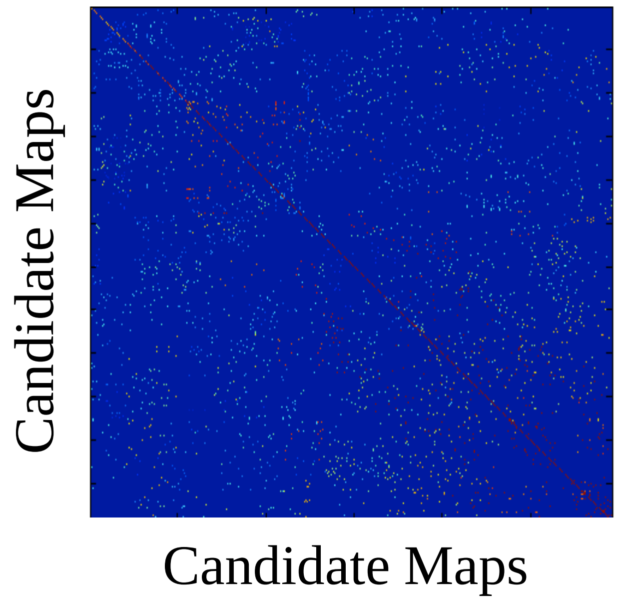

Set of candidate maps

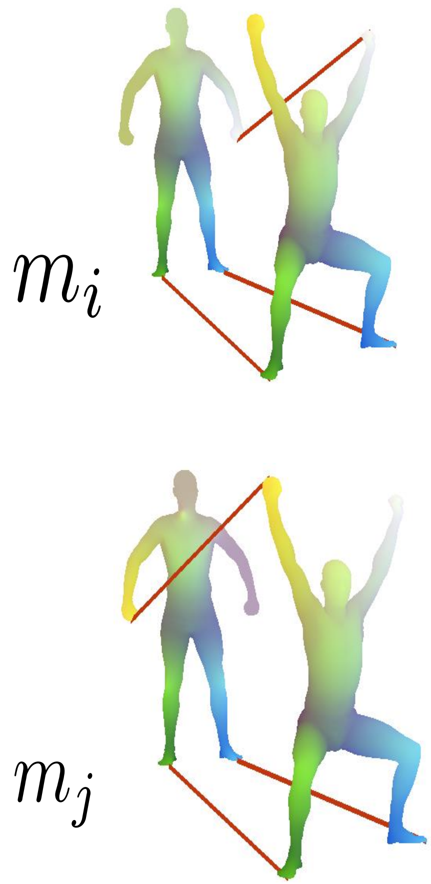


# Computing Blended Maps

2a. Define a matrix **B** where every entry (i,j) indicates the distortion of $m_i$ and $m_j$ and their pairwise similarity $S_{i,j}$

