Some recent methods in non-rigid shape matching, with and without learning

GAMES 2019 webinar

Maks Ovsjanikov

Based on joint work with: *E. Corman, Michael Bronstein, Emanuele Rodolà, Justin Solomon, Adrien Butscher, Mirela Ben-Chen, Leonidas Guibas, Frederic Chazal ….*

Laboratoire d'Informatique de l'École polytechnique
ÉCOLE POLYTECHNIQUE
UNIVERSITÉ PARIS-SACLAY
cnrs

My Background

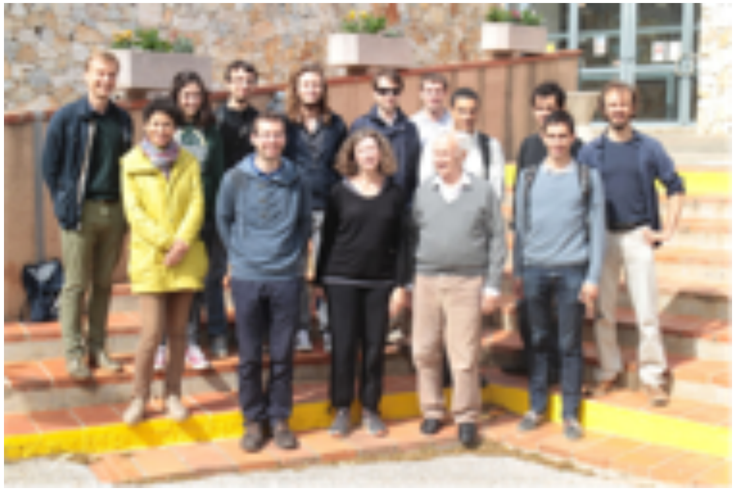
- 2005 – 2010: PhD from Stanford University (advisor Leonidas Guibas).
- 2011: engineer at Google Inc.
- Since 2012 Professor in the Computer Science Department at Ecole Polytechnique in France.

University

- Ecole Polytechnique: located very near Paris.
- Ranked 2nd best small university in the world in 2019.
- Est. 1794. Students and professors such as Ampère, Cauchy, Fourier, Hermite, Lagrange, Monge, Poincaré, Poisson…
- Very international campus.

Our group

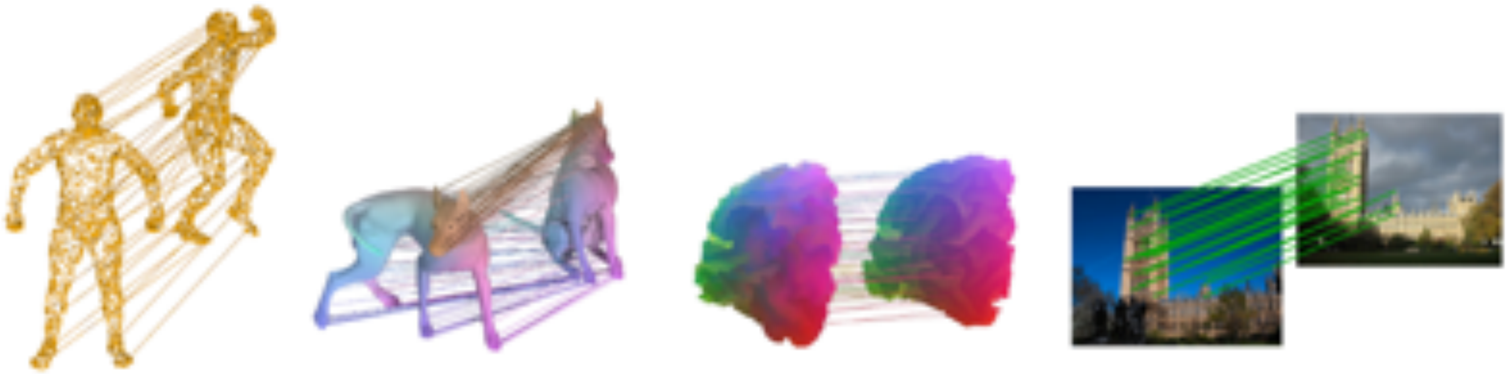
- Currently 5 PhD students and 2 PostDocs.
- Part of the larger STREAM team dedicated to visual computing with 3 other professors.
- Many international collaborations: Stanford, MIT, UCL, KAUST, Univ. Toronto, Univ. Rome, etc.
- Funding for **PhD students and postdocs !**



General Overview

Overall Objective:

Create tools for computing and analyzing *mappings* between geometric objects.



Talk Overview

- [Related] Course Website:

  http://www.lix.polytechnique.fr/~maks/fmaps_SIG17_course/

  or http://bit.do/fmaps2017

- Course Notes:

  Linked from the website. Or use http://bit.do/fmaps2017_notes
  Attention: (significantly) more material than in the lectures

- Sample Code:

  See **Sample Code** link on the website.

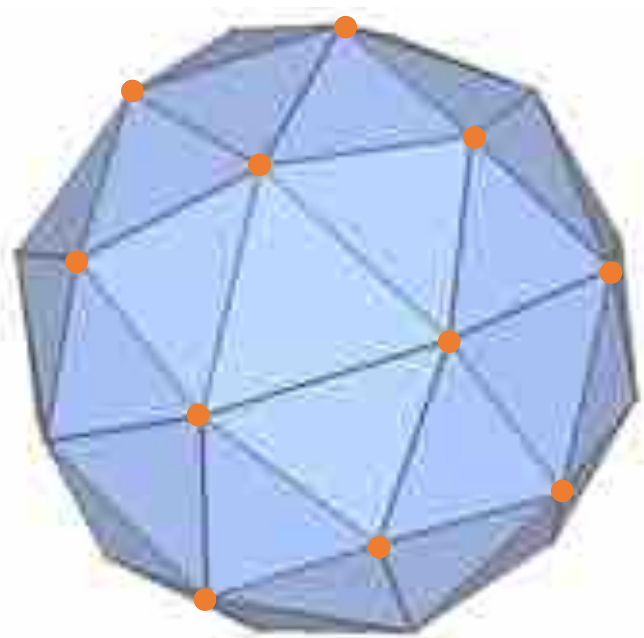
  Demo code for basic operations.

Talk Overview

- Motivation and Problem Taxonomy
- Rigid Matching: ICP
- Functional Map representation, properties
- Basic pipeline for non-rigid matching
- Recent extensions, improvements
- Open problems, Q&A

What is a Shape?

- Continuous: a surface embedded in 3D.
- Discrete: a graph embedded in 3D (triangle mesh).

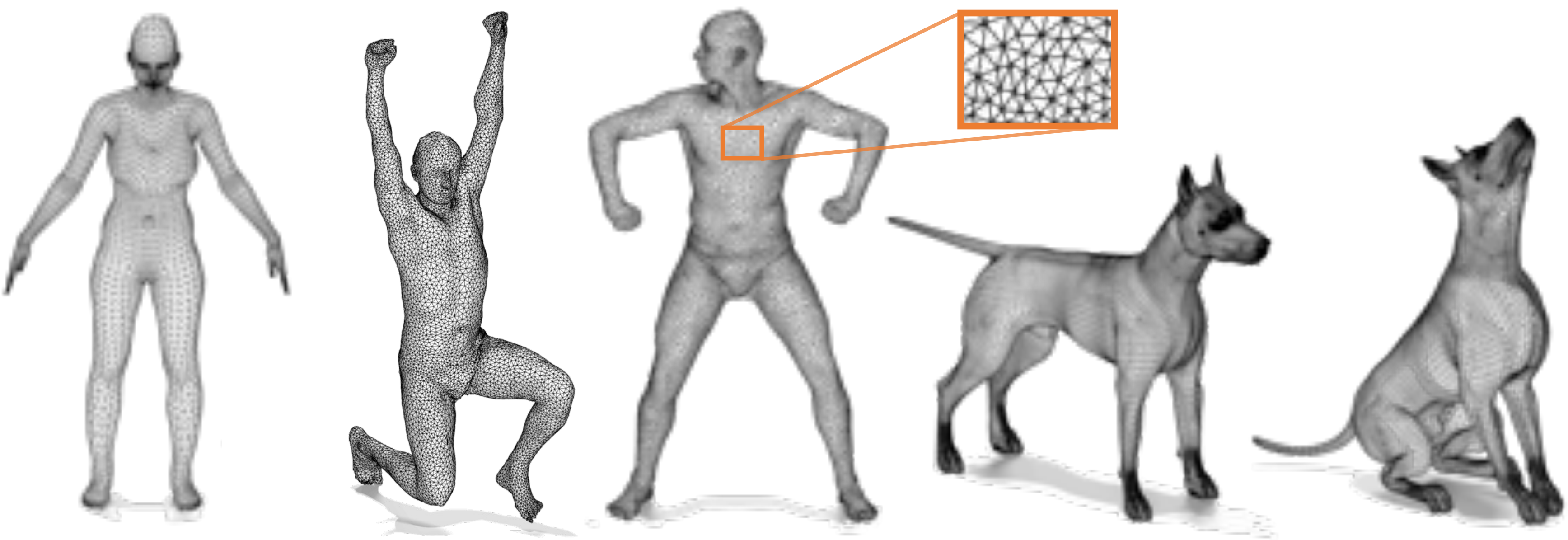


Common assumptions:

- Connected.
- Manifold.
- Without Boundary.

What is a Shape?

- Continuous: a surface embedded in 3D.
- Discrete: a graph embedded in 3D (triangle mesh).



5k – 200k triangles

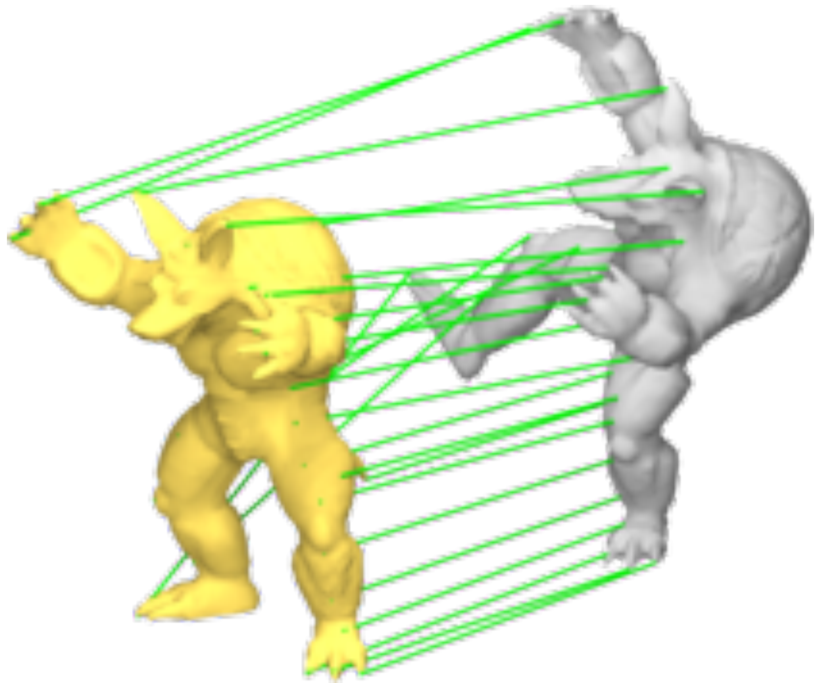
Shapes from the FAUST, SCAPE, and TOSCA datasets

Overall Goal

- Given two shapes, find **correspondences** between them.

Overall Goal

- Given two shapes, find **correspondences** between them.



- Finding the **best** map between a pair of shapes.

Problem Taxonomy

Local vs. Global

refinement (e.g. ICP) | alignment (search)

Rigid vs. Deformable

rotation, translation | general deformation

Semi vs. Fully Automatic

given landmarks | a priori model

Learning-Based vs. Direct

known examples | unseen data

Problem Taxonomy

Local vs. $\boxed{\text{Global}}$

refinement (e.g. ICP) | alignment (search)

Rigid vs. $\boxed{\text{Deformable}}$

rotation, translation | general deformation

Semi vs. $\boxed{\text{Fully Automatic}}$

given landmarks | a priori model

$\boxed{\text{Learning-Based vs. Direct}}$

known examples | unseen data

Why Shape Matching?

Given a correspondence, we can transfer:

texture and
parametrization

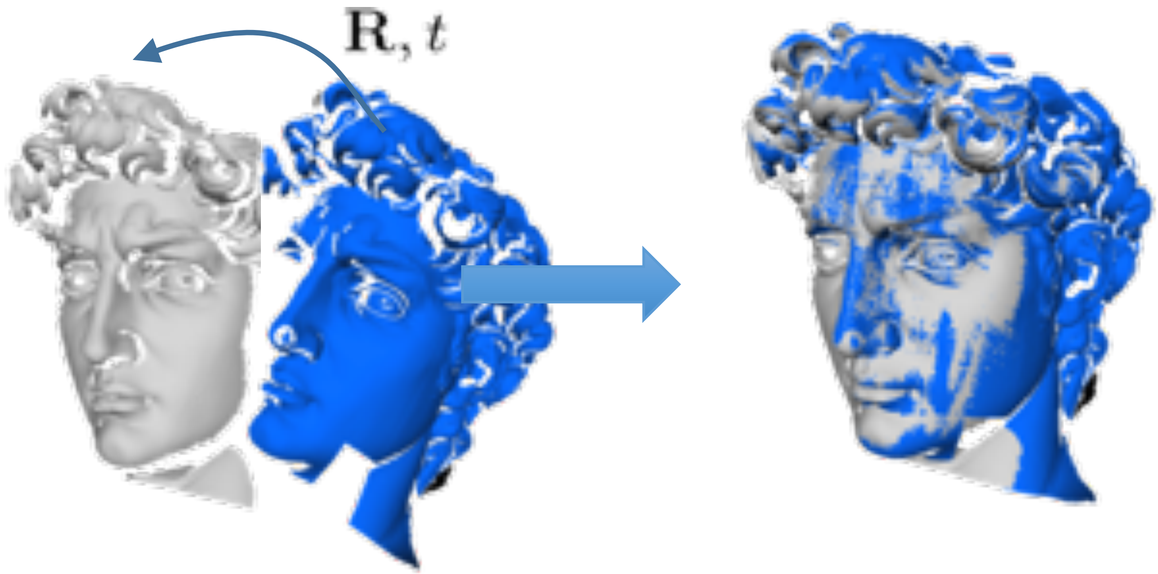
segmentation and labels

deformation

Sumner et al. ‘04.

Other applications: shape interpolation, reconstruction ...

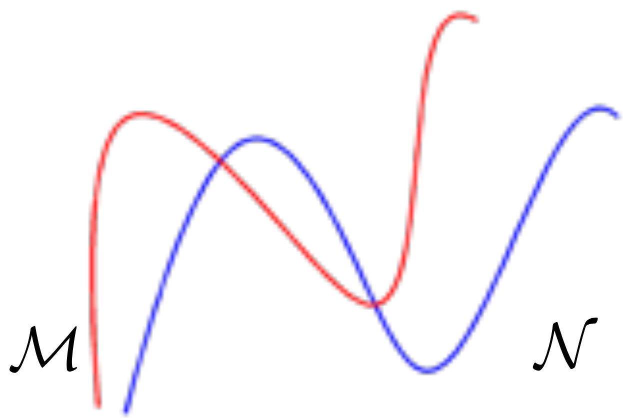
Rigid Shape Matching



- Given a pair of shapes, find the optimal *Rigid Alignment* between them.
- The unknowns are the rotation/translation parameters of the source onto the target shape.

