Efficient regularization of functional map computations

Flows, mappings and shapes VMVW03 Workshop
December 12th, 2017

Maks Ovsjanikov

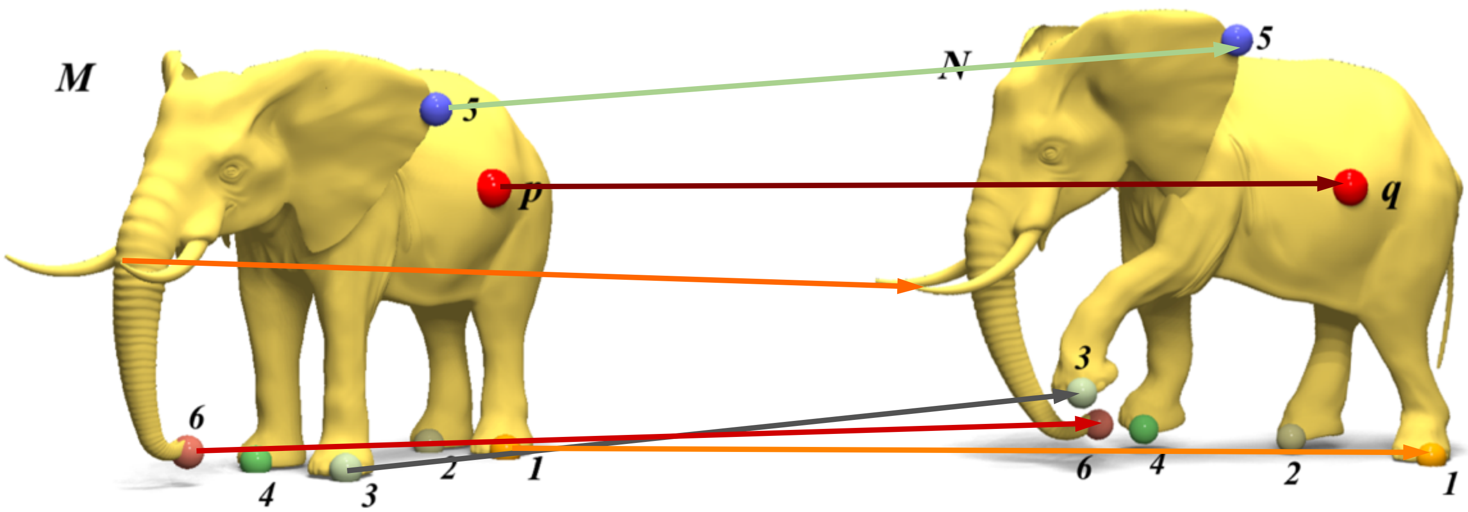
Joint with: *E. Corman, D. Nogneng, R. Huang, M. Ben-Chen, J. Solomon, A. Butscher, L. Guibas…*

ÉCOLE POLYTECHNIQUE
UNIVERSITÉ PARIS-SACLAY

Shape Matching

Problem:

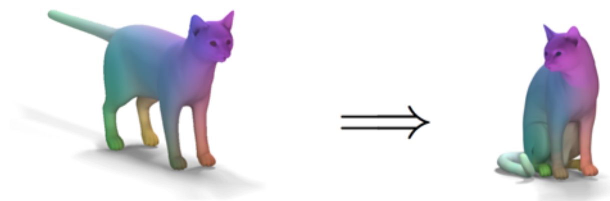
Given a pair of shapes, find corresponding points.



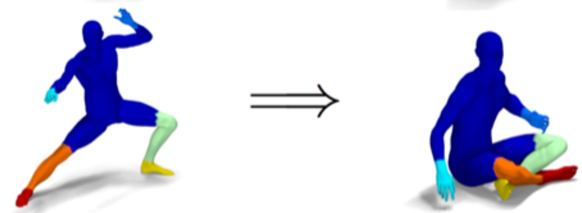
Why Shape Matching

Given a correspondence, we can transfer:

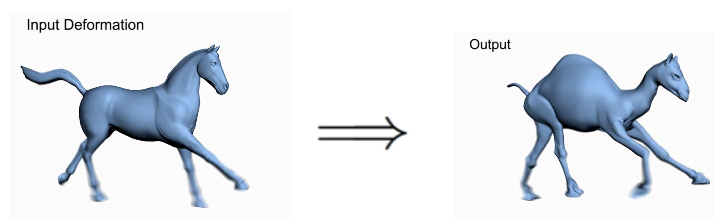
texture and parametrization



segmentation and labels



deformation

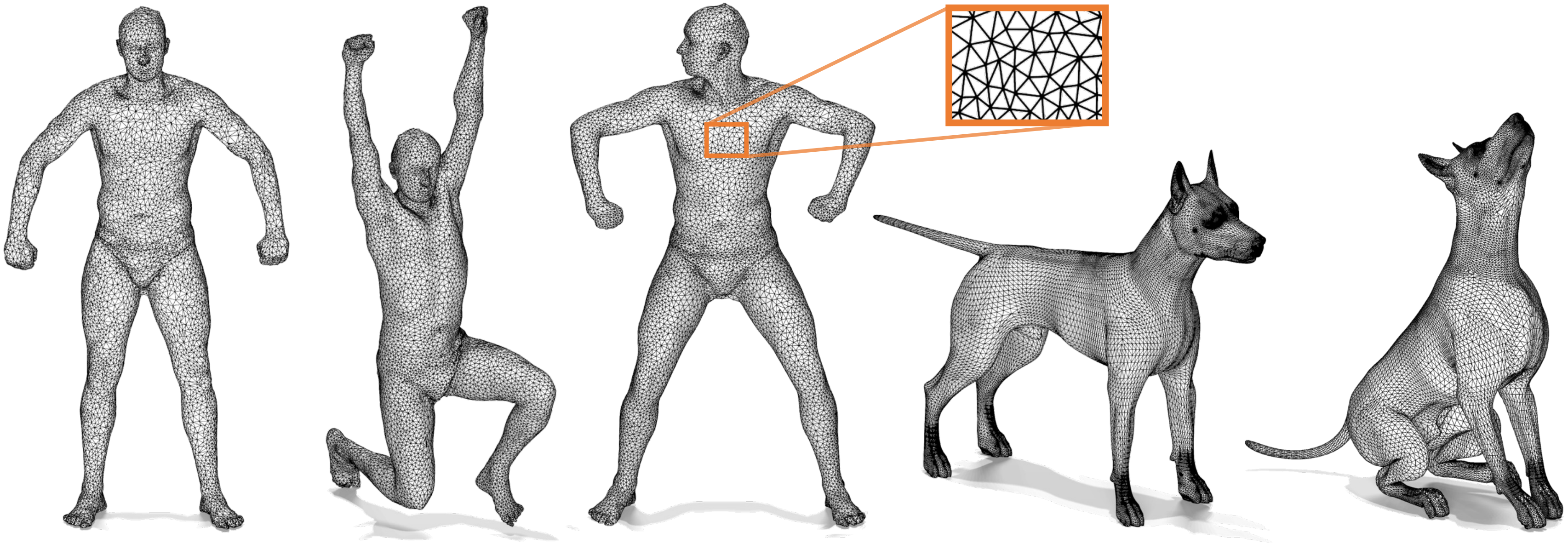


Sumner et al. '04.

Other applications: shape interpolation, reconstruction ...

