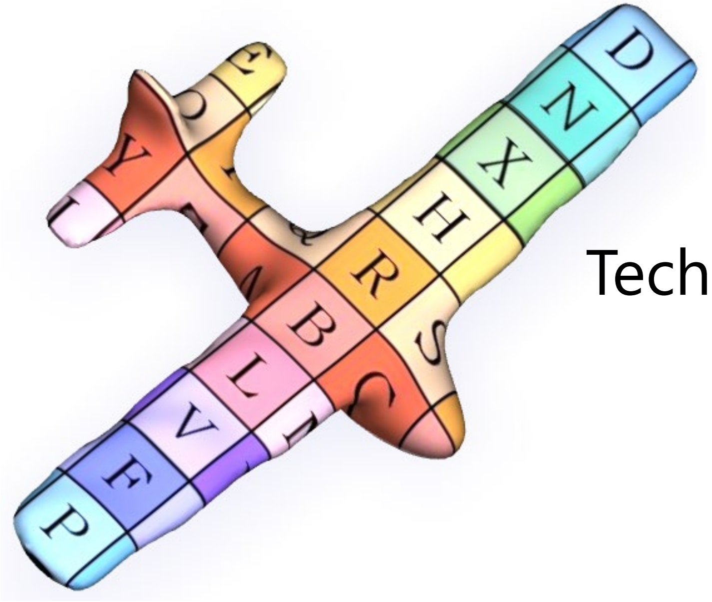
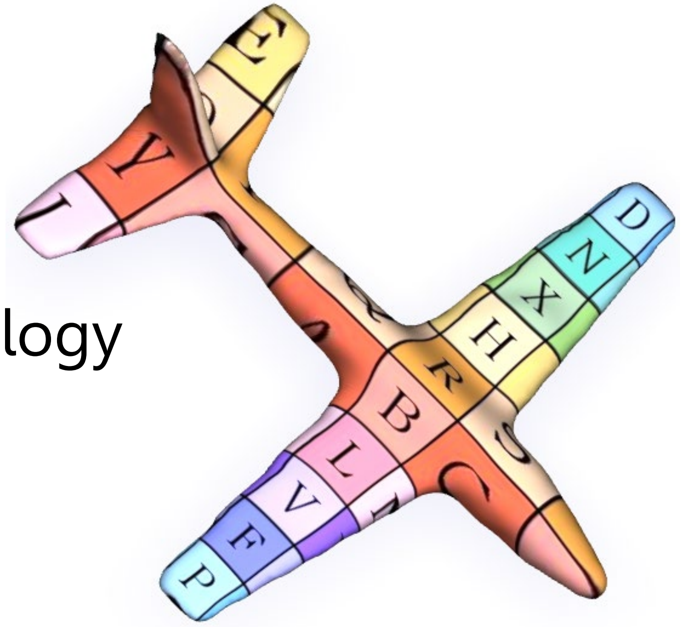
Maps between Shapes
(shape correspondence / matching)



Danielle Ezuz

Technion – Israel Institute of Technology

SGP Graduate School 2019

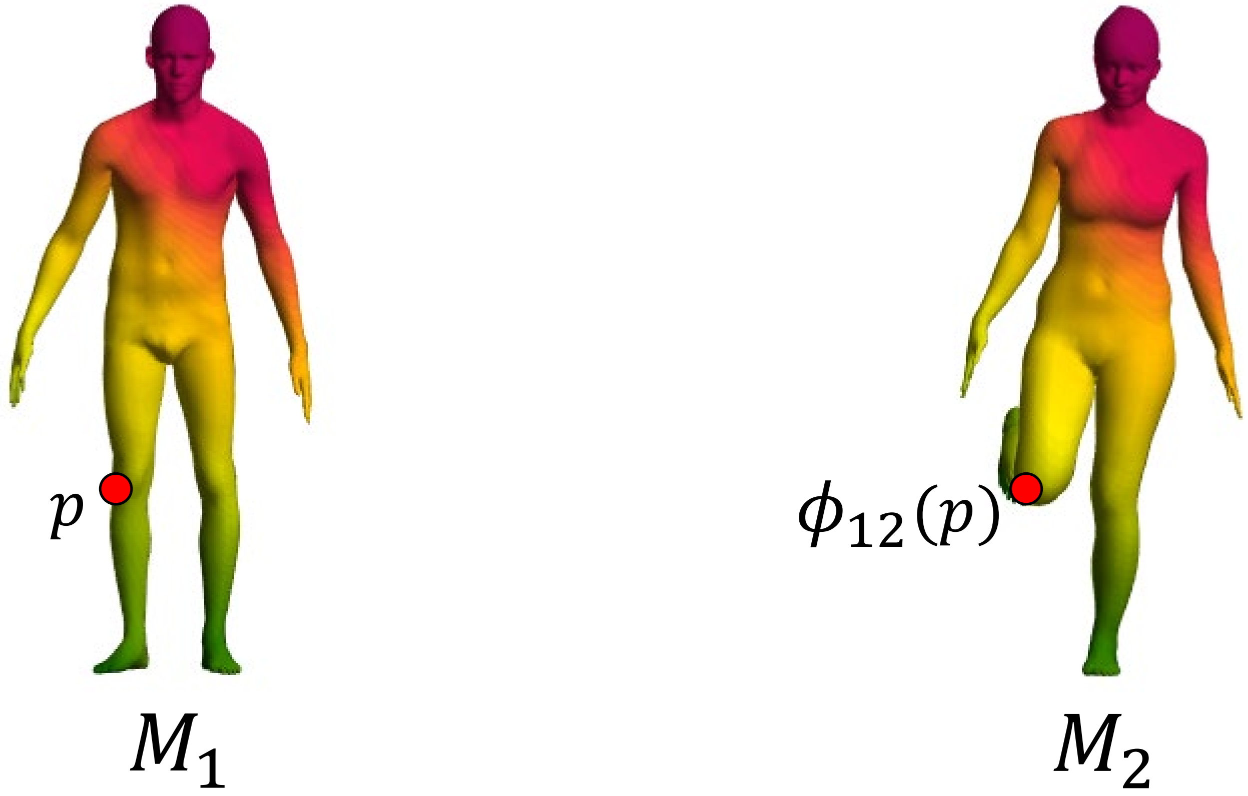


Outline

- Introduction
  - Problem formulation
  - Applications
- Semi-automatic Matching
- Fully automatic methods
  - Functional maps
  - Fuzzy maps

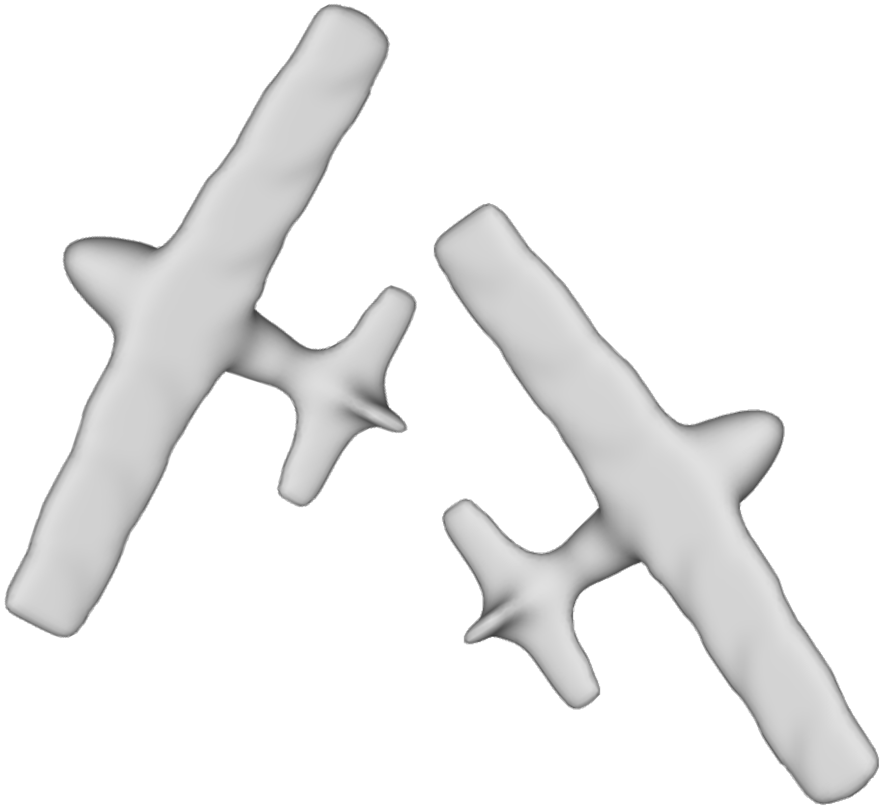
Shape Correspondence

Given two shapes M_1 and M_2, compute a semantic map $\phi_{12}: M_1 \to M_2$:

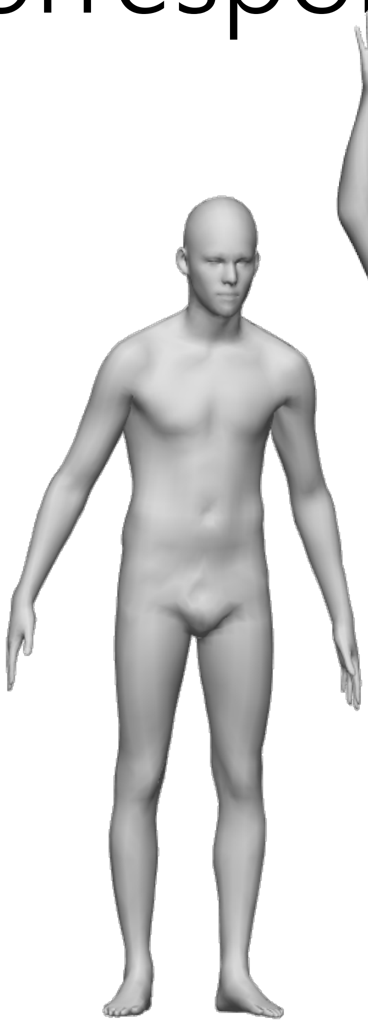


Shapes from FAUST dataset, Bogo et al.

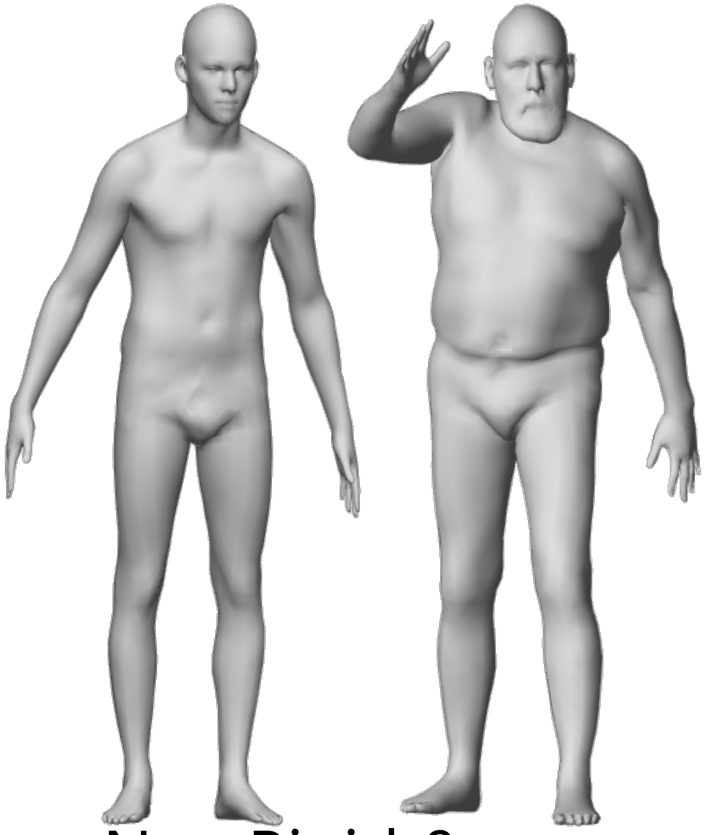
Categories (full correspondence)



Rigid



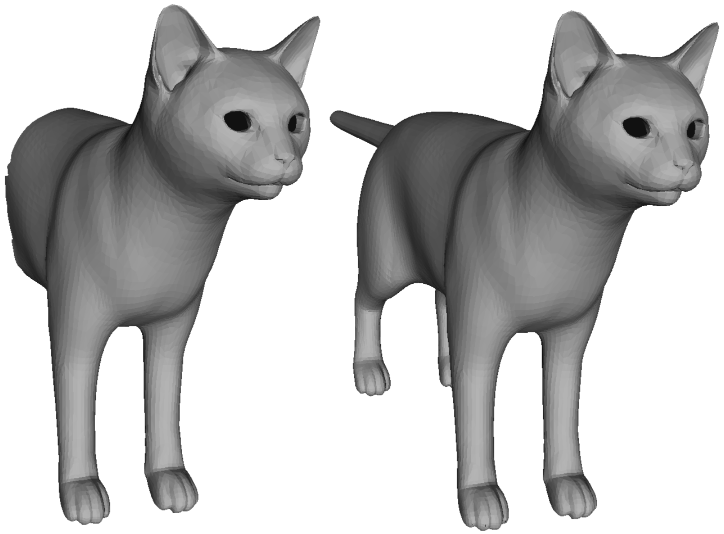
Non Rigid,
(nearly) Isometric



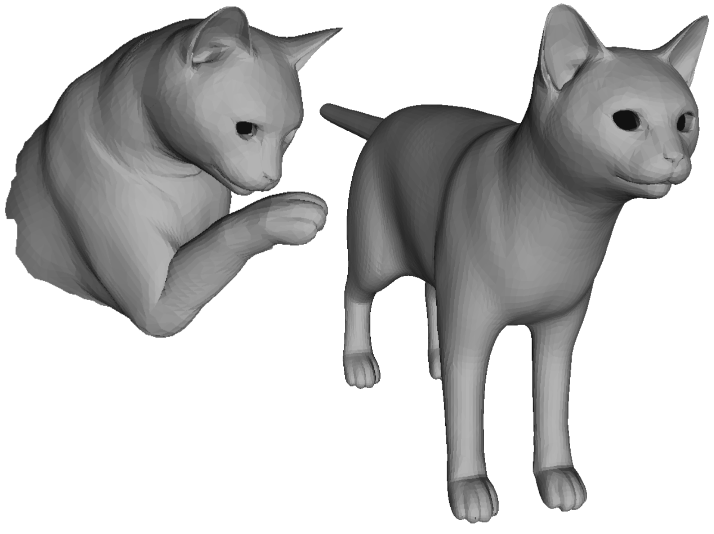
Non Rigid &
Non Isometric

Shapes from Shrec'07 & FAUST datasets

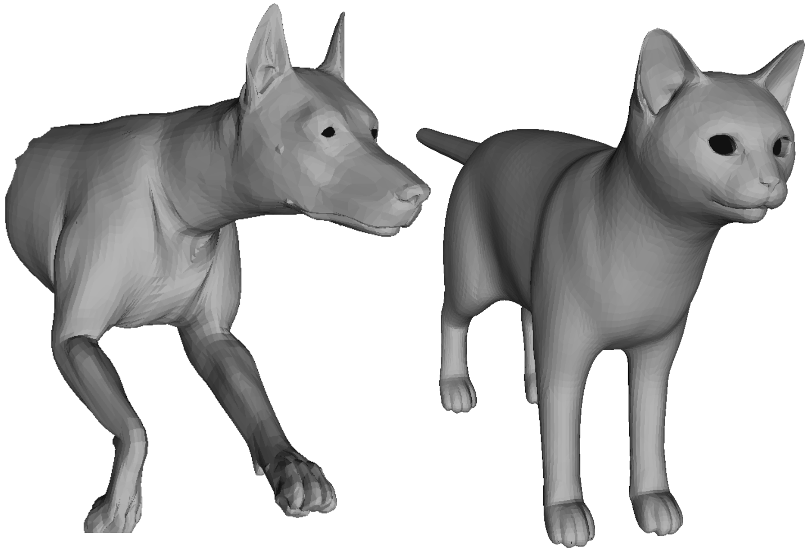
Categories (partial correspondence)



Rigid



Non Rigid,
(nearly) Isometric



Non Rigid &
Non Isometric

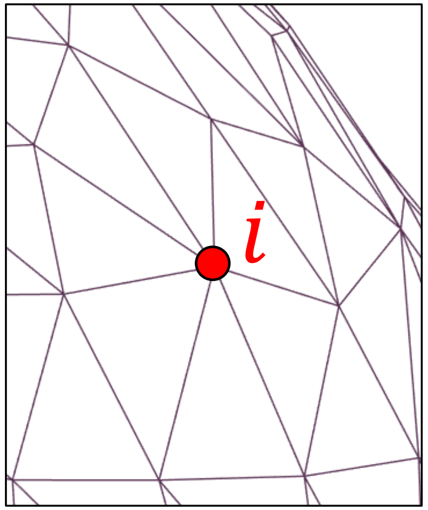
Shapes from partial TOSCA, Rodola et al.

Discrete Shape Correspondence

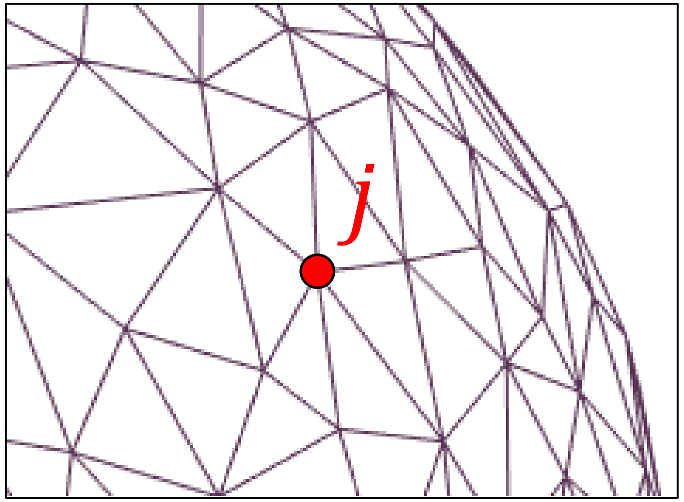
How are discrete maps between shapes represented?

Vertex-to-vertex maps: list of target vertex indices (n_1 integers)

$\boxed{\text{Combinatorial}}$



M_1



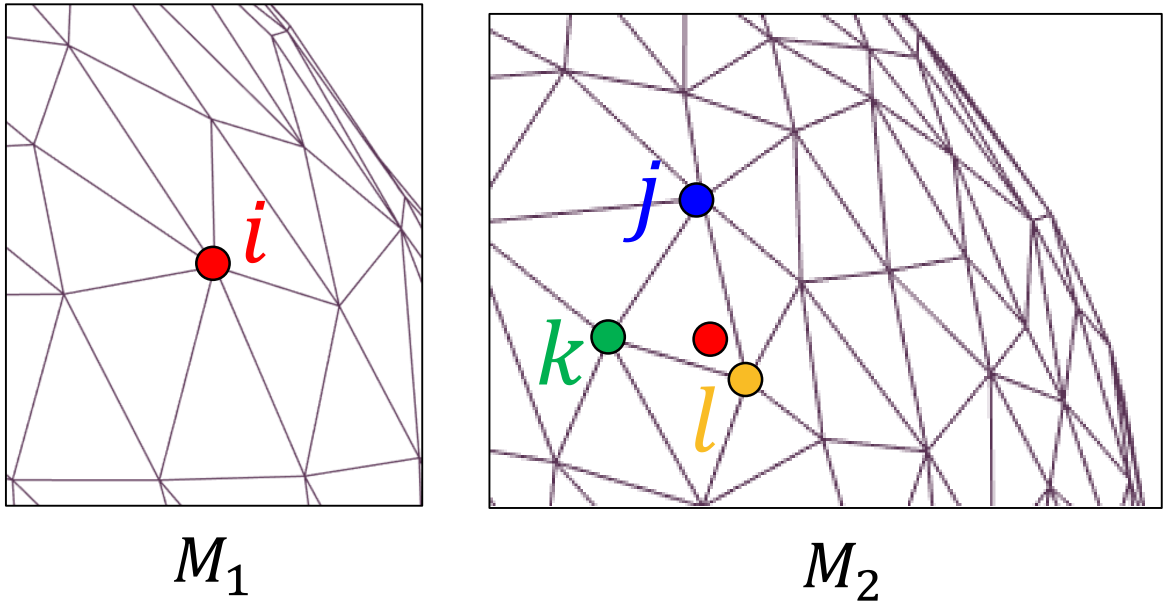
M_2

Discrete Shape Correspondence

How are discrete maps between shapes represented?

Precise maps: target face index and three barycentric coordinates for each source vertex

Continuous



Non Trivial Shape Correspondence

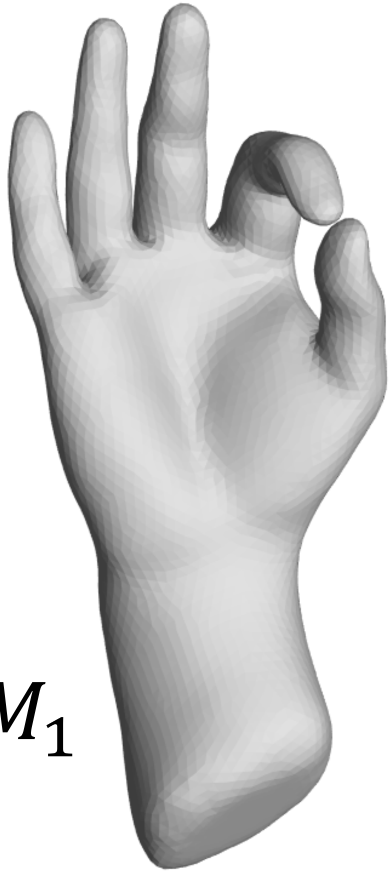
How should the correspondence look like?

Fingers, palm – straightforward

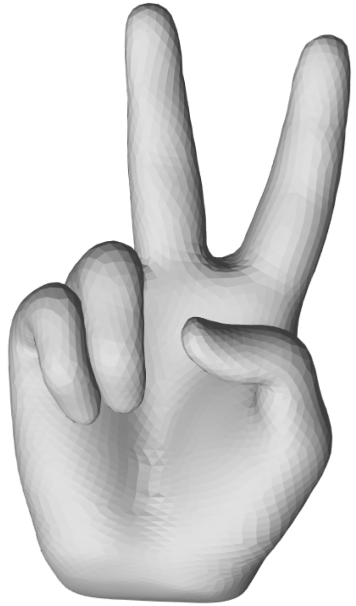
Where should we map the bottom part?