Iterative Closest Point (ICP)

- Classical approach: iterate between finding correspondences and finding the transformation:



example in 2D

Besl, McKay (1992). "A Method for Registration of 3-D Shapes".

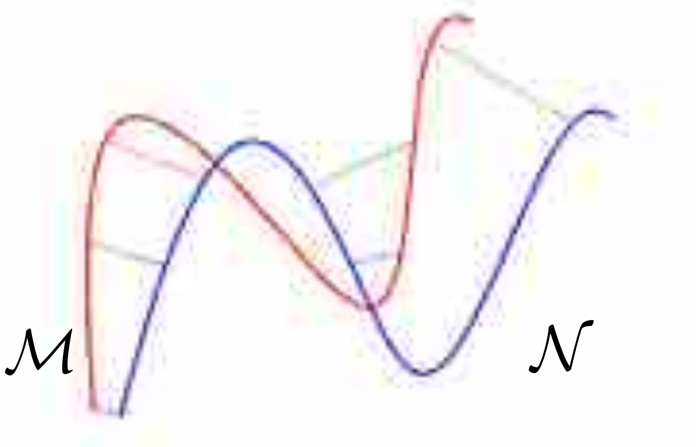
Given a pair of shapes, $\mathcal{M}$ and $\mathcal{N}$, iterate:

1. For each $x_i \in \mathcal{M}$ find **nearest** neighbor $y_i \in \mathcal{N}$.
2. Find optimal transformation $\mathbf{R}, t$ minimizing:

$$\arg\min_{R,t} \sum_i \|Rx_i + t - y_i\|_2^2$$

Iterative Closest Point

- Classical approach: iterate between finding correspondences and finding the transformation:



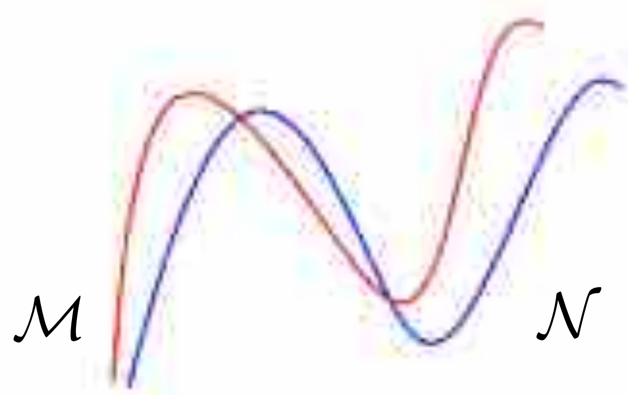
Given a pair of shapes, $\mathcal{M}$ and $\mathcal{N}$, iterate:

1. For each $x_i \in \mathcal{M}$ find **nearest** neighbor $y_i \in \mathcal{N}$.
2. Find optimal transformation $\mathbf{R}, t$ minimizing:

$$\arg\min_{R,t} \sum_i \|Rx_i + t - y_i\|_2^2$$

Iterative Closest Point

- Classical approach: iterate between finding correspondences and finding the transformation:



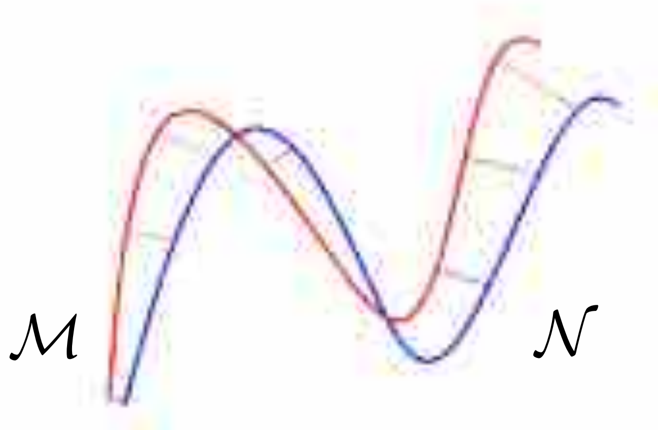
Given a pair of shapes, $\mathcal{M}$ and $\mathcal{N}$, iterate:

1. For each $x_i \in \mathcal{M}$ find **nearest** neighbor $y_i \in \mathcal{N}$.
2. Find optimal transformation $\mathbf{R}, t$ minimizing:

$$\arg\min_{R,t} \sum_i \|Rx_i + t - y_i\|_2^2$$

Iterative Closest Point

- Classical approach: iterate between finding correspondences and finding the transformation:



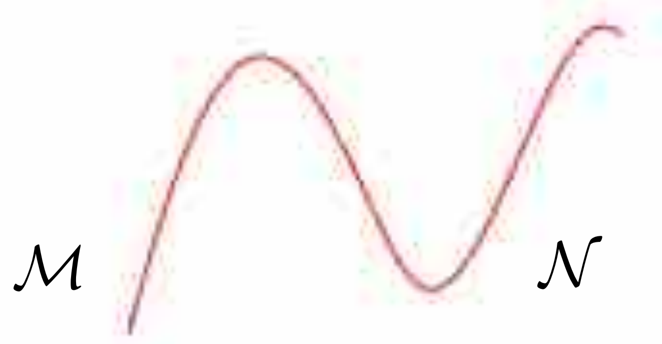
Given a pair of shapes, $\mathcal{M}$ and $\mathcal{N}$, iterate:

1. For each $x_i \in \mathcal{M}$ find **nearest** neighbor $y_i \in \mathcal{N}$.
2. Find optimal transformation $\mathbf{R}, t$ minimizing:

$$\arg\min_{R,t} \sum_i \|Rx_i + t - y_i\|_2^2$$

Iterative Closest Point

- Classical approach: iterate between finding correspondences and finding the transformation:



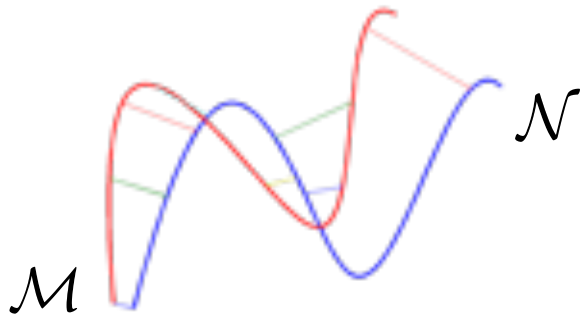
Given a pair of shapes, $\mathcal{M}$ and $\mathcal{N}$, iterate:

1. For each $x_i \in \mathcal{M}$ find **nearest** neighbor $y_i \in \mathcal{N}$.
2. Find optimal transformation $\mathbf{R}, t$ minimizing:

$$\arg\min_{R,t} \sum_i \|Rx_i + t - y_i\|_2^2$$

Iterative Closest Point

- Classical approach: iterate between finding correspondences and finding the transformation:



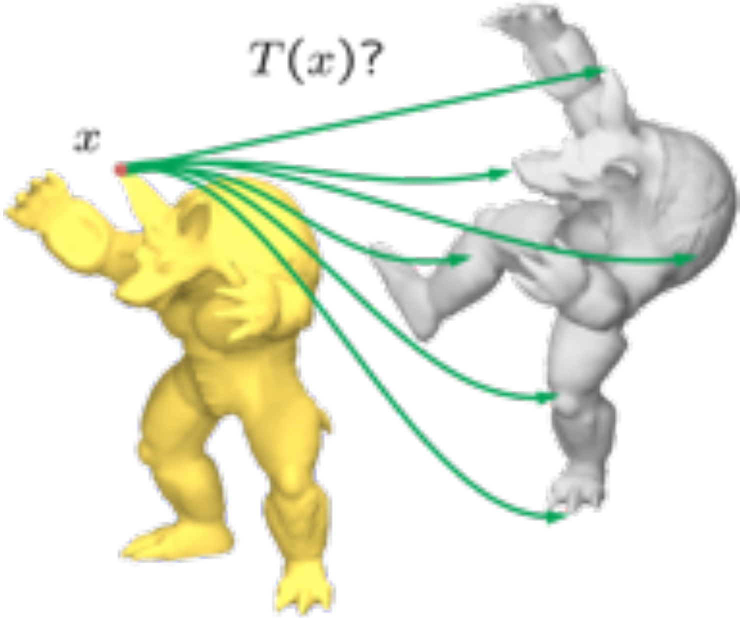
1. Finding nearest neighbors: can be done with space-partitioning data structures (e.g., KD-tree).
2. Finding the optimal transformation $\mathbf{R}, t$ minimizing:

$$\arg\min_{R,t} \sum_i \|Rx_i + t - y_i\|_2^2$$

Can be done efficiently via SVD decomposition.

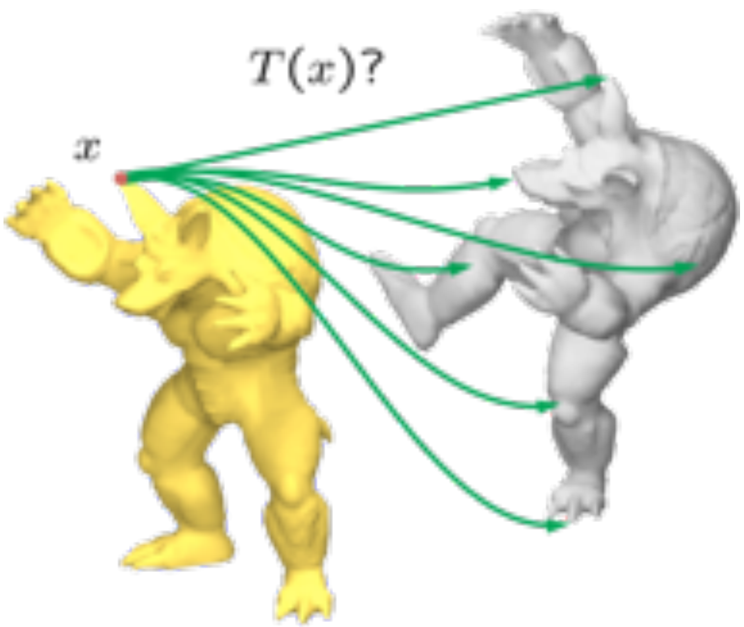
Arun et al., Least-Squares Fitting of Two 3-D Point Sets

Non-Rigid Shape Matching



Unlike rigid matching with rotation/translation, there is no compact representation to optimize for in non-rigid matching.

Non-Rigid Shape Matching



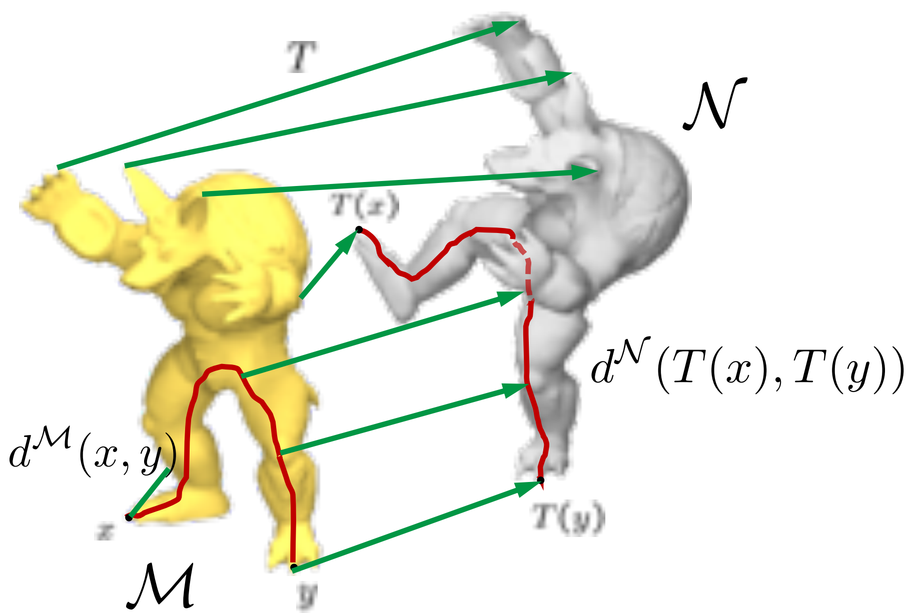
Main Questions:

- What does it mean for a correspondence to be “good”?
- How to compute it efficiently in practice?

Isometric Shape Matching

Deformation Model:

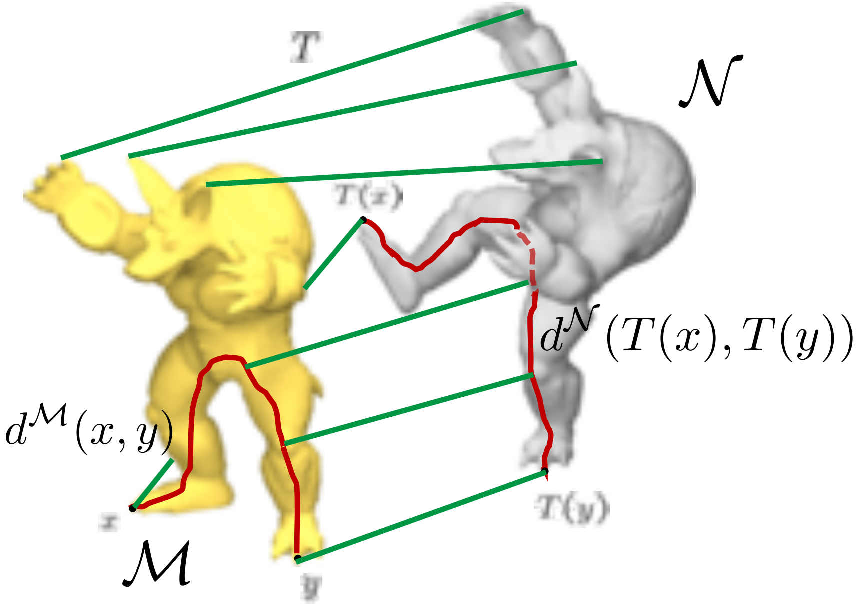
Good maps must preserve geodesic distances.



Geodesic: length of shortest path lying entirely on the surface.

Isometric Shape Matching

Approach:



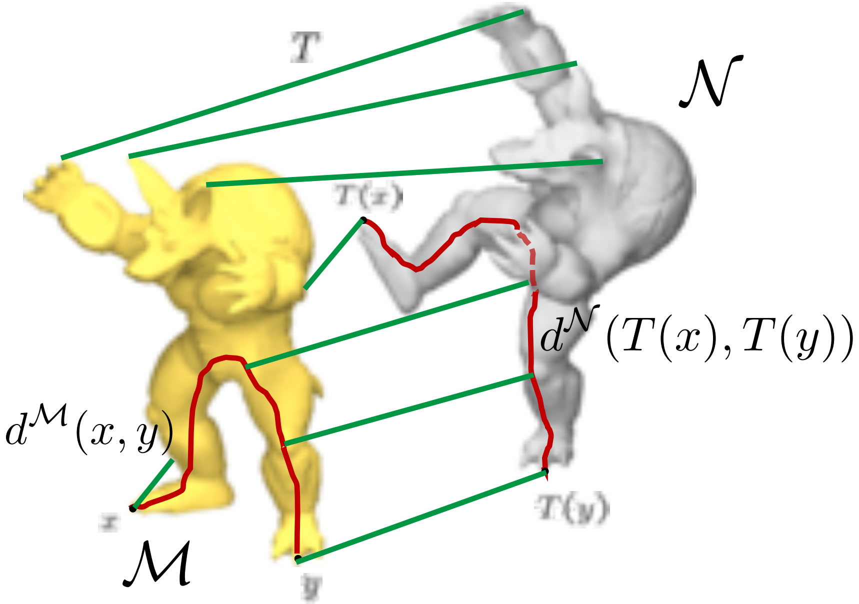
Find the point mapping by minimizing the distance distortion:

$$T_{\text{opt}} = \arg\min_{T} \sum_{x,y} \| d^{\mathcal{M}}(x, y) - d^{\mathcal{N}}(T(x), T(y)) \|$$

The unknowns are point correspondences.