What is a Shape?

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

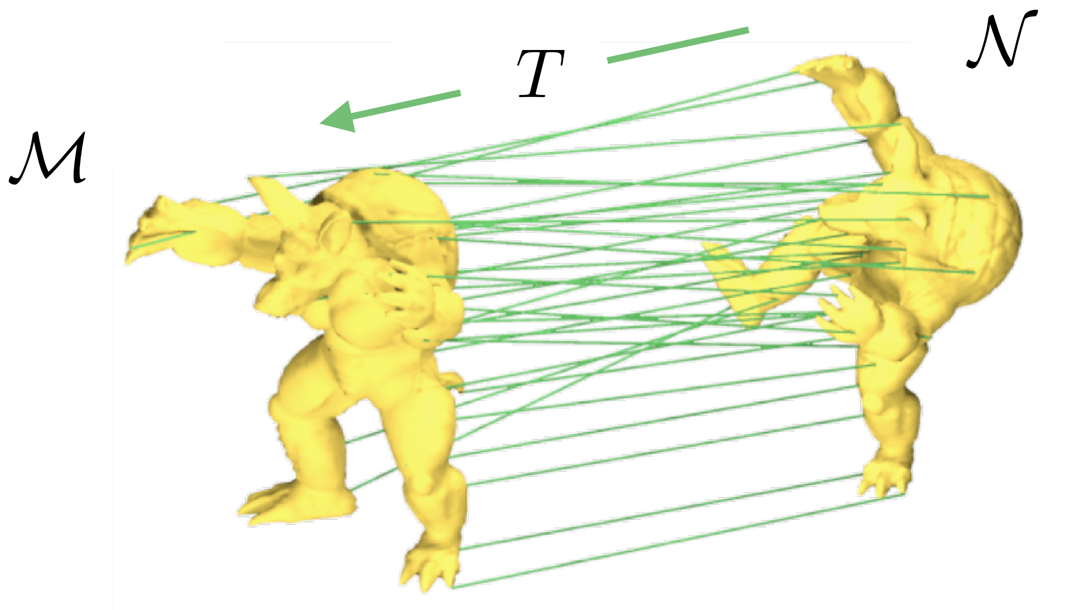


5k – 200k triangles

Shapes from the SCAPE, TOSCA and FAUST datasets

Functional Approach to Mappings

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



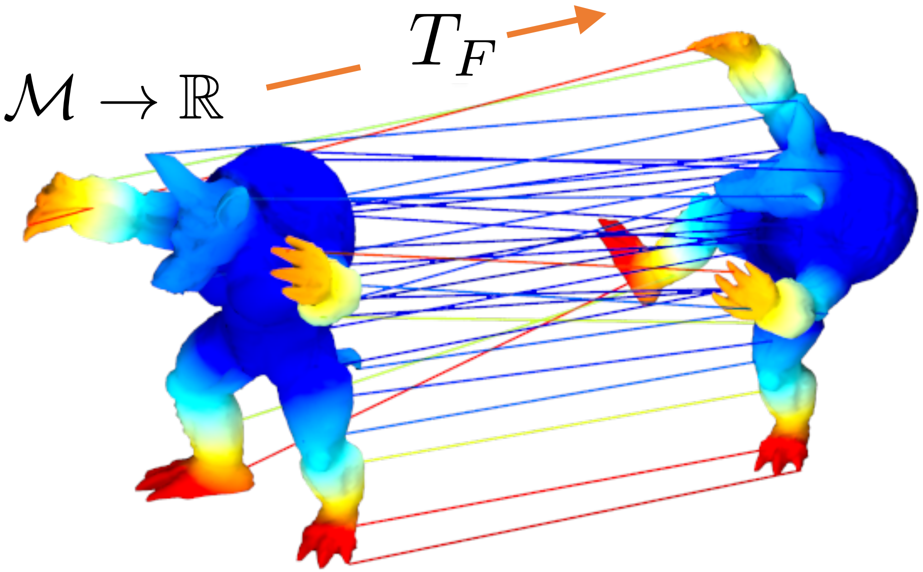
The map T induces a functional correspondence:

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

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 bijection $T : \mathcal{N} \to \mathcal{M}$

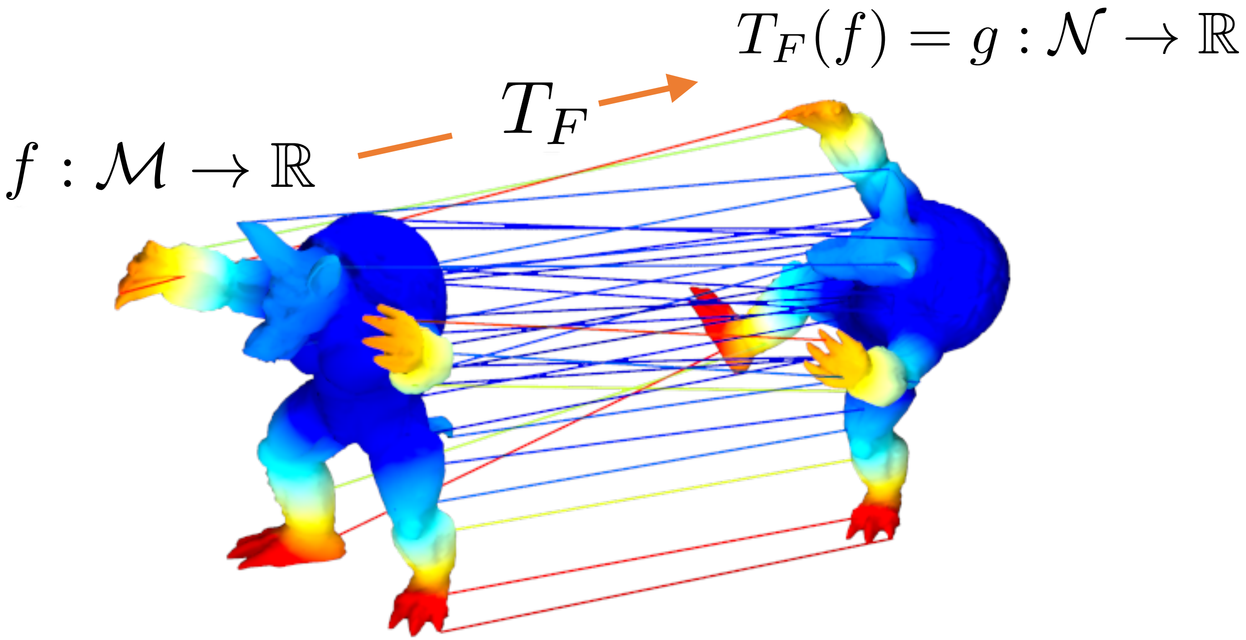


The map T induces a functional correspondence:

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

Functional Approach to Mappings

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

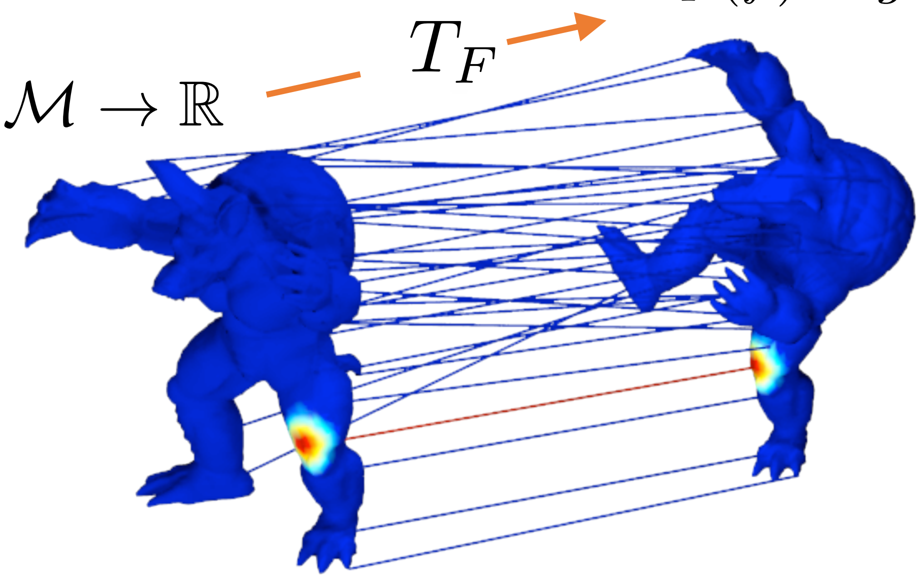


The induced functional correspondence is linear:

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

Functional Map Representation

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

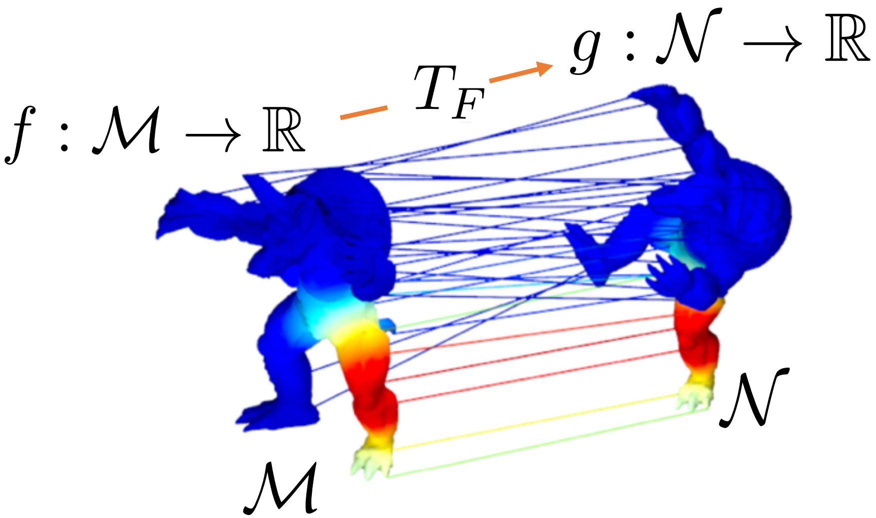


The induced functional correspondence is complete.

Can recover T from T_F.

Observation

Assume that both: $f \in \mathcal{H}_0^1(\mathcal{M}), g \in \mathcal{H}_0^1(\mathcal{N})$



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:

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

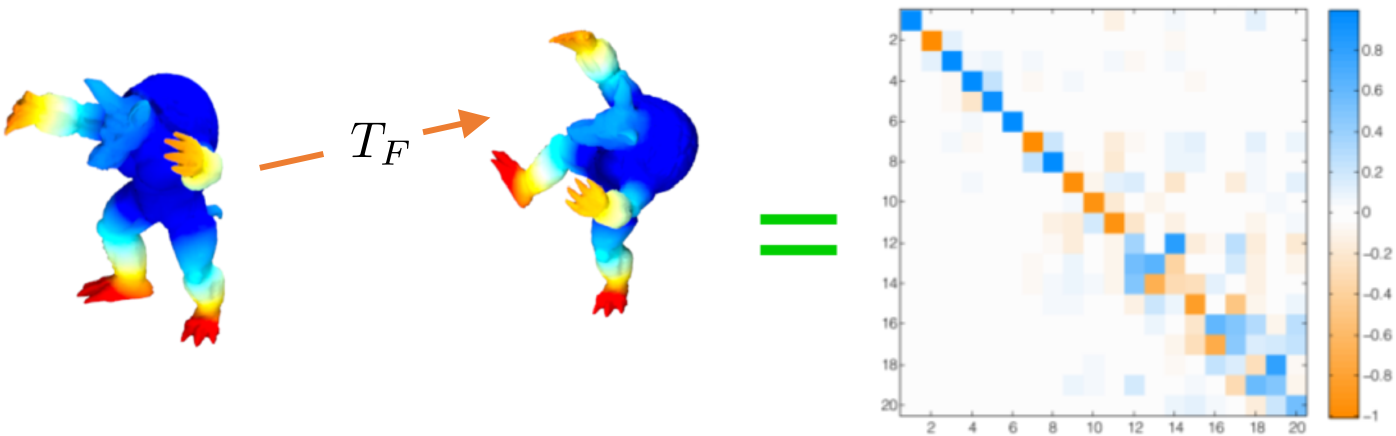
- Minimize Dirichlet energy: $\int_M \|\nabla \phi_i(x)\|^2 d\mu$
- Ordered by eigenvalues and provide a natural notion of *scale*.

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

Functional Map Representation

Since the functional Mapping T_F is linear:

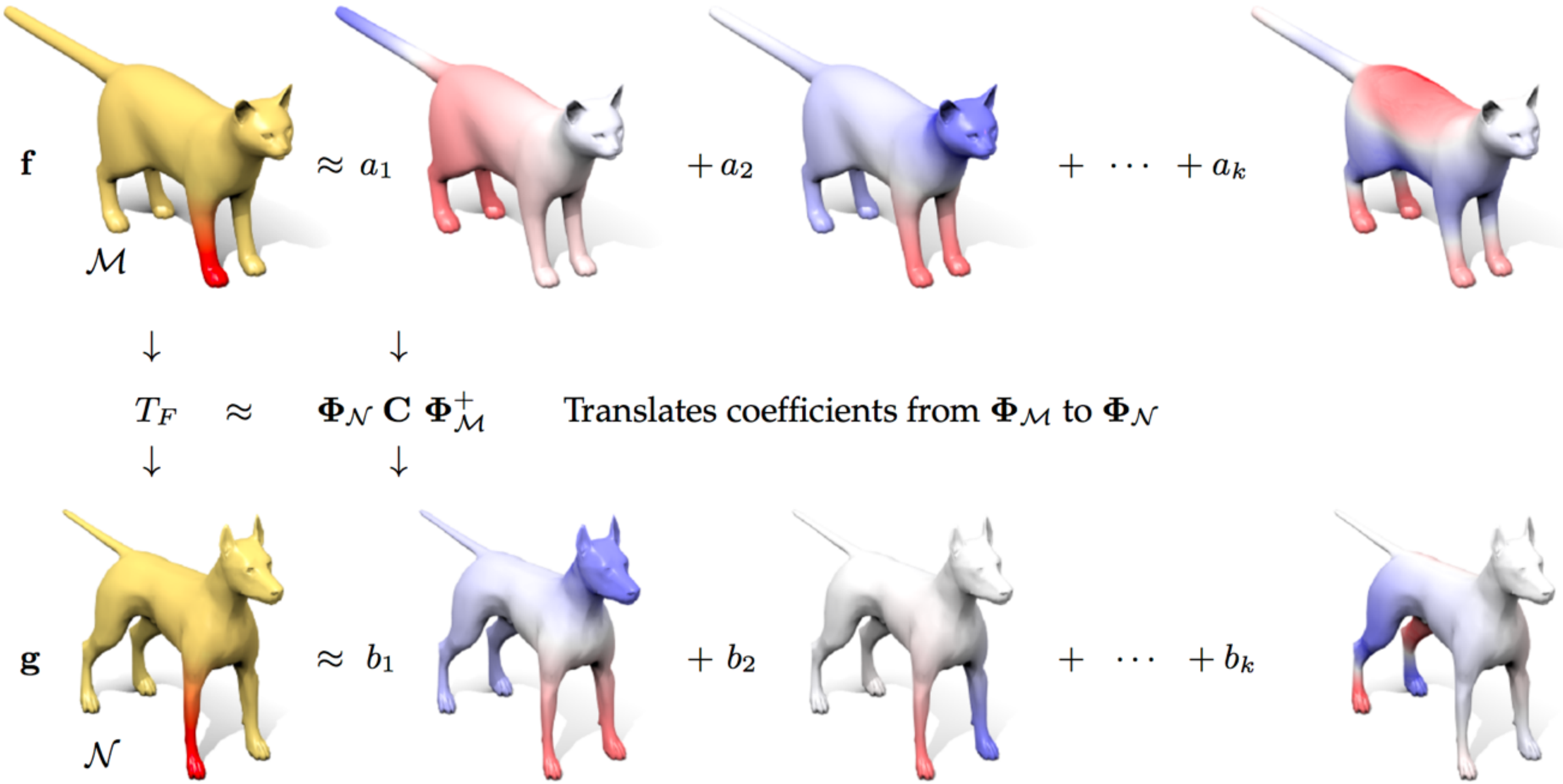
$$T_F(\alpha_1 f_1 + \alpha_2 f_2) = \alpha_1 T_F(f_1) + \alpha_2 T_F(f_2)$$