$$\mathbf{B}_{i,j} = \int_{M_1} c_i(p)c_j(p)S_{i,j}(p)dA(p)$$

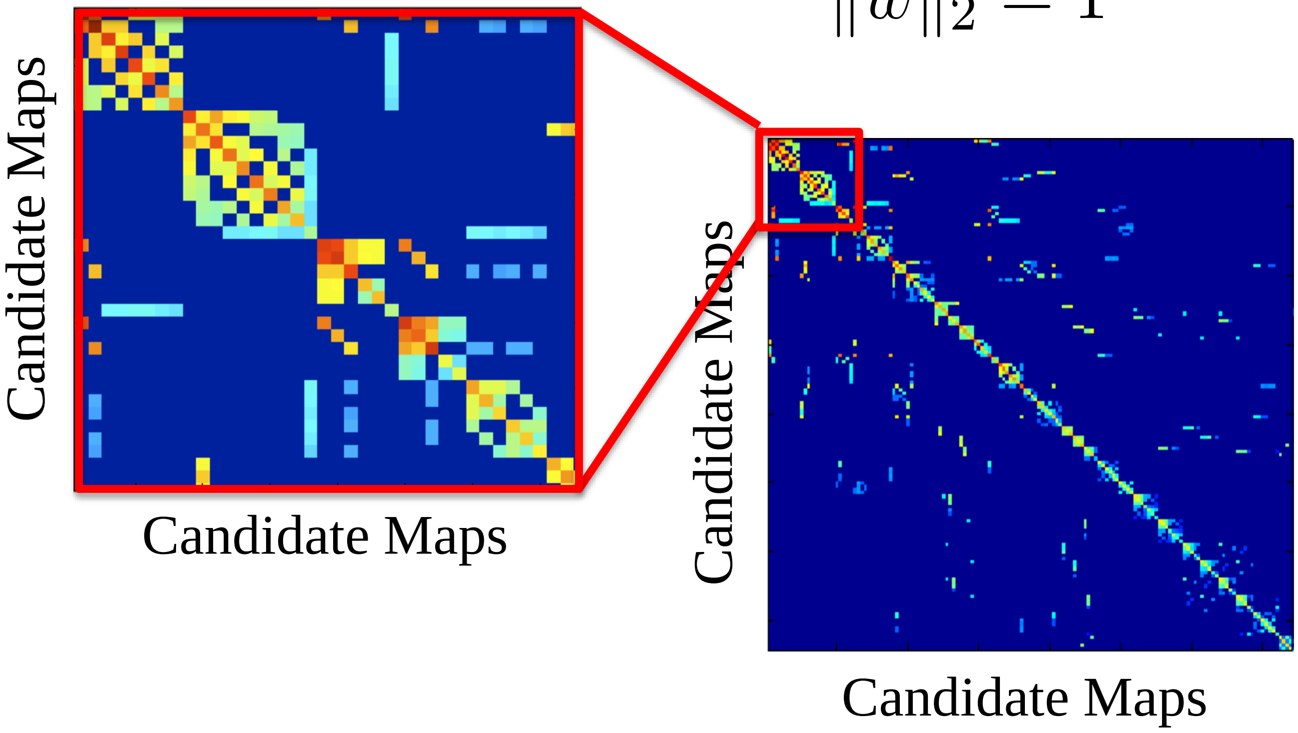


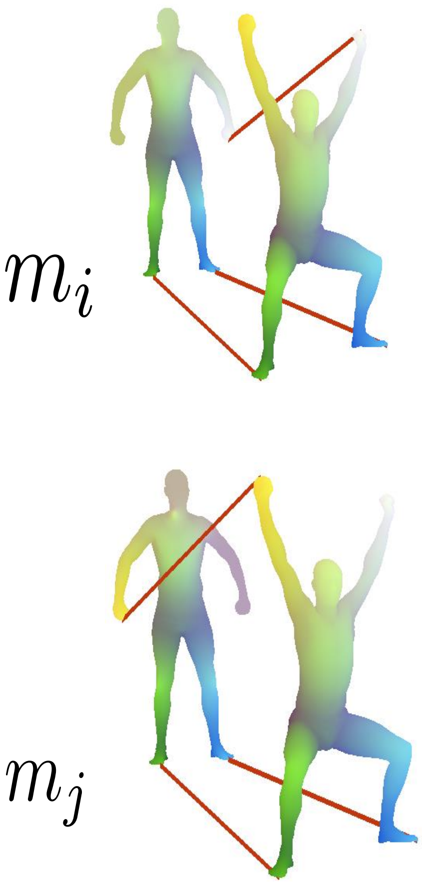



# Computing Blended Maps

2b. Find block of consistent, low-distortion maps using top eigenvector(s) of $\mathbf{B}$

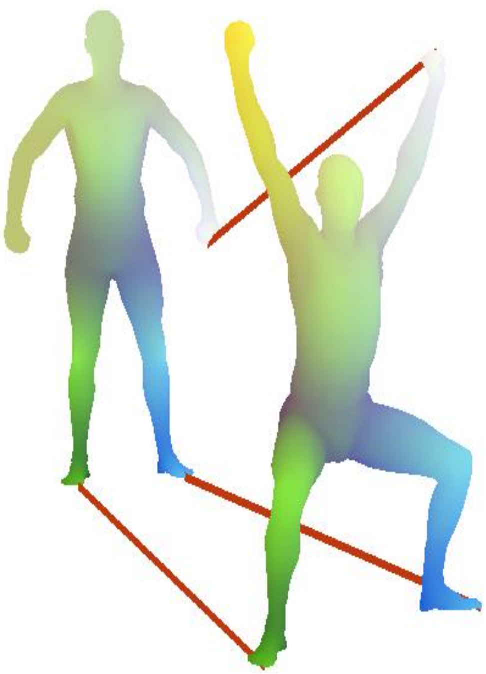
$$E_{\mathcal{M}}(\vec{w}) = \vec{w}^T \mathbf{B} \vec{w}$$
$$\|\vec{w}\|_2 = 1$$