Isometric Shape Matching

Approach:



Find the point mapping by minimizing the distance distortion:

$$T_{\text{opt}} = \arg\min_{T} \sum_{x,y} \| d^{\mathcal{M}}(x, y) - d^{\mathcal{N}}(T(x), T(y)) \|$$

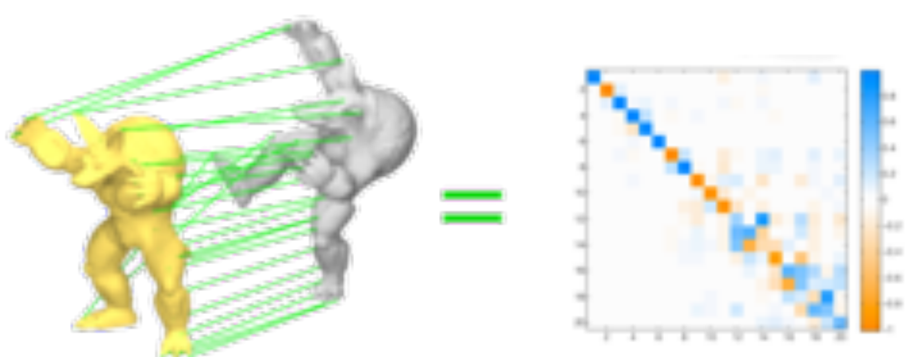
Problem:

The space of possible solutions is highly non-linear, non-convex.

Functional Map Representation

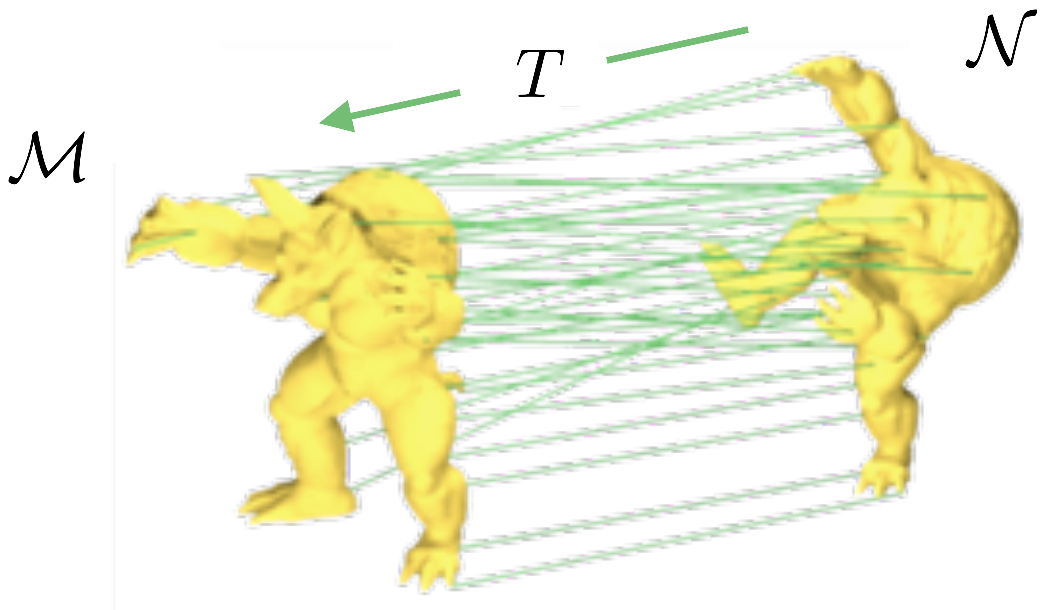
We would like to define a representation of shape maps that is more amenable to direct optimization.

1. A compact representation for “natural” maps.
2. Inherently global and multi-scale.
3. Handles uncertainty and ambiguity gracefully.
4. Allows efficient manipulations (averaging, composition).
5. Leads to simple (linear) optimization problems.



Functional Approach to Mappings

Given two shapes and a pointwise map $T : \mathcal{N} \to \mathcal{M}$



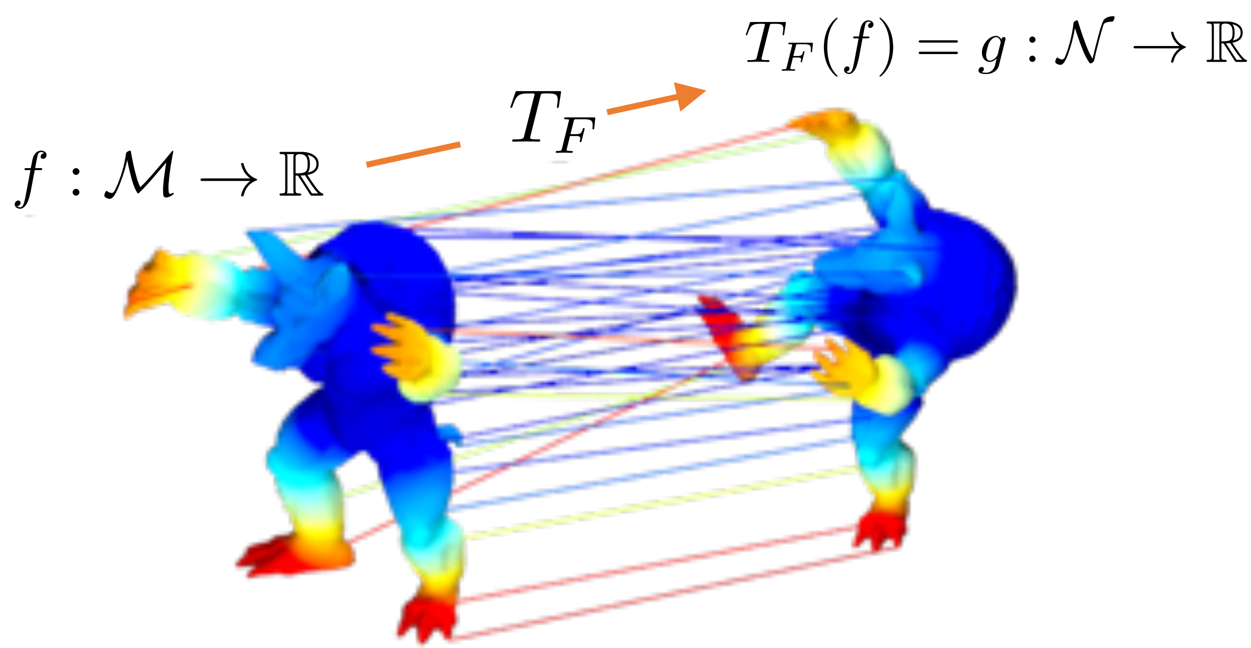
The map T induces a functional correspondence:

$$T_F(f) = g, \text{ where } g = f \circ T$$

Functional maps: a flexible representation of maps between shapes, *O., Ben-Chen, Solomon, Butscher, Guibas*, SIGGRAPH 2012

Functional Approach to Mappings

Given two shapes and a pointwise map $T : \mathcal{N} \to \mathcal{M}$



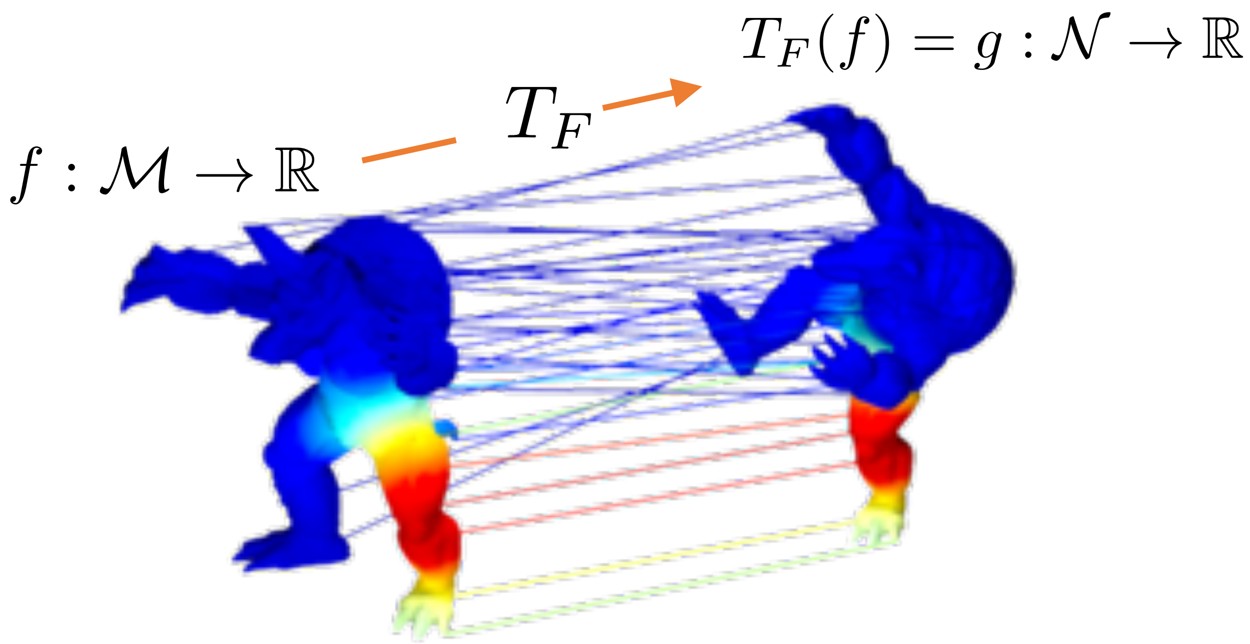
The map T induces a functional correspondence:

$$T_F(f) = g, \text{ where } g = f \circ T$$

Functional maps: a flexible representation of maps between shapes, *O., Ben-Chen, Solomon, Butscher, Guibas*, SIGGRAPH 2012

Functional Approach to Mappings

Given two shapes and a pointwise map $T : \mathcal{N} \to \mathcal{M}$



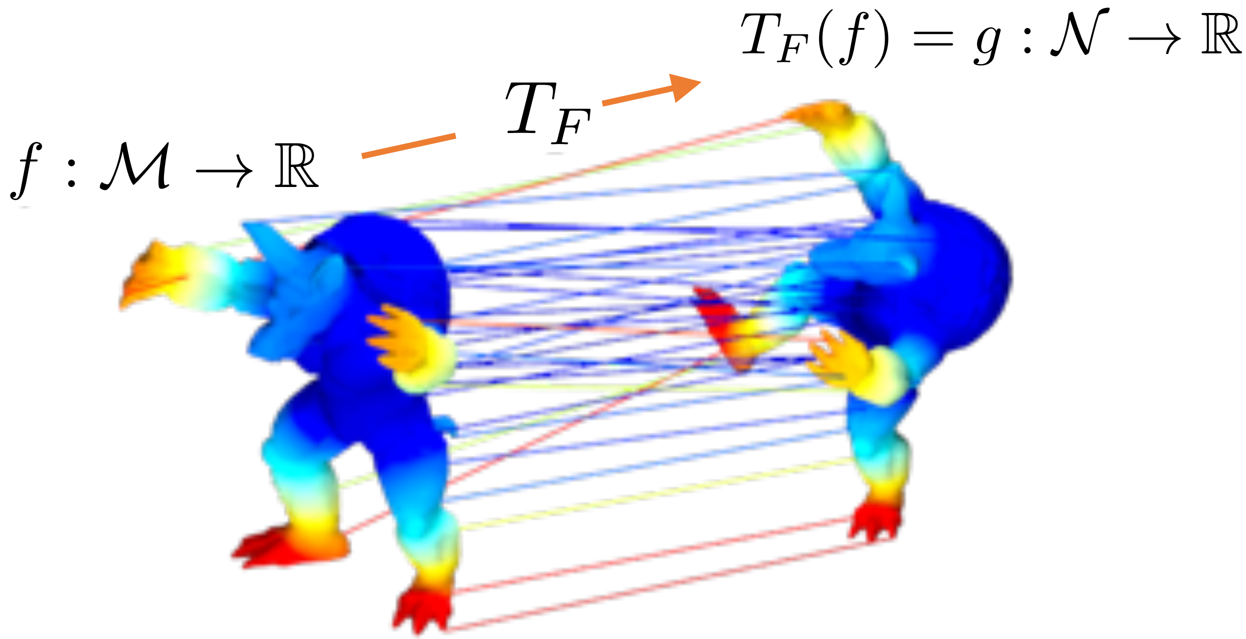
The map T induces a functional correspondence:

$$T_F(f) = g, \text{ where } g = f \circ T$$

Functional maps: a flexible representation of maps between shapes, *O., Ben-Chen, Solomon, Butscher, Guibas*, SIGGRAPH 2012

Functional Approach to Mappings

Given two shapes and a pointwise map $T : \mathcal{N} \to \mathcal{M}$

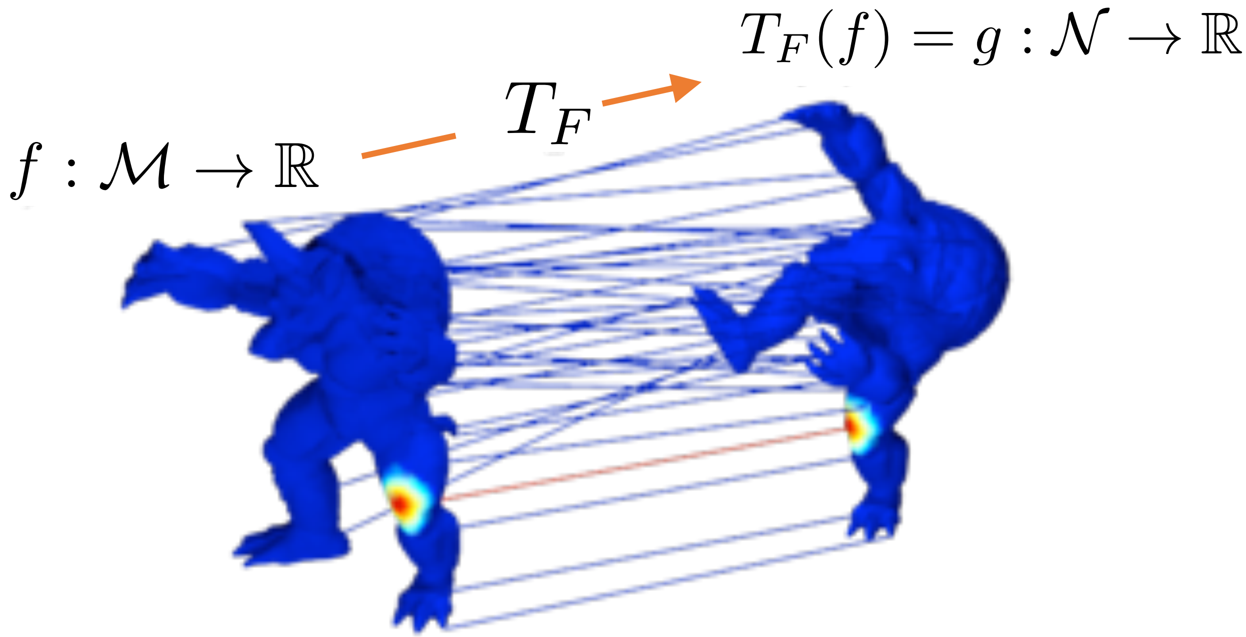


The induced functional correspondence is linear:

$$T_F(\alpha_1 f_1 + \alpha_2 f_2) = \alpha_1 T_F(f_1) + \alpha_2 T_F(f_2)$$

Functional Map Representation

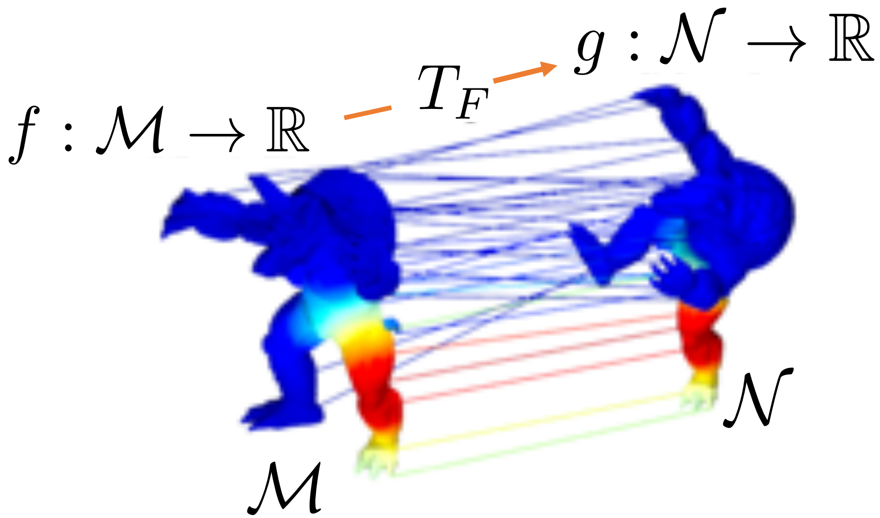
Given two shapes and a pointwise map $T : \mathcal{N} \to \mathcal{M}$



The induced functional correspondence is complete.

Observation

Assume that both: $f \in \mathcal{L}_2(\mathcal{M}), g \in \mathcal{L}_2(\mathcal{N})$



Express both f and $T_F(f)$ in terms of *basis functions:*

$$f = \sum_i a_i \phi_i^{\mathcal{M}} \qquad g = T_F(f) = \sum_j b_j \phi_j^{\mathcal{N}}$$

Since T_F is linear, there is a linear transformation from $\{a_i\}$ to $\{b_j\}$.

Functional Map Representation

Choice of Basis:

Eigenfunctions of the Laplace-Beltrami operator:

$$\Delta \phi_i = \lambda_i \phi_i \qquad \Delta(f) = -\mathrm{div}\nabla(f)$$

- Generalization of *Fourier bases* to surfaces.
- Ordered by eigenvalues and provide a natural notion of *scale*.

$\lambda_0 = 0$ $\quad \lambda_1 = 2.6$ $\quad \lambda_2 = 3.4$ $\quad \lambda_3 = 5.1$ $\quad \lambda_4 = 7.6$

Functional Map Representation

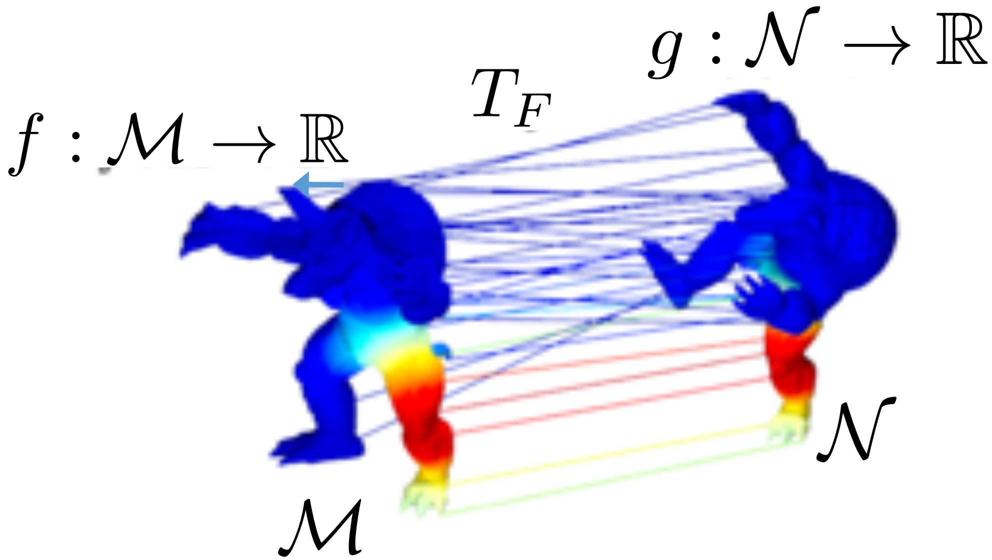
Choice of Basis:

Eigenfunctions of the Laplace-Beltrami operator:

$$\Delta\phi_i = \lambda_i\phi_i$$

- Generalization of *Fourier bases* to surfaces. Form an orthonormal basis for $L^2(\mathcal{M})$.
- Ordered by eigenvalues and provide a natural notion of *scale*.
- Can be computed efficiently, with a sparse matrix eigensolver.

Observation



Express both f and $T_F(f)$ in terms of *basis functions:*

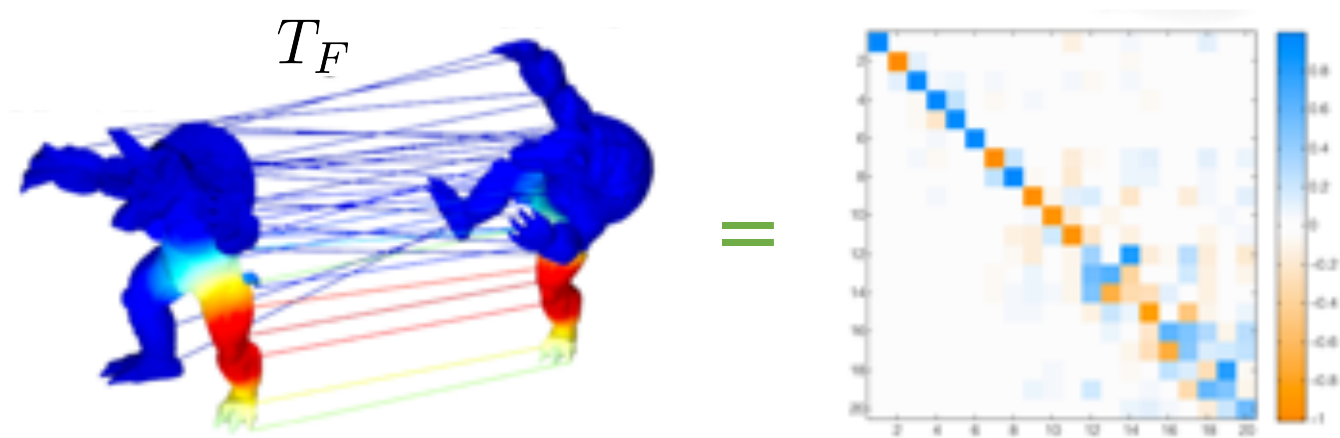
$$f = \sum_i a_i \phi_i^{\mathcal{M}} \qquad g = T_F(f) = \sum_i b_i \phi_i^{\mathcal{N}}$$

Since T_F is linear, there is a linear transformation from $\{a_i\}$ to $\{b_j\}$.

Functional Map Representation

Since the functional mapping T_F is linear:

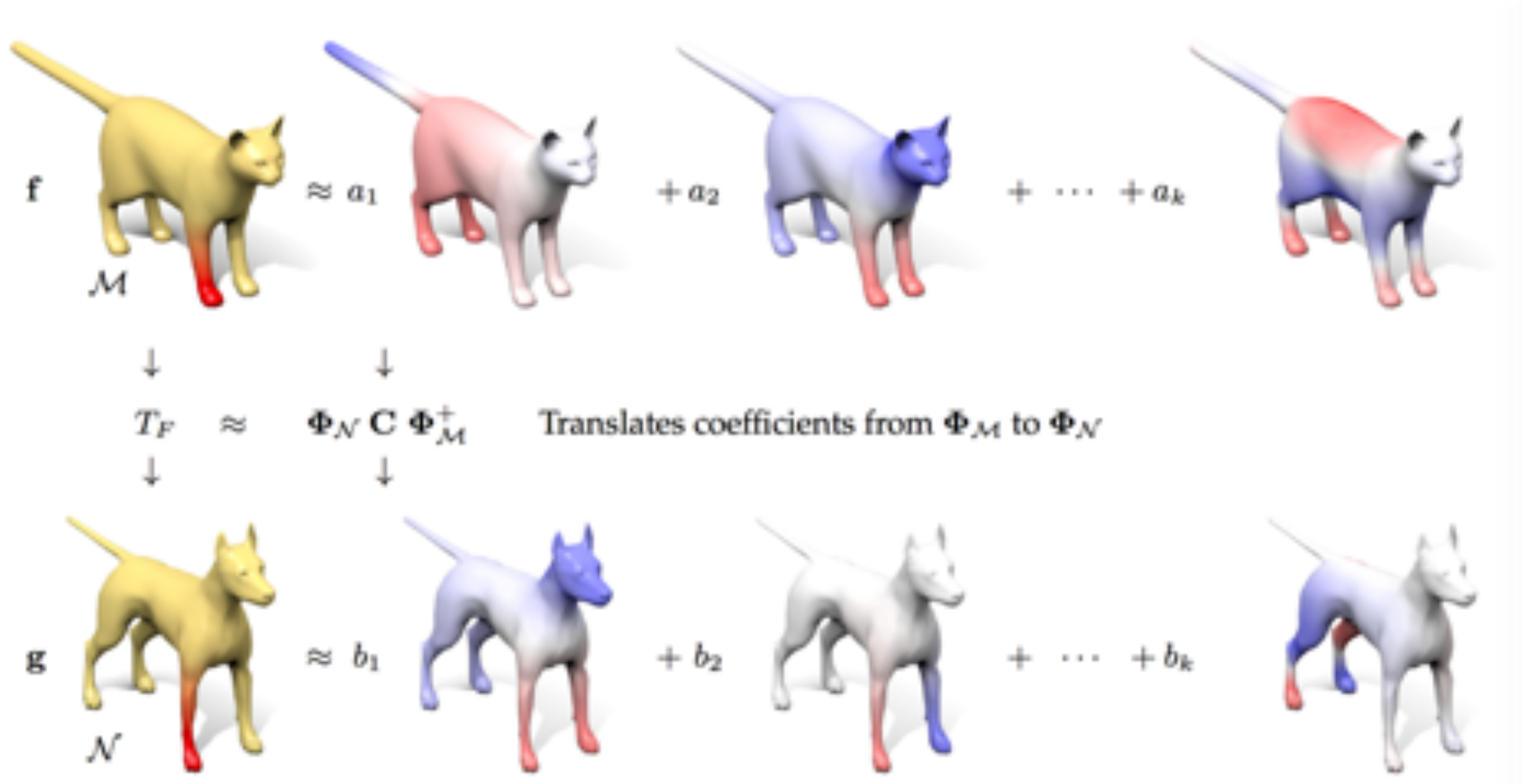
$$T_F(\alpha_1 f_1 + \alpha_2 f_2) = \alpha_1 T_F(f_1) + \alpha_2 T_F(f_2)$$



T_F can be represented as a matrix C, given a choice of basis for function spaces.

Functional maps: a flexible representation of maps between shapes, *O., Ben-Chen, Solomon, Butscher, Guibas*, SIGGRAPH 2012

Functional Map Definition

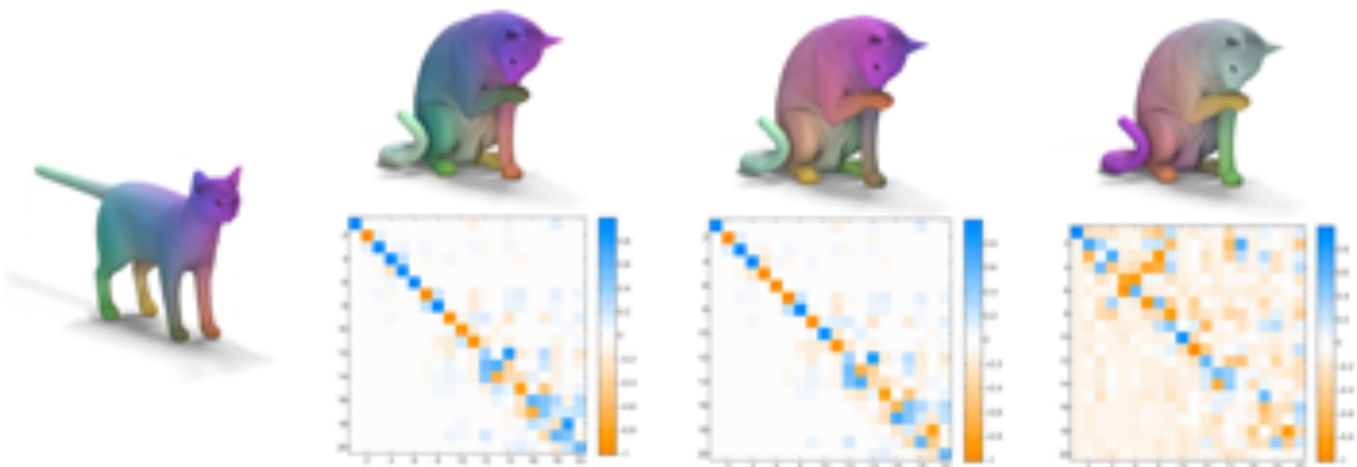


Functional map:

matrix C that translates coefficients from $\Phi_{\mathcal{M}}$ to $\Phi_{\mathcal{N}}$.

Example Maps in a Reduced Basis

Triangle meshes with pre-computed pointwise maps



(a) source (b) ground-truth map (c) left-to-right map (d) head-to-tail map

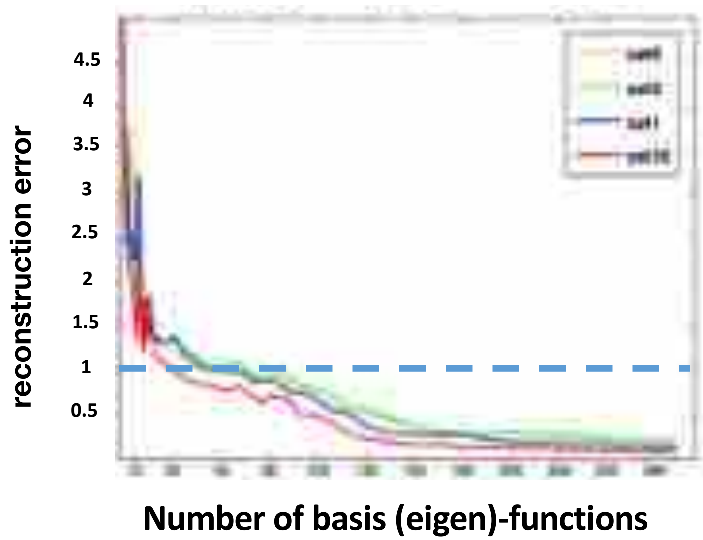
"Good" maps are close to being diagonal

Try [fmap_computation_demo] on the course website

Reconstructing from LB basis

Map reconstruction error using a fixed size matrix.

27.9k vertices



Try [fmap_reconstruction_demo]() on the course website

Functional Map algebra

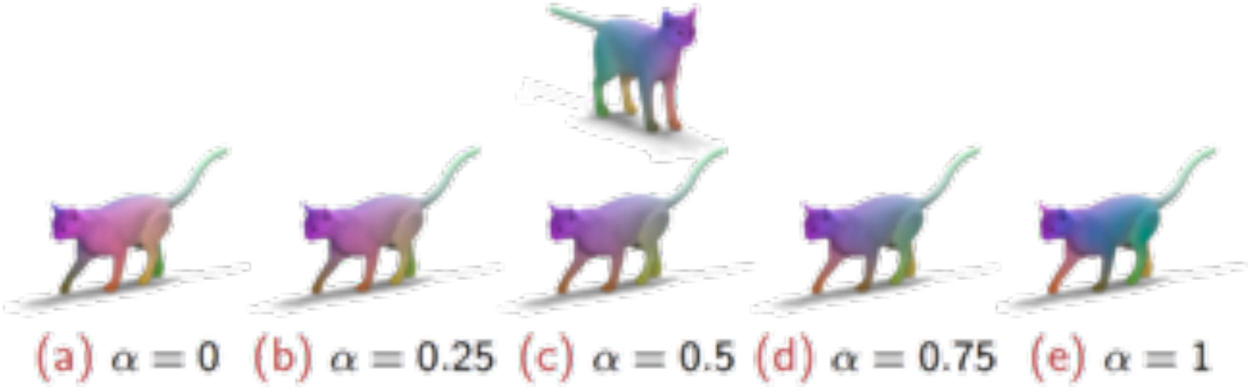
1. Map composition becomes matrix multiplication.

2. Map inversion is matrix inversion (in fact, transpose).

3. Algebraic operations on functional maps are possible.

E.g. interpolating between two maps with

$$C = \alpha C_1 + (1-\alpha) C_2.$$



Talk Overview

- Motivation and Problem Taxonomy
- Rigid Matching: ICP
- Functional Map Representation, properties
- Basic pipeline for non-rigid matching
- Extensions, Improvements
- Open Problems, Q&A

Shape Matching

In practice we do not know C. Given two objects our goal is to find the correspondence.

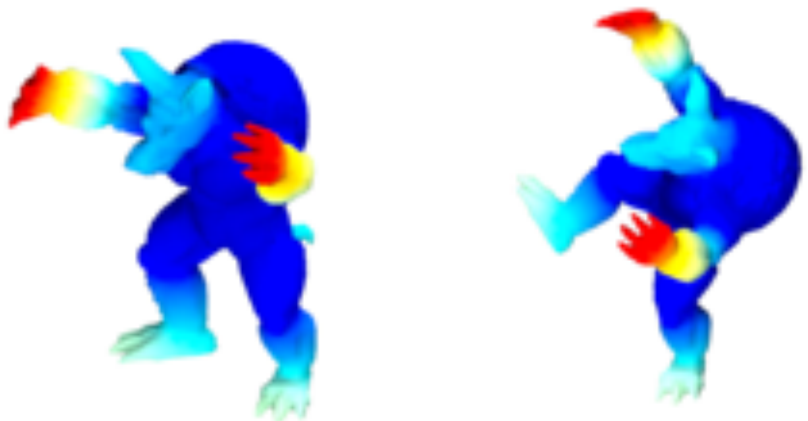
How can the functional representation help to compute the map in practice?

Matching via Function Preservation

Suppose we don't know C. However, we expect a pair of functions $f : \mathcal{M} \to \mathbb{R}$ and $g : \mathcal{N} \to \mathbb{R}$ to correspond. Then, C must be s.t.

$$C\mathbf{a} \approx \mathbf{b}$$

where $f = \sum_i a_i \phi_i^{\mathcal{M}}, \quad g = \sum_i b_i \phi_i^{\mathcal{N}}.$



Given enough $\{\mathbf{a}, \mathbf{b}\}$ pairs, we can recover C through *a linear least squares system*.

Map Constraints

Suppose we don't know C. However, we expect a pair of functions $f : \mathcal{M} \to \mathbb{R}$ and $g : \mathcal{N} \to \mathbb{R}$ to correspond. Then, C must be s.t.

$$C\mathbf{a} \approx \mathbf{b}$$

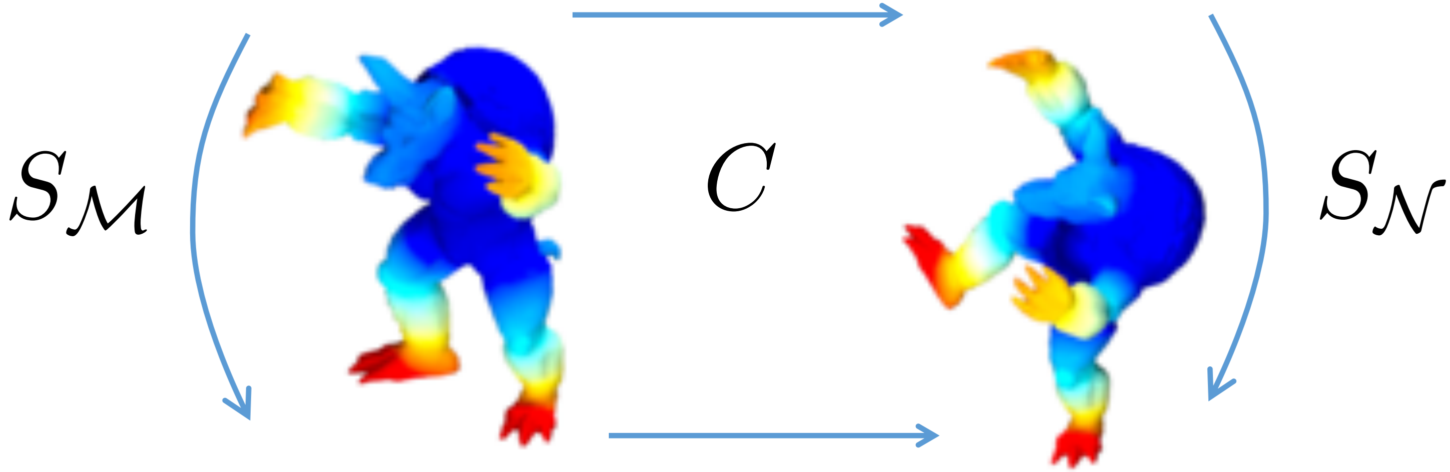
Function preservation constraint is general and includes:

- Attribute (e.g., color) preservation.
- Descriptor preservation (e.g. Gauss curvature).
- Landmark correspondences (e.g. distance to the point).
- Part correspondences (e.g. indicator function).

Commutativity Constraints

Regularizations:

Commutativity with other operators:



$$CS_{\mathcal{M}} = S_{\mathcal{N}}C$$

Note that the energy: $\|CS_{\mathcal{M}} - S_{\mathcal{N}}C\|_F^2$ is *quadratic* in C.

Regularization

Linking functional and point-to-point maps

Lemma 1:

The mapping is *isometric*, if and only if the functional map matrix commutes with the Laplacian:

$$C\Delta_{\mathcal{M}} = \Delta_{\mathcal{N}}C$$

Implies that exact isometries result in *diagonal functional maps*.

Functional maps: a flexible representation of maps between shapes, *O., Ben-Chen, Solomon, Butscher, Guibas*, SIGGRAPH 2012

Basic Pipeline

Given a pair of shapes $\mathcal{M}, \mathcal{N}$:

1. Compute the first *k* (~80-100) eigenfunctions of the Laplace-Beltrami operator. Store them in matrices: $\Phi_{\mathcal{M}}, \Phi_{\mathcal{N}}$

2. Compute descriptor functions (e.g., Wave Kernel Signature) on $\mathcal{M}, \mathcal{N}$. Express them in $\Phi_{\mathcal{M}}, \Phi_{\mathcal{N}}$, as columns of : $\mathbf{A}, \mathbf{B}$

3. Solve $C_{\text{opt}} = \underset{C}{\arg\min} \|C\mathbf{A} - \mathbf{B}\|^2 + \|C\Delta_{\mathcal{M}} - \Delta_{\mathcal{N}}C\|^2$

   $\Delta_{\mathcal{M}}, \Delta_{\mathcal{N}}$: diagonal matrices of eigenvalues of LB operator

4. Convert the functional map C_{opt} to a point to point map T.

Recent Implementation

Recent implementation incorporating efficient spatial and spectral constraints.

Continuous and Orientation-preserving Correspondences via Functional Maps Jing Ren, Adrien Poulenard, Peter Wonka, Maks Ovsjanikov, SIGGRAPH Asia 2018

https://github.com/llorz/SGA18_orientation_BCICP_code

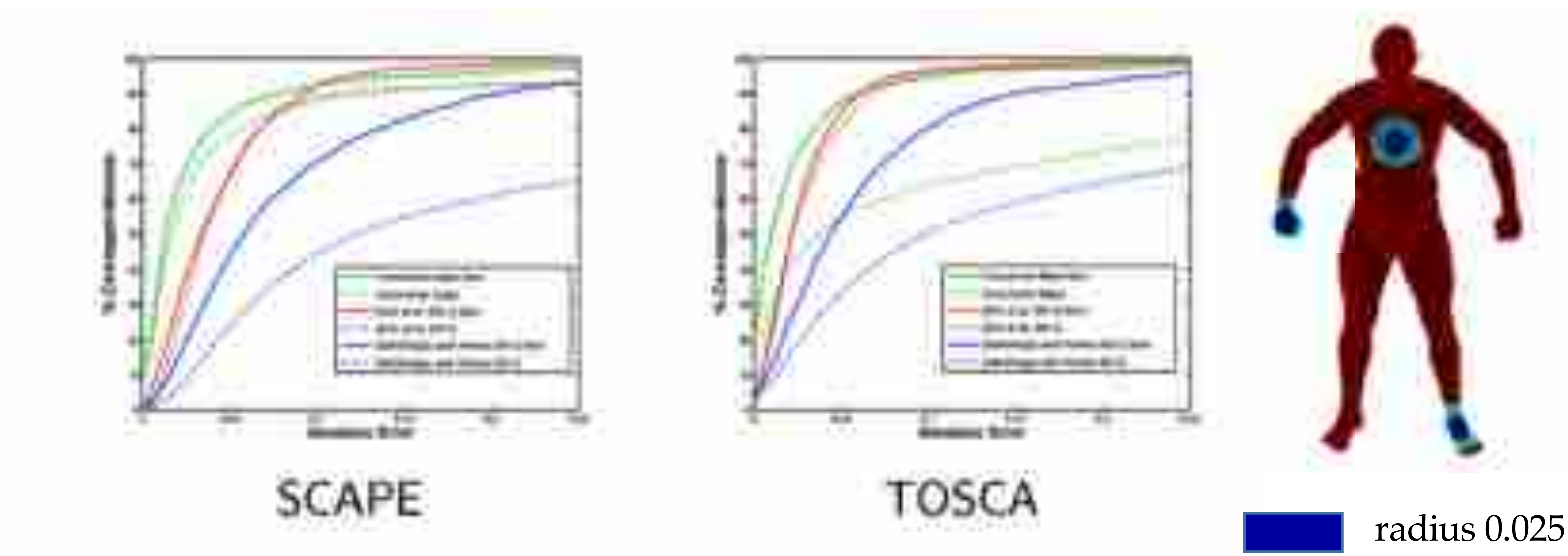
Results

A very simple method that puts together many constraints and uses 100 basis functions gives reasonable results:

Functional maps: a flexible representation of maps between shapes, *O., Ben-Chen, Solomon, Butscher, Guibas*, SIGGRAPH 2012

Results

A very simple method that puts together many constraints and uses 100 basis functions gives reasonable results:



radius 0.025

radius 0.05

Functional maps: a flexible representation of maps between shapes, *O., Ben-Chen, Solomon, Butscher, Guibas*, SIGGRAPH 2012

Segmentation Transfer without P2P

To transfer functions we do not need to convert functional to pointwise maps.

E.g. we can also transfer segmentations: for each segment, transfer its indicator function, and for each point pick the segment that gave the highest value.

Talk Overview

- Motivation and Problem Taxonomy
- Rigid Matching: ICP
- Functional Map Representation, properties
- Basic pipeline for non-rigid matching
- Extensions, Improvements
- Open Problems, Q&A

Some Recent Extensions

- **Unsupervised Learning**

  Unsupervised Deep Learning for Structured Shape Matching
  Roufosse, Sharma, Ovsjanikov, ICCV, 2019 (oral).

- **Efficient Refinement**

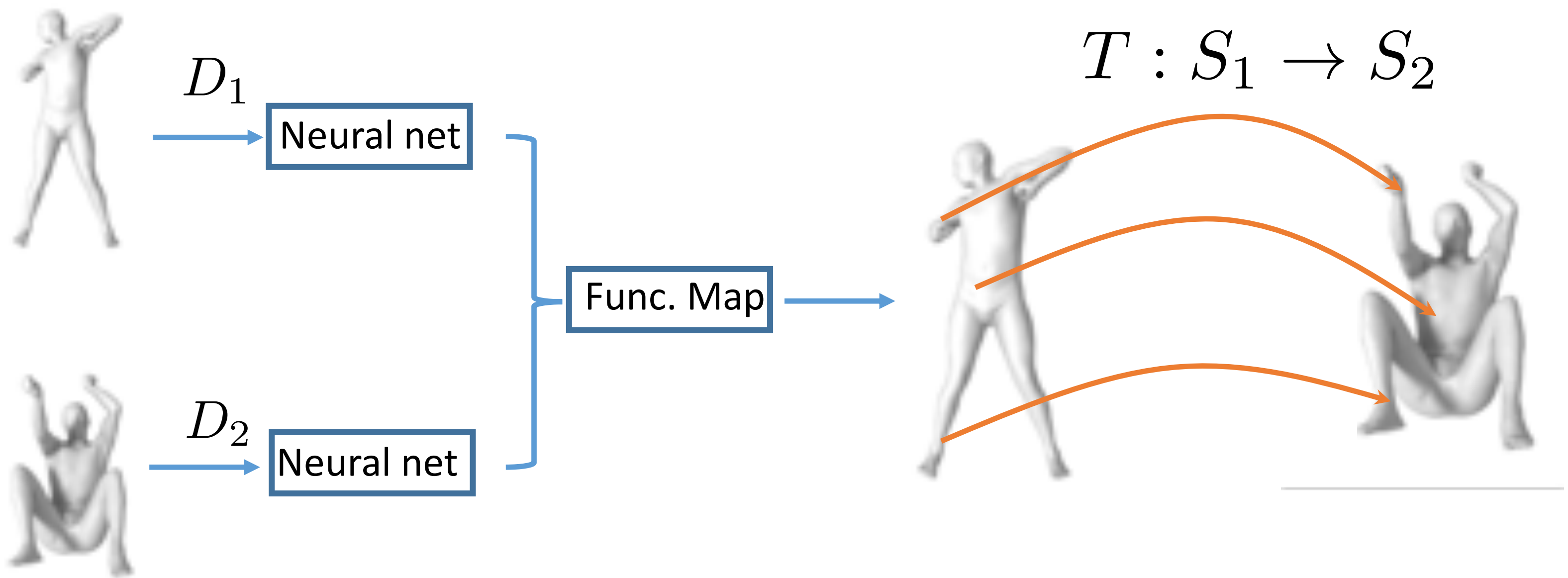
  ZoomOut: Spectral Upsampling for Efficient Shape Correspondence
  Melzi, Ren, Rodolà, Sharma, Wonka, Ovsjanikov, SIGGRAPH Asia 2019

Main Question

What happens if the descriptors are bad?

FMNet

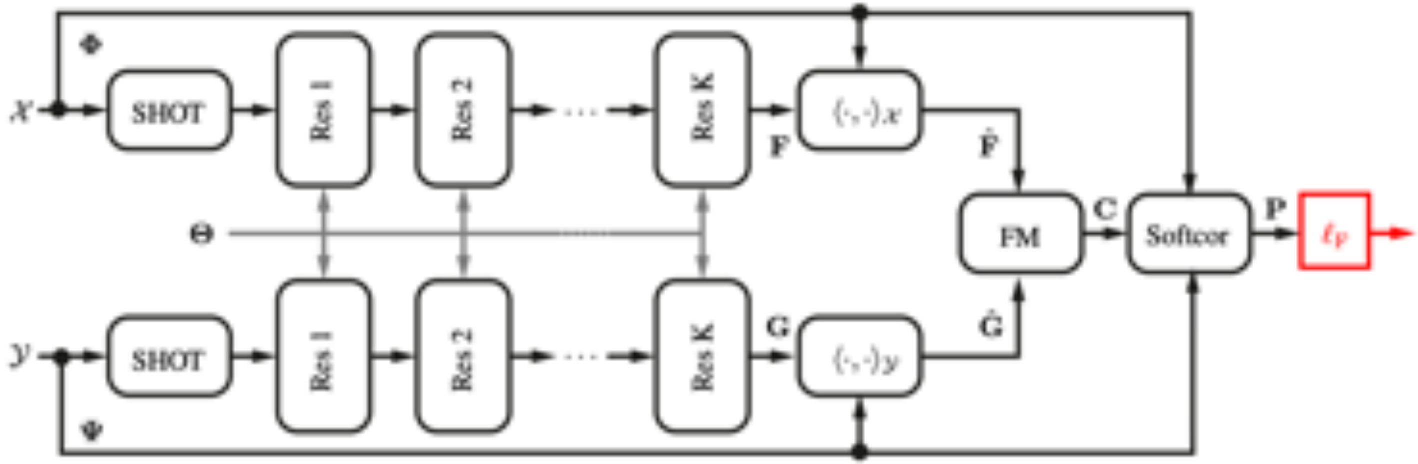
Learning approach to computing descriptors.



O. Litany, T. Remez, E. Rodolà, A. Bronstein, M. Bronstein: Deep functional maps: Structured prediction for dense shape correspondence. In Proc. ICCV (2017).

FMNet

Learning approach to computing descriptors.



Training loss: $\ell_F = \sum_{(x,y)\in(\mathcal{X},\mathcal{Y})} P(x,y) d_{\mathcal{Y}}(y, \pi^*(x))$ $\quad \mathbf{P} = |\mathbf{\Psi}\mathbf{C}\mathbf{\Phi}^\top\mathbf{A}|^\wedge$

O. Litany, T. Remez, E. Rodolà, A. Bronstein, M. Bronstein: Deep functional maps: Structured prediction for dense shape correspondence. In Proc. ICCV (2017).

Our Goals

1. Avoid using ground truth correspondences
   - Replace supervised loss with *unsupervised* one

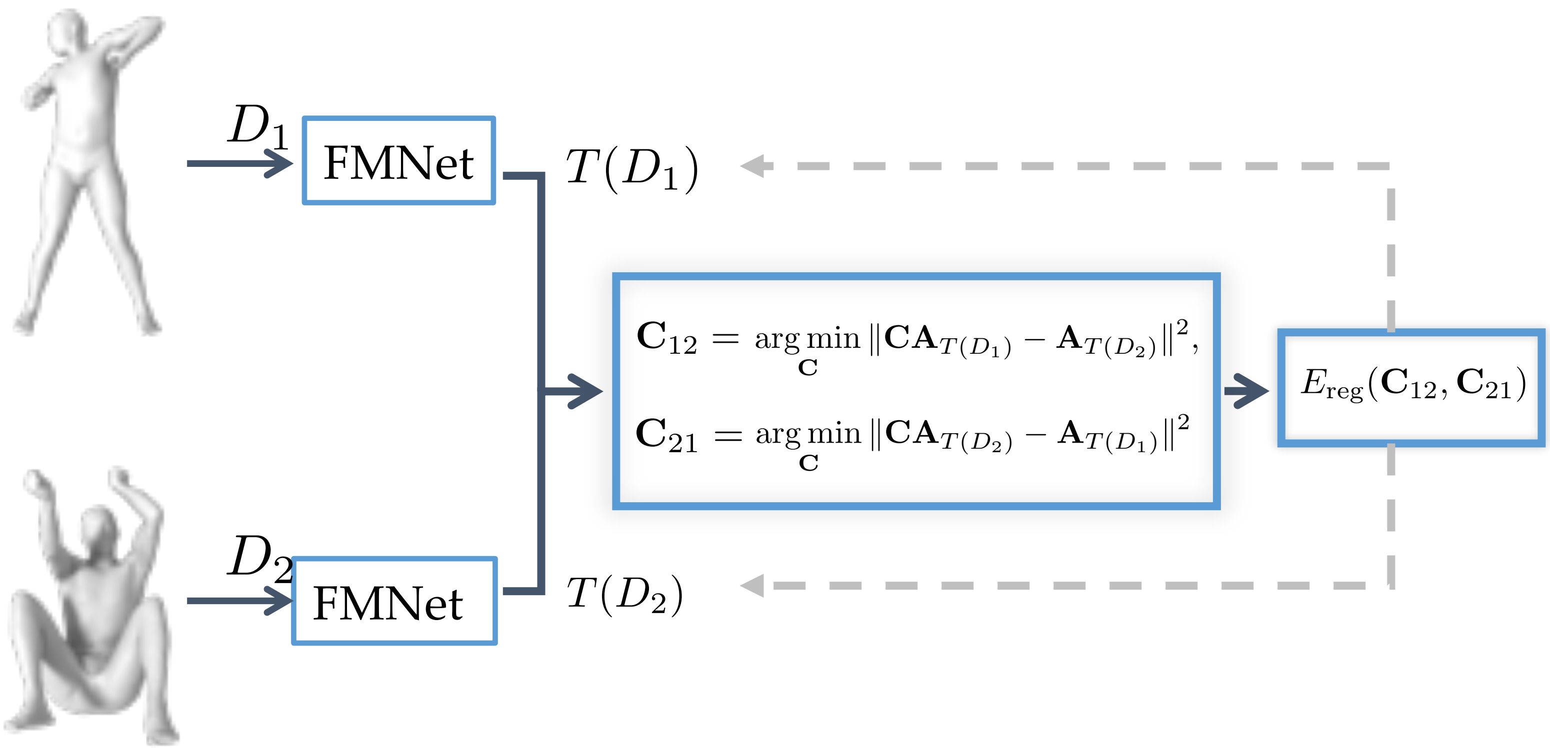
2. Avoid using geodesic distances
   - Perform all computations in the *spectral domain*

Main question: how to measure the quality of a map?

Note: related concurrent paper by Halimi et al. *Unsupervised learning of dense shape correspondence*. In CVPR, 2019

Our approach

Replace supervised loss with unsupervised one



Unsupervised Deep Learning for Structured Shape Matching, J.-M. Rouffosse, A. Sharma, M. O., ICCV 2019

Our approach

Replace supervised loss with unsupervised one

$$loss_{unsupervised} = \sum_{i \in penalties} w_i E_i(C_{1\text{-}2}, C_{2\text{-}1})$$

$$\begin{cases} E_1(C_{12}, C_{21}) = \|C_{12}C_{21} - Id\|^2 \\ E_1(C_{12}, C_{21}) = \|C_{21}C_{12} - Id\|^2 \end{cases}$$ Bijectivity

$$E_2(C) = \|C^T C - Id\|^2$$ Area-preservation

$$E_3(C) = \|\Lambda_2 C - C\Lambda_1\|^2$$ Near-isometry

$$E_4(C) = \sum_i \|CX_{f_i} - Y_{g_i}C\|^2$$ Functional map close to pointwise one.

All penalties are in the reduced basis. 50x faster than FMNet

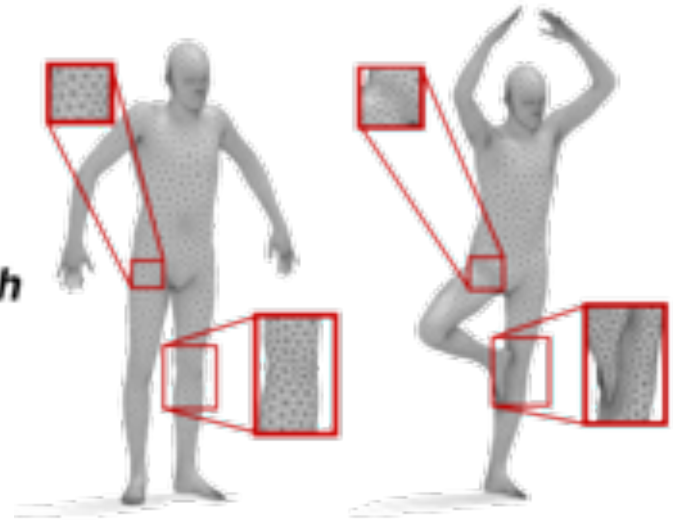
Datasets

FAUST :

- Subset: train on 80 and test on 20
- Whole set : train on 100 shapes, ***without ground truth***

SCAPE :

- Subset: train on 50 and test on 10
- Whole set : train on 60 shapes, ***without ground truth***

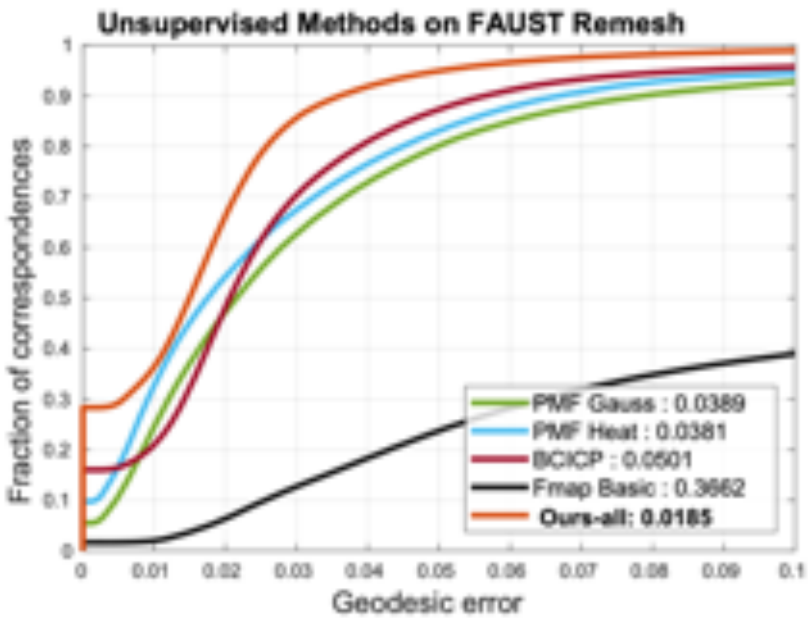


Remeshed FAUST - 5000 vertices

Datasets released as part of: *Continuous and Orientation-preserving Correspondences via Functional Maps*, J. Ren, A. Poulenard, P. Wonka, M. O, SIGGRAPH Asia 2018

Results

Comparison to unsupervised methods



State-of-the art among unsupervised methods.

Unsupervised Deep Learning for Structured Shape Matching, J.-M. Rouffosse,
A. Sharma, M. O., ICCV 2019

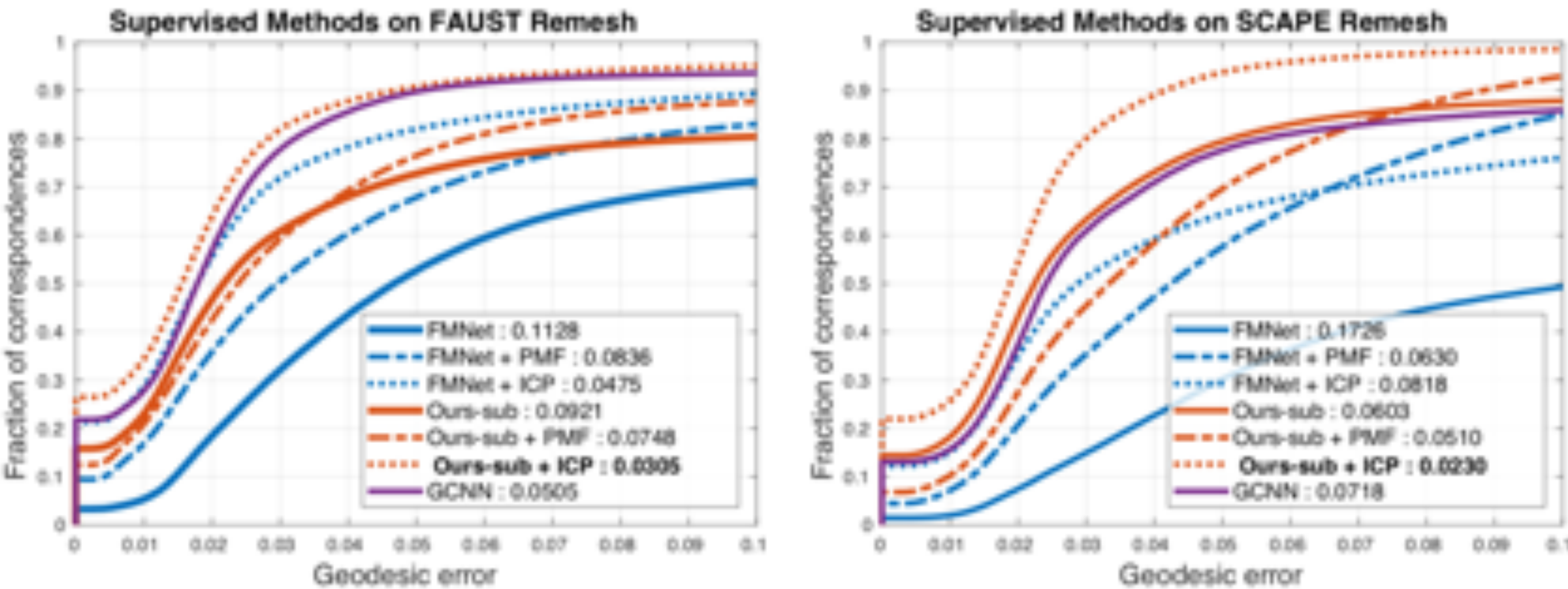
Results

State-of-the art among unsupervised methods.

Unsupervised Deep Learning for Structured Shape Matching, J.-M. Rouffosse, A. Sharma, M. O., ICCV 2019

Results

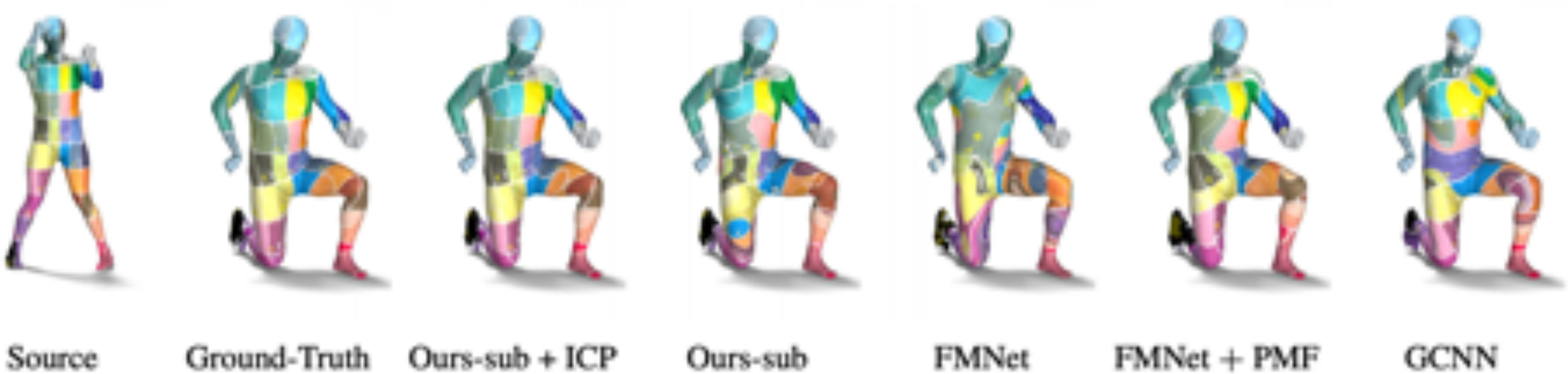
Comparison to supervised methods



Comparable results even to supervised methods

Unsupervised Deep Learning for Structured Shape Matching, J.-M. Rouffosse, A. Sharma, M. O., ICCV 2019

Results



Comparable results even to supervised methods

Unsupervised Deep Learning for Structured Shape Matching, J.-M. Rouffosse, A. Sharma, M. O., ICCV 2019

Results

Original vs. learned descriptors.

Source descriptor before | Target descriptor before | Source descriptor after | Target descriptor after

Unsupervised Deep Learning for Structured Shape Matching, J.-M. Rouffosse, A. Sharma, M. O., ICCV 2019

Several related questions

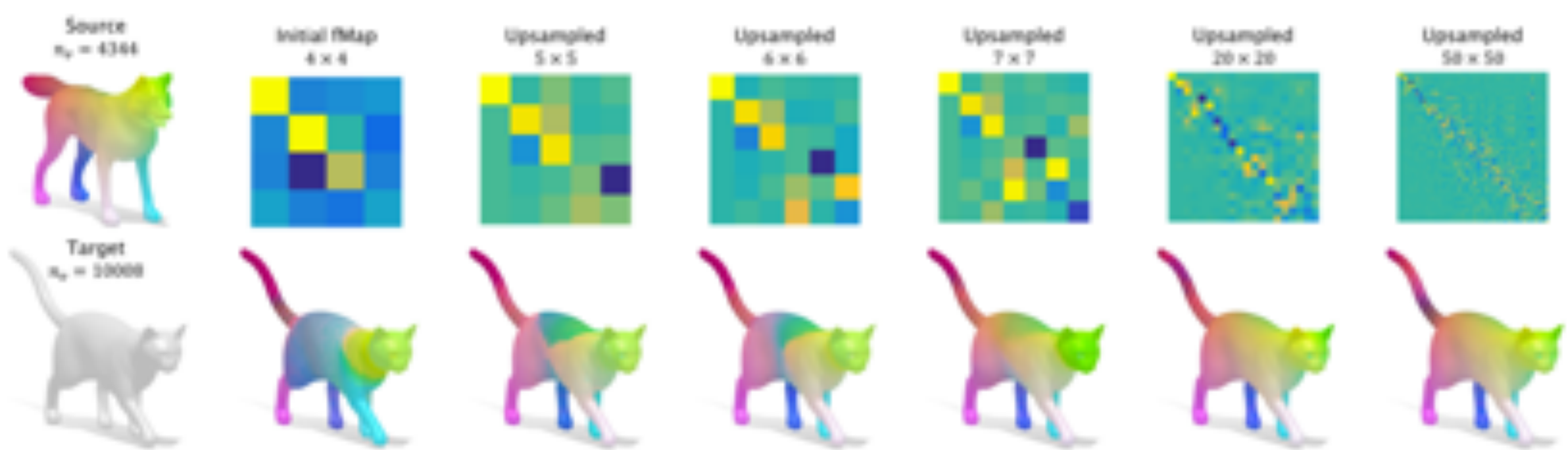
1. How can we build up a functional map *progressively*?
2. Given a small functional map, can we use it to transfer *high frequency functions*?
3. Simplify and speed-up *functional map refinement*?

ZoomOut

A two-lines-of-code algorithm:

1) Given a functional map C1 of size k x k convert it to a p2p map T.
2) Convert T to C2 of size (k+1) x (k+1)

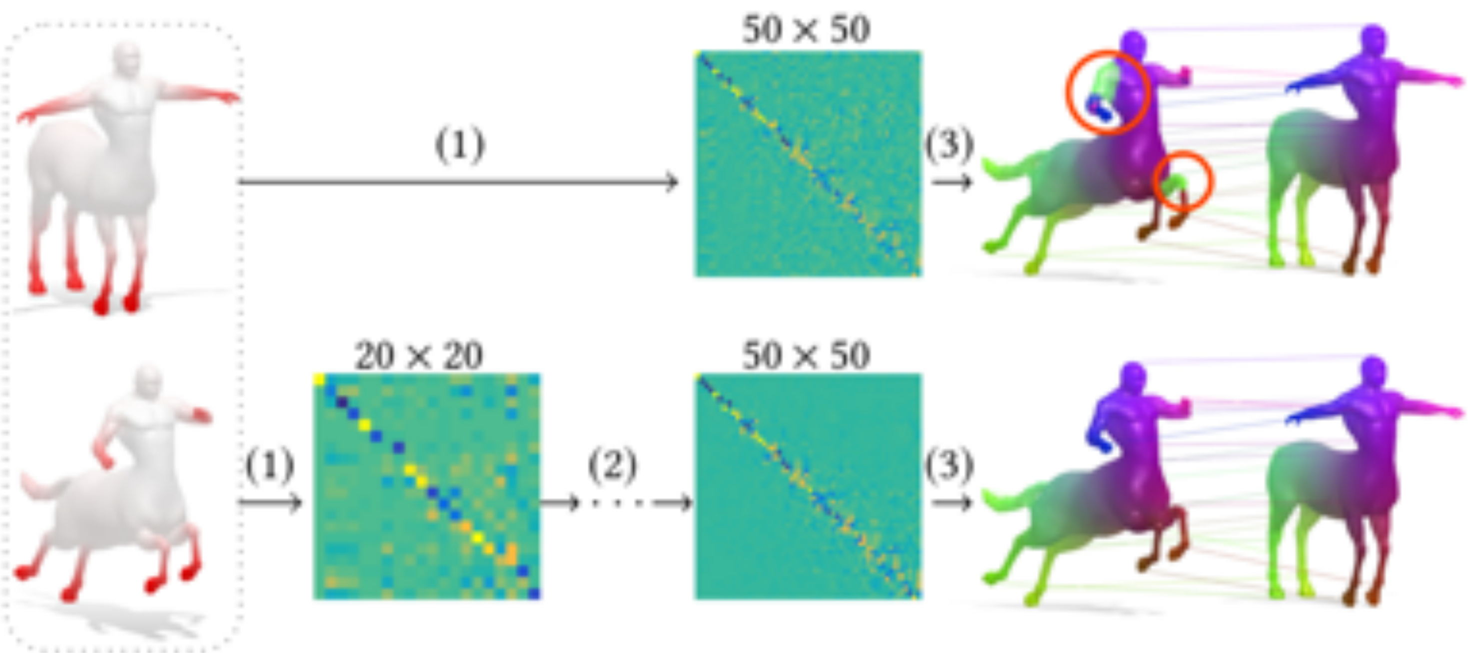
Repeat for progressively larger k



ZoomOut: Spectral Upsampling for Efficient Shape Correspondence, S. Melzi, J. Ren, A. Sharma, E. Rodolà, P. Wonka, M. O., SIGGRAPH Asia 2019

ZoomOut

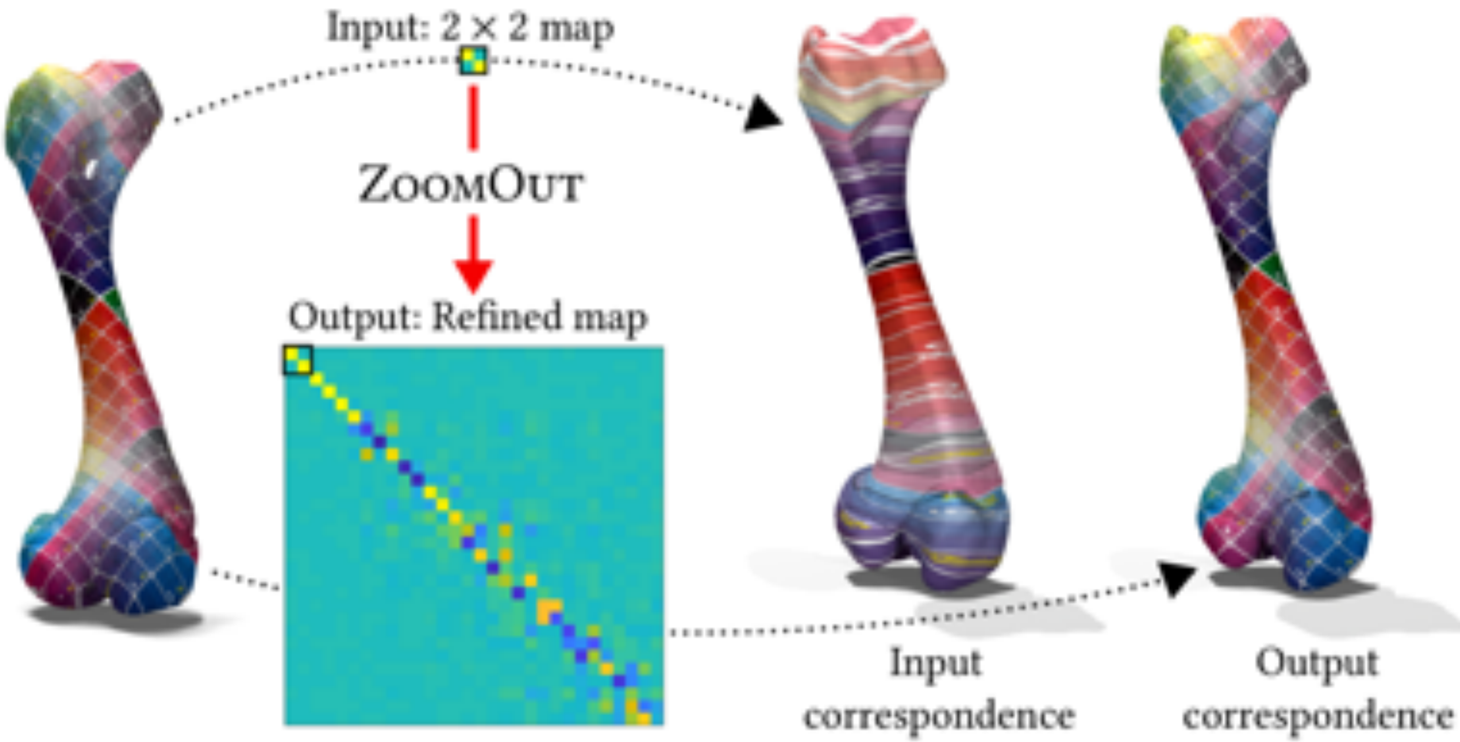
Upsampling vs. computing directly:



ZoomOut: Spectral Upsampling for Efficient Shape Correspondence, S. Melzi, J. Ren, A. Sharma, E. Rodolà, P. Wonka, M. O., SIGGRAPH Asia 2019

ZoomOut – Results

Extreme case, from 2x2 to 100x100



Dataset provided by the Natural History Museum in Paris.

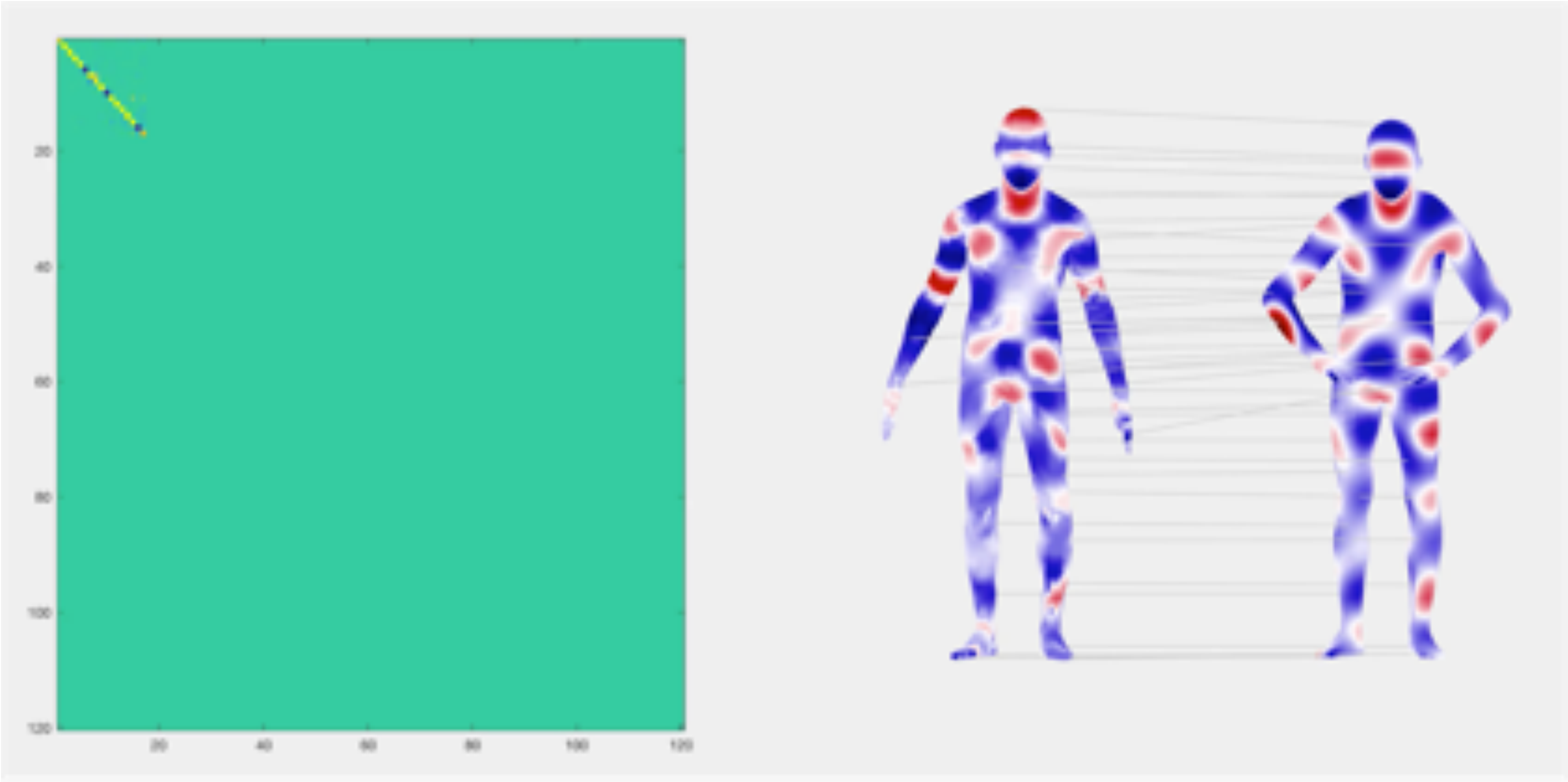
ZoomOut – Results

From 5x5 to 50x50

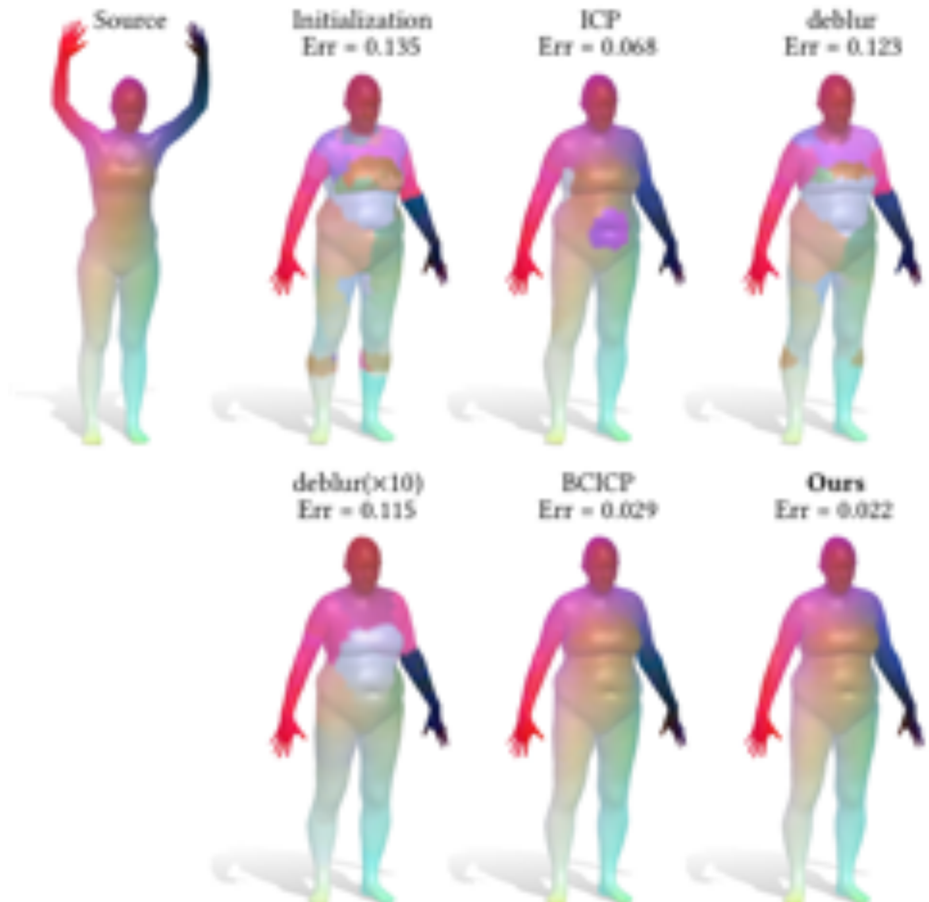
ZoomOut: Spectral Upsampling for Efficient Shape Correspondence, S. Melzi, J. Ren, A. Sharma, E. Rodolà, P. Wonka, M. O., SIGGRAPH Asia 2019

ZoomOut – Results

From 20x20 to 120x120



ZoomOut – Results



Evaluated on:
- Intrinsic symmetry detection
- Complete matching
- Partial matching
- Function transfer

…

Compared against 14 baselines

Ours is 50-300x faster than state-of-the-art with higher accuracy

ZoomOut – Rationale

Consider the optimization problem:

$$\min_{\mathbf{C} \in \mathcal{P}} E(\mathbf{C}), \;\; \text{where } E(\mathbf{C}) = \sum_k \frac{1}{k} \| \mathbf{C}_k^T \mathbf{C}_k - I_k \|_F^2 \,.$$

$\mathbf{C} \in \mathcal{P}$: functional map arising from some pointwise map.

$\mathbf{C}_k$: leading principal $k \times k$ submatrix of $\mathbf{C}$.

Theorem:

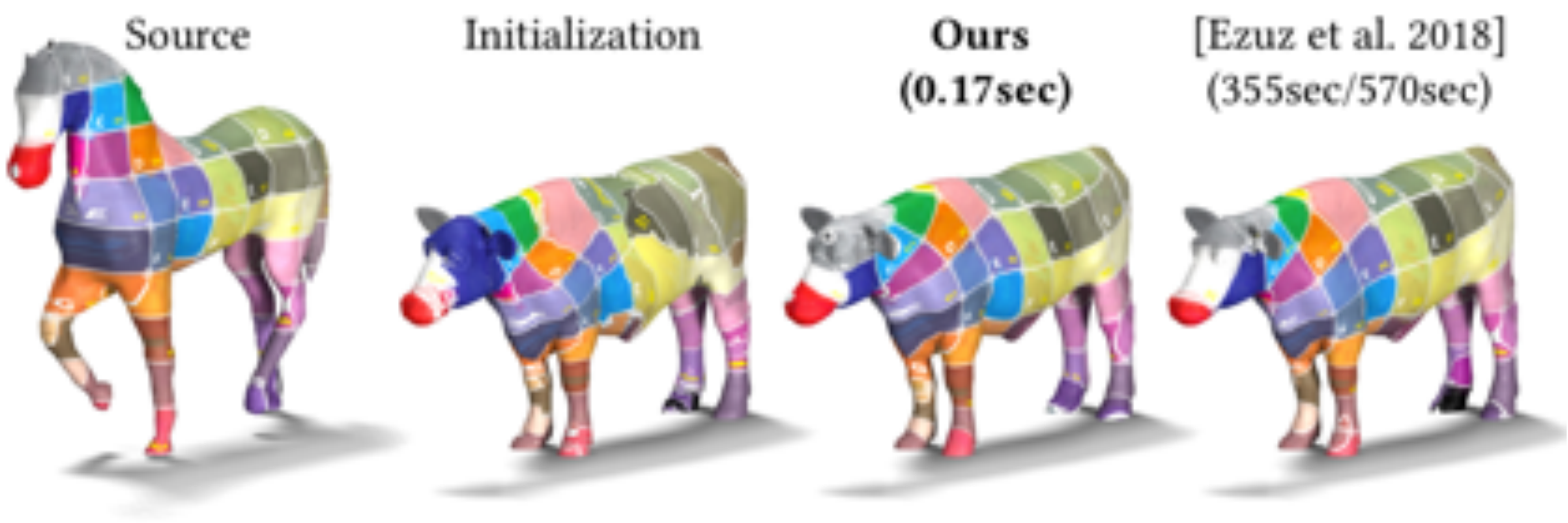
$E(\mathbf{C}) = 0$ if and only if the point-to-point map is an isometry.

ZoomOut can be derived as a iterative method for solving this optimization problem.

ZoomOut: Spectral Upsampling for Efficient Shape Correspondence, S. Melzi, J. Ren, A. Sharma, E. Rodolà, P. Wonka, M. O., SIGGRAPH Asia 2019

ZoomOut – Non-isometric

In some cases also works for non-isometric shapes



Other Extensions

- ## Maps in Collections

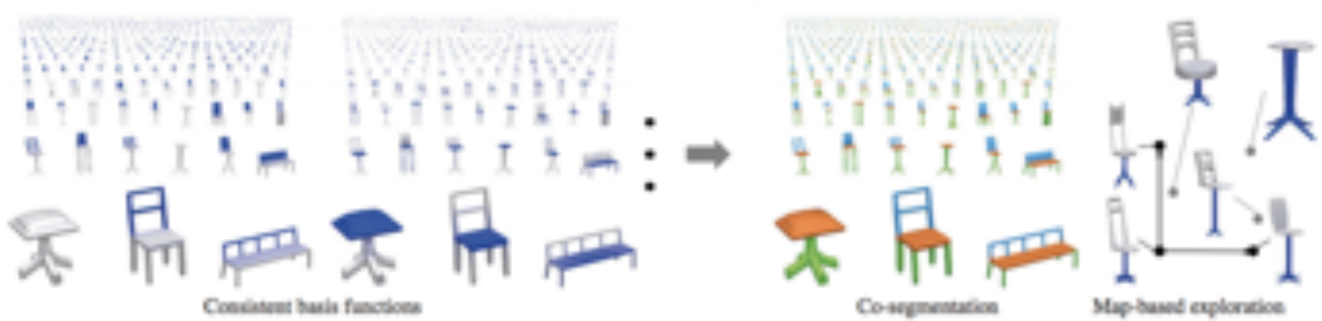
  Functional map networks for analyzing and exploring large shape collections
  Huang, Wang, Guibas, SIGGRAPH 2014

- ## Promoting Pointwise Maps

  Informative Descriptor Preservation via Commutativity for Shape Matching,
  Nogneng, O., Eurographics 2017

- ## Manifold Optimization

  MADMM: A generic algorithm for non-smooth optimization on manifolds.
  Kovnatsky, Glashoff, M. Bronstein, ECCV, 2016.



Huang et al., SIGGRAPH 2014

Consistency via Latent Space Optimization

- Application to Co-segmentation:

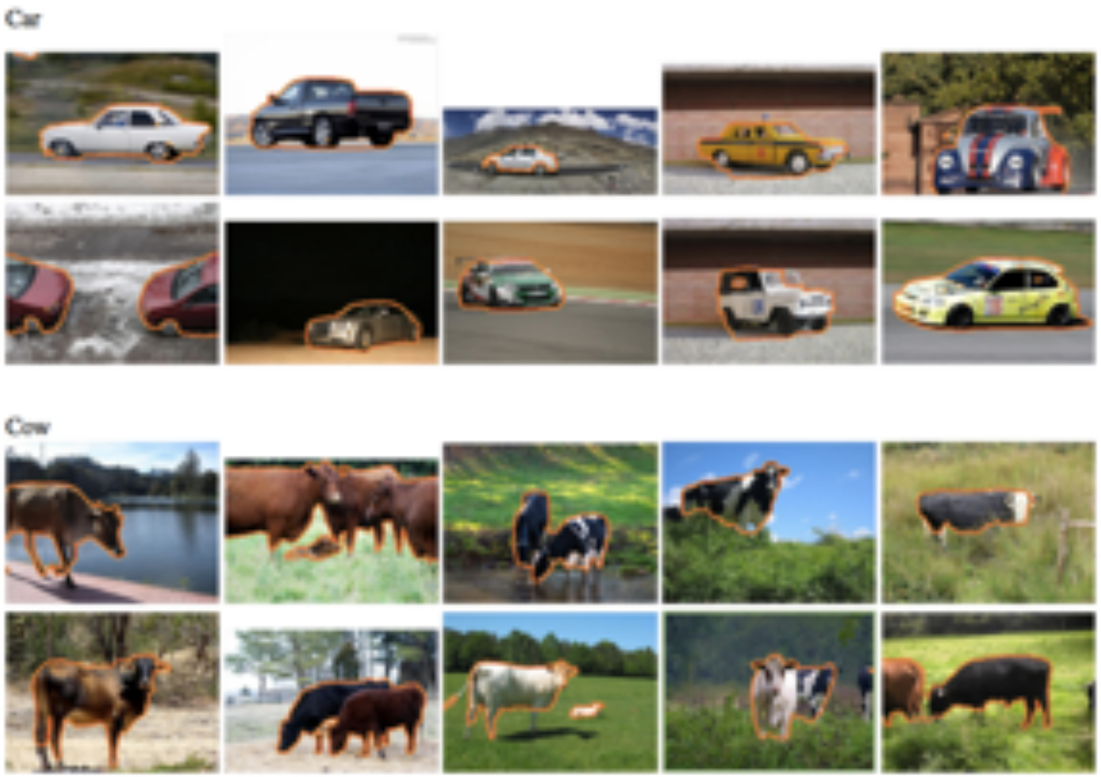


Image Co-Segmentation via Consistent Functional Maps
Wang, Huang, Guibas, CVPR 2013

Other Extensions

- **Tangent Vector Field processing**

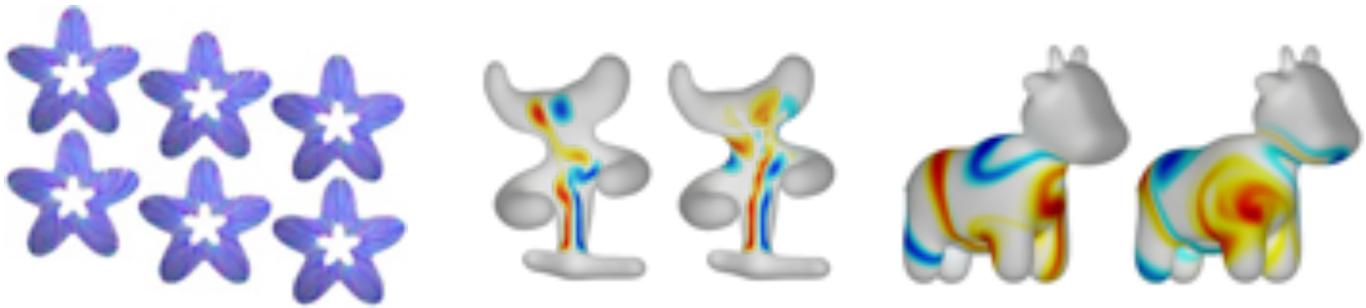
  An Operator Approach to Tangent Vector Field Processing Azencot, Ben-Chen, Chazal, Ovsjanikov, SGP, 2013.

- **Measuring Differences between shapes**

  Map-Based Exploration of Intrinsic Shape Differences and Variability Rustamov, Ovsjanikov, Azencot, Ben-Chen, Chazal, Guibas, SIGGRAPH 2014

- **Maps Between Partial shapes**

  Partial Functional Correspondence, Rodolà, Cosmo, A. Bronstein, Torsello, Cremers, CGF 2017



Azencot et al., SGP 2014

Some Open Problems

- What is the optimal choice of basis?
- How to guarantee a *continuous* pointwise map?
- What are better *deformation models?*
- Shape interpolation *without* converting to p2p?

Conclusions

- Functional maps provide an efficient way to encode "generalized" mappings.
- Can be computed in practice with simple (least squares) optimization.
- Many different constraints can be incorporated: pointwise maps, consistency in collections, etc.
- Recent work incorporating learning of descriptors.

Thank You

Questions?

Acknowledgements:

A. Poulenard, M.-J. Rakotosaona, Y. Ponty, J.-M. Rouffosse, A. Sharma, S. Melzi, E. Rodolà, J. Ren, P. Wonka …. Work supported by KAUST OSR Award No. CRG-2017-3426, a gift from Nvidia and the ERC Starting Grant StG-2017-758800 (EXPROTEA)