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

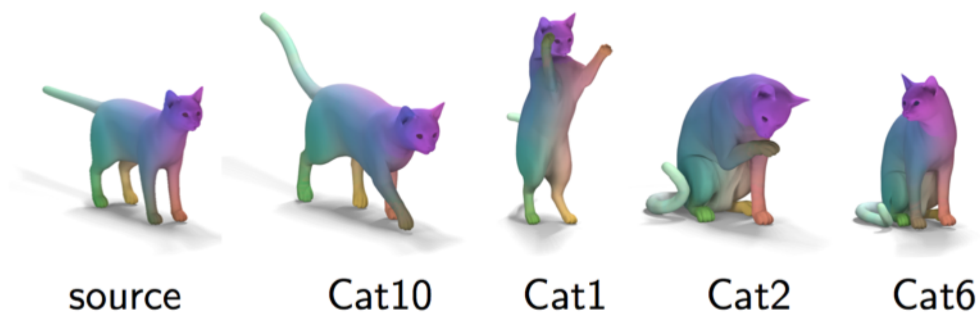


Functional map:

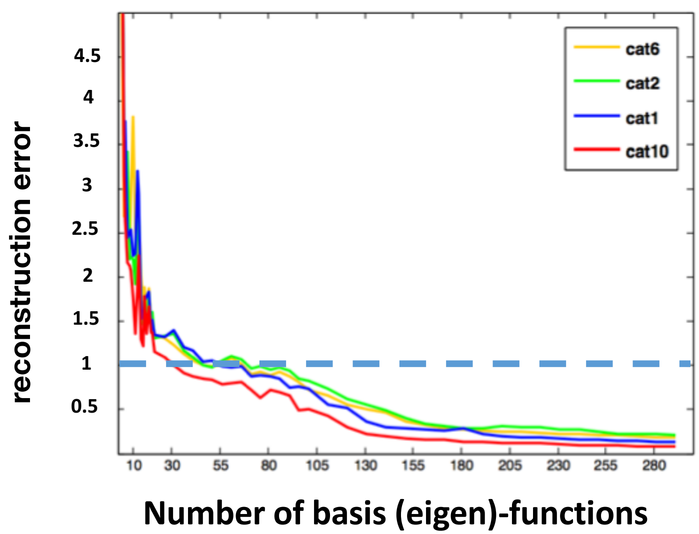
matrix C that translates coefficients from $\Phi_{\mathcal{M}}$ to $\Phi_{\mathcal{N}}$.

Reconstructing from LB basis

Map reconstruction error using a fixed size matrix.



27.9k vertices



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 do not 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 $f = \sum_i a_i \phi_i^{\mathcal{M}}, \quad g = \sum_j b_j \phi_j^{\mathcal{N}}$

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

Basic Pipeline

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

1. Compute the multi-scale bases for functions on the two shapes. Store them in matrices: $\Phi_{\mathcal{M}}, \Phi_{\mathcal{N}}$

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

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

   $\Delta_{\mathcal{M}}, \Delta_{\mathcal{N}}$: Laplacian operators.

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

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

Structural Questions for Today

Can we promote functional maps to be:

1. Closer to **point-to-point maps**?
2. Closer to being **bijective**?
3. Encode **extrinsic** (embedding-dependent) information?

While retaining the computational advantages.

Making Functional Maps Point-to-Point

Question 1:

When does a linear functional mapping correspond to a pull-back by a point-to-point map?

Making Functional Maps Point-to-Point

Question 1a:

When does a linear functional mapping correspond to a pull-back by a point-to-point map?

Question 1b:

Given a *single perfect* descriptor that identifies each point uniquely, why does our system not recover the map?

$$C_{\text{opt}} = \arg\min_{C} \|C\mathbf{a} - \mathbf{b}\|$$

We're not using the full information from the descriptors!

Making Functional Maps Point-to-Point

(Main) Question:

When does a linear functional mapping correspond to a pull-back by a point-to-point map?

(Known) Theoretical result:

A functional map is point-to-point iff it preserves pointwise products of functions:

$$C(fh) = C(f)C(h) \ \forall \ f, h \quad (fh)(x) = f(x)h(x)$$

J. von Neumann, *Zur operatoren methode in der klassichen Mechanik*, Ann. of Math.(2) 33 (1932)

Making Functional Maps Point-to-Point

(Known) Theoretical result:

A functional map is point-to-point iff it preserves pointwise products of functions:

$$C(fh) = C(f)C(h) \; \forall \; f, h \quad (fh)(x) = f(x)h(x)$$

Can we use this in the algorithm?

Challenges:

1) Leads to non-convex energy.
2) Large number of constraints
   (mixing primal and spectral domains)

Would like to exploit this fact without losing convexity.