Solution #1: the bottom part has no match

Solution #2: the bottom part should be mapped smoothly



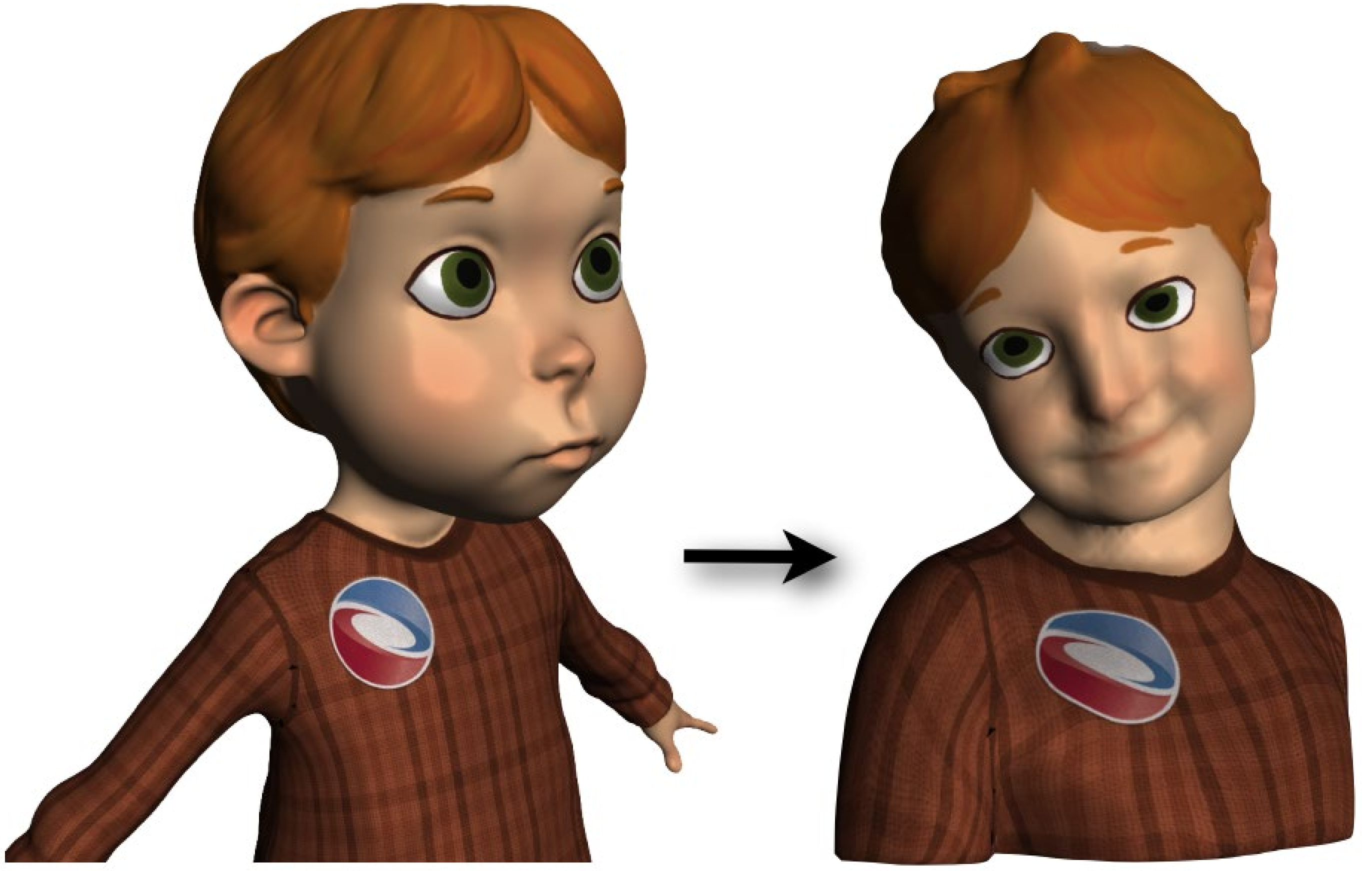
M_1



M_2

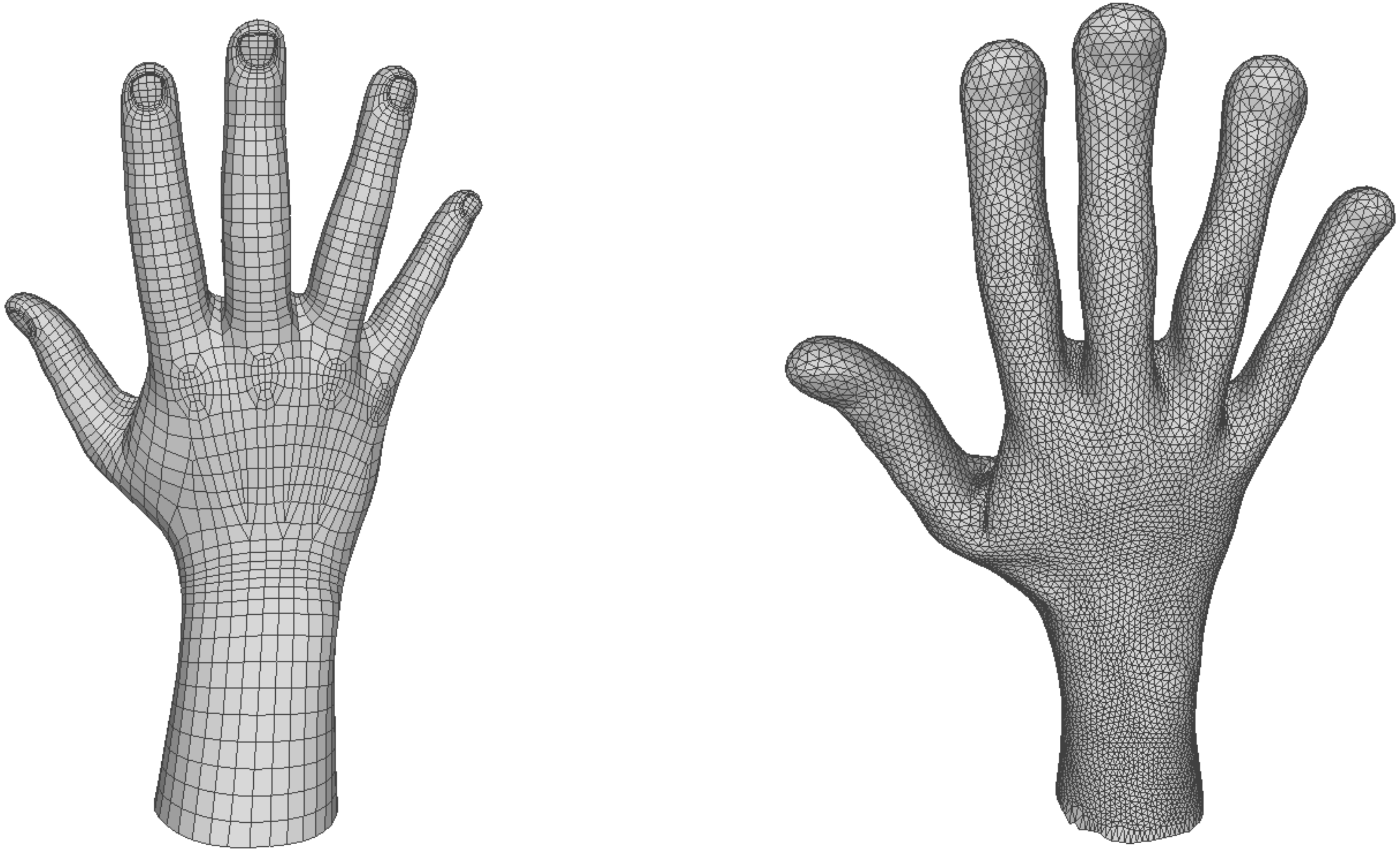
What do we need it for?

Application – Texture Transfer



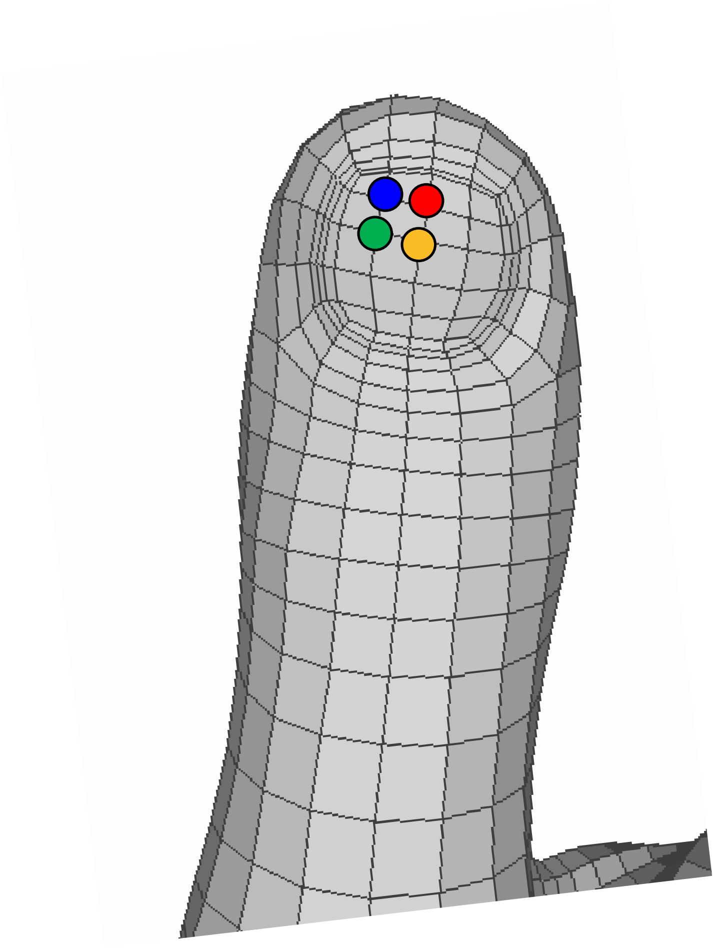
From: “Weighted Averages on Surfaces”, Panozzo et al. 2013

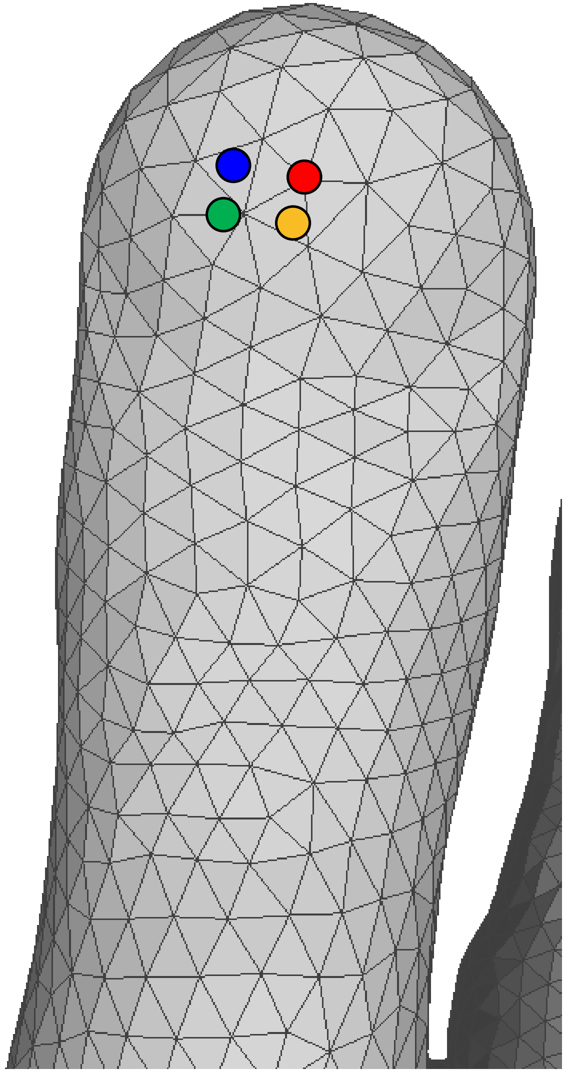
Joint Remeshing



From: Ezuz et al. 2019

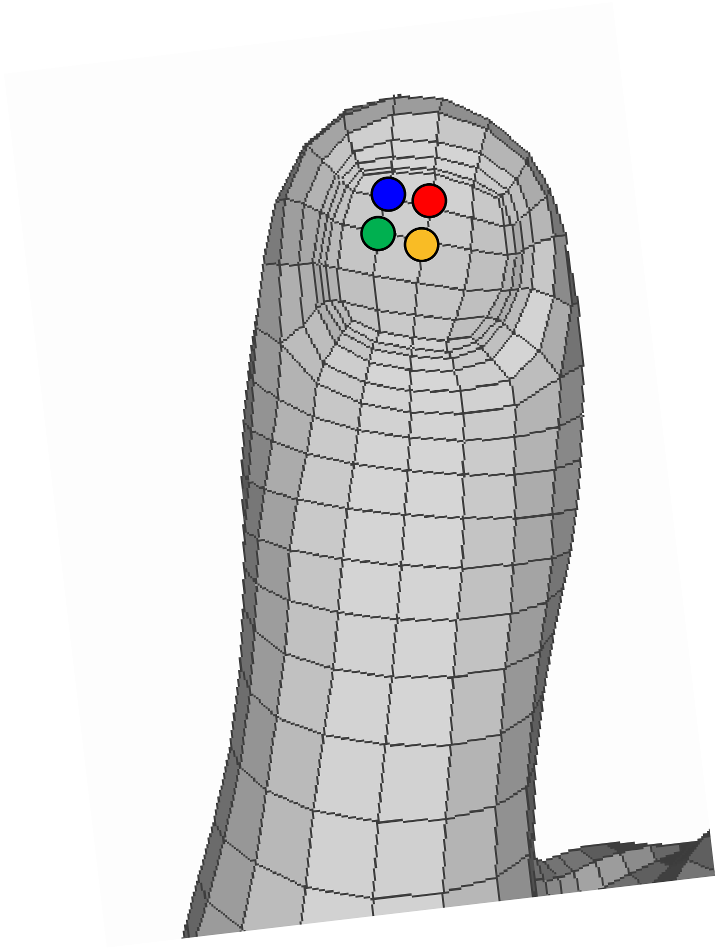
Joint Remeshing

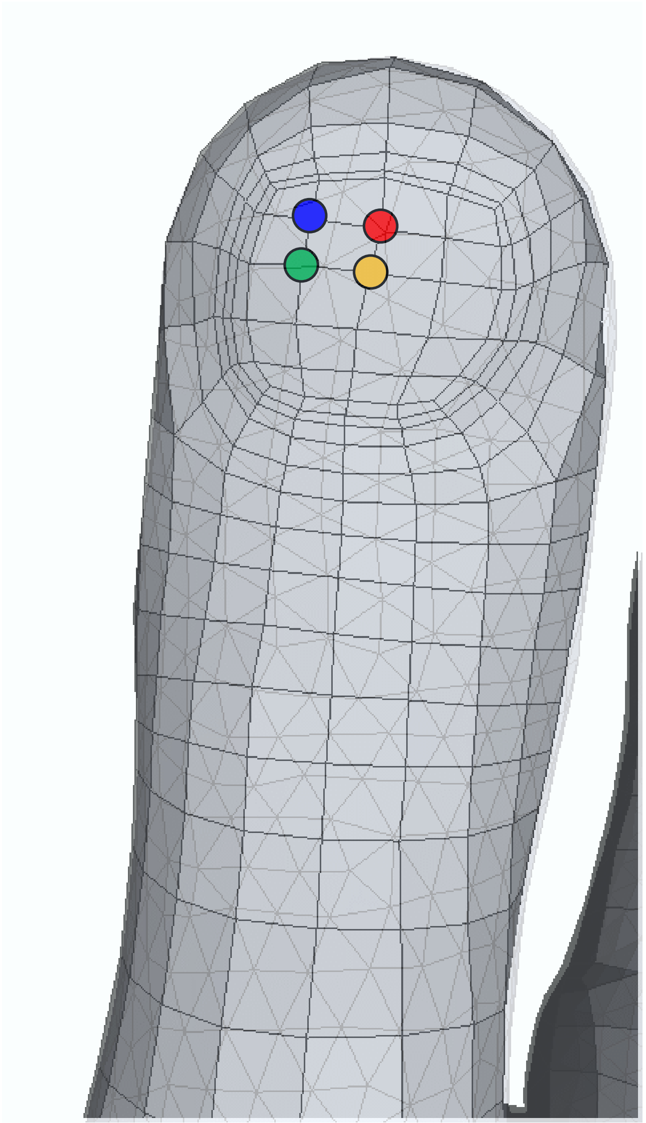




From: Ezuz et al. 2019

Joint Remeshing





From: Ezuz et al. 2019

Application – Shape Interpolation

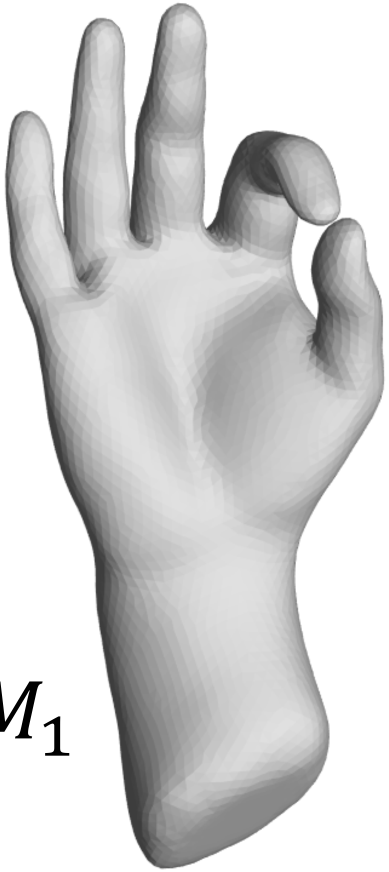
From: “Time-Discrete Geodesics in the Space of Shells”, Heeren et al. 2012

More Applications

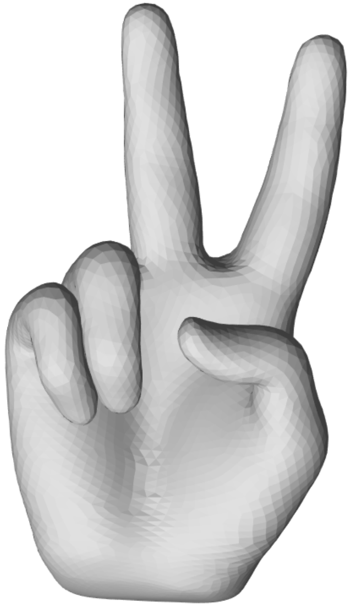
- Shape alignment for application of DNNs on 3D
- Deformation transfer
- Joint remeshing
- Statistical shape analysis
- Registration
- Object recognition
- …

Non Trivial Shape Correspondence

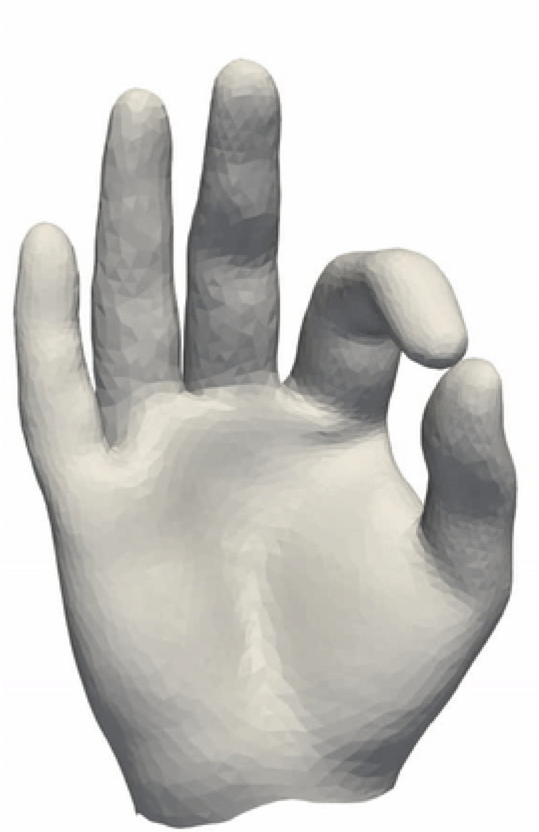
What do we need it for? Interpolation:



M_1



M_2

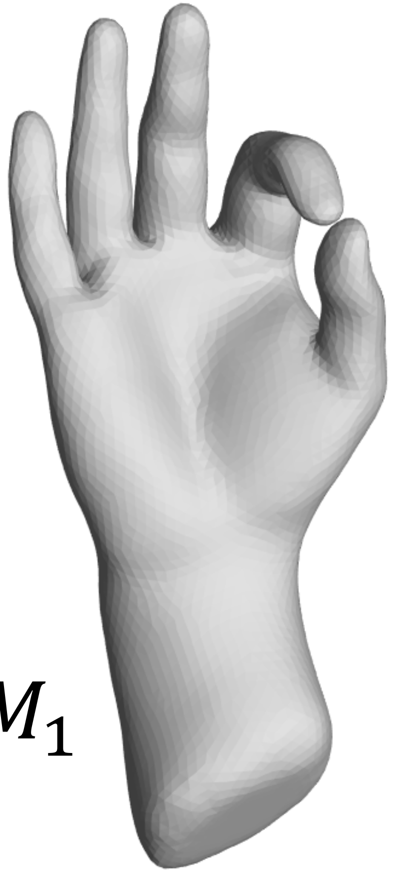


With partial correspondence

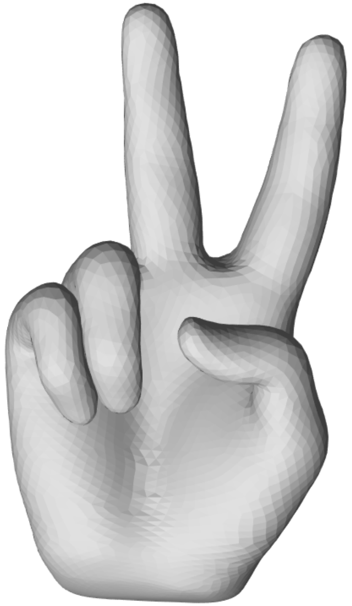
Interpolation: Behrend Heeren

Non Trivial Shape Correspondence

What do we need it for? Interpolation:



M_1



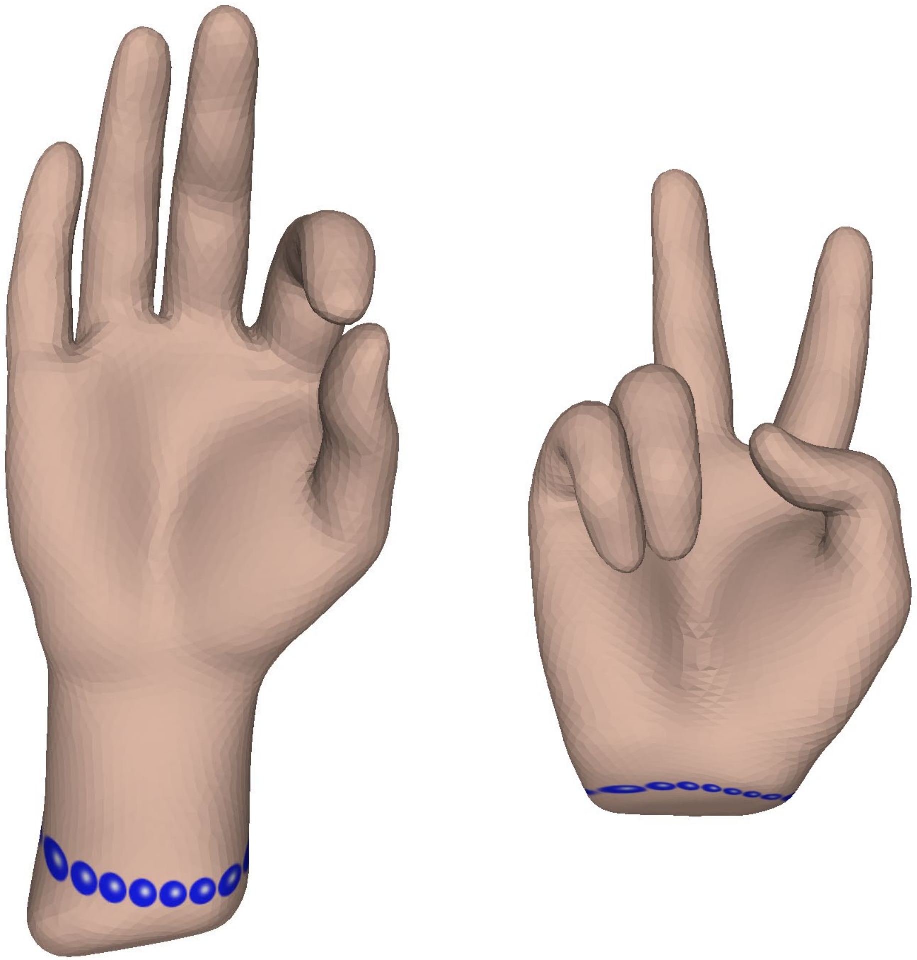
M_2

With full correspondence

Interpolation: Behrend Heeren

Non Trivial Shape Correspondence

What do we need it for? Texture transfer:

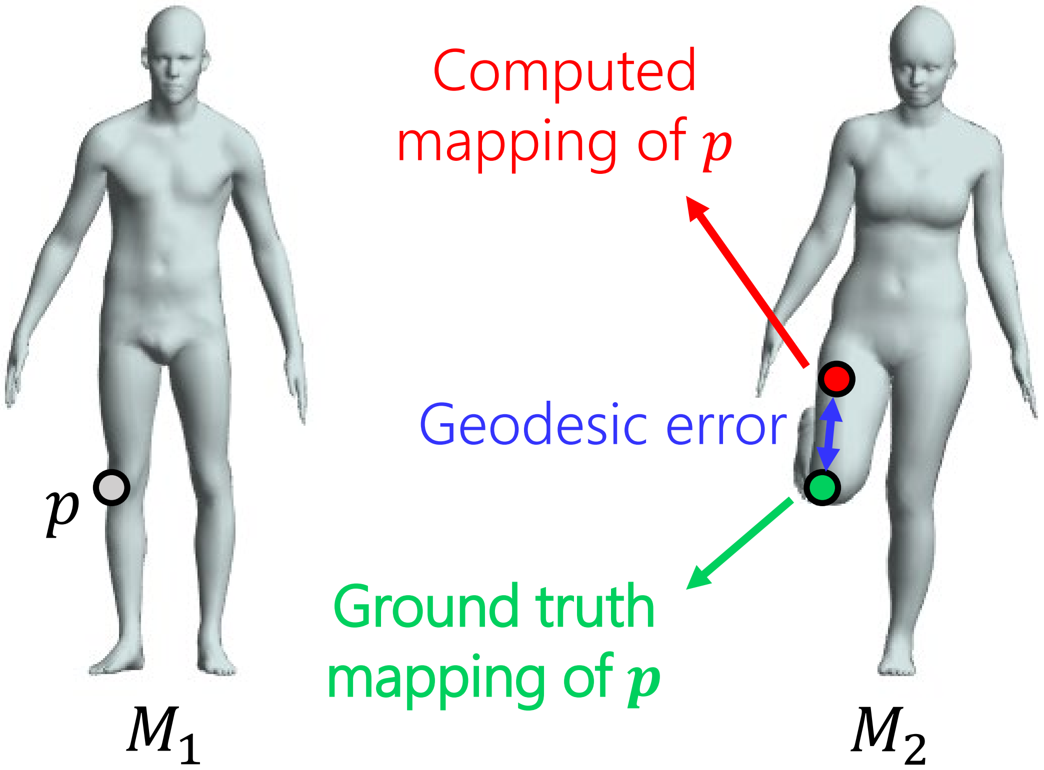


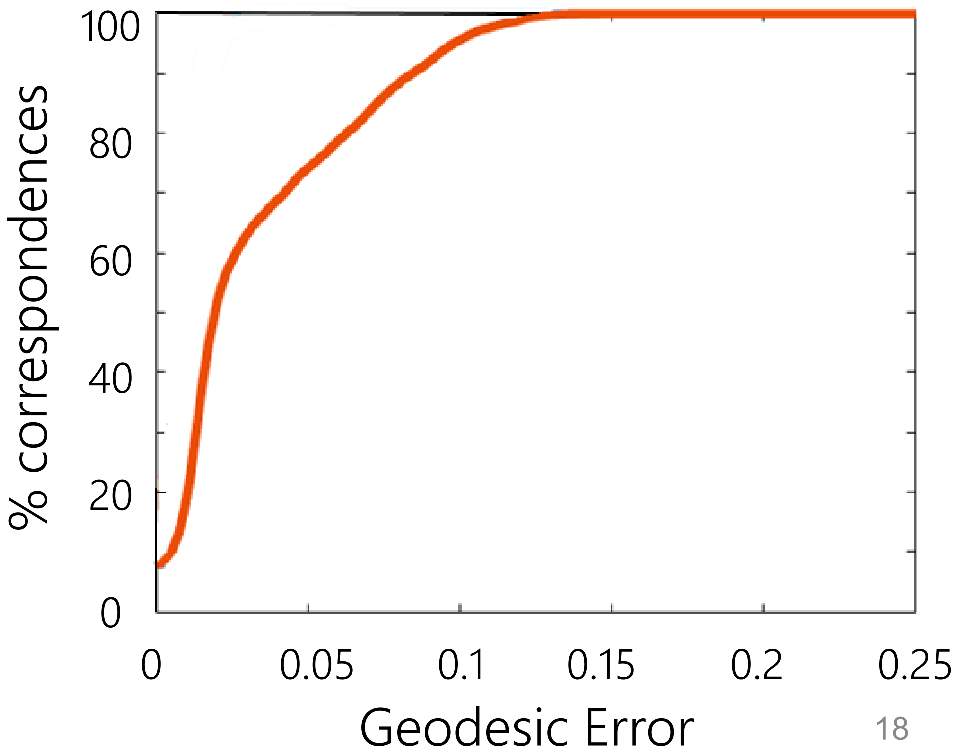
The desired correspondence is application dependent

Evaluation

How can we evaluate a given map quality?

Given a ground truth map, compute the cumulative error graph

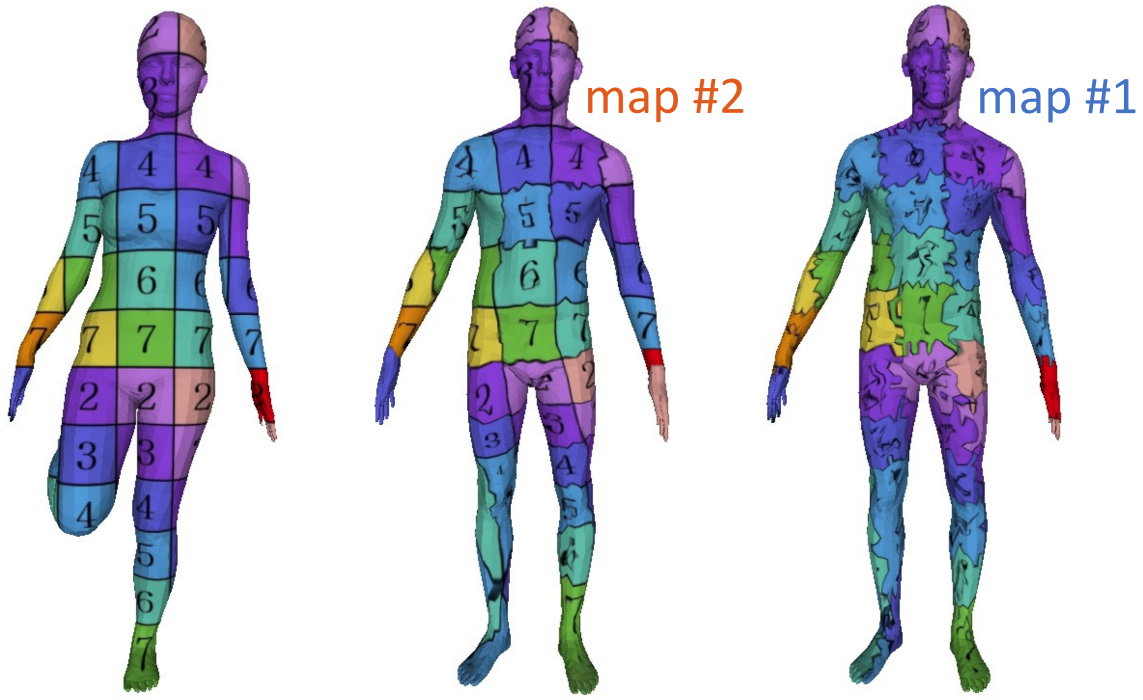




Evaluation

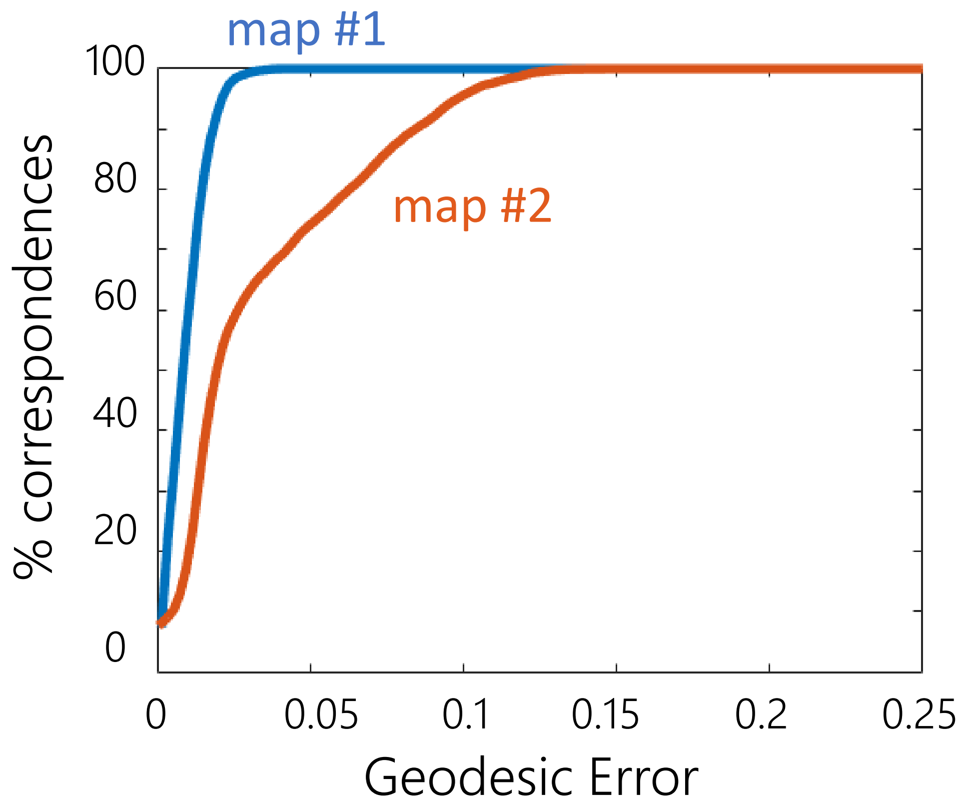
How can we evaluate a given map quality?

Given a ground truth map, compute the cumulative error graph



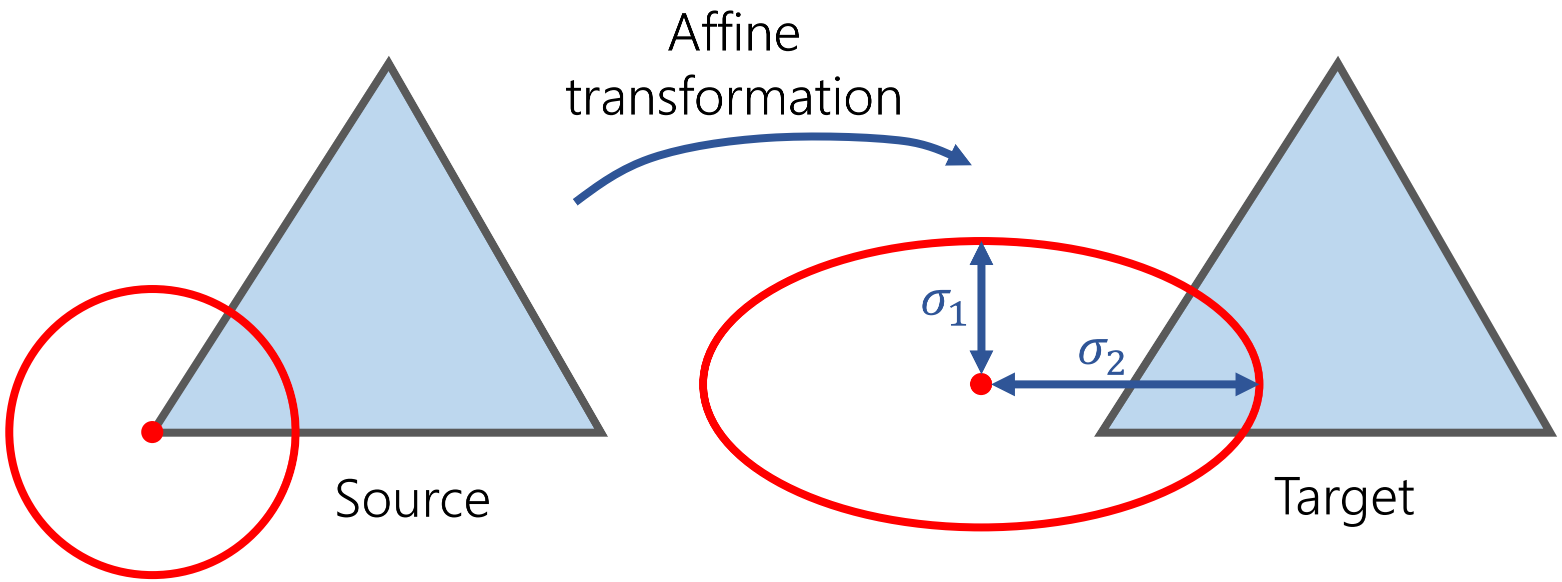
Target Texture

Which one is map #1?



Conformal Distortion

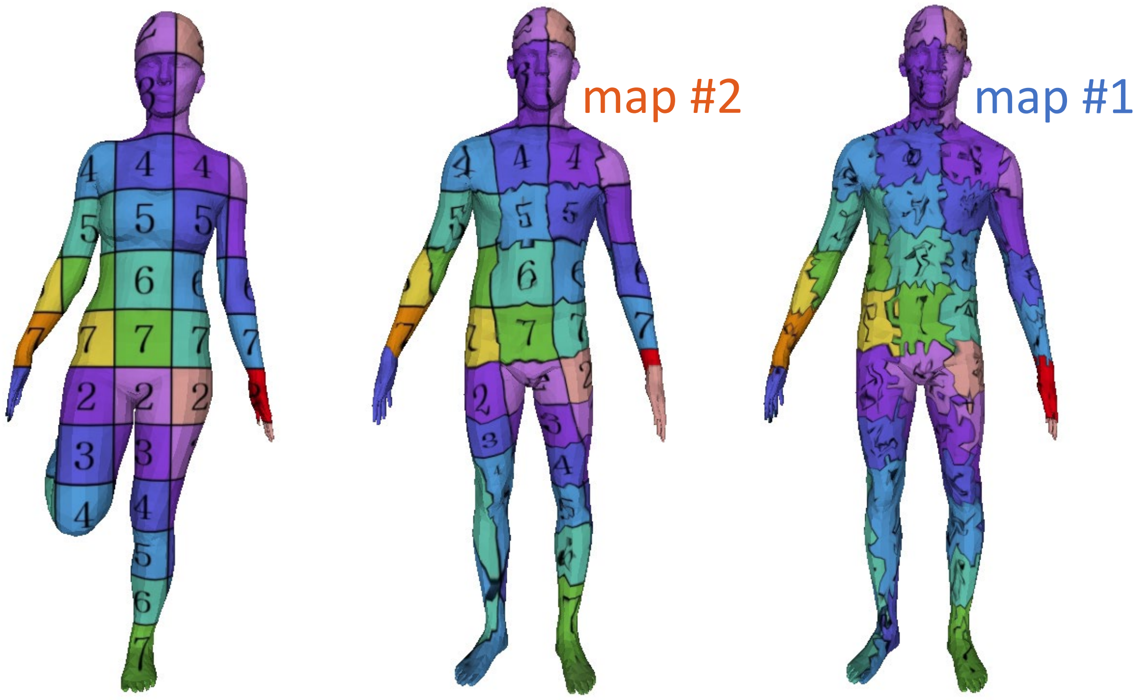
A conformal map preserves angles



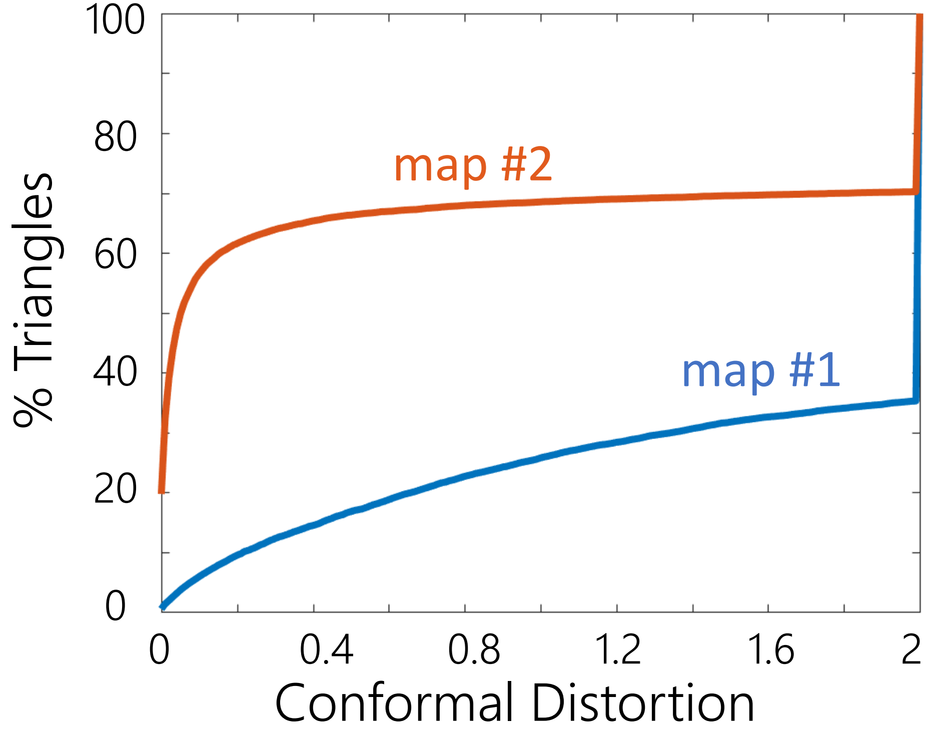
Evaluation

How can we evaluate a given map quality?

Measure *conformal distortion* (angle preservation)



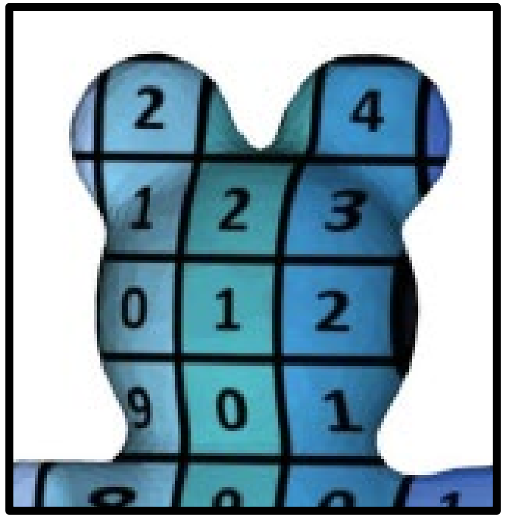
Target Texture



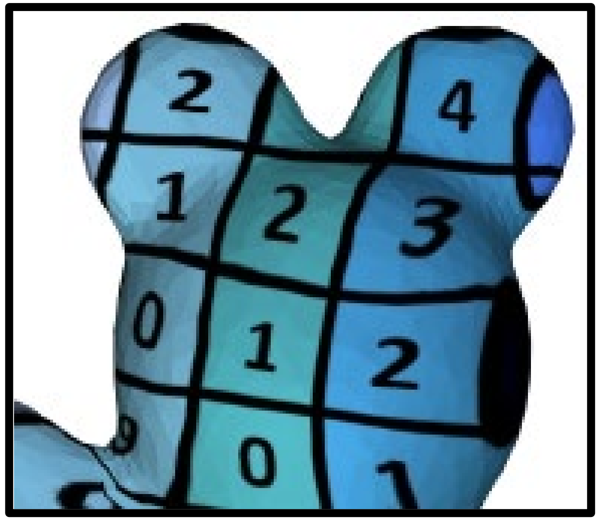
Qualitative Evaluation

Visualization using texture transfer:

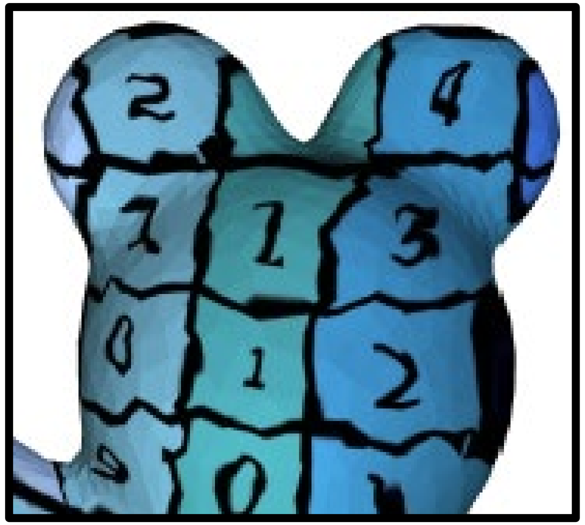
vertex-to-vertex



Target Texture (projection)