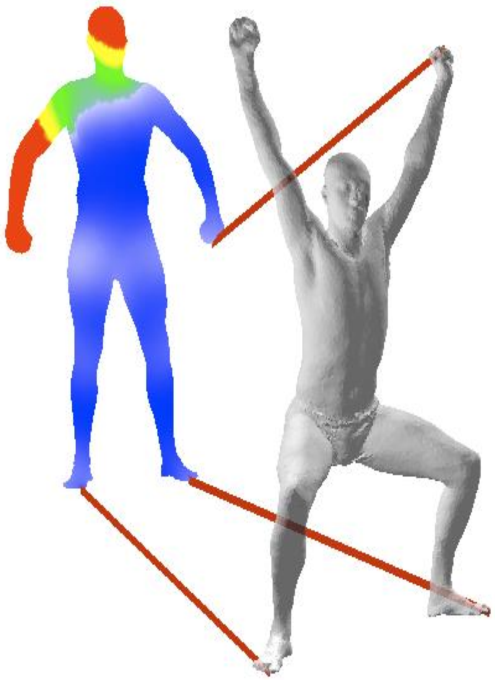



# Computing Blended Maps

3. Compute blending weight $c_i(p)$ for every map $m_i$ at every point $p$ based on distortion of $m_i$ at $p$



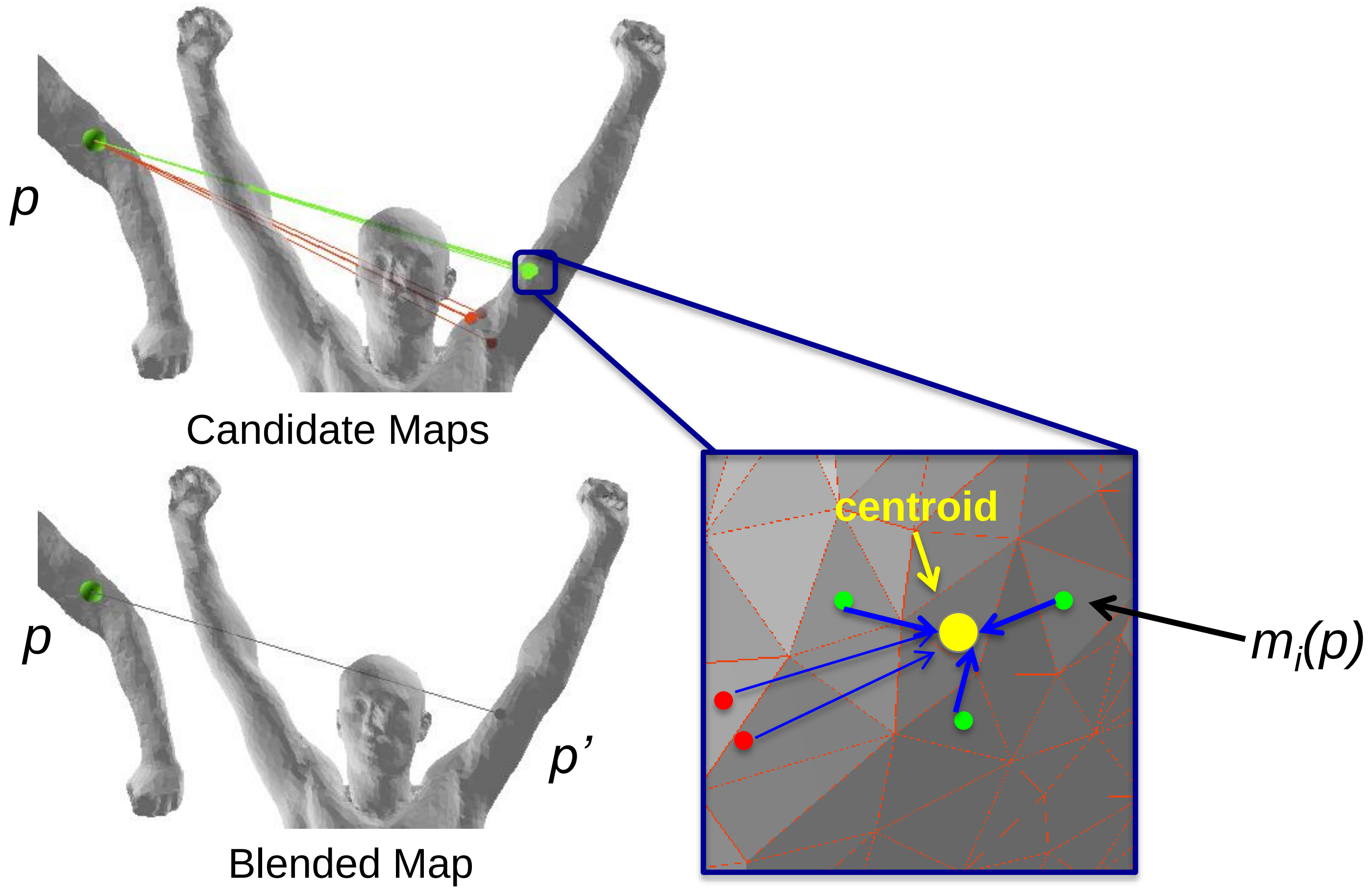