Making Functional Maps Point-to-Point

(Known) Theoretical result:

A functional map is point-to-point iff it preserves pointwise products of functions:

$$C(fh) = C(f)C(h) \ \forall \ f, h \quad (fh)(x) = f(x)h(x)$$

Approach:

Consider the linear operator:

$$S_f(h) = fh \quad S_f(h)(x) = f(x)h(x)$$

Making Functional Maps Point-to-Point

(Known) Theoretical result:

A functional map is point-to-point iff it preserves pointwise products of functions:

$$C(fh) = C(f)C(h) \ \forall \ f, h \quad (fh)(x) = f(x)h(x)$$

Approach:

Consider the linear operator:

$$S_f(h) = fh \quad S_f(h)(x) = f(x)h(x)$$

given a pair of functions f_k, g_k for which we expect: $C(f_k) = g_k$, the above implies:

$$C \circ S_{f_k}(h) = S_{g_k} \circ C(h) \ \forall \ h$$

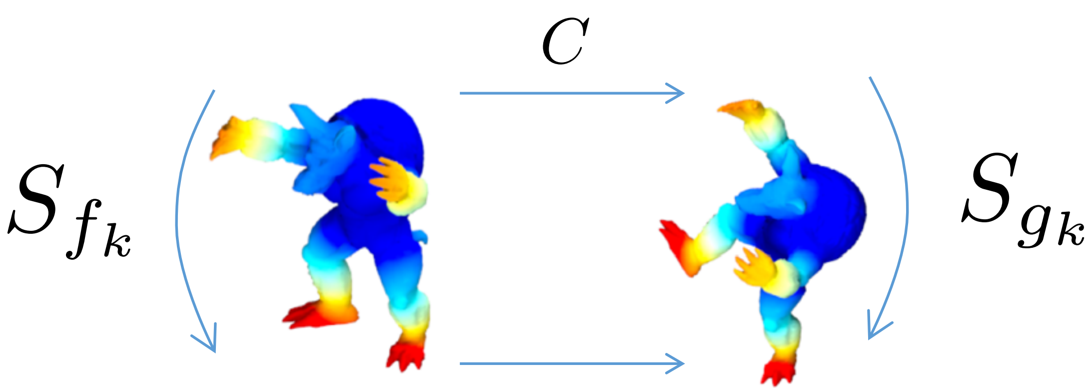
Making Functional Maps Point-to-Point

(Known) Theoretical result:

A functional map is point-to-point iff it preserves pointwise products of functions.

Approach

Represent descriptor functions via their action on functions through multiplication.



$$C(f_k h) = g_k C(h) \iff C S_{f_k} = S_{g_k} C$$

Making Functional Maps Point-to-Point

Approach

Represent descriptor functions via their action on functions through multiplication.

Theorem 1 (even in the reduced basis):

$$CF = GC, \text{ and } C\mathbf{1} = \mathbf{1} \Longrightarrow Cf = g$$

where F, G are the multiplicative operators of *f*, *g*.

Theorem 2:

If *f*, *g* have the same values, then in the full basis for any doubly stochastic matrix Π :

$$\Pi f = g \Longleftrightarrow \Pi F = G\Pi$$

where F, G are the multiplicative operators of *f*, *g*.

Extended Basic Pipeline

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

1. Compute the multi-scale bases for functions on the two shapes. Store them in matrices: $\Phi_{\mathcal{M}}, \Phi_{\mathcal{N}}$

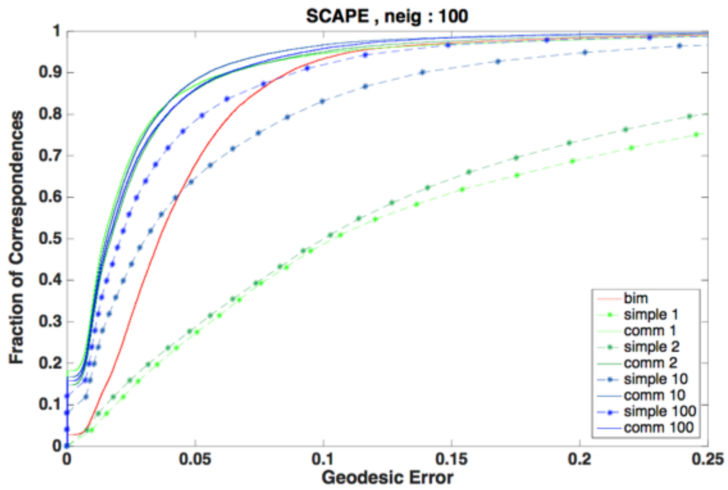
2. Compute descriptor functions (e.g., Gauss curvature) on $\mathcal{M}, \mathcal{N}$. Express them in $\Phi_{\mathcal{M}}, \Phi_{\mathcal{N}}$, as columns of : $\mathbf{A}, \mathbf{B}$

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

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

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

Results with extended pipeline



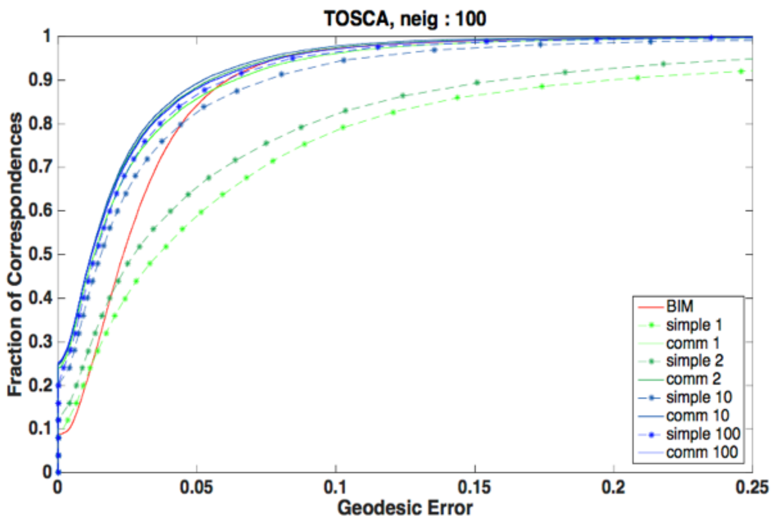
before

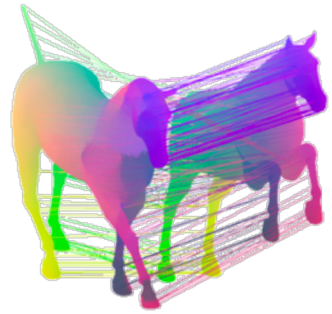
after

Incorporating multiplicative operators improves results significantly.

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

Results with extended pipeline





before

after

Incorporating multiplicative operators improves results significantly.

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

Improving Map Bi-directionality

Question 2a:

Can we remove the direction bias?

$$C_{\text{opt}} = \arg\min_{C} \|C\mathbf{A} - \mathbf{B}\|^2 + \|C\Delta_{\mathcal{M}} - \Delta_{\mathcal{N}}C\|^2$$

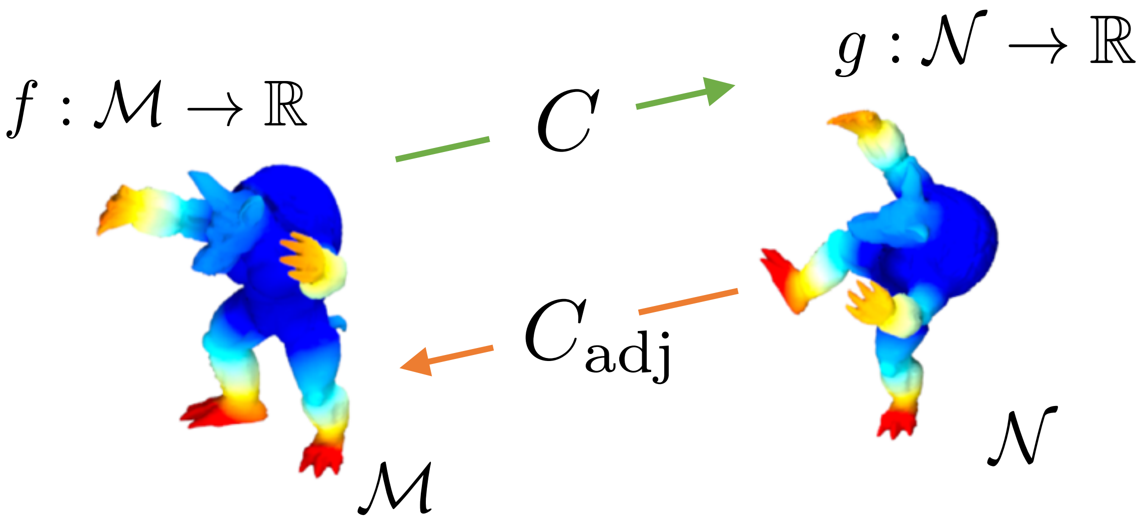
Source/target shapes are not interchangeable.