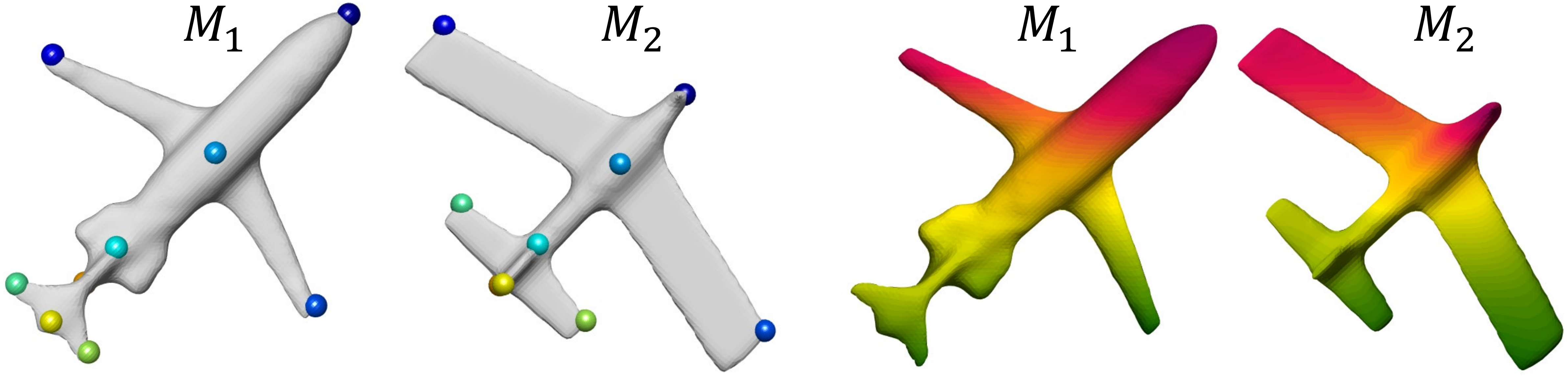


Locally and globally accurate map



Globally accurate, locally distorted map

Semi-automatic Matching



Input

Output

Parameterization based

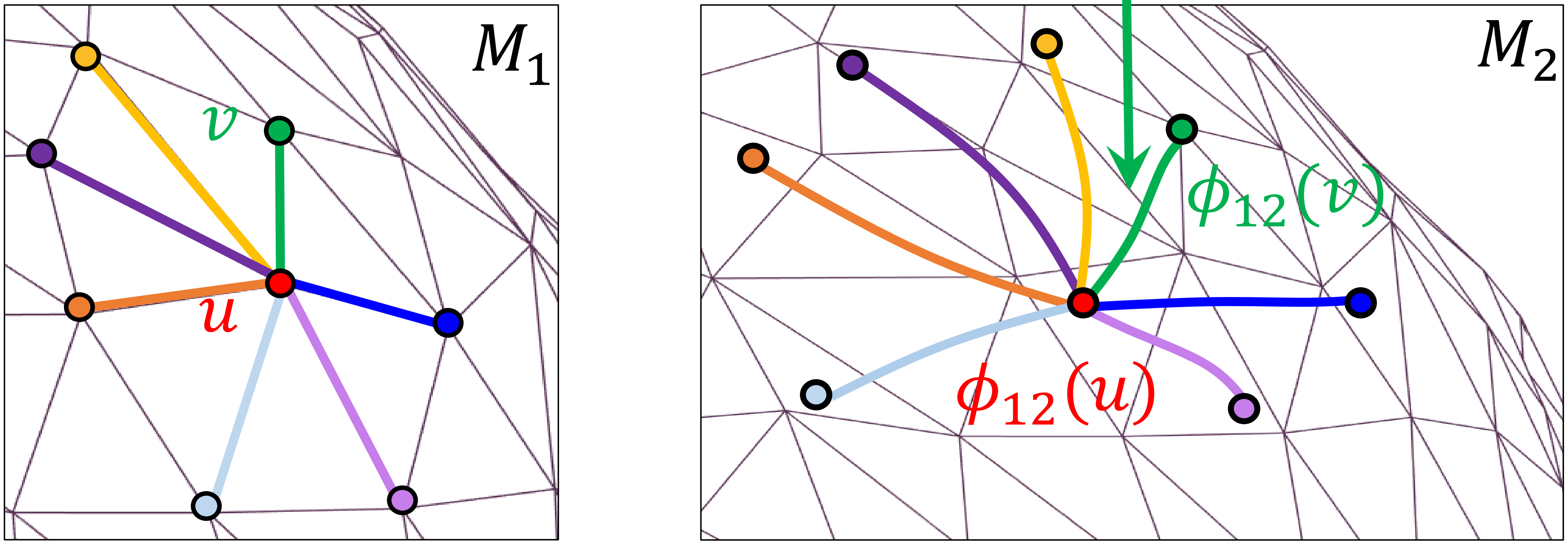
Motivation: the Dirichlet energy measures smoothness

$$E(\phi_{12}) = \frac{1}{2} \int_{M_1} |d\phi_{12}|^2$$

A map is **harmonic** if it is a critical point of the Dirichlet energy

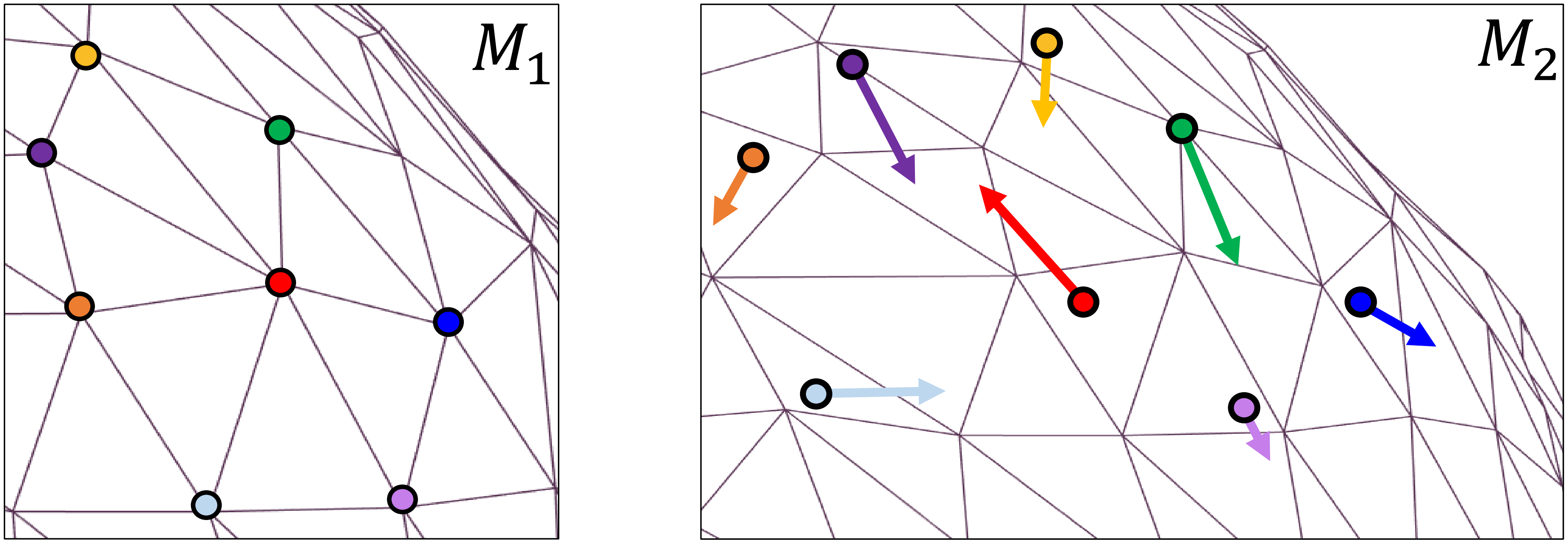
Discrete Dirichlet Energy

$$E_D(\phi_{12}) = \sum_{(u,v)\in \mathrm{E}_1} w_{uv}\, \underbrace{d^2_{M_2}(\phi_{12}(u), \phi_{12}(v))}$$



Discrete Dirichlet Energy

The gradient of a matching energy:



Discrete Dirichlet Energy

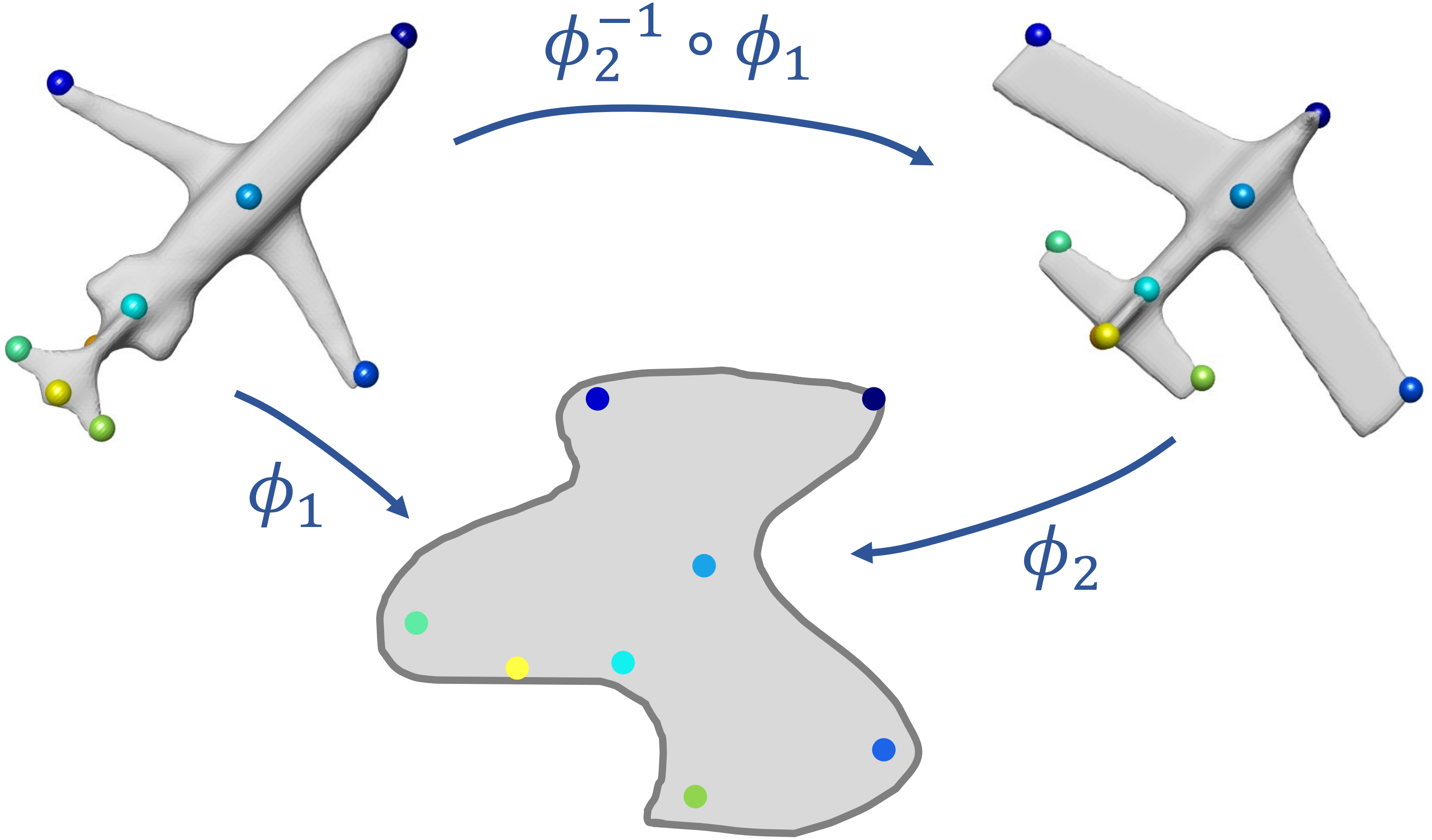
$$E_D(\phi_{12}) = \sum_{(u,v)\in \mathrm{E}_1} w_{uv} d^2_{M_2}(\phi_{12}(u), \phi_{12}(v))$$

Gradient of the Dirichlet energy:

- Computationally expensive (geodesics on **target**)
- Well defined only for hyperbolic **target**

Parameterization-based Correspondence

Motivation: for some **target** domains the problem is much simpler



Parameterization-based Correspondence

Common domain examples:

- Plane
- Sphere
  - Uniformization theorem: any genus zero surface can be mapped conformally to the unit sphere
- Orbifolds [Aigerman et al.]
  - Spherical
  - Hyperbolic

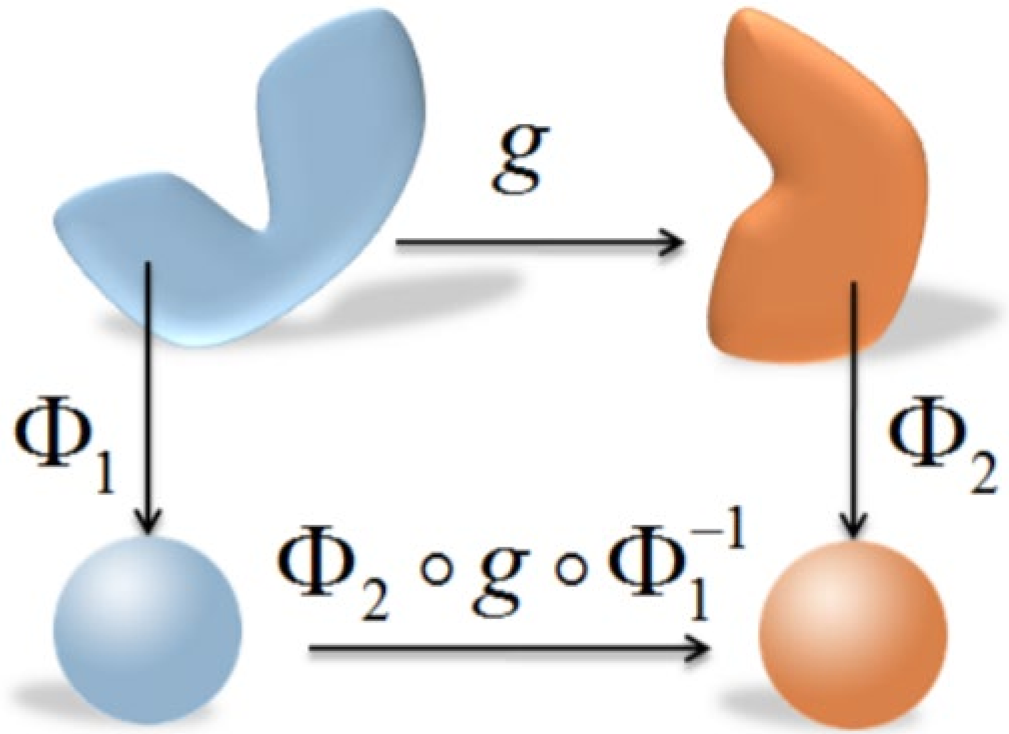
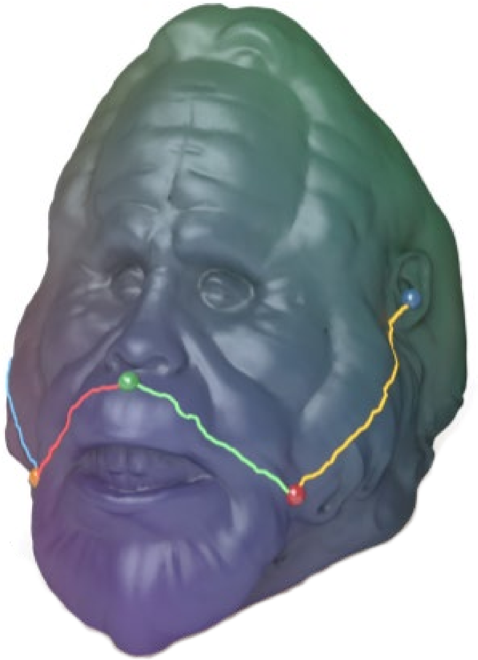
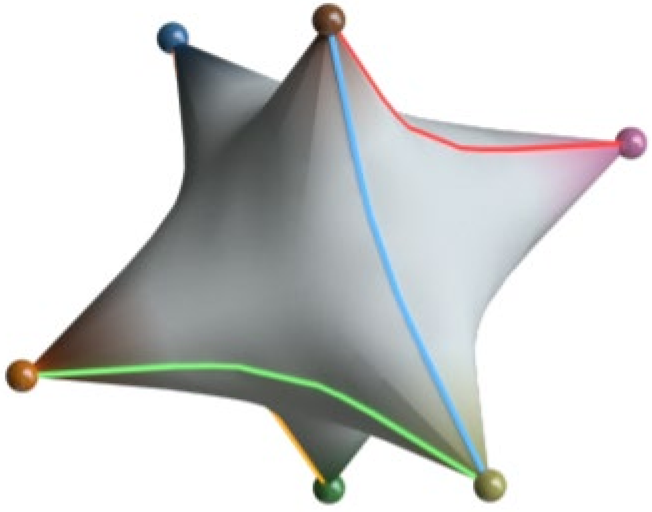


Image from: “Mobius Voting For Surface Correspondence”, Lipman & Funkhouser, SIGGRAPH 2009

Parameterization-based Correspondence

Aigerman & Lipman, “Hyperbolic Orbifold Tutte Embeddings”:
Use a hyperbolic orbifold as the common domain





Hyperbolic orbifold illustration

Example result

Parameterization-based Correspondence

Hyperbolic orbifold common domain:

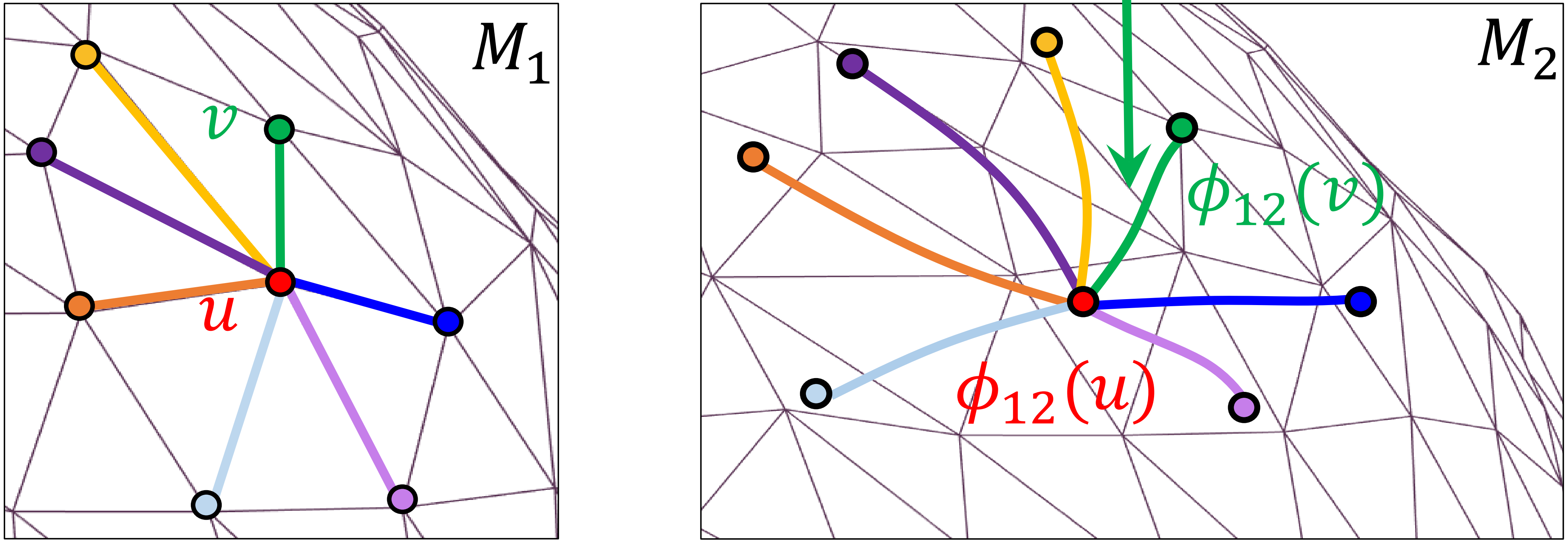
+ Gradient of the Dirichlet energy is well defined
+ **Bijectivity** is well defined and guaranteed (continuous setting)
+ Distances can be computed analytically
+ Not limited to vertex-to-vertex output

However:

- Continuous guarantees do not always hold in discrete cases
- Composition of maps might not be with minimal distortion
- Topological constraints (same genus)

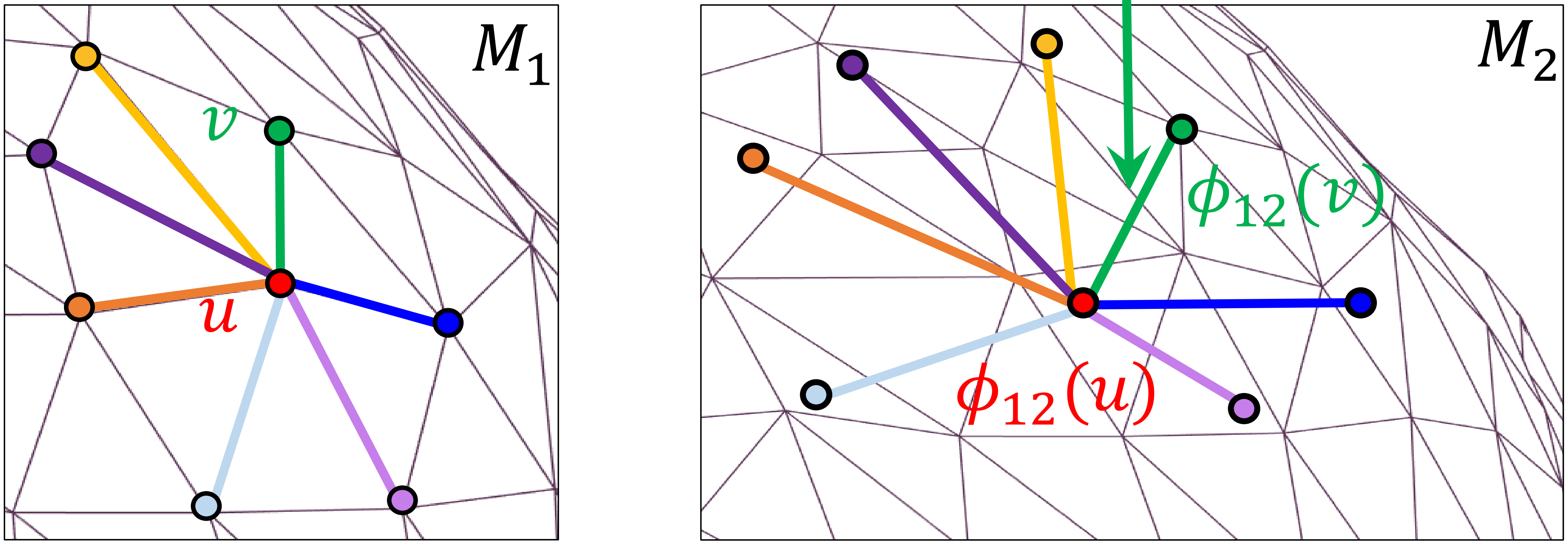
Discrete Dirichlet Energy

$$E_D(\phi_{12}) = \sum_{(u,v)\in \mathrm{E}_1} w_{uv} d^2_{M_2}(\phi_{12}(u), \phi_{12}(v))$$



Discrete Dirichlet Energy

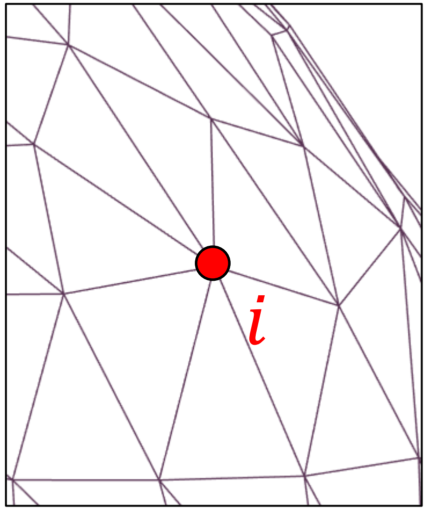
$$E_D^{Euc}(\phi_{12}) = \sum_{(u,v)\in E_1} w_{uv} \underbrace{\|\phi_{12}(u) - \phi_{12}(v)\|^2}$$



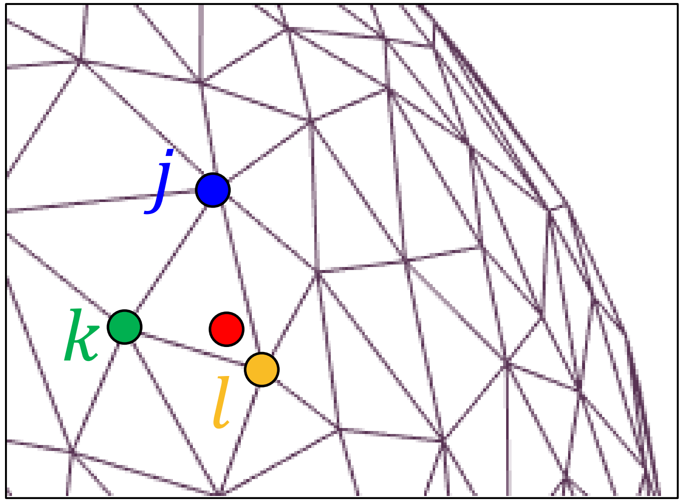
Discrete Precise Maps

Stochastic matrices with barycentric coordinates at each row:

$$P_{12} = \begin{array}{c} \begin{matrix} \quad j & \quad k & \quad l \end{matrix} \\ \begin{pmatrix} & \vdots & \\ 0.1 & 0.2 & 0.7 \\ & \vdots & \end{pmatrix} \end{array} \text{row } i$$



