Improved Functional Mappings via Product Preservation

Workshop: Imaging and Vision
March 15th, 2018

Maks Ovsjanikov

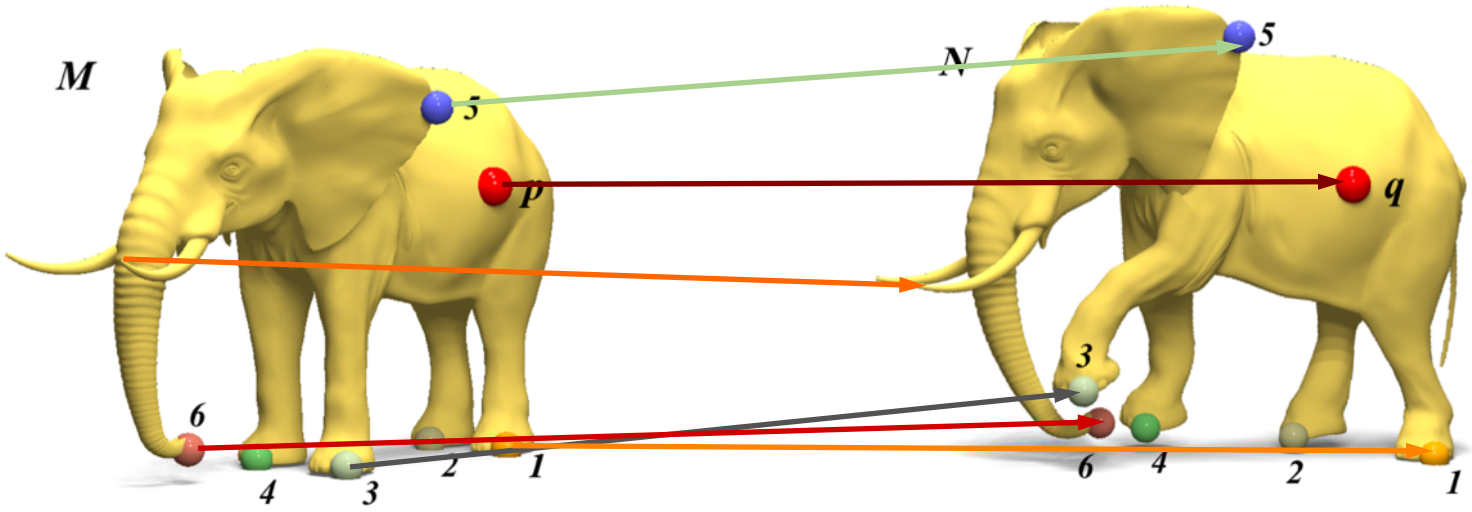
Joint with: *D. Nogneng, Simone Melzi, Emanuele Rodolà, Umberto Castellani, Michael Bronstein…*

ÉCOLE POLYTECHNIQUE
UNIVERSITÉ PARIS-SACLAY

Shape Matching

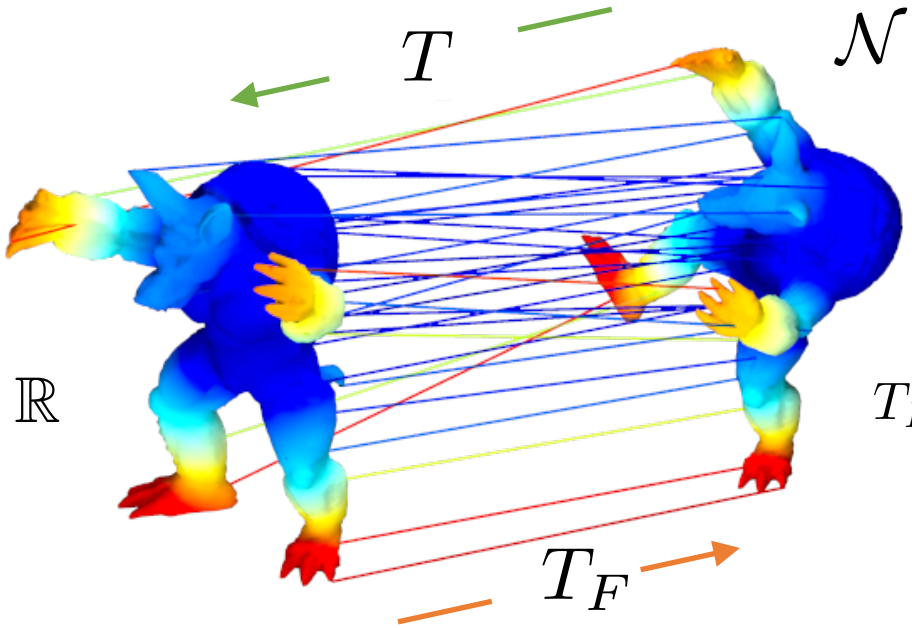
Problem:

Given a pair of shapes, find corresponding points.