Candidate Map



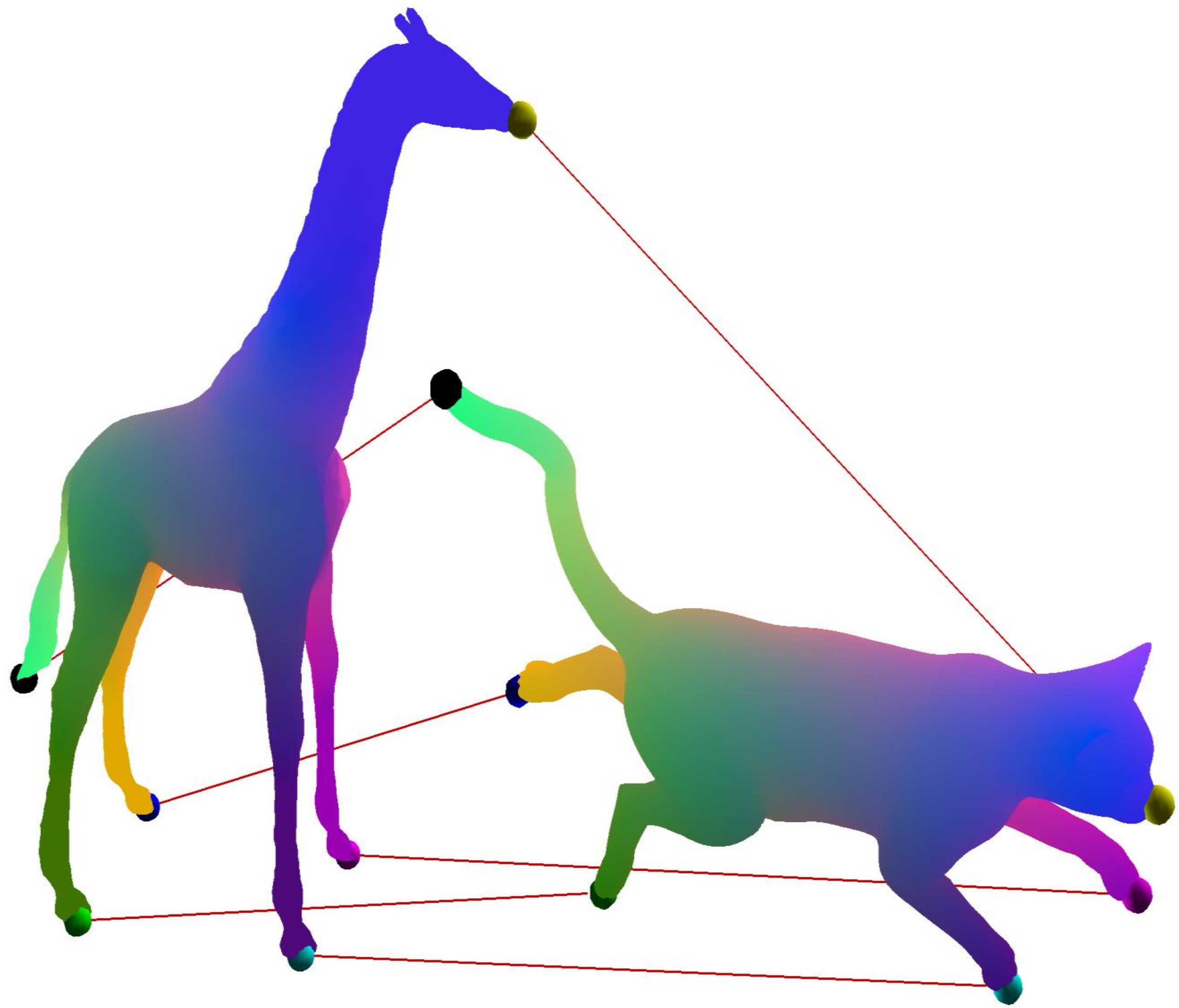
Blending Weight
$c_i(p)$

# Computing Blended Maps

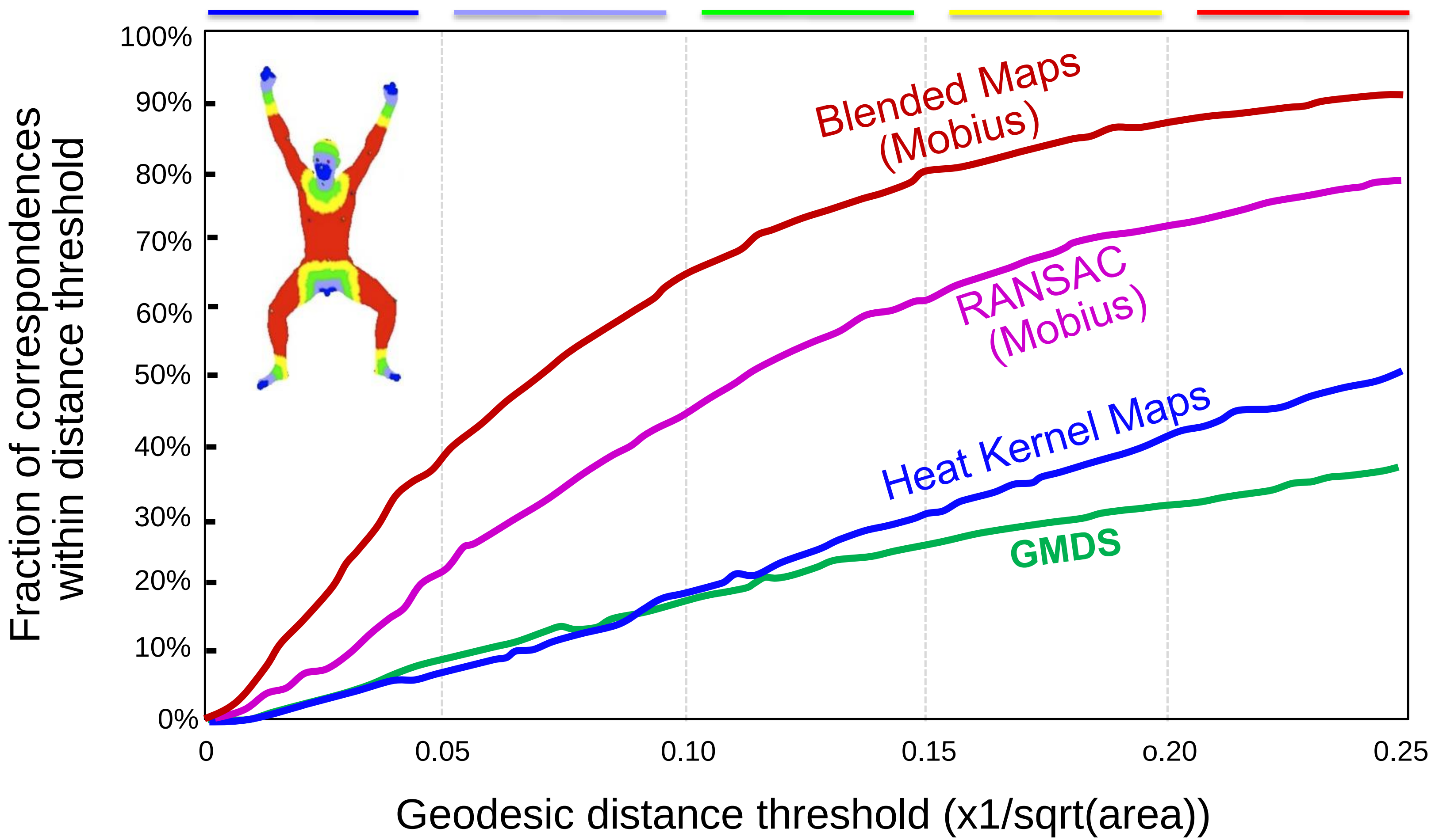
4. Define image *p'* of every point *p* as the weighted geodesic centroid of $m_i(p)$