M_1



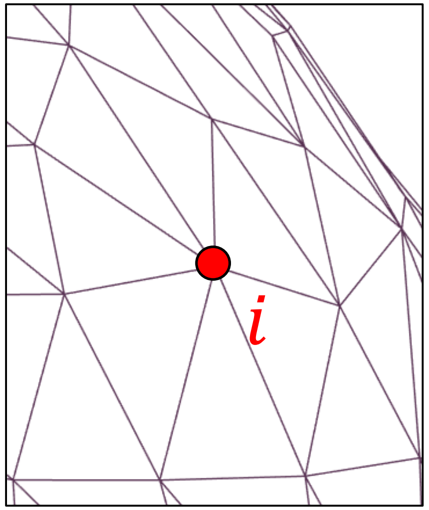
M_2

Discrete Precise Maps

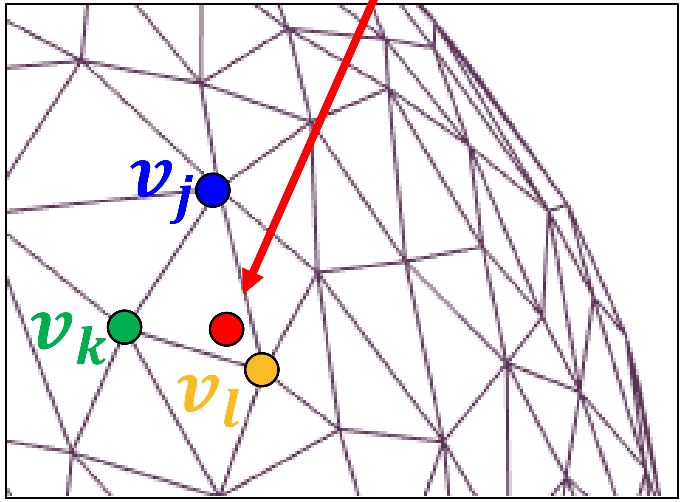
Stochastic matrices with barycentric coordinates at each row:

i^{th} row of $P_{12}V_2$

$$\underbrace{\begin{array}{c c} & \begin{array}{ccc} j & k & l \end{array} \\ i & \begin{pmatrix} & & \vdots & & \\ - & 0.1 - & 0.2 - & 0.7 - \\ & & \vdots & & \end{pmatrix} \end{array}}_{P_{12}} \underbrace{\begin{pmatrix} - v_j - \\ - v_k - \\ - v_l - \end{pmatrix}}_{V_2}$$



M_1



M_2

$V_2 \in \mathbb{R}^{n_2 \times 3}$ is a matrix with vertex coordinates of M_2

Discretization – Dirichlet Energy

If we replace the geodesic distances by Euclidean distances, the discrete Dirichlet energy is:

$$E_D^{Euc}(\phi_{12}) = \sum_{(u,v)\in \mathrm{E}_1} w_{uv} \|\phi_{12}(u) - \phi_{12}(v)\|^2$$

$$= \cdots = \|P_{12}V_2\|_{W_1}^2$$

Discrete Dirichlet Energy

We use a *high dimensional embedding* where Euclidean distances approximate geodesic distances (MDS)

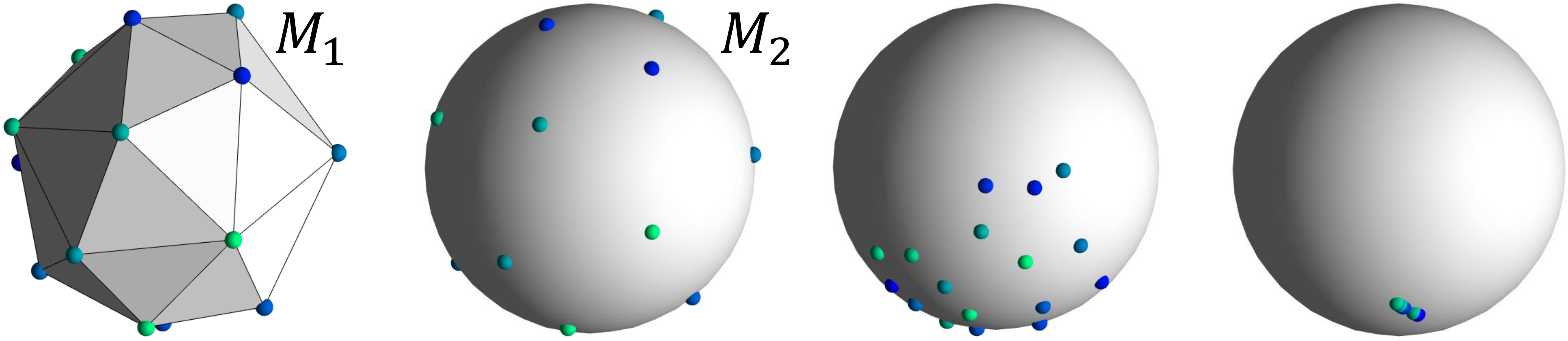
$$X_2 \in \mathbb{R}^{n_2 \times 8}$$

Then the discrete Dirichlet energy is approximated by:

$$E_D(P_{12}) = \|P_{12}X_2\|_{W_1}^2$$

From: “Reversible Harmonic Maps between Discrete Surfaces”, Ezuz et al., TOG 2019

Minimizing the Dirichlet Energy



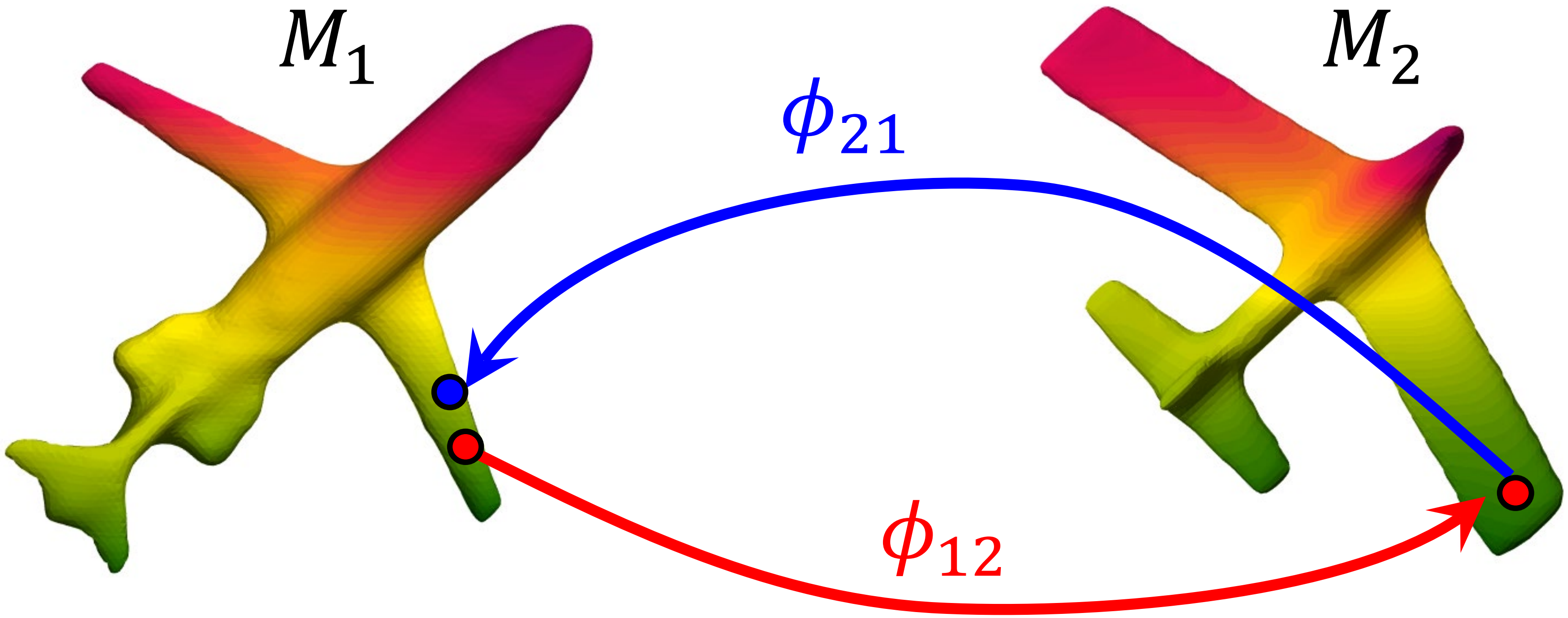
Initial map (Id)

optimize E_D

From: "Reversible Harmonic Maps between Discrete Surfaces", Ezuz et al., TOG 2019

Reversibility

- We add a **reversibility term** to prevent the map from shrinking



$$E_R(P_{12}, P_{21}) = \|P_{21}P_{12}X_2 - X_2\|_{M_2}^2 + \|P_{12}P_{21}X_1 - X_1\|_{M_1}^2$$

Reversible Harmonic Energy

“Reversible Harmonic Maps between Discrete Surfcaes”, Ezuz et al. 2019:

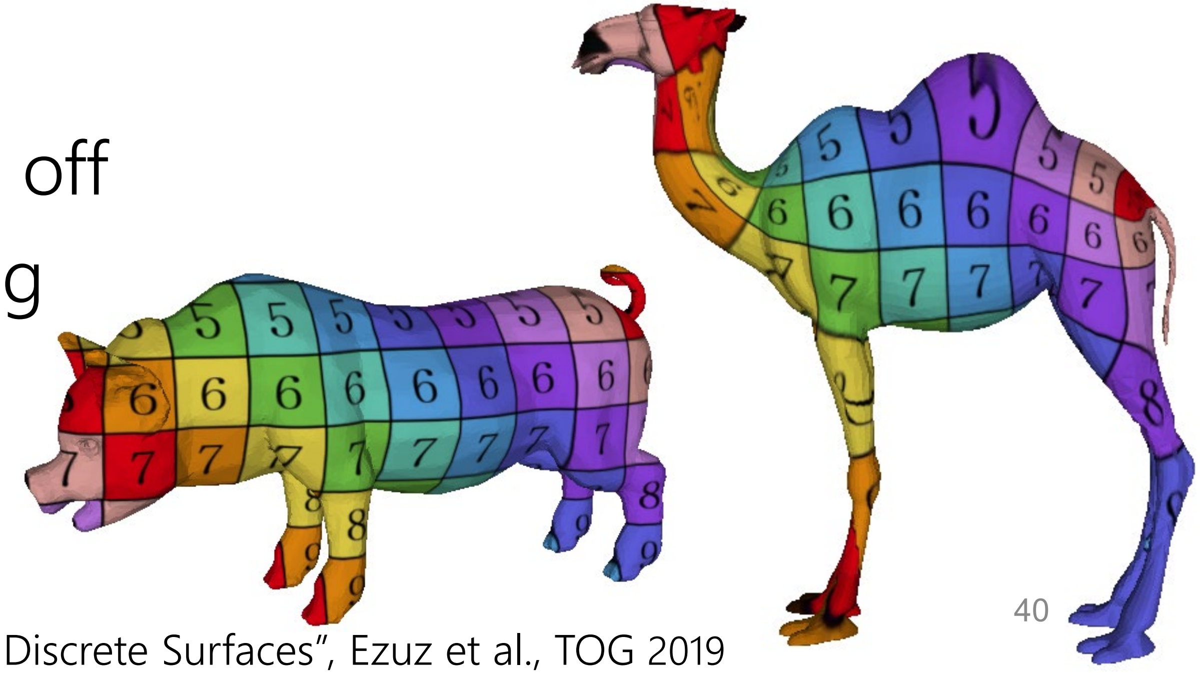
Combine the Dirichlet energy and the reversibility term:

$$E(P_{12}, P_{21}) = \alpha E_D(P_{12}) + \alpha E_D(P_{21}) + (1 - \alpha) E_R(P_{12}, P_{21})$$

The parameter α controls the trade off

Optimization: half quadratic splitting

and alternating minimization

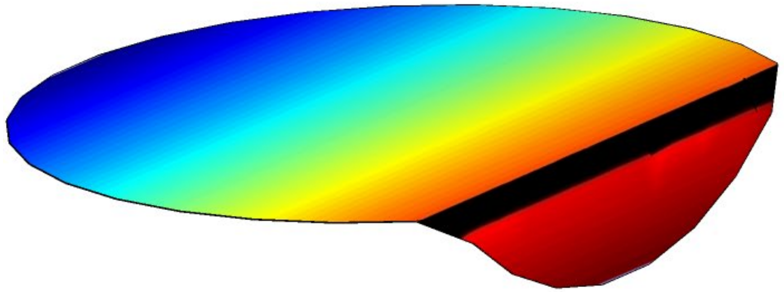


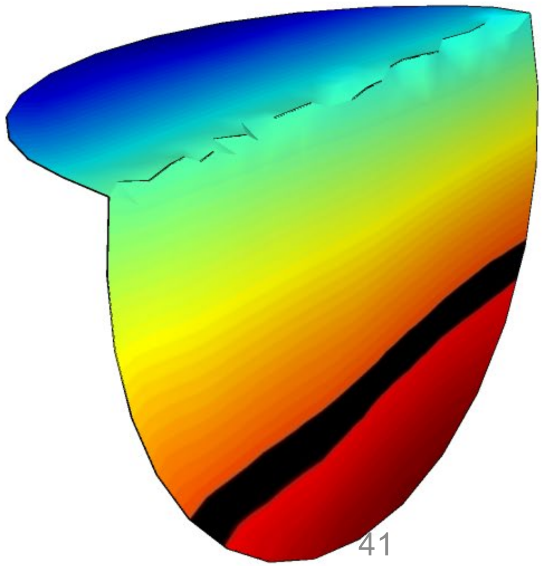
From: “Reversible Harmonic Maps between Discrete Surfaces”, Ezuz et al., TOG 2019

Reversible Harmonic Energy

+ Distortion is minimized directly (results have low conformal distortion)
+ **Bijectivity** is promoted using reversibility
+ No topological constraints
+ Not limited to vertex-to-vertex output

However:
Intrinsic methods do not align extrinsic features





From: “Reversible Harmonic Maps between Discrete Surfaces”, Ezuz et al., TOG 2019

Elastic Energy

The discrete thin-shell energy consists of two terms:

- Membrane energy: E_M
- Bending energy: E_B

$$E_{elastic} = \alpha E_M + \eta E_B$$

Frequently used for shape deformation, tricky

to optimize for shape correspondence

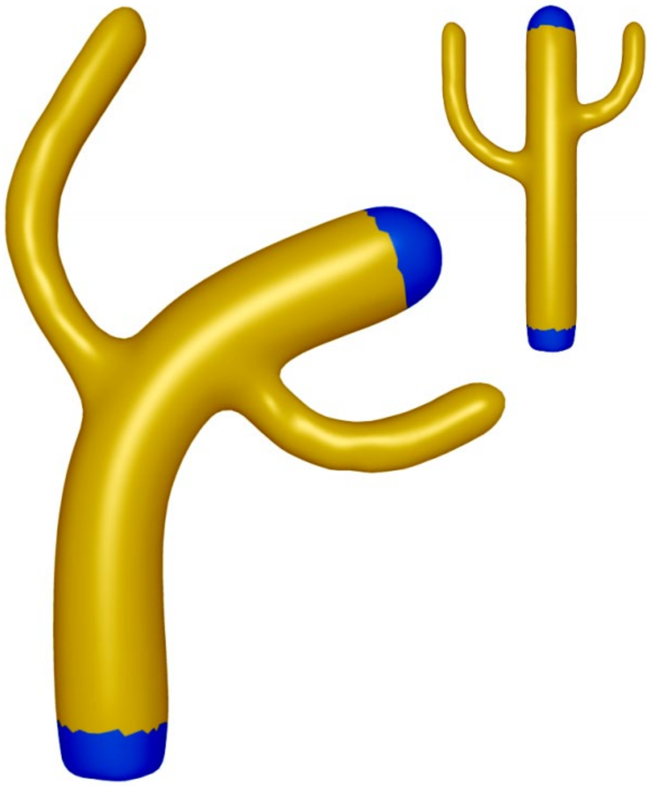


Image from: "Interactive Surface Modeling using Modal Analysis", Hilderbrandt et al., SIGGRAPH 2012

Shape Correspondence and Deformation

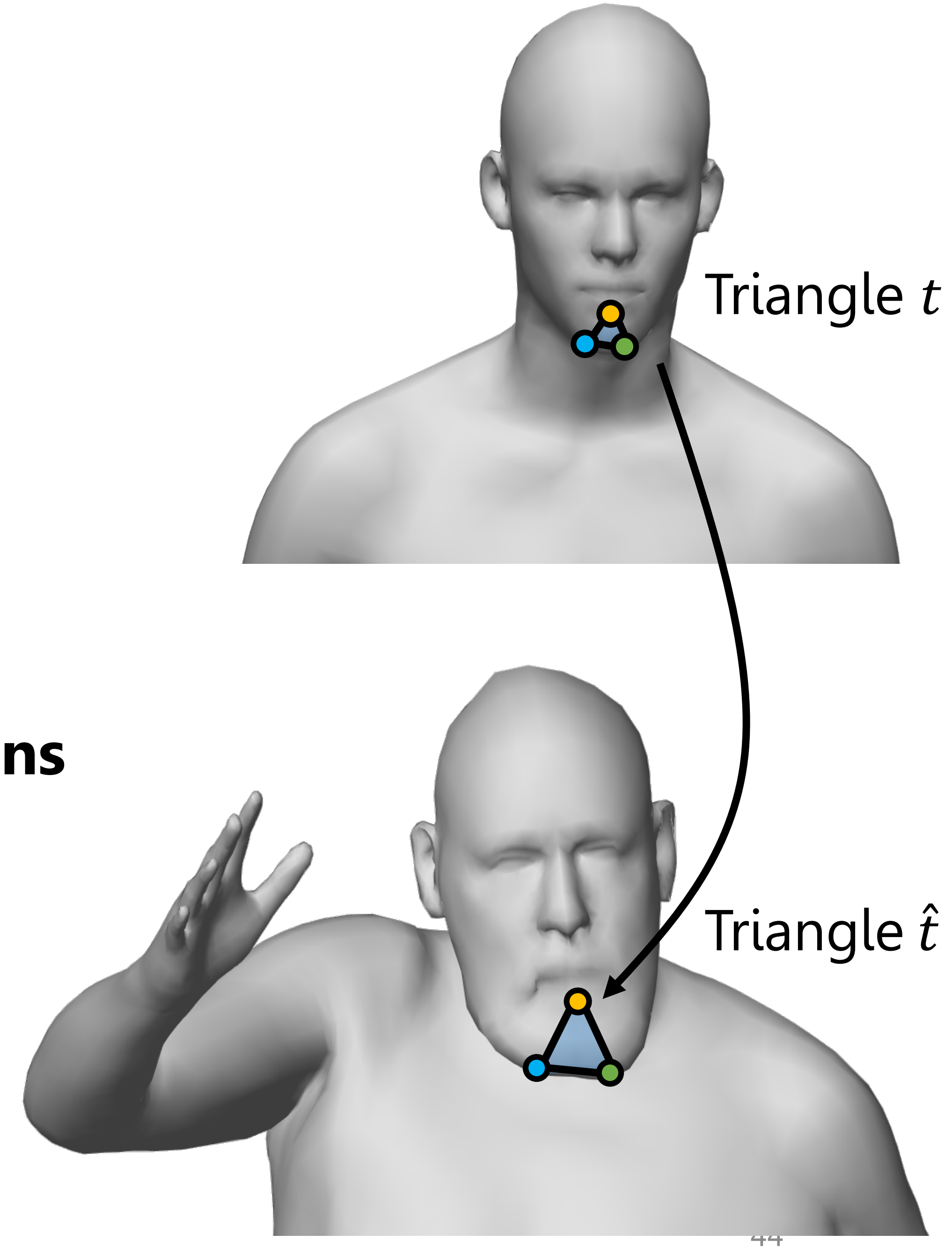
Correspondence is a constrained deformation

Membrane Energy

Sum of area distortion per triangle:

$$E_M = \sum_{t \in faces} \hat{a}_t W(t, \hat{t})$$

$W(t, \hat{t})$ is a non linear functional
that is **minimal** only for **rigid transformations**

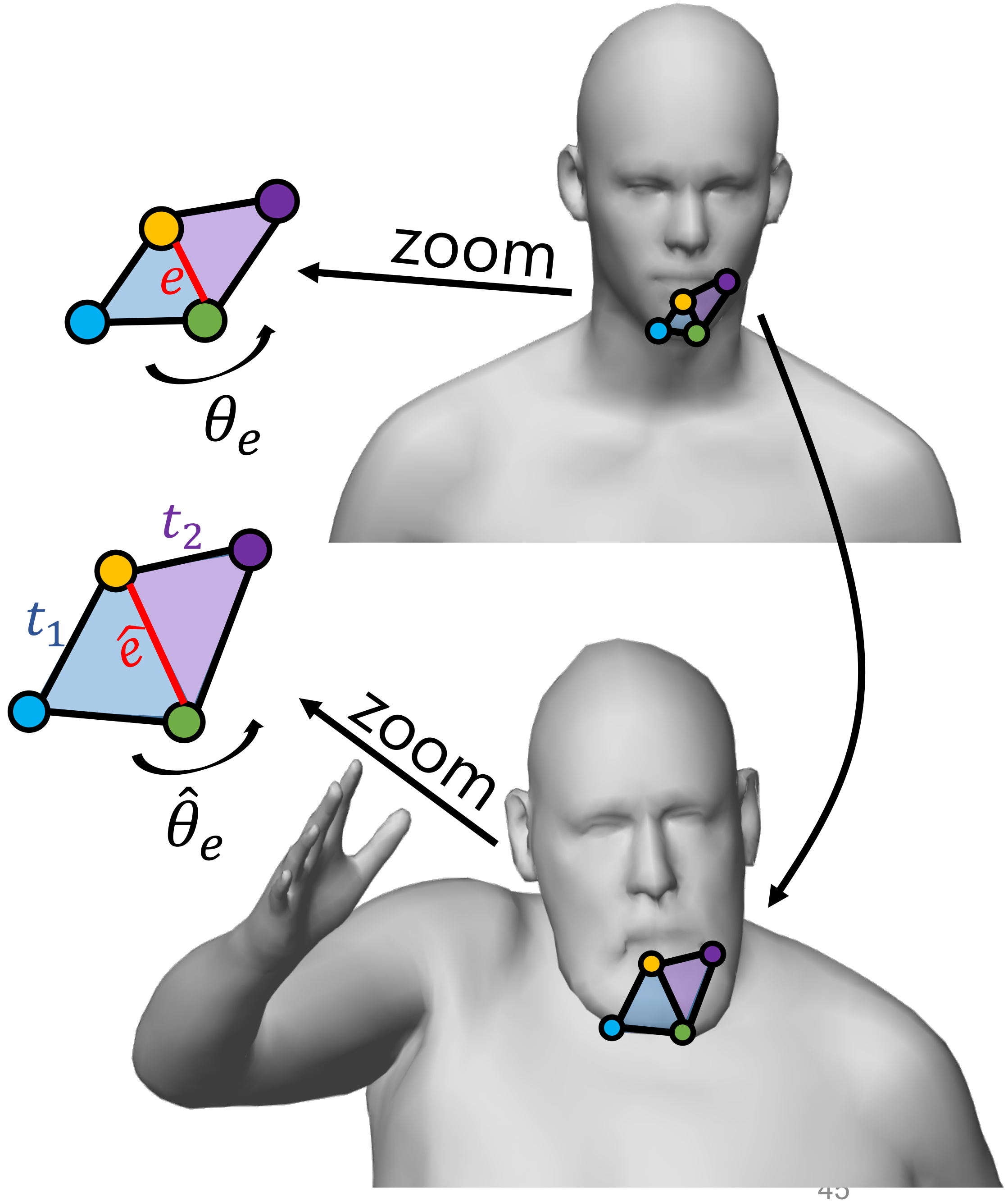


Bending Energy

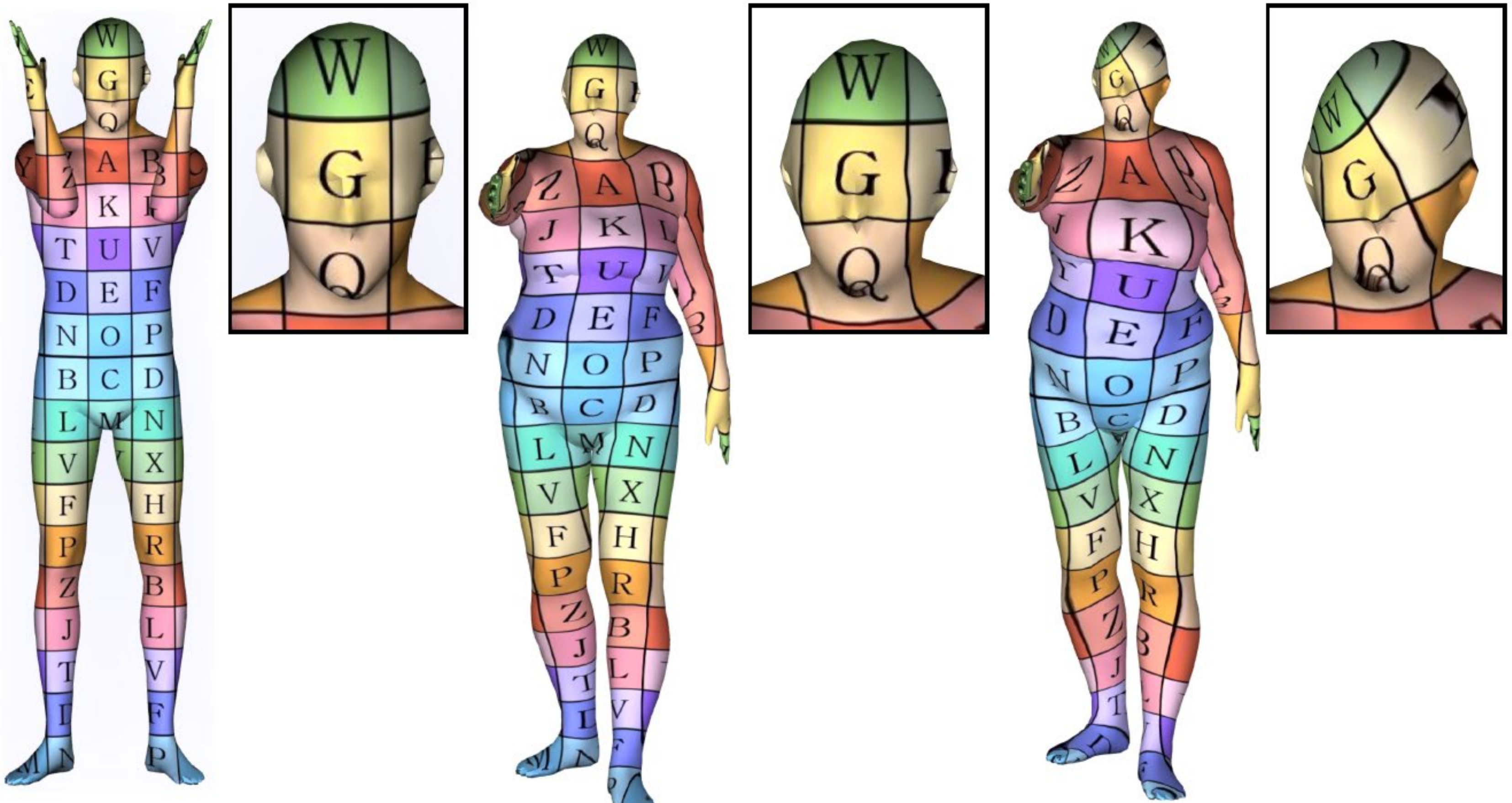
The bending energy compares dihedral angles:

$$E_B(X,Y) = \sum_{e \in edges} w_e\left(\theta_e - \hat{\theta}_e\right)^2$$

Aligns **mean curvature**



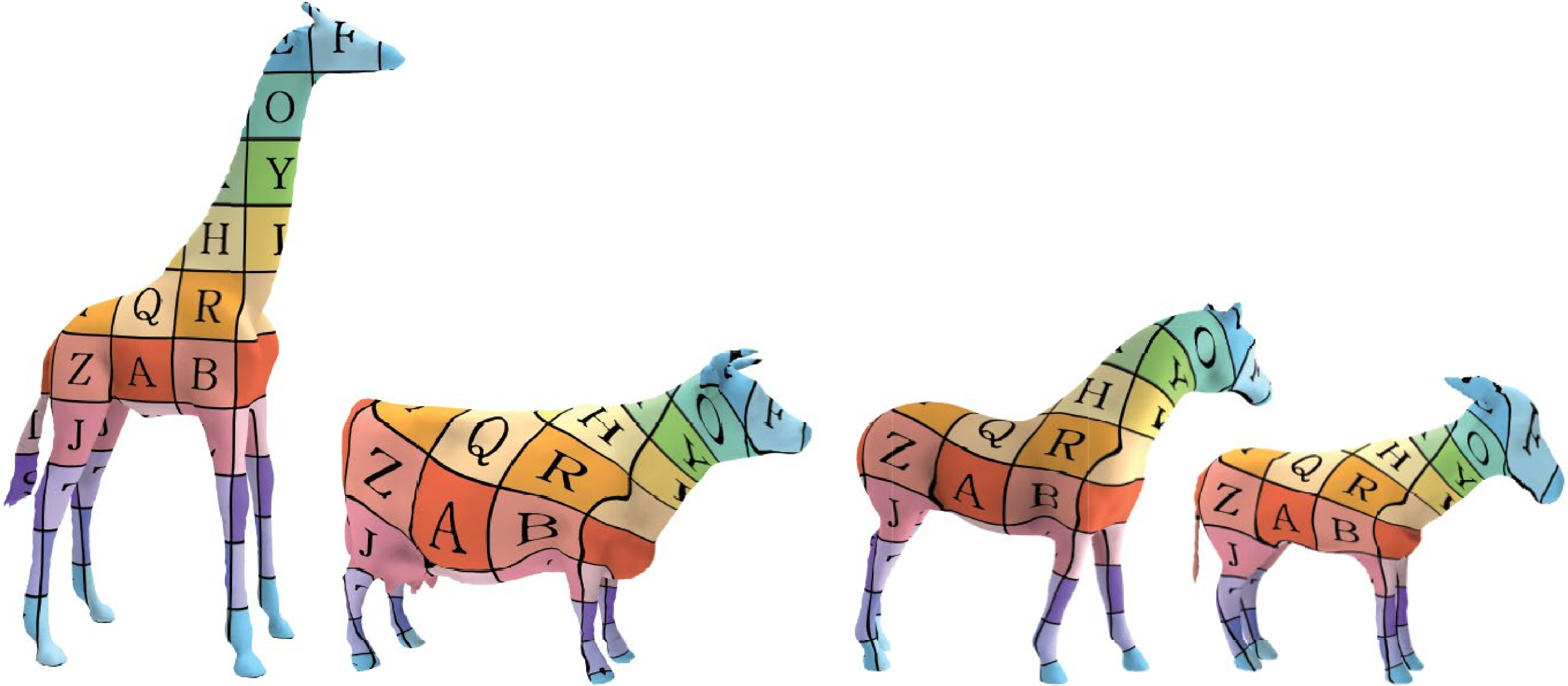
Elastic Matching



Target HOT+elastic Initial Map (HOT)

HOT: “Hyperbolic orbifold tutte embeddings”, Aigerman & Lipman, SIGGRAPH Asia 2016
Elastic: “Elastic Correspondence between Triangle Meshes”, Ezuz et al., Eurographics 2019

Elastic Matching



Elastic: “Elastic Correspondence between Triangle Meshes”, Ezuz et al., Eurographics 2019

Is the problem solved?

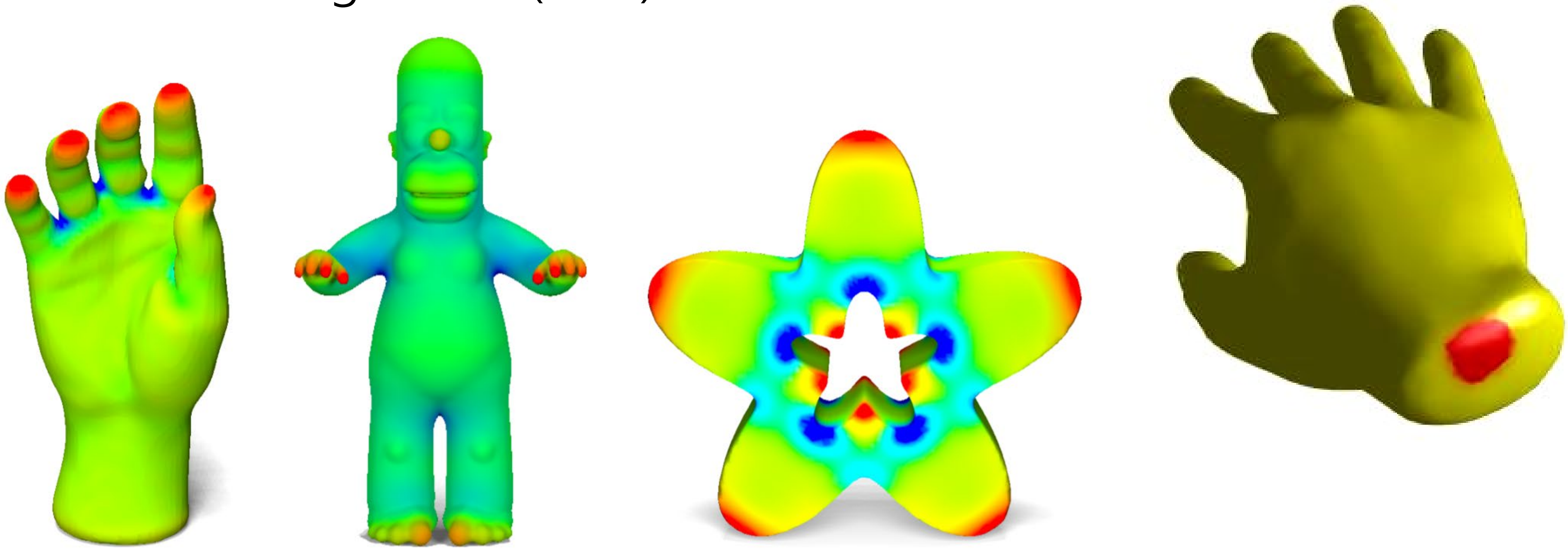
Nice results of semi-automatic methods, but:

- Full matching, what about partiality?
- What if we do not have landmarks?

Next: fully automatic correspondence

Descriptor Matching

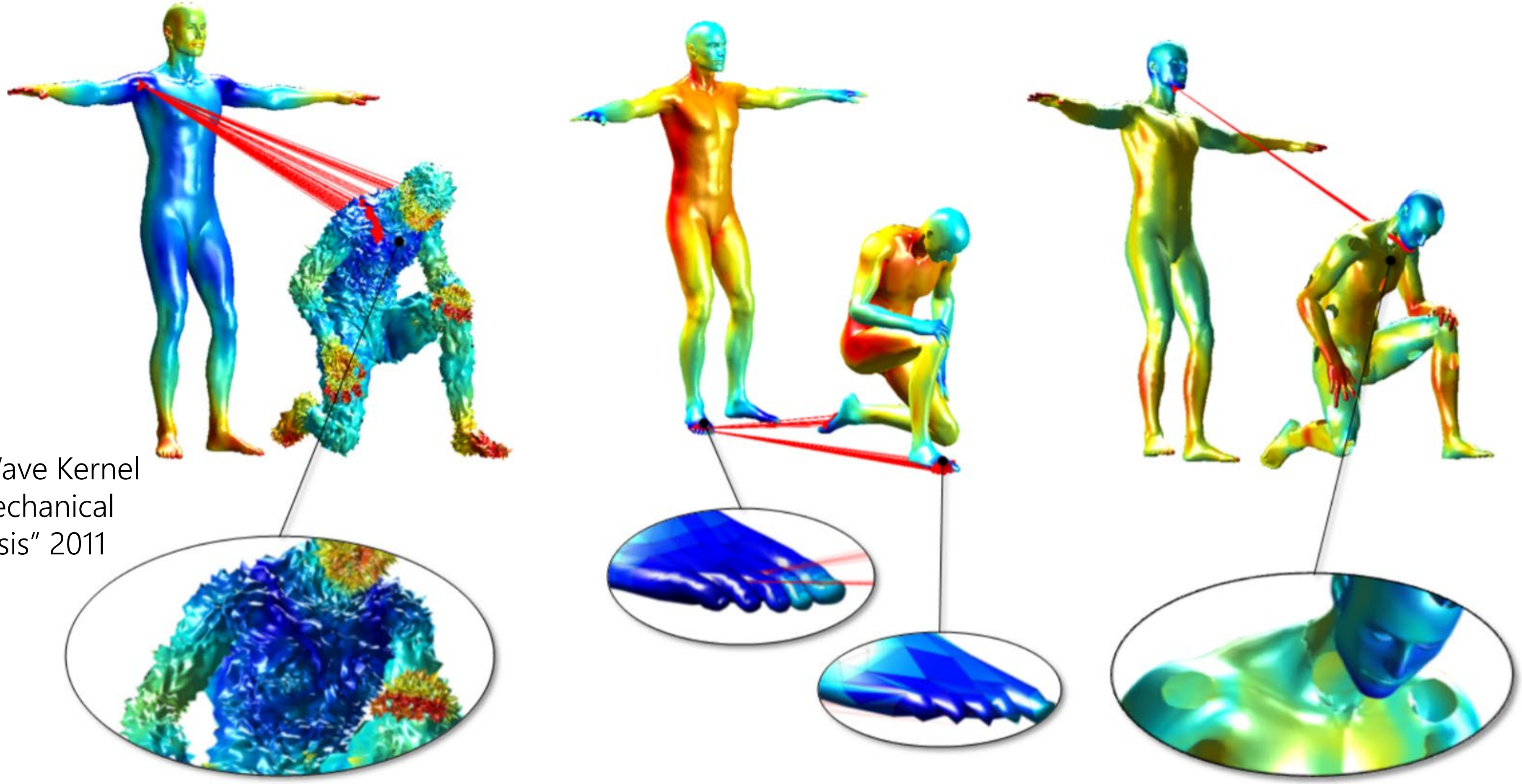
Heat Kernel Signature (HKS)



From: Sun et al. "A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion" 2009

Descriptor Matching

Wave Kernel Signature (WKS)



From: Aubry et al. “The Wave Kernel Signature: A Quantum Mechanical Approach to Shape Analysis” 2011

Descriptor Matching

What about non isometric shapes?

Various methods use deep learning to learn consistent descriptors

Image from: "Geodesic Convolutional Neural Networks on Riemannian Manifolds", Masci et al., 3DRR 2015

Descriptor Matching

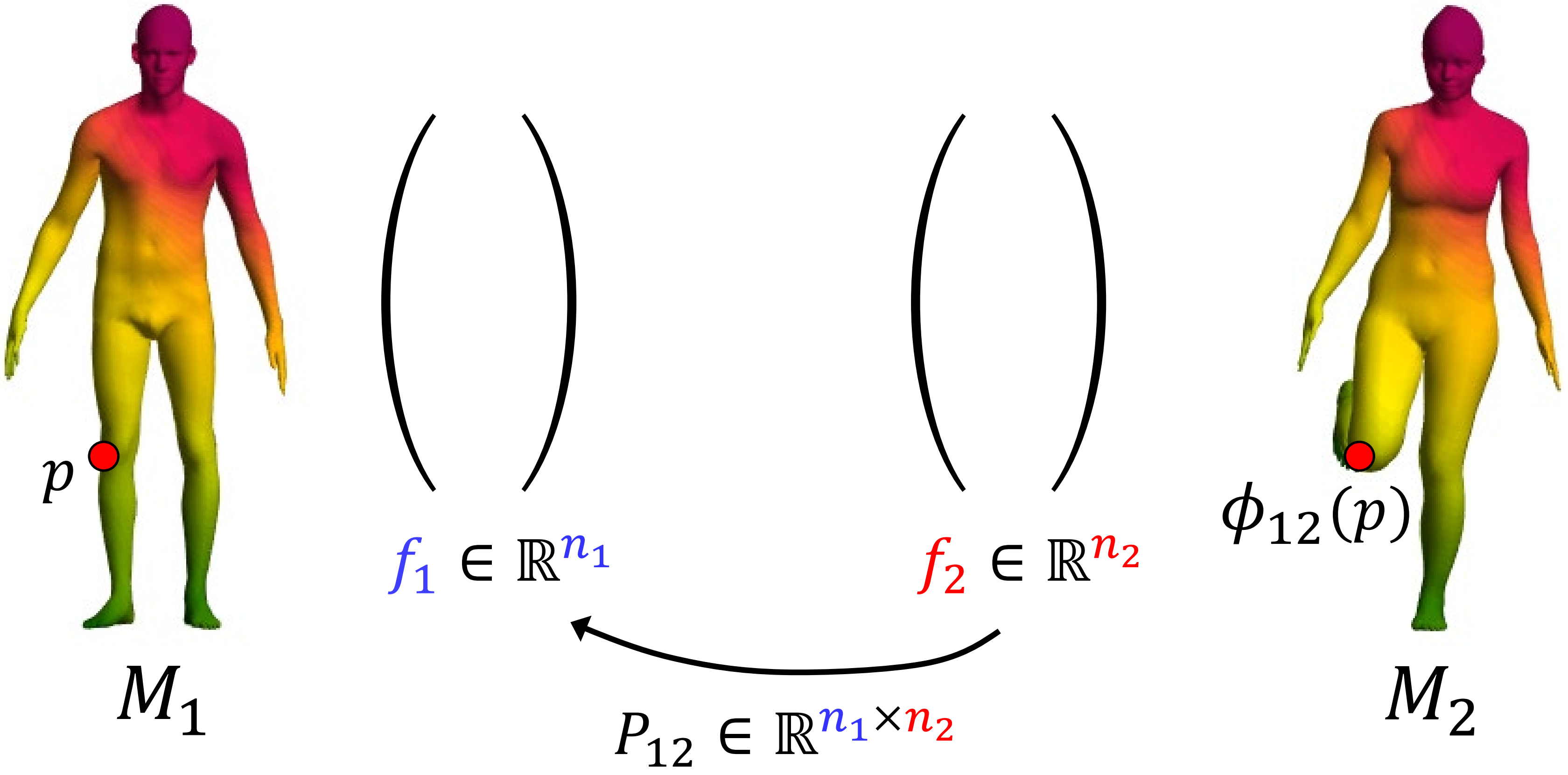
Say we computed matching descriptors, how do we compute correspondence?

- "The Wave Kernel Signature: A Quantum Mechanical Approach to Shape Analysis", Aubry et al. 2011: an iterative approach based on nearest neighbors in descriptor space

- "Efficient Deformable Shape Correspondence via Kernel Matching", Vestner et al., 3DV 2017: global combinatorial approach

Functional Maps

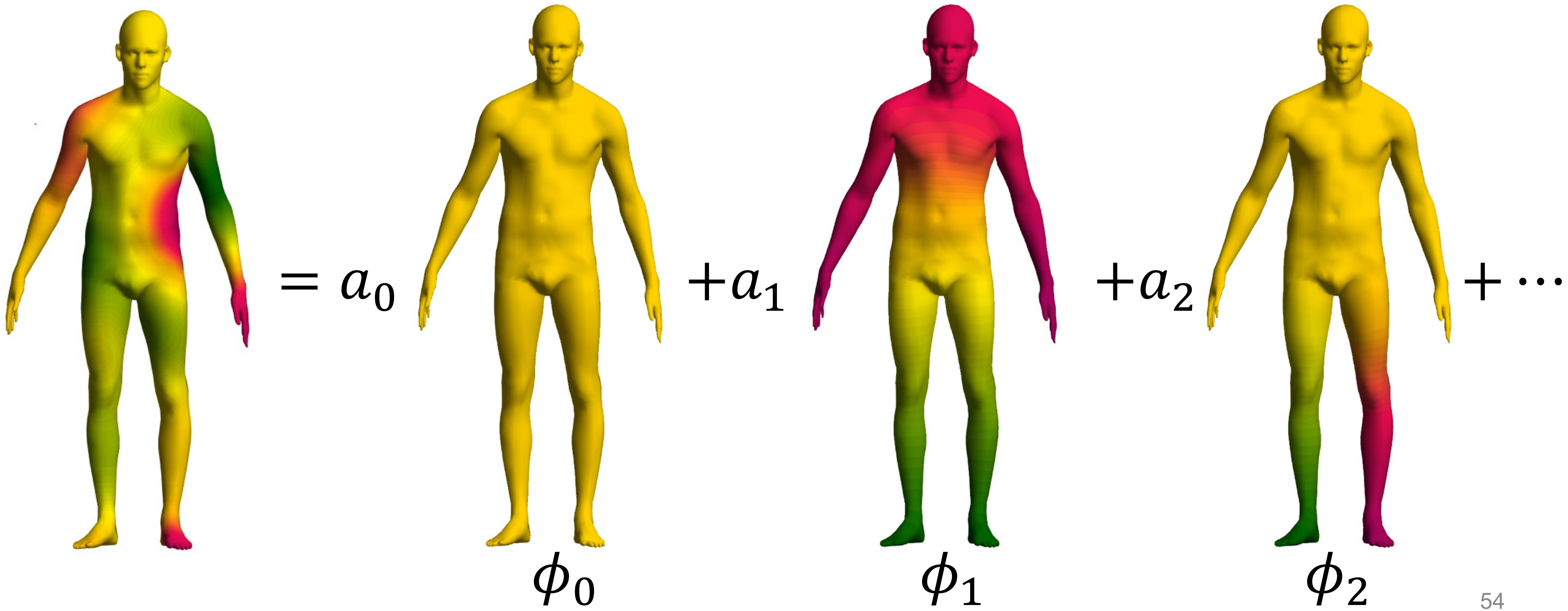
Function correspondence is linear



From: “Functional Maps: A Flexible Representation of Maps Between Shapes”, Ovsjanikov et al., SIGGRAPH 2012

Functional Maps

The eigenfunctions of the Laplace-Beltrami operator form the basis:



From: “Functional Maps: A Flexible Representation of Maps Between Shapes”, Ovsjanikov et al., SIGGRAPH 2012

Functional Maps – Reduced Basis

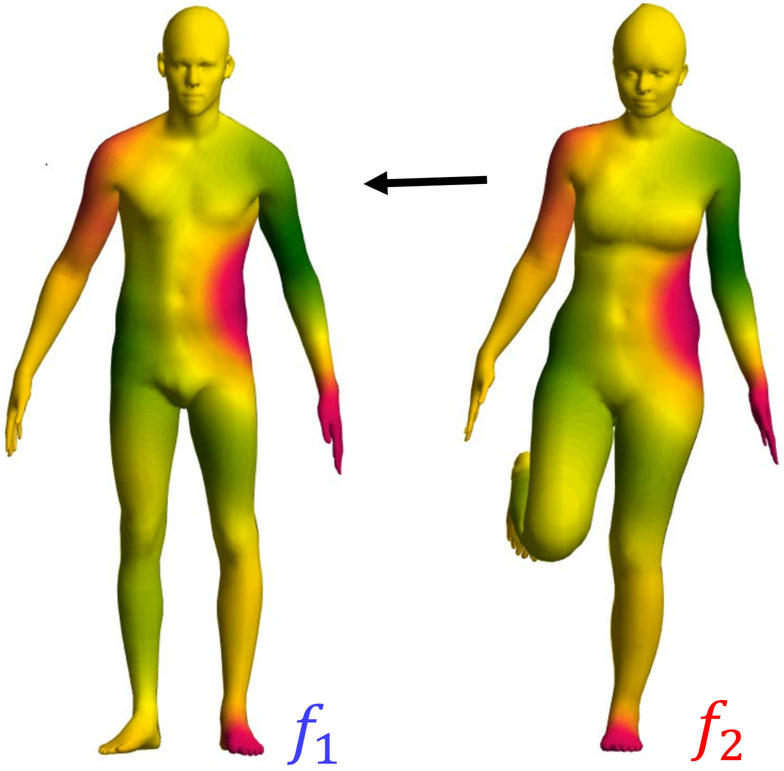
The **reduced** functional map is C_{12}

$$\begin{pmatrix} f_1 \end{pmatrix}_{k_1} = \begin{pmatrix} \quad C_{12} \quad \end{pmatrix}_{k_1 \times k_2} \begin{pmatrix} f_2 \end{pmatrix}_{k_2}$$

Coefficients of basis Φ_1

Change of basis matrix

Coefficients of basis Φ_2



From: "Functional Maps: A Flexible Representation of Maps Between Shapes", Ovsjanikov et al., SIGGRAPH 2012