Functional Approach to Mappings

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



The map T induces a functional correspondence:

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

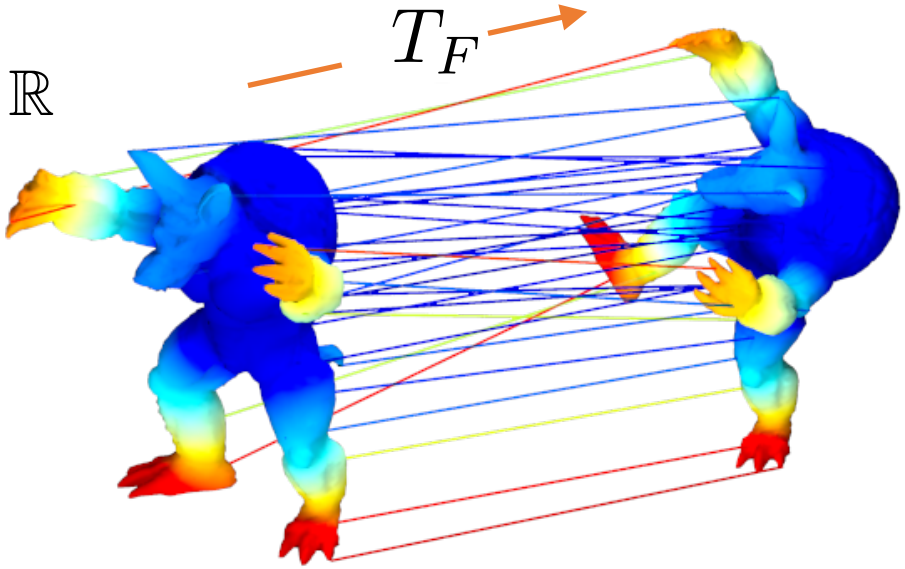
Functional maps: a flexible representation of maps between shapes, *O., Ben-Chen, Solomon, Butscher, Guibas*, SIGGRAPH 2012

Functional Approach to Mappings

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

$$T_F(f) = g : \mathcal{N} \to \mathbb{R}$$

$$f : \mathcal{M} \to \mathbb{R}$$



The induced functional correspondence is linear:

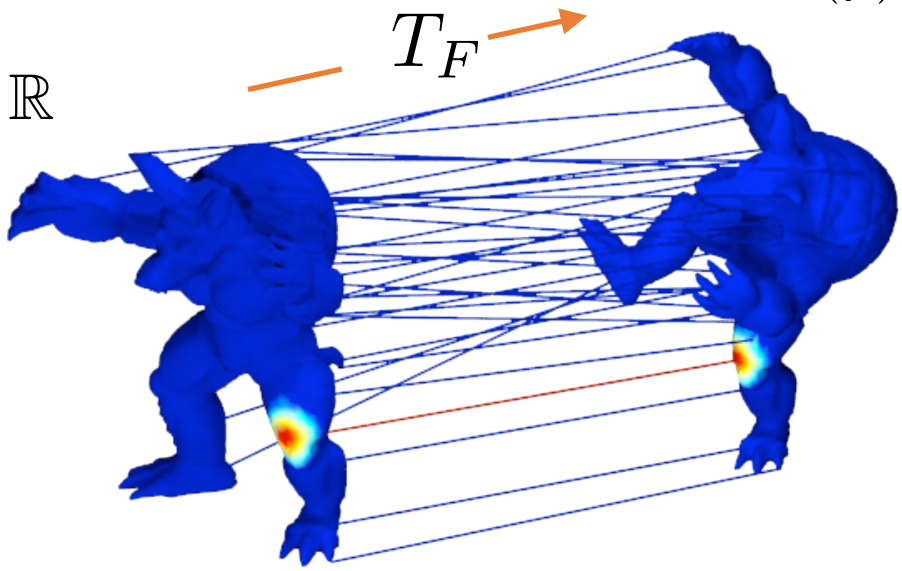
$$T_F(\alpha_1 f_1 + \alpha_2 f_2) = \alpha_1 T_F(f_1) + \alpha_2 T_F(f_2)$$

Functional Map Representation

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

$T_F(f) = g : \mathcal{N} \to \mathbb{R}$

$f : \mathcal{M} \to \mathbb{R}$



The induced functional correspondence is complete.