# Computing Blended Maps



# Experimental Results



# Outline

Introduction

Some surface mapping algorithms

- Feature correspondence search
- High-dimensional embedding
- Möbius transformations
- Blended maps

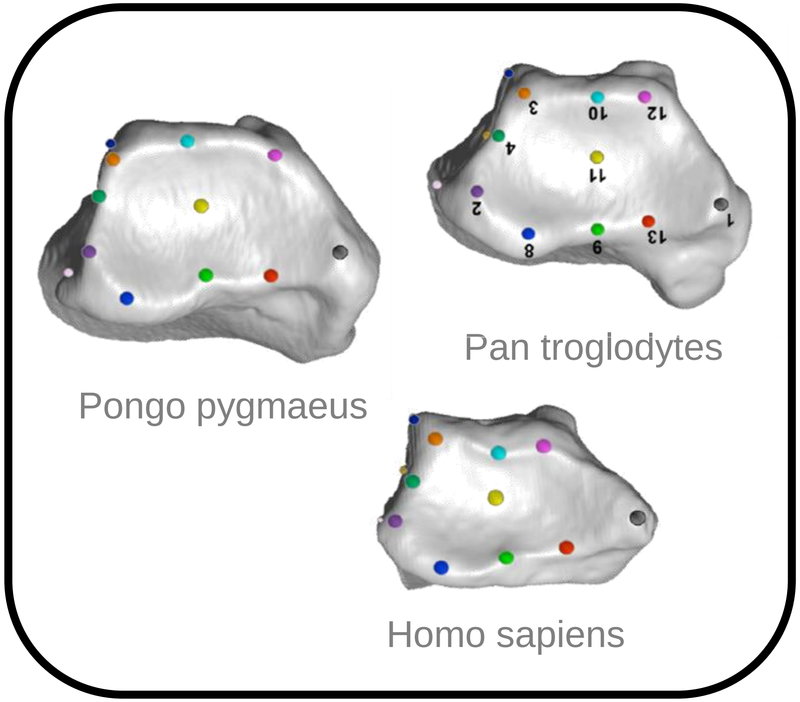
Example Application

Conclusion

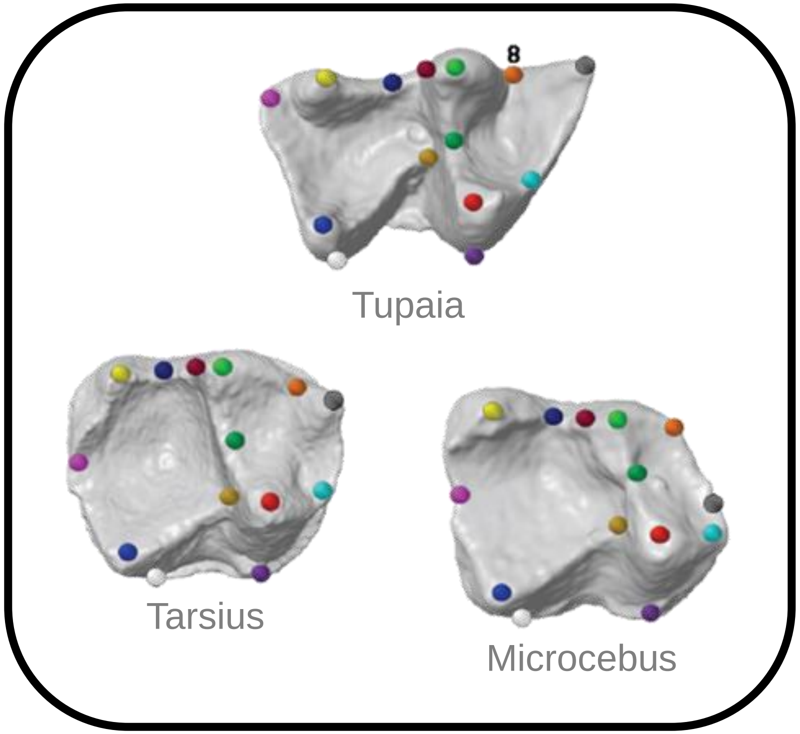
Future work

# Application

Automatically quantify the geometric similarity of anatomical surfaces




Distal Radius



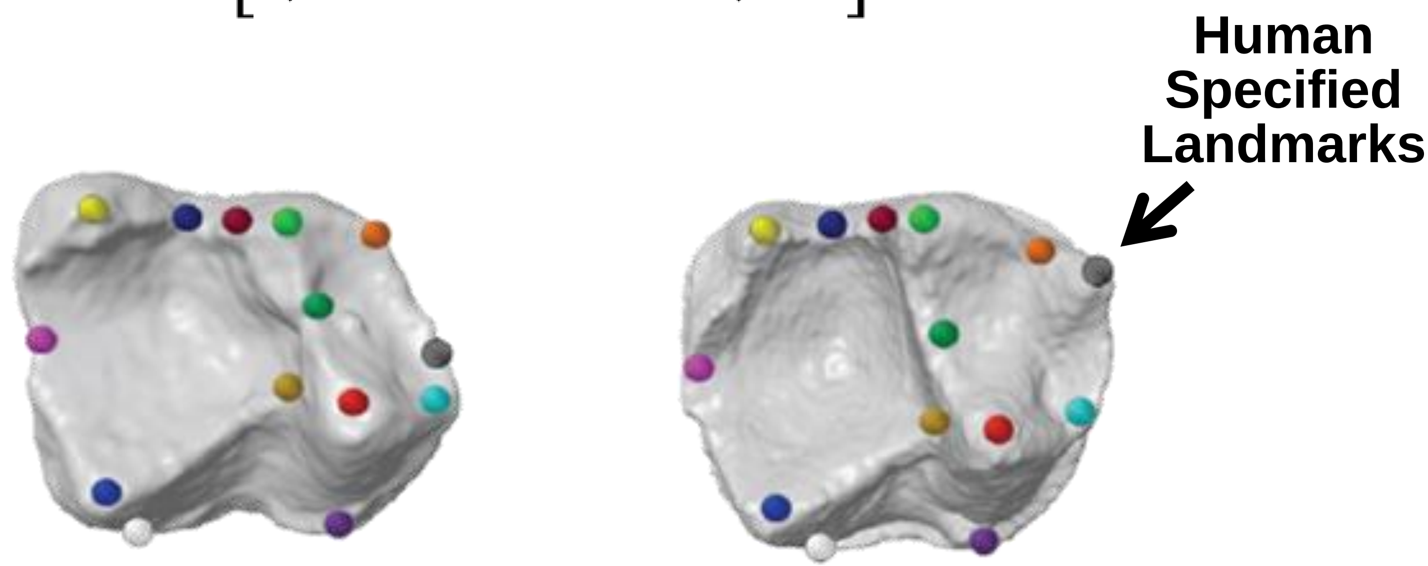

Mandibular Molar

[Boyer, Lipman, St. Clair, Puente, Patel, Funkhouser, Jernvall, and Daubechies, 2011]

# Application

Traditional Procrustes distance:

$$d(X,Y) = min_R \left[ \left( \sum_{i=1}^{N} \|R(X_i) - Y_i\|^2 \right)^{1/2} \right]$$