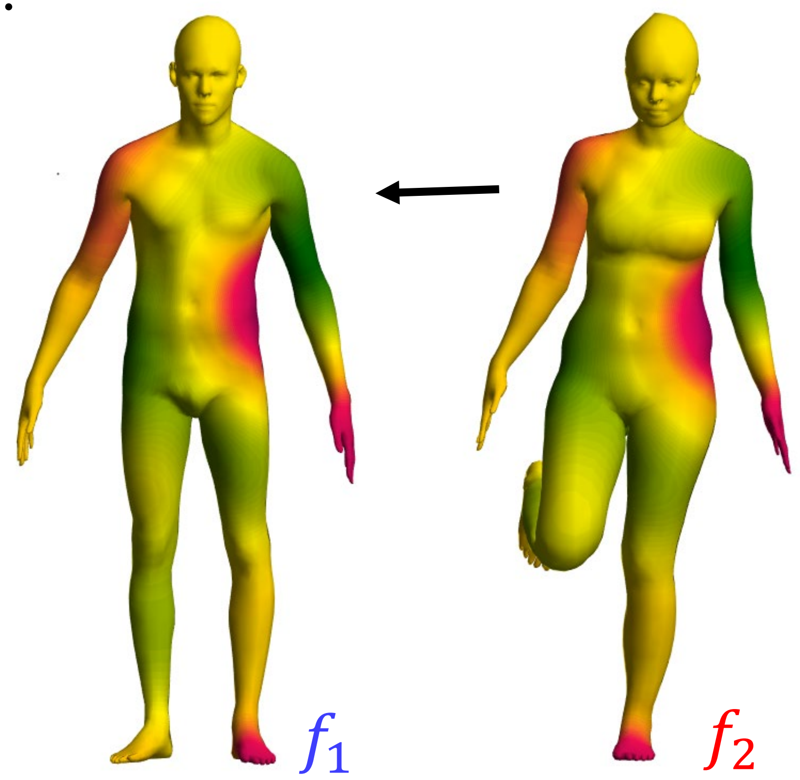
Functional Map Computation

Descriptor correspondence is easy to formulate:

$$argmin_{C_{12}} ||f_1 - C_{12} f_2||^2$$

$f_1 \in \mathbb{R}^{k_1}$ descriptor on M_1 (reduced basis coefficients)

$f_2 \in \mathbb{R}^{k_2}$ descriptor on M_2 (reduced basis coefficients)



From: "Functional Maps: A Flexible Representation of Maps Between Shapes", Ovsjanikov et al., SIGGRAPH 2012

Functional Map Computation

Laplacian commutativity promotes isometries:

$$argmin_{C_{12}} ||C_{12}\Delta_2 - \Delta_1 C_{12}||^2$$

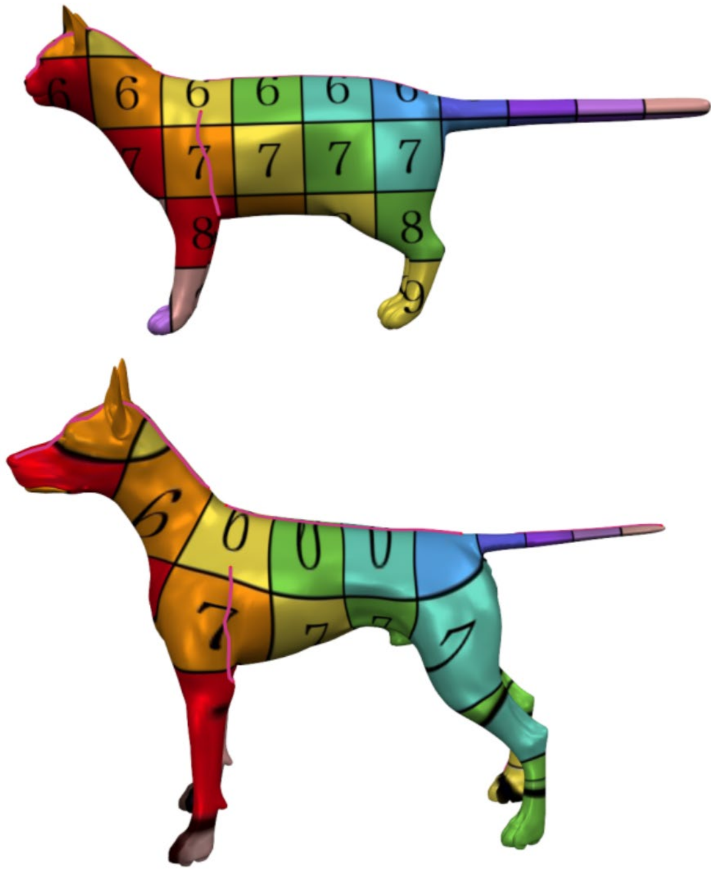
$\Delta_1 \in \mathbb{R}^{k_1 \times k_1}, \Delta_2 \in \mathbb{R}^{k_2 \times k_2}$ Laplace Beltrami operators, projected on the reduced bases Φ_1, Φ_2

From: "Functional Maps: A Flexible Representation of Maps Between Shapes", Ovsjanikov et al., SIGGRAPH 2012

Functional Map Computation

Laplacian commutativity promotes isometries:

$$argmin_{C_{12}} ||C_{12}\Delta_2 - \Delta_1 C_{12}||^2$$



"Interactive Curve Constrained Functional Maps",
Gehre et al., SGP 2018:
Curve constraints + Laplacian commutativity
Works well for non isometric shapes as well

Image from: "Interactive Curve Constrained Functional Maps", Gehre et al., SGP 2018

Functional Maps & Partiality

"Partial Functional Correspondence", Rodolà et al., SGP 2017:

The diagonal angle depends on the area ratio

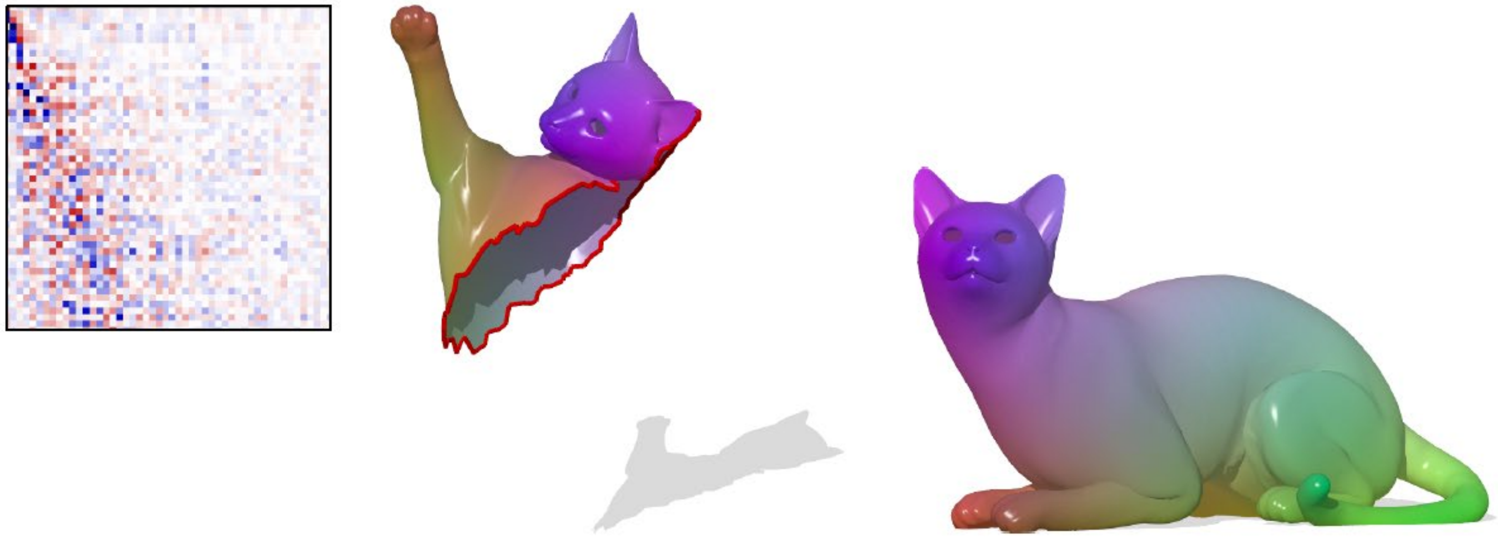


Image from: "Partial Functional Correspondence", Rodolà et al., SGP 2017

Functional Map Computation

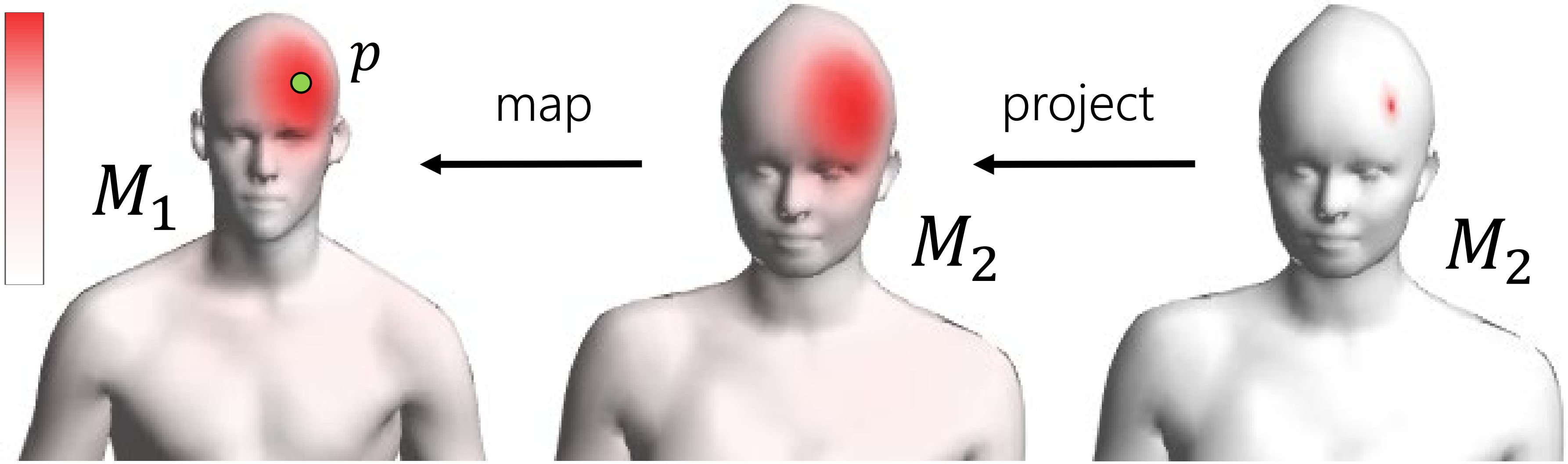
We computed a functional map, what’s next?

- For some applications we can use the functional map as is
  - For example, texture transfer just requires transferring functions
- For some applications we must **convert** the functional map to a pointwise map

Functional Map to Pointwise Map

A possible simple approach [Ovsjanikov et al. 2012]:

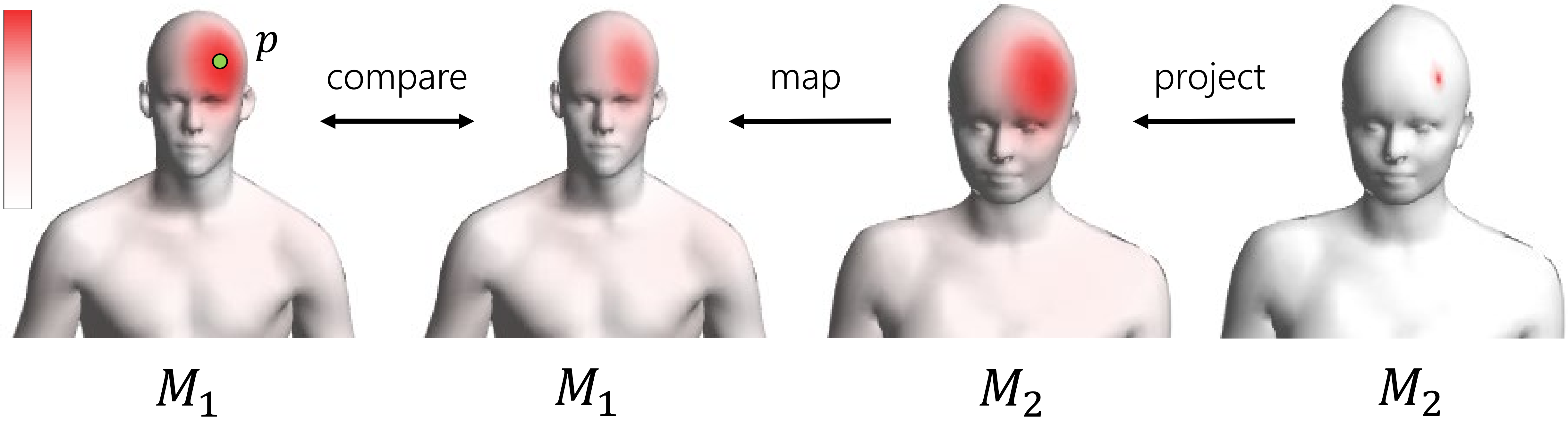
- For each vertex of M_2
  - Map an indicator function at this vertex to M_1
  - Find the vertex of M_1 where the value of the mapped function is maximal



Functional Map to Pointwise Map

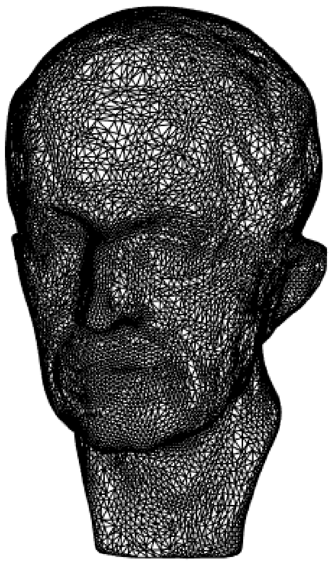
A better approach [Ovsjanikov et al. 2012]:

Compute indicator functions on both shapes, compare

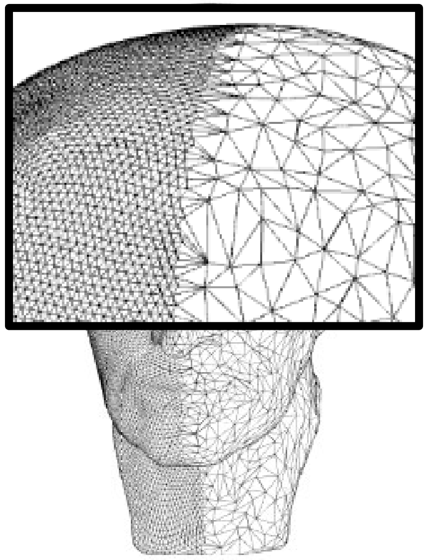


Functional Map to Pointwise Map

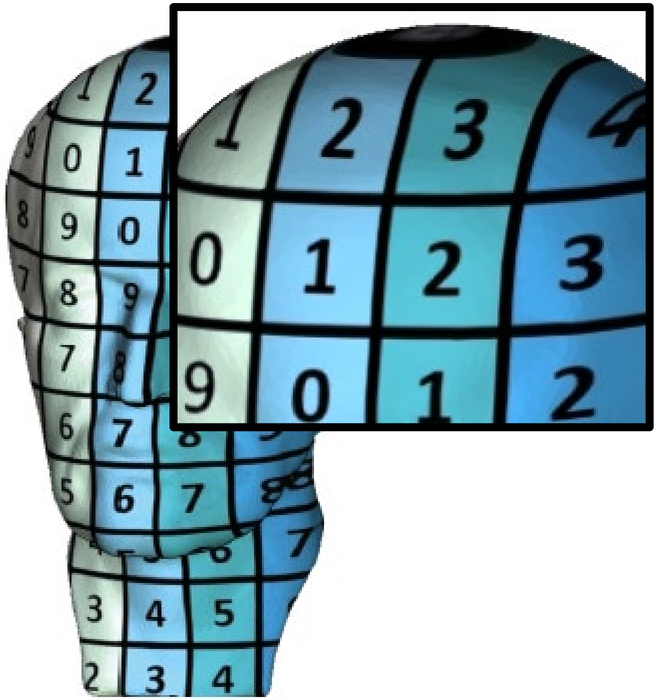
Finding nearest indicator function generates vertex-to-vertex output



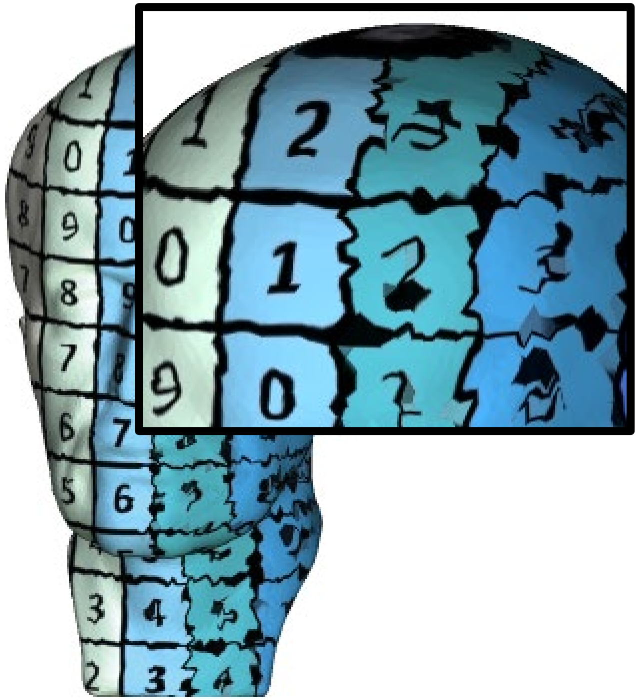
M_1



M_2



Target



Vertex-to-vertex

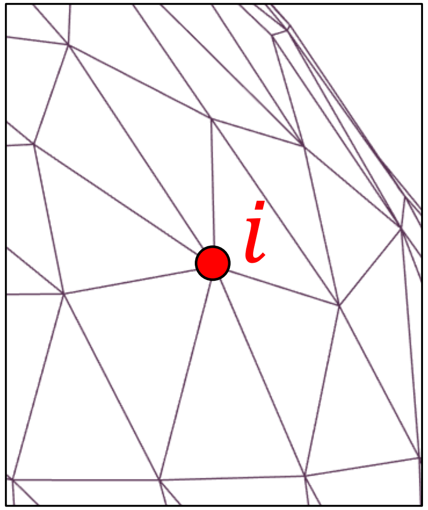
From: “Deblurring and Denoising of Maps between Shapes”, Ezuz & Ben-Chen, SGP 2017

Functional Map to Pointwise Map

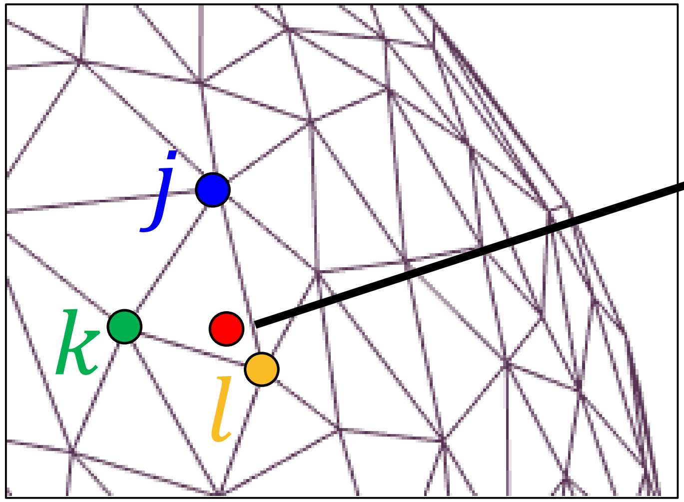
Finding nearest indicator function generates vertex-to-vertex output

"Deblurring and Denoising of Maps between Shapes", Ezuz & Ben-Chen, SGP 2017:

Instead of nearest neighbors among vertices only, find nearest neighbors on faces



M_1



M_2

Linear combination of indicator functions at vertices

Functional Map to Pointwise Map

An algebraic approach:

$$\underset{P_{12}\in S}{\arg\min} \left\| C_{12} - \Phi_1^{\dagger} P_{12} \Phi_2 \right\|_F^2$$

$S = \{$*functional maps that uniquely define pointwise maps*$\}$

Difficult to solve, usually underconstrained

From: "Deblurring and Denoising of Maps between Shapes", Ezuz & Ben-Chen, SGP 2017

Functional Map to Pointwise Map

We add a regularizer:

$$P_{12}^{*} = \arg\min_{P_{12} \in S} \left\| C_{12} - \Phi_1^{\dagger} P_{12} \Phi_2 \right\|_F^2 \quad + R(P_{12})$$

Should guarantee:

- Unique global minimizer
- Efficient optimization
- Maps with “good” properties

From: “Deblurring and Denoising of Maps between Shapes”, Ezuz & Ben-Chen, SGP 2017

Functional Map to Pointwise Map

We add a regularizer:

$$P_{12}^{*} = \arg\min_{P_{12}\in S} \underbrace{\left\| C_{12} - \Phi_1^{\dagger} P_{12} \Phi_2 \right\|_F^2}_{} + \underbrace{R(P_{12})}_{}$$

Constrains the projection of $P_{12}\Phi_2$ on Φ_1

Should constrain the projection of $P_{12}\Phi_2$ on the **orthogonal complement** of Φ_1

Should guarantee:

- **Unique global minimizer**
- Efficient optimization
- Maps with "good" properties

From: "Deblurring and Denoising of Maps between Shapes", Ezuz & Ben-Chen, SGP 2017

Functional Map to Pointwise Map

We add a regularizer:

$$P_{12}^{*} = \arg\min_{P_{12} \in S} \left\| C_{12} - \Phi_1^{\dagger} P_{12} \Phi_2 \right\|_F^2 \quad + \underbrace{R(P_{12})}$$

Should constrain the projection of $P_{12}\Phi_2$ on the **orthogonal complement** of Φ_1

As small as possible

Should guarantee:

- Unique global minimizer
- Efficient optimization
- **Maps with "good" properties**

From: "Deblurring and Denoising of Maps between Shapes", Ezuz & Ben-Chen, SGP 2017

Functional Map to Pointwise Map

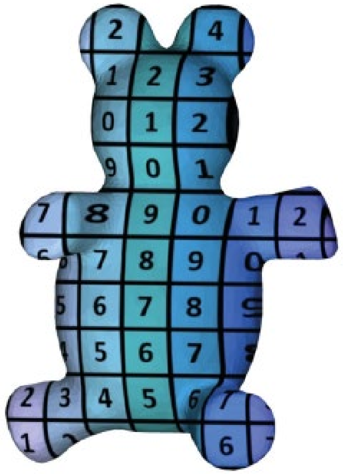
Regularize to favor smoother maps: $P_{12}\Phi_2 \in span(\Phi_1)$

$$P_{12}^{*} = \arg\min_{P_{12}\in S} \underbrace{\left\| C_{12} - \Phi_1^{\dagger} P_{12}\Phi_2 \right\|_F^2}_{\text{Similarity term}} + \underbrace{\left\| \left(Id - \Phi_1\Phi_1^{\dagger}\right) P_{12}\Phi_2 \right\|_{M_1}^2}_{\text{Smoothness term}}$$

- Unique minimizer
- Efficient minimization
- Pointwise map does not induce spurious high frequencies

From: “Deblurring and Denoising of Maps between Shapes”, Ezuz & Ben-Chen, SGP 2017

Functional Map to Pointwise Map

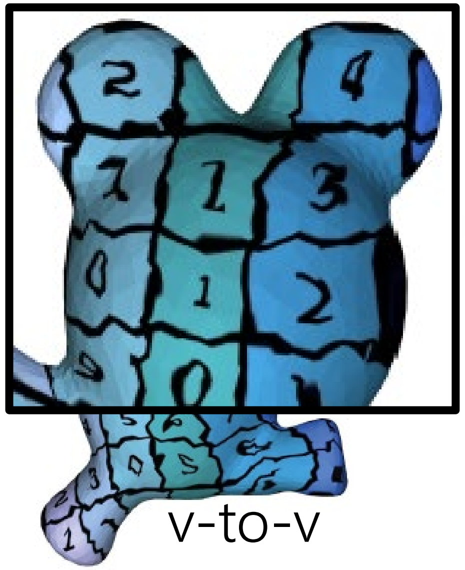


Target

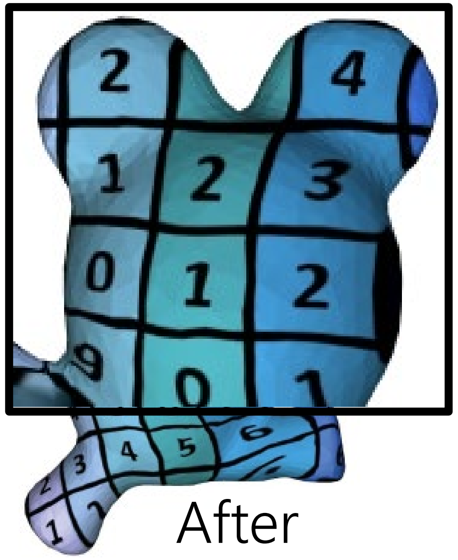
Useful even for pointwise maps!