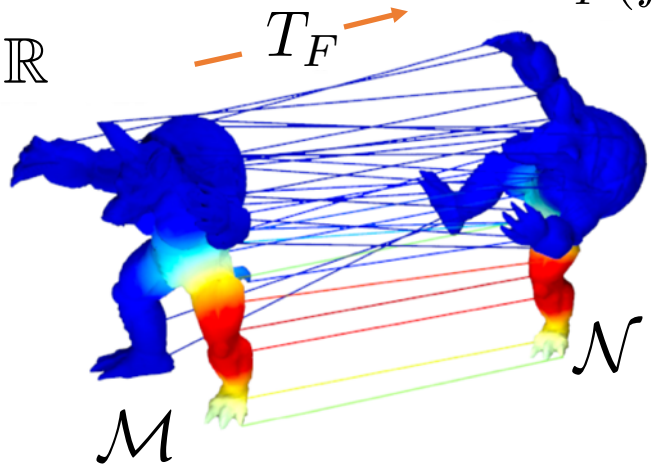
Can recover T from T_F.

Observation

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

$T_F(f) = g : \mathcal{N} \to \mathbb{R}$

$f : \mathcal{M} \to \mathbb{R}$

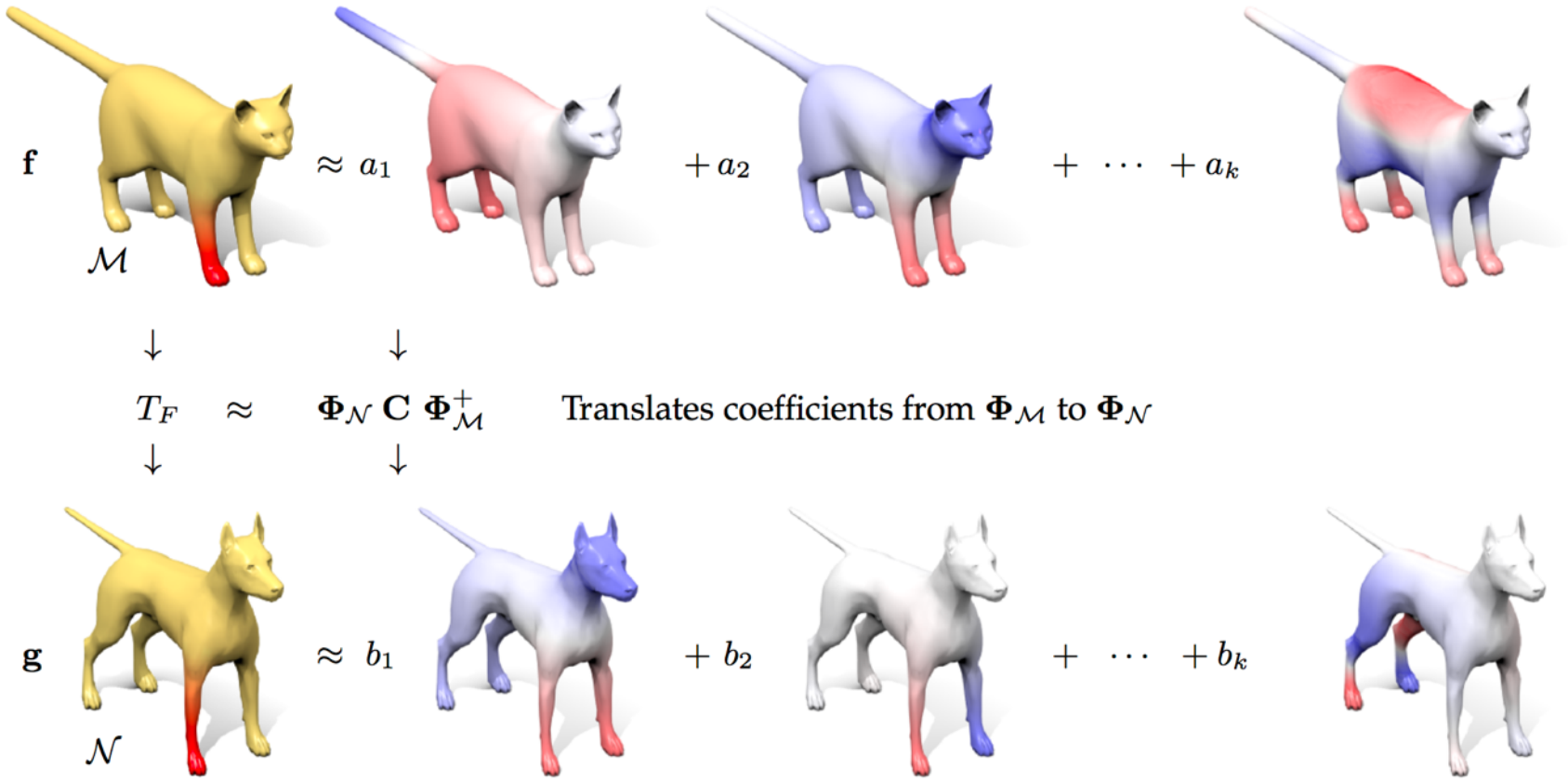


Express both f and $T_F(f)$ in terms of *basis functions:*

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

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

Functional Map Definition



Functional map:

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

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.
- Ordered by eigenvalues and provide a natural notion of *scale*.
- Can be computed efficiently, with a sparse matrix eigensolver.

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

   $\Delta_{\mathcal{M}}, \Delta_{\mathcal{N}}$: 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

Questions for Today

Can we enforce functional maps to be:

1. Closer to **point-to-point maps**?
2. Be able to transfer **higher-frequency 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 linear transformation across function spaces is a pull-back of a point-to-point map iff it preserves pointwise products of functions:

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

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:

$$C(f \odot h) = C(f) \odot C(h), \ \forall \ f, h$$

Making Functional Maps Point-to-Point

(Known) Theoretical result:

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

$$C(f \odot h) = C(f) \odot C(h), \ \forall \ f, h$$

Approach:

Suppose we are given a pair of (descriptor) functions f_k, g_k for which we expect: $C(f_k) = g_k$.

Then, the above implies:

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

Let: $S_f(h) = f \odot h$

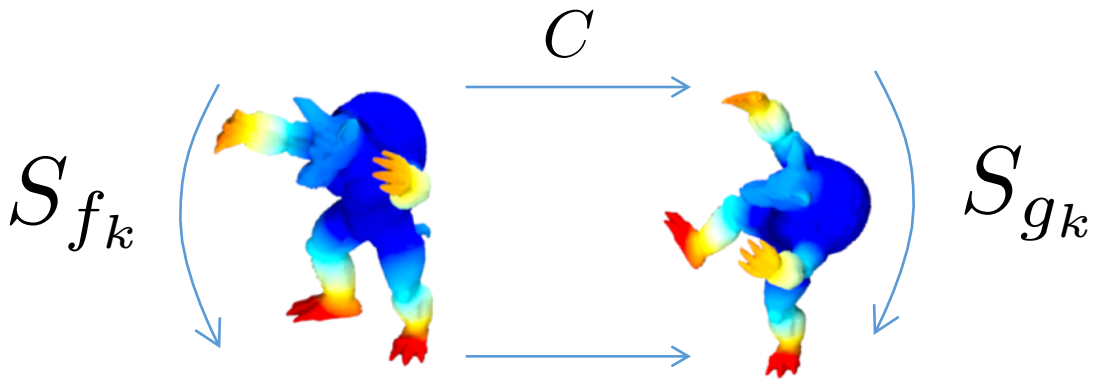
Making Functional Maps Point-to-Point

(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.



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

Making Functional Maps Point-to-Point

Approach

Represent descriptor functions via their action on functions through multiplication.

Theorem 1 (in both full **or** 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: $\Phi_{\mathcal{M}}, \Phi_{\mathcal{N}}$

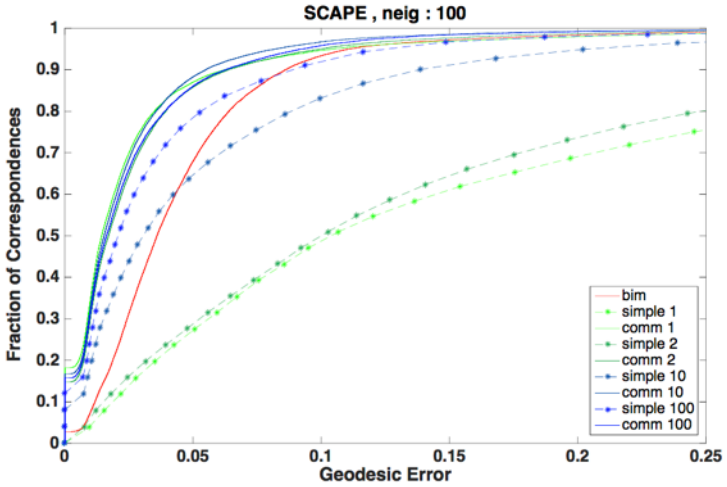
2. Compute descriptor functions (e.g., Gauss curvature) on $\mathcal{M}, \mathcal{N}$. Express them in $\Phi_{\mathcal{M}}, \Phi_{\mathcal{N}}$, as columns of : $\mathbf{A}, \mathbf{B}$

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

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

Questions for Today

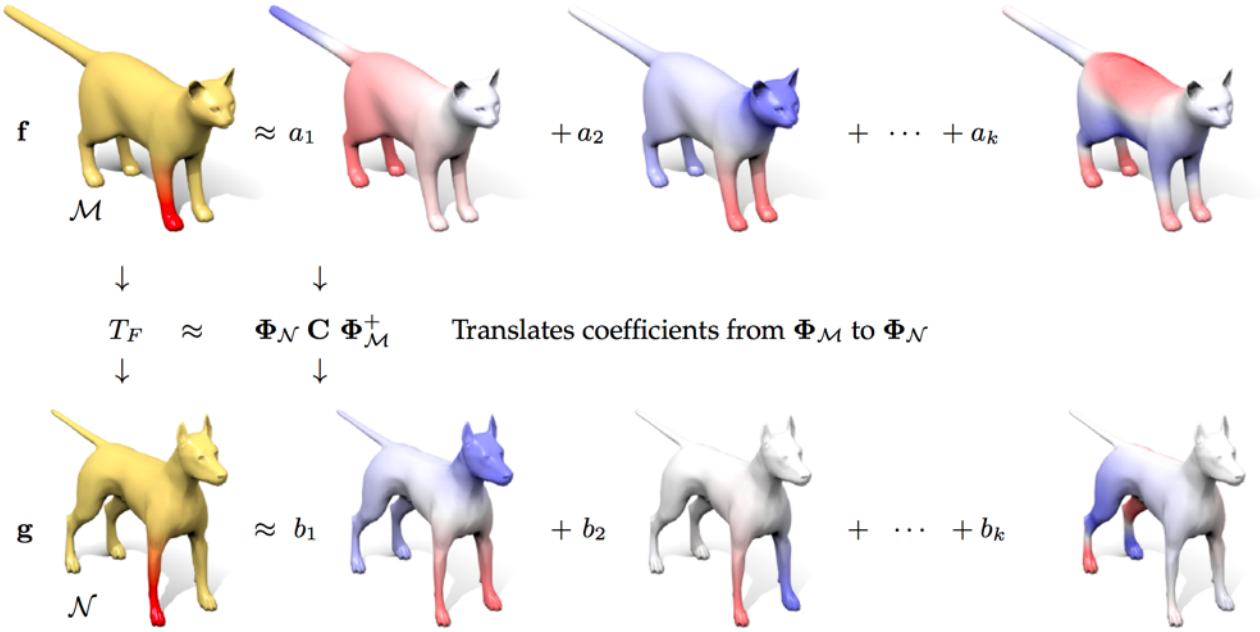
Can we enforce functional maps to be:

1. Closer to **point-to-point maps**?
2. Be able to transfer **higher-frequency information**?

While retaining the computational advantages.

Function transfer without p2p

In practice conversion to p2p is costly and error-prone.



Given C, can transfer **f** without pointwise map.

Improved Functional Mappings via Product Preservation, Nogneng, Melzi, Rodolà, Castellani, Bronstein, O. Eurographics 2018

Function transfer without p2p

Example:

Segmentation transfer via characteristic functions.

Improved Functional Mappings via Product Preservation, Nogneng, Melzi, Rodolà, Castellani, Bronstein, O. Eurographics 2018

Function transfer without p2p

Problem:

Only works if the function is in the span of the basis:

$$\mathbf{f} \longrightarrow \Phi_{\mathcal{N}} \mathbf{C} \Phi_{\mathcal{M}}^{+} \mathbf{f}$$

Question:

Can use $\mathbf{C}$ to transfer higher frequency functions?

Main idea:

If $\mathbf{C}$ is "good", it should preserve products. Thus, knowing how to transfer basis functions, allows us to *also transfer their pointwise products*!

Improved Functional Mappings via Product Preservation, Nogneng, Melzi, Rodolà, Castellani, Bronstein, O. Eurographics 2018

Function transfer without p2p

Main idea:

Express f as a combination of basis and their products:

$$f = \sum_i a_i \phi_i^{\mathcal{M}} + \sum_{i,j} a_i b_j \phi_i^{\mathcal{M}} \odot \phi_j^{\mathcal{M}}$$

If we know how to transfer $\phi_i^{\mathcal{M}}$ (using $\mathbf{C}$), and assuming preservation of products:

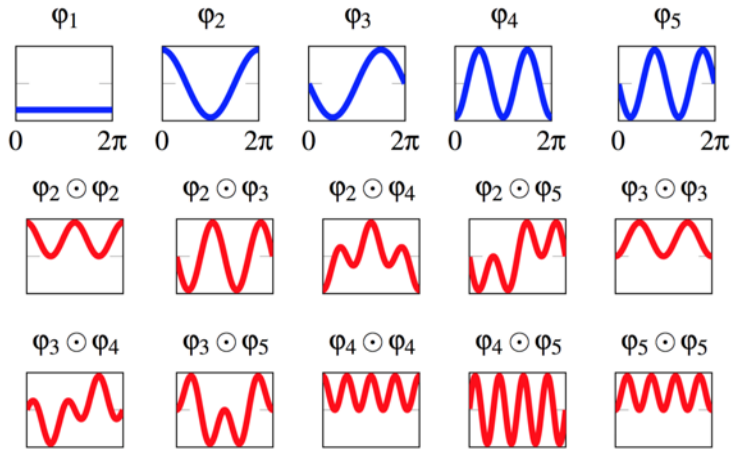
$$\mathbf{C}(f) = \sum_i a_i \mathbf{C}(\phi_i^{\mathcal{M}}) + \sum_{i,j} a_i b_j \mathbf{C}(\phi_i^{\mathcal{M}}) \odot \mathbf{C}(\phi_j^{\mathcal{M}})$$

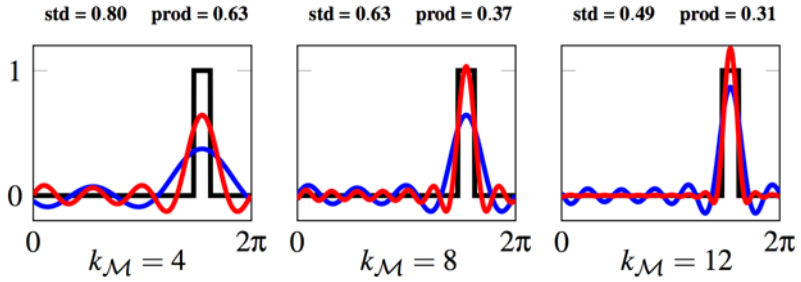
Same idea applies to higher order products.

Improved Functional Mappings via Product Preservation, Nogneng, Melzi, Rodolà, Castellani, Bronstein, O. Eurographics 2018

Function transfer without p2p

Example in 1D





Using pointwise products allows to capture higher frequency information.

Improved Functional Mappings via Product Preservation, Nogneng, Melzi, Rodolà, Castellani, Bronstein, O. Eurographics 2018

Function transfer without p2p

Using higher order products can introduce noise. Can we find the "best" approximation?

(unfortunate) **Theorem**

> The problem of approximating a function to a given precision, with the smallest number of products of basis functions is NP-hard.

(somewhat) related to minimum L_0 norm approximation.

Improved Functional Mappings via Product Preservation, Nogneng, Melzi, Rodolà, Castellani, Bronstein, O. Eurographics 2018

Function transfer without p2p

In practice, construct an ***extended basis*** using basis functions and (all) their first-order pointwise products.

$$\mathbf{B}_{\mathcal{M}} = \left(\Phi_{\mathcal{M}}, \widetilde{\Phi}_{\mathcal{M}}\right), \text{ s.t. } \widetilde{\phi}_i = \phi_l^{\mathcal{M}} \odot \phi_m^{\mathcal{M}}$$

Approximate and transfer functions expressed in the ***extended basis.***