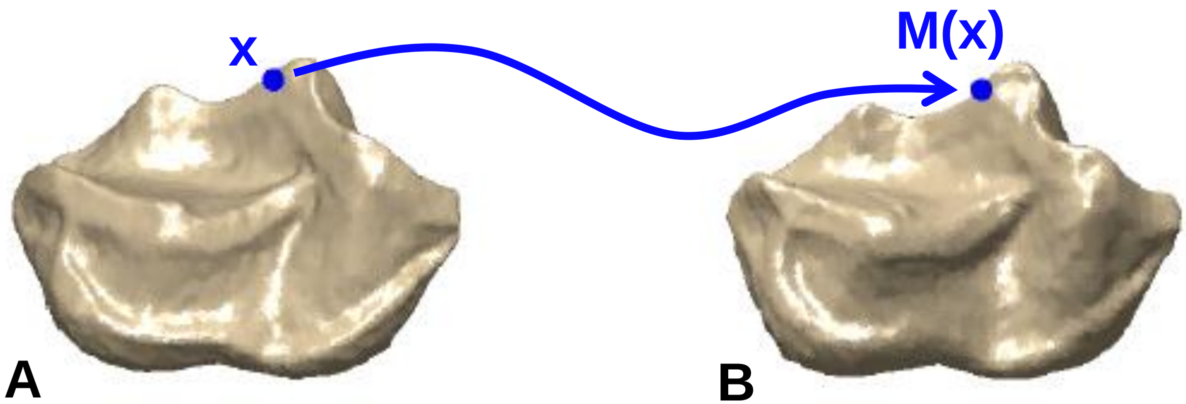



$X = \{ X_i \}$ $Y = \{ Y_i \}$

# Application

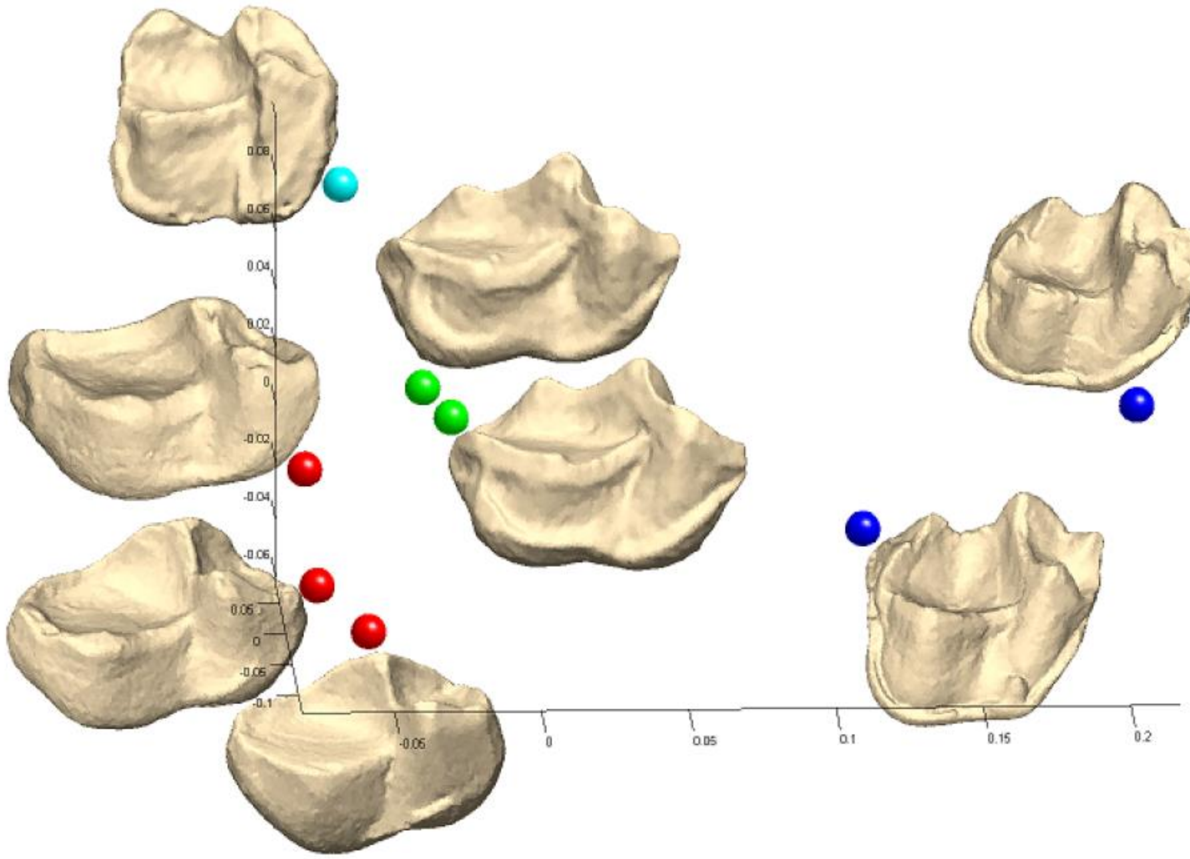
New continuous Procrustes distance:

$$d(A,B) = min_{R,M}\left[\left(\int_A \|R(x) - M(x)\|^2\, dx\right)^{1/2}\right]$$