Question 2b:

What does the functional map adjoint/transpose encode?

Functional Map Adjoint

Given a functional map C and choice of inner products on the source/target, the adjoint C_{adj} is defined implicitly:



$$< C(f), g) >_{\mathcal{N}} = < f, C_{\text{adj}}(g) >_{\mathcal{M}} \; \forall f, g$$

Improving Map Bi-directionality

We define the adjoints based on two inner products:

$$< f, g >_{L^2} = \int fg d\mu \qquad C_{\text{adj}}^{L^2} = C^T$$

$$< f, g >_{H^1} = \int < \nabla f, \nabla g > d\mu \qquad C_{\text{adj}}^{H^1} = \Delta_{\mathcal{M}}^{+} C^T \Delta_{\mathcal{N}}$$

Theorem:

If a functional map C comes from a pointwise bijection T between surfaces then:

- T is locally area-preserving if and only if $C^{-1} = C_{\text{adj}}^{L^2}$
- T is conformal if and only if $C^{-1} = C_{\text{adj}}^{H^1}$

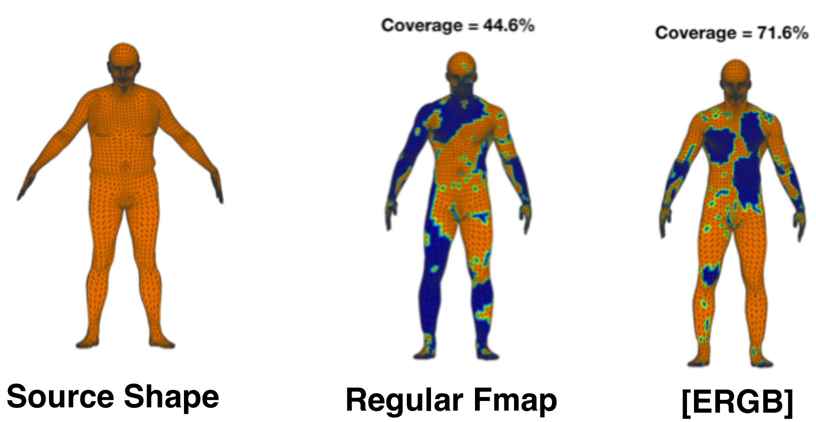
Adjoint Map Representation for Shape Analysis and Matching, Huang, O., SGP 2017

Improving Map Bi-directionality

Removing direction bias:

$$\underset{C_{MN},C_{NM}}{\arg\min} \quad E_1(C_{MN}) + E_2(C_{NM}) + \|C_{MN}C_{NM} - Id\|$$

Leads to a non-linear, non-convex energy.



Coupled functional maps, Eynard et al., 3DV 2016

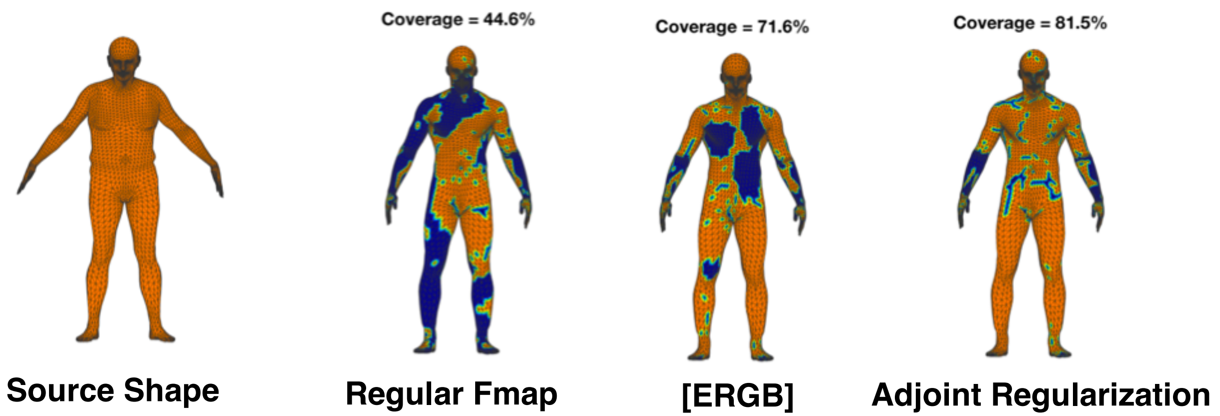
Improving Map Bi-directionality

Removing direction bias:

$$\arg\min_{C_{MN}, C_{NM}} \quad E_1(C_{MN}) + E_2(C_{NM}) + \|C_{MN} - C_{NM}^T\| + \|\Delta_N C_{MN} - C_{NM}^T \Delta_M\|$$

Overall energy remains quadratic in C_{MN}, C_{NM}.

Tends to promote invertibility and near-isometry.



Adjoint Map Representation for Shape Analysis and Matching, Huang, O., SGP 2017

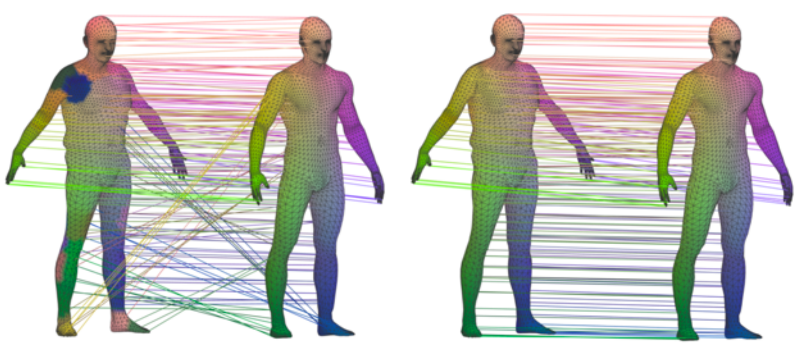
Improving Map Bi-directionality

Removing direction bias:

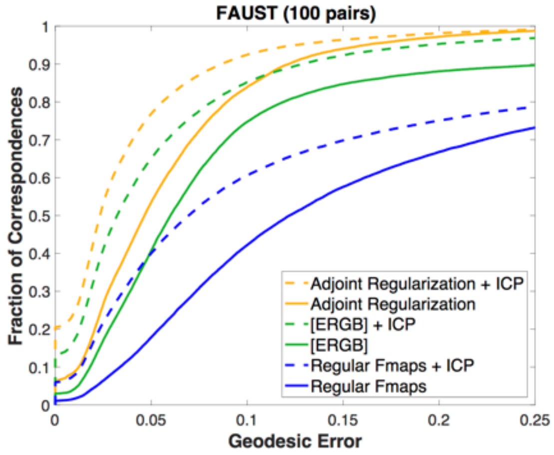
$$\arg\min_{C_{MN}, C_{NM}} \; E_1(C_{MN}) + E_2(C_{NM}) + \|C_{MN} - C_{NM}^T\| + \|\Delta_N C_{MN} - C_{NM}^T \Delta_M\|$$

Overall energy remains quadratic in C_{MN}, C_{NM}.