Given a noisy pointwise map:

- Convert it to a functional map (project on reduced bases)
  - Removes high frequencies
- Convert back to a pointwise map



v-to-v
(ICSKM)



After
“denoising”

From: “Deblurring and Denoising of Maps between Shapes”,
Ezuz & Ben-Chen, SGP 2017

Functional Map Summary

Functional map computation is efficient, but relation to discrete pointwise maps is not completely clear

What subset of functional maps actually represents pointwise maps?

Functional Map Summary

"Informative Descriptor Preservation via Commutativity for Shape Matching",

Nogneng & Ovsjanikov, Eurographics 2017:

A functional map corresponds to a pointwise map iff it preserves pointwise products:

$$Cf \otimes Ch = C(f \otimes h)$$

Nogneng & Ovsjanikov used soft commutativity constraints

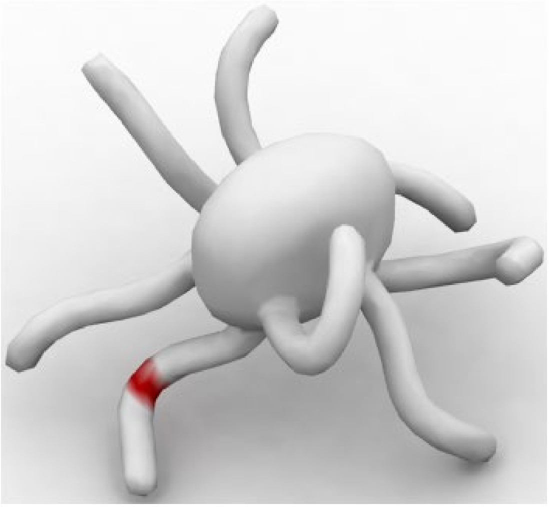
Can we formulate hard constraints for a functional map to correspond to pointwise?

Soft (Fuzzy) Maps

Compute the **probability** that a pair of points correspond

M_1

M_2, probability to be matched to p_0

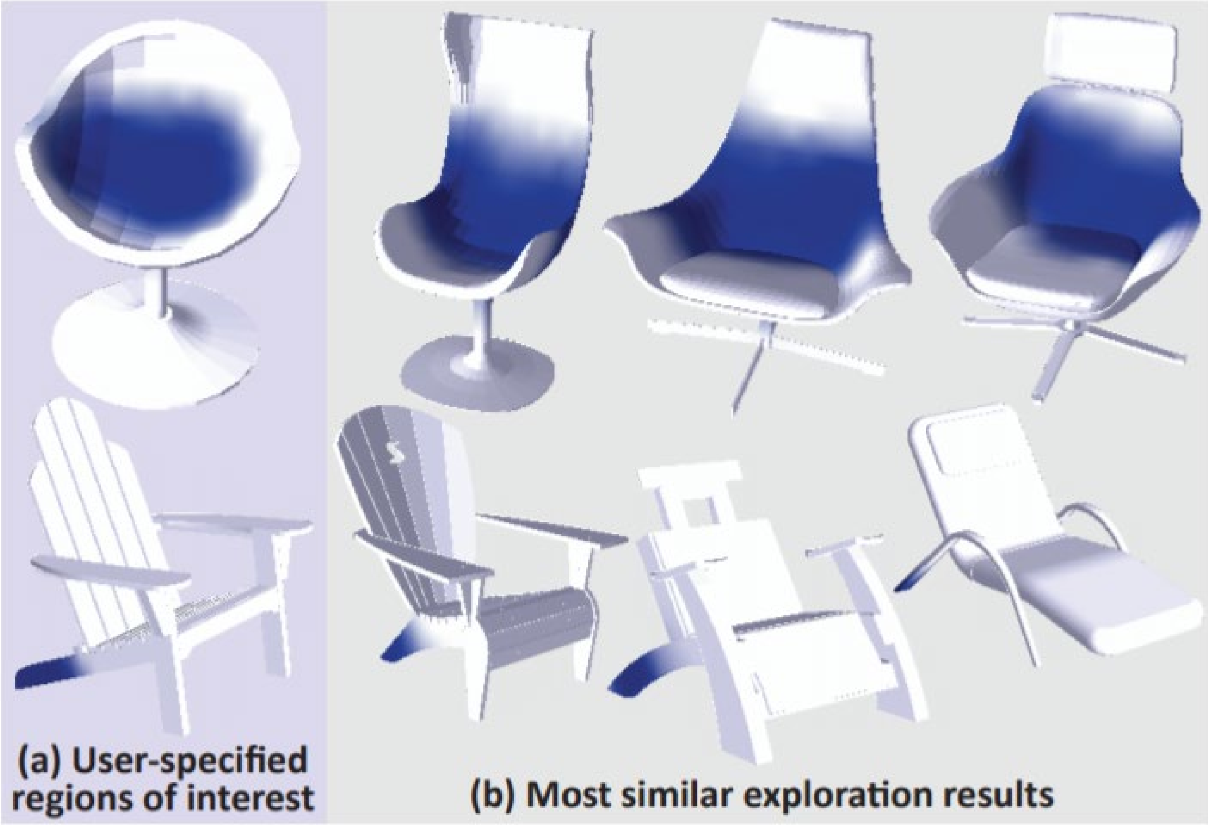


M_2, probability to be matched to q_0

From: Solomon et al. “Entropic Metric Alignment for Correspondence Problems” 2016

Soft (Fuzzy) Maps

“Exploring Collections of 3D Models using Fuzzy Correspondences”, Kim et al., SIGGRAPH 2012:



Soft (Fuzzy) Maps

Set of possible fuzzy maps:

$$\bar{\mathcal{M}}(\mu_1,\mu_2) = \left\{\Gamma \in \mathbb{R}_+^{n_1\times n_2} : \Gamma\mu_2 = \mathbf{1} \qquad and \qquad \mu_1^T\Gamma = \mathbf{1}^T\right\}$$

$$\mathrm{n}_1 \underbrace{\begin{pmatrix} — & \Gamma(p_1) & — \end{pmatrix}}_{n_2} \begin{pmatrix} | \\ \mu_2 \\ | \end{pmatrix} \qquad \begin{pmatrix} — & \mu_1^T & — \end{pmatrix} \underbrace{\begin{pmatrix} | \\ \Gamma(p_2) \\ | \end{pmatrix}}_{n_2} \mathrm{n}_1$$

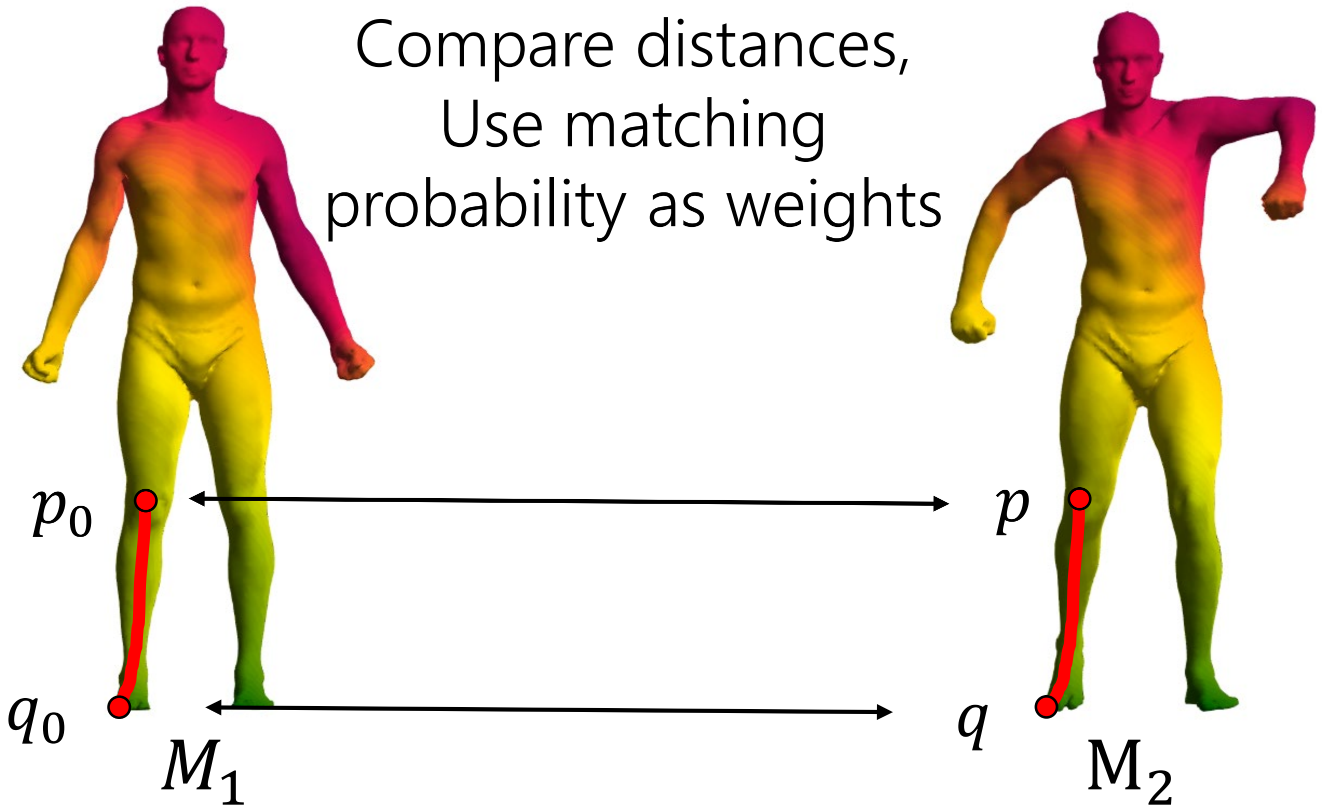
$\mu_1 \in \mathbb{R}^{n_1}, \mu_2 \in \mathbb{R}^{n_2}$ vertex areas

From: Solomon et al. "Entropic Metric Alignment for Correspondence Problems" 2016

Matric Alignment

$$GW(\Gamma, D^1, D^2) = \underset{\Gamma \in \bar{\mathcal{M}}}{\arg\min} \sum_{ijkl} \left(\boldsymbol{D}^{\boldsymbol{1}}_{ij} - \boldsymbol{D}^{\boldsymbol{2}}_{kl}\right)^2 \Gamma_{ik}\Gamma_{jl}$$

Compare distances,
Use matching
probability as weights



From: Solomon et al. “Entropic Metric Alignment for Correspondence Problems” 2016

Soft Map Entropy

$$H(\Gamma) = -\sum_{ik} \Gamma_{ik} \ln(\Gamma_{ik})$$

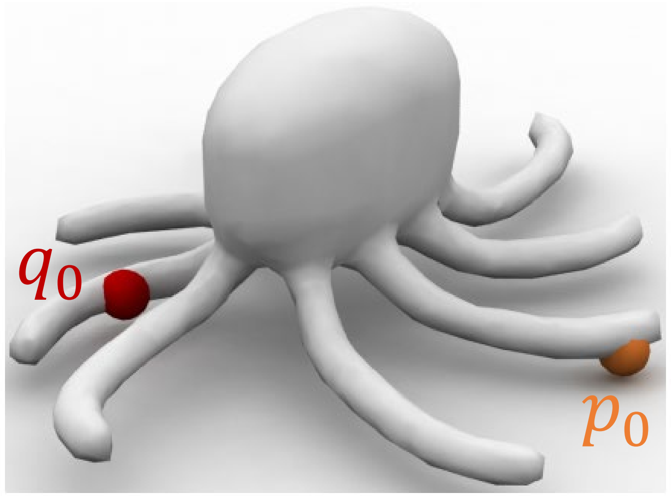
Small entropy ⟶ Γ is sparse (close to a permutation)

Fuzziness ⟶ faster convergence

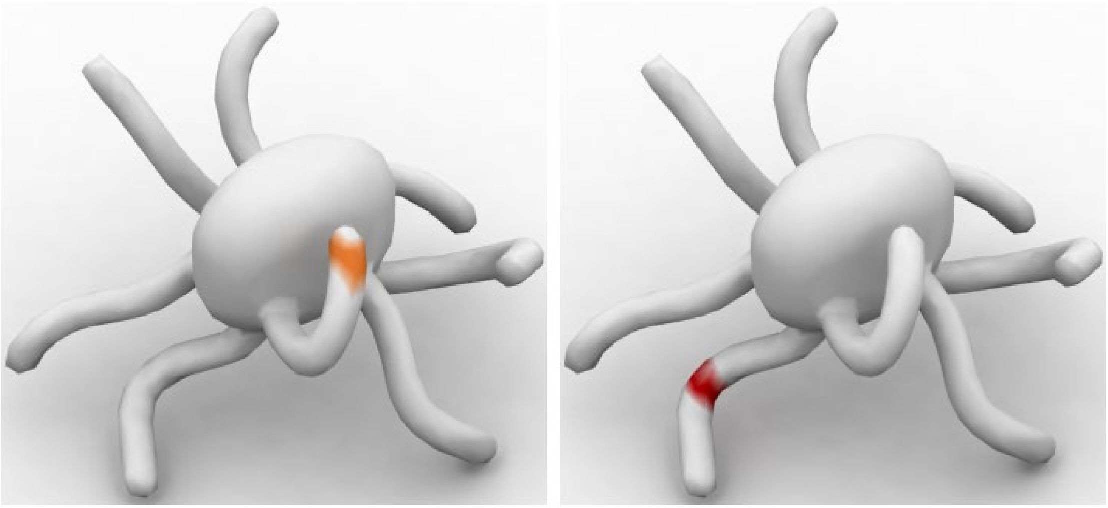
$$\arg\min_{\Gamma \in \bar{\mathcal{M}}} GW\left(\Gamma, D^1, D^2\right) - \alpha H(\Gamma)$$

From: Solomon et al. "Entropic Metric Alignment for Correspondence Problems" 2016

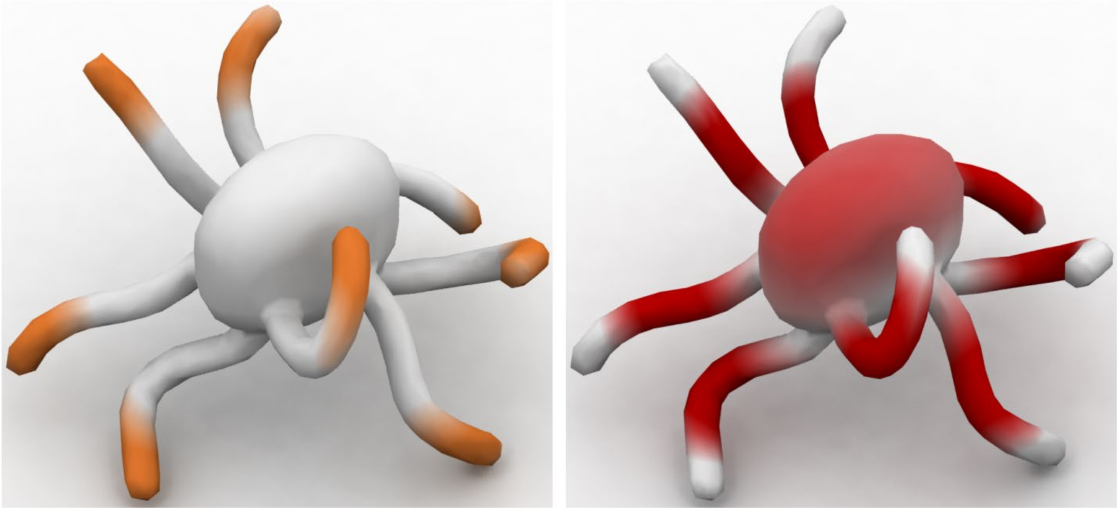
Parameter Selection



$\alpha = 7 \cdot 10^{-4}$

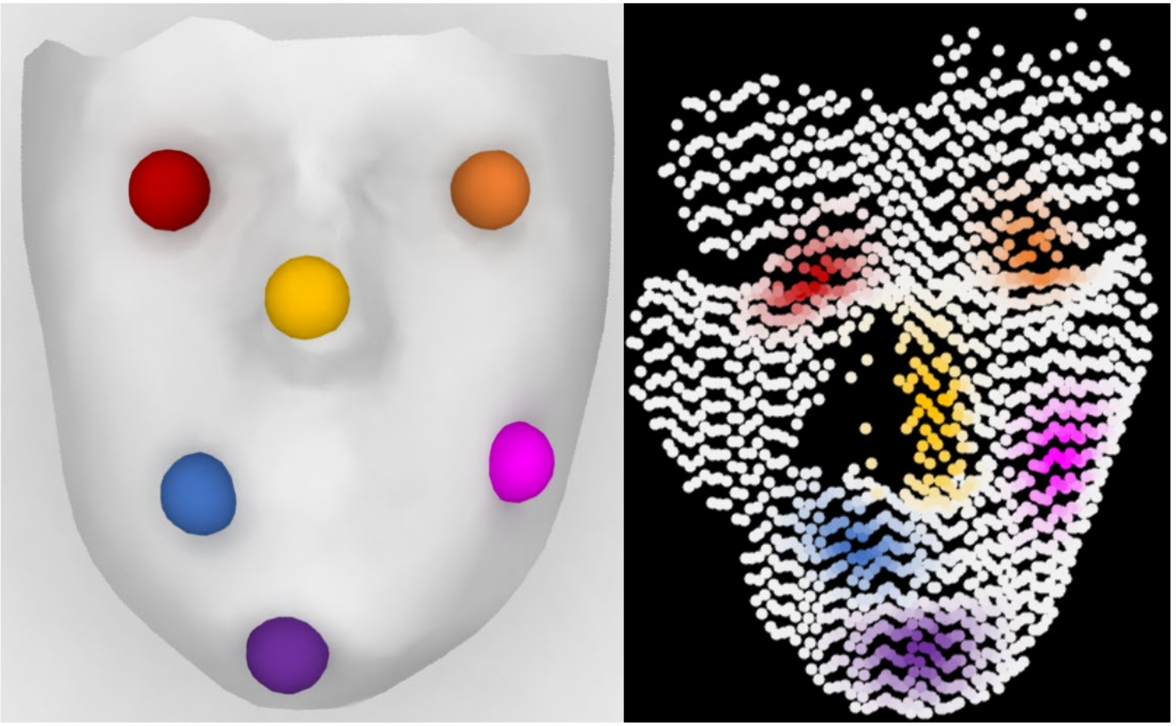


$\alpha = 8 \cdot 10^{-3}$

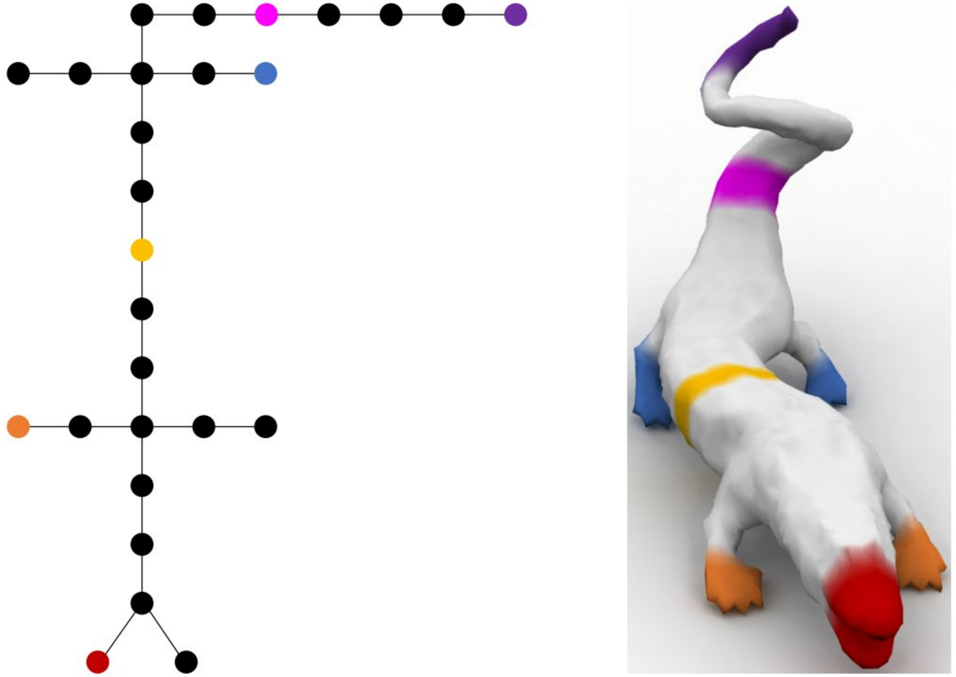


From: Solomon et al. “Entropic Metric Alignment for Correspondence Problems” 2016

Matric Alignment



Mesh-point cloud correspondence



Mesh-graph correspondence

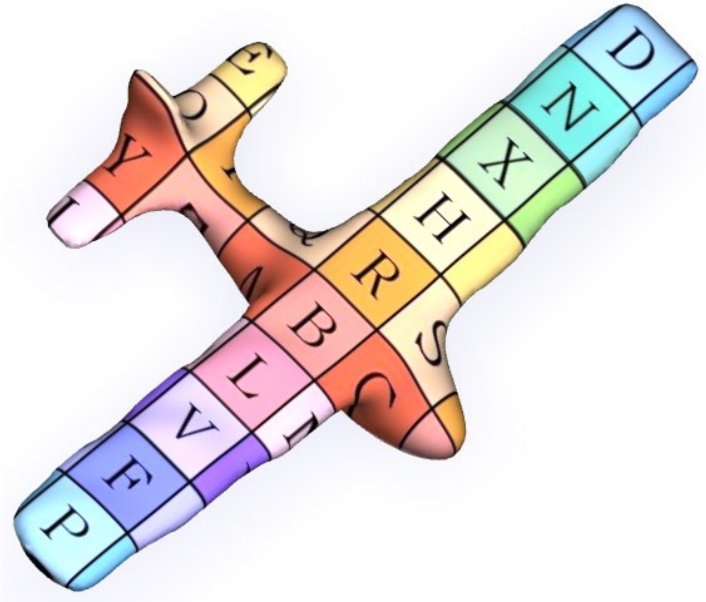
From: Solomon et al. “Entropic Metric Alignment for Correspondence Problems” 2016

Conclusion

- There are MANY different approaches to compute correspondence
- The discretization is crucial
  - Continuous vs. combinatorial
  - Output accuracy
  - Intermediate discretizations (e.g. functional maps) are useful

Conclusion

- There are MANY different approaches to compute correspondence
- More categories:
  - Volumetric correspondence
  - Partial correspondence
  - Machine learning methods



Thank you!

Questions?