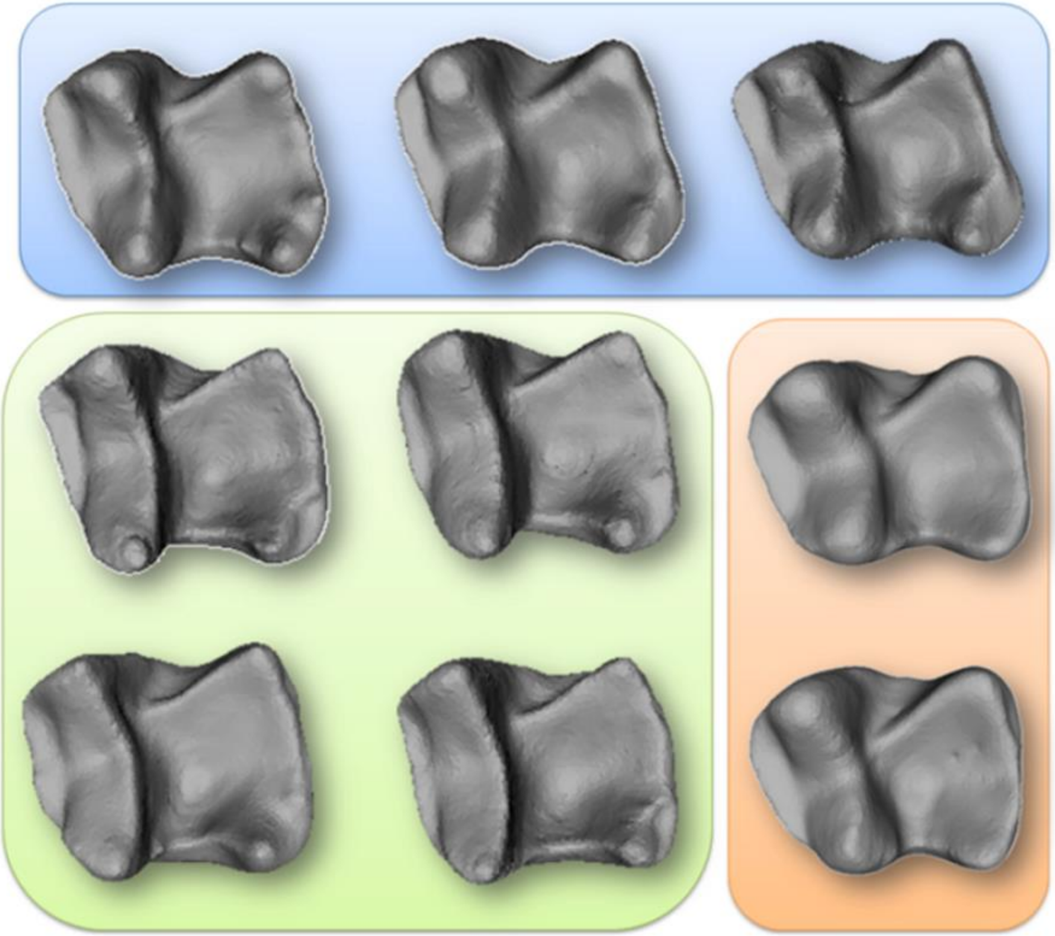
# Application

Embedding based on new distance



# Application

Clustering based on new distance



Species Groups of Galaga Genus

# Application

## Classification based on nearest-neighbors

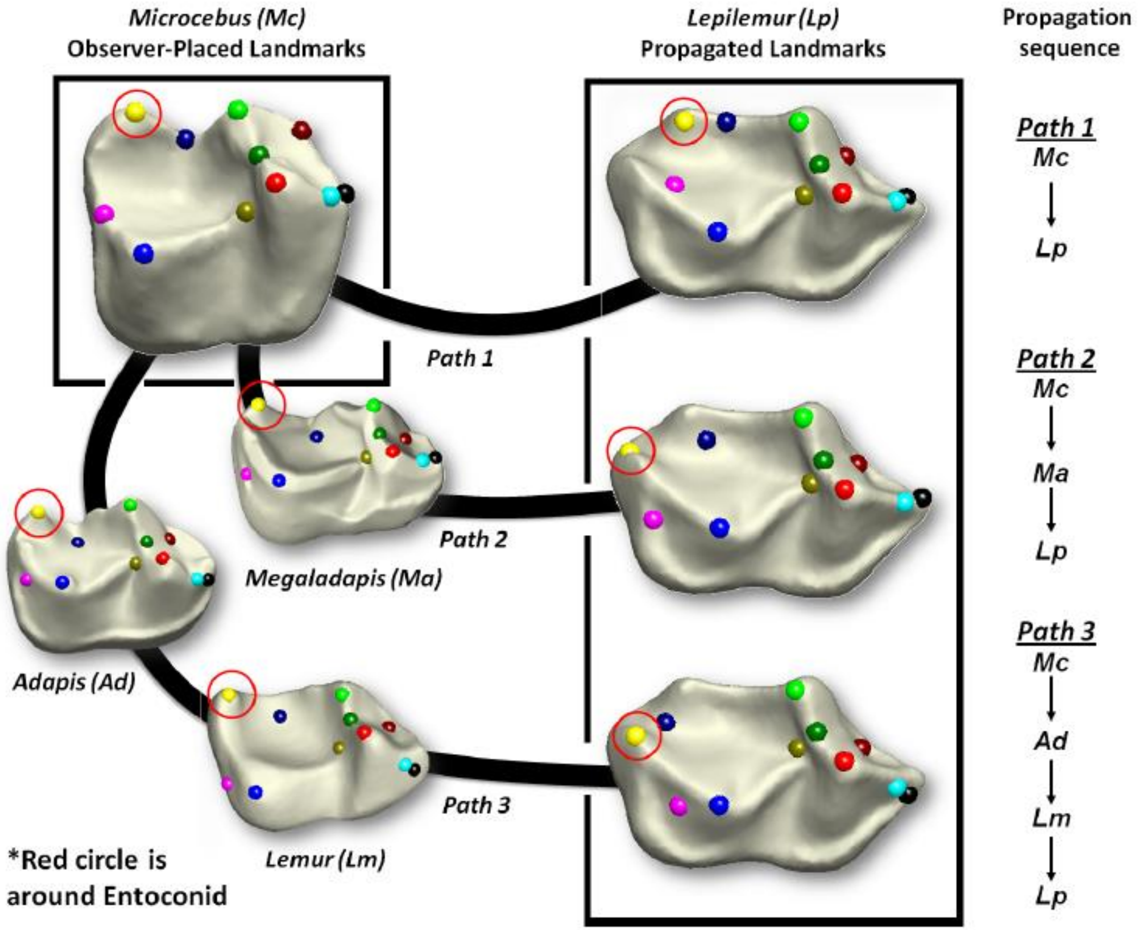
| **Mandibular Molar** | # Groups | # Objects | New Distance | Human Landmarks |
|---|---|---|---|---|
| Genus | 24 | 99 | 90.9% | 91.9% |
| Family | 17 | 106 | 92.5% | 94.3% |
| Order | 5 | 116 | 94.8% | 95.7% |

| **First Metatarsal** | # Groups | # Objects | New Distance | Human1 Landmarks | Human2 Landmarks |
|---|---|---|---|---|---|
| Genus | 13 | 59 | 79.9% | 76.3% | 88.1% |
| Family | 9 | 61 | 91.8% | 83.6% | 93.4% |
| Superfamily | 2 | 61 | 100% | 100% | 100% |

| **Distal Radius** | # Groups | # Objects | New Distance | Human Landmarks |
|---|---|---|---|---|
| Genus | 4 | 45 | 84.4% | 77.7% |

# Application

## Propagating correspondences



# Acknowledgments

Test data

- Giorgi et al. (SHREC Watertight), Anguelov et al. (SCAPE), Bronstein et al. (TOSCA)

Test code:

- Ovsjanikov et al. (HKM), Bronstein et al. (GMDS)

Application

- Boyer, St. Clair, Patel, Jernvall, Puente, Daubechies

Funding:

- NSERC, NSF, AFOSR, Intel, Adobe, Google

**Thank You!**