Tends to promote invertibility and near-isometry.



Regular Fmap   Adjoint Regularization



Adjoint Map Representation for Shape Analysis and Matching, Huang, O., SGP 2017

Encoding Extrinsic Information

Question 3a:

Can we encode extrinsic (embedding-dependent) information?

Question 3b:

Surfaces are encoded via:

- First fundamental form (intuitively: geodesics).
- Second fundamental form (intuitively: principal curvatures).

Can we translate this into functional representation?

Encoding More Complex Data

Recall our earlier result:

Lemma 1:

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

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

The first fundamental form (geodesics) is fully encoded by the Laplacian.

Encoding Extrinsic Information

Intuitively:

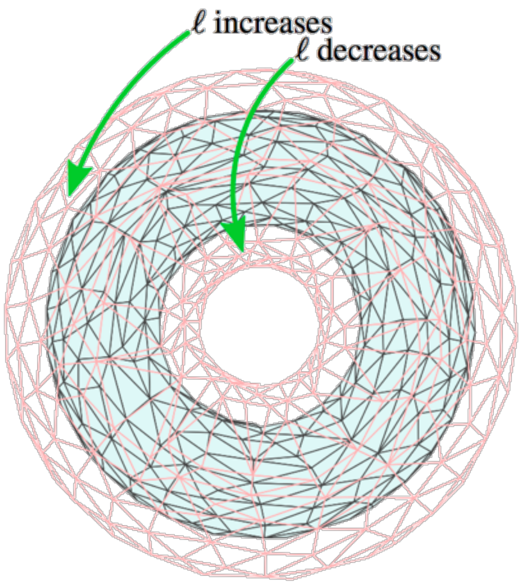
Curvature is encoded by the change of the local lengths along the normal direction.

A bit more formally:

Given a family of shapes

$$F_t : M \to \mathbb{R}^3 \quad \text{s.t.} \quad \frac{\partial F_t}{\partial t}(p) = n(p)$$

then: $$\left.\frac{\partial g_{ij}}{\partial t}\right|_{t=0} = 2h_{ij}|_{t=0}$$



Functional Characterization of Intrinsic and Extrinsic Geometry, Corman et al. ACM TOG 2017

Encoding the Second Fundamental Form

Lemma 2:

A map preserves *the second fundamental form*, if and only if the functional map commutes with the *derivative of the Laplacian* along the normal:

$$C \left. \frac{\partial \Delta_{\mathcal{M}+t\mathbf{n}}}{\partial t} \right|_{t=0} = \left. \frac{\partial \Delta_{\mathcal{N}+t\mathbf{n}}}{\partial t} \right|_{t=0} C$$

The second fundamental form (principal curvatures) is fully encoded by the Laplacian *in the normal direction*.

Functional Characterization of Deformation Fields (major revision, arxiv 1709.09701)

Encoding the Second Fundamental Form

Define a functional operator E^n implicitly such that:

$$\int_M \langle \nabla g, \nabla E^n(f) \rangle \, d\mu = \int_M \mathcal{L}_{\mathbf{n}}\mathbf{g}(\nabla g, \nabla f) d\mu \quad \forall f, g$$

Where $\mathcal{L}$ is the infinitesimal strain tensor:

$$\mathcal{L}_{\mathbf{n}}\mathbf{g}(x, y) = \langle x, \nabla_y \mathbf{n} \rangle + \langle \nabla_x \mathbf{n}, y \rangle$$

We can now require the functional map to commute:

$$\| C_{MN} E^n_M - E^n_N C_{MN} \|$$

Functional Characterization of Deformation Fields (major revision, arxiv 1709.09701)

Encoding the Second Fundamental Form

Theorem

Two surfaces are related by a rigid motion if and only if:

$$\|C_{MN}\Delta_M - \Delta_N C_{MN}\| + \|C_{MN}E^n_M - E^n_N C_{MN}\| = 0$$

- The Laplacian encodes the first fundamental form.
- Metric change along the normal encodes the second.

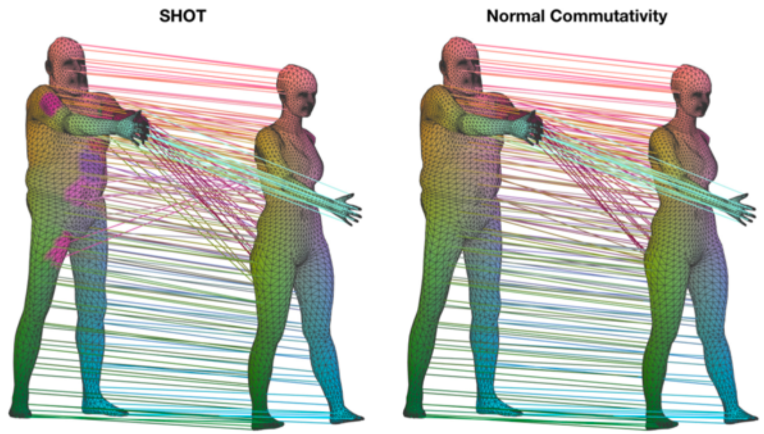
Encoding the Second Fundamental Form

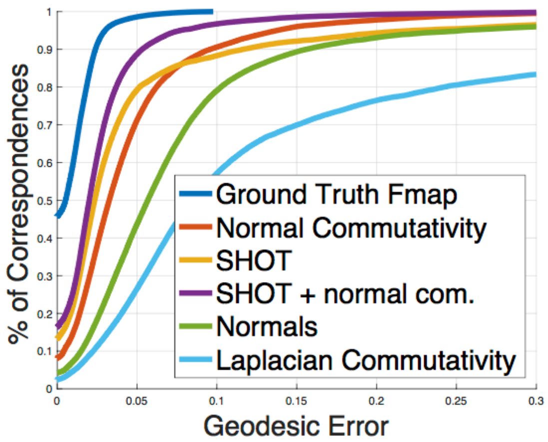
Theorem

Two surfaces are related by a rigid motion if and only if:

$$\|C_{MN}\Delta_M - \Delta_N C_{MN}\| + \|C_{MN}E^n_M - E^n_N C_{MN}\| = 0$$

Objective remains quadratic in *C*.
Promotes preservation of mean/principal curvatures.

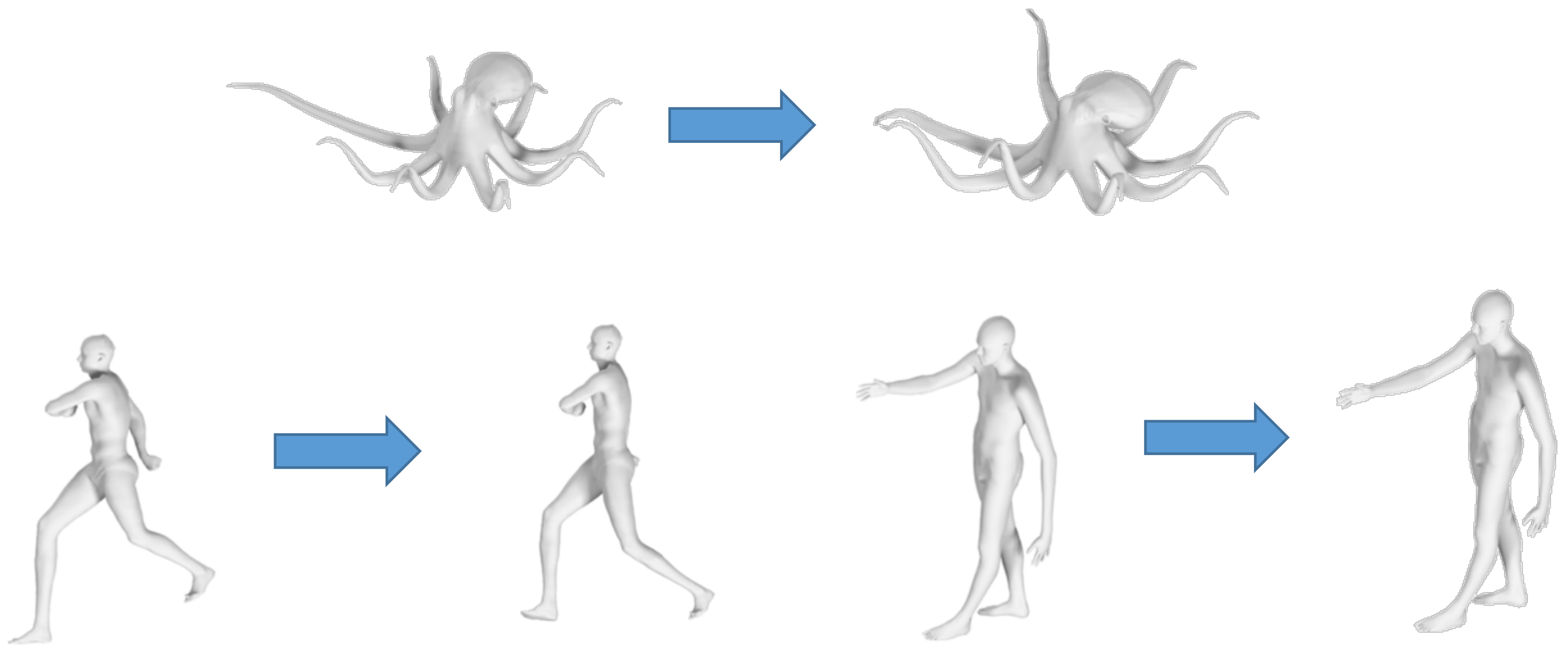




Functional Characterization of Deformation Fields (major revision, arxiv 1709.09701)

Metric-Prescribed Shape Deformation

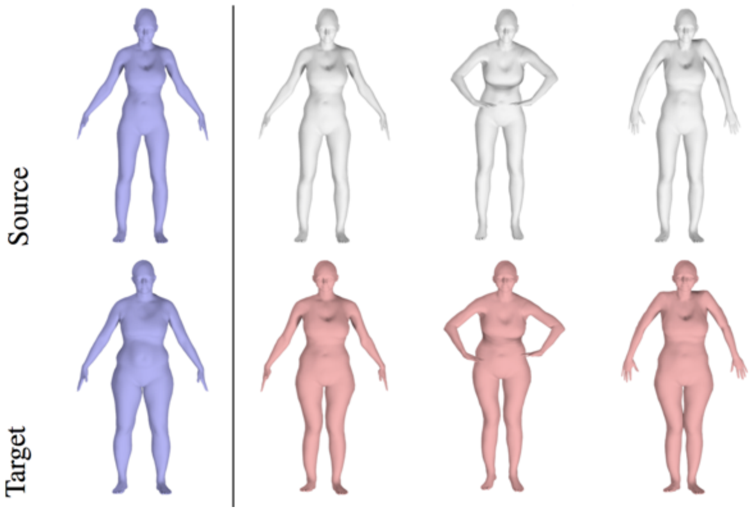
Same machinery allows *intrinsic symmetrization* without pointwise correspondences.

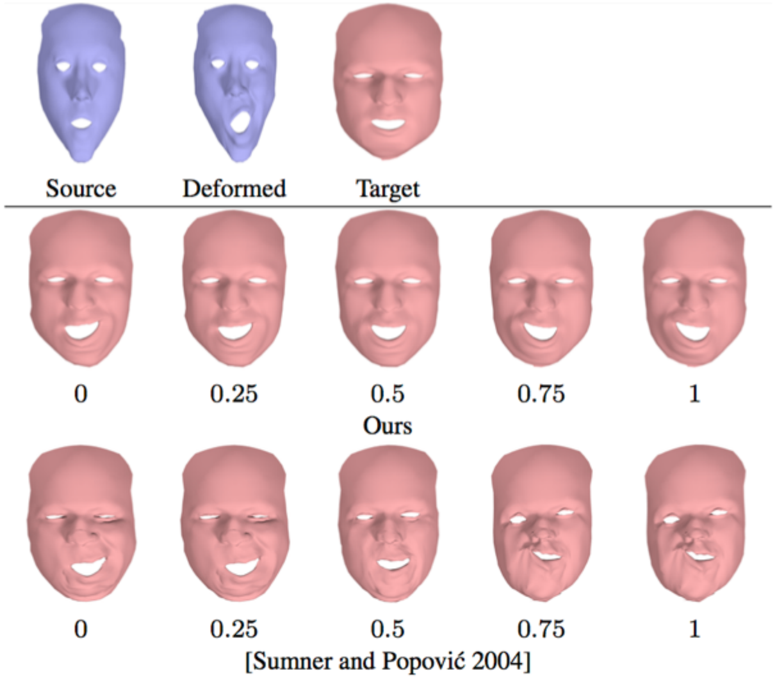


Functional Characterization of Deformation Fields (major revision, arxiv 1709.09701)

Functional Deformation Fields

… and deformation transfer *without pointwise correspondences.*





Functional Characterization of Deformation Fields (major revision, arxiv 1709.09701)

Follow-ups and References

- SIGGRAPH Course Website:

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

  or http://bit.do/fmaps2017

- Contains detailed course notes and **sample code**

  http://bit.do/fmaps2017_notes

- Some references and follow-up works:

  Functional maps: a flexible representation of maps between shapes, ACM SIGGRAPH 2012

  Informative Descriptor Preservation via Commutativity for Shape Matching, Eurographics 2017

  Adjoint Map Representation for Shape Analysis and Matching, SGP 2017

  Functional Characterization of Deformation Fields, arxiv 1709.09701, 2017

Thank you!

Questions?

Acknowledgements:

Etienne Corman, Dorian Nogneng, Ruqi Huang, Mirela Ben-Chen, Justin Solomon, Leonidas Guibas, Adrien Butscher, Omri Azencot, Frédéric Chazal….

Work supported by the CNRS chaire d'excellence, Marie Curie Career Integration Grant, chaire Jean Marjoulet, and a Google Focused Research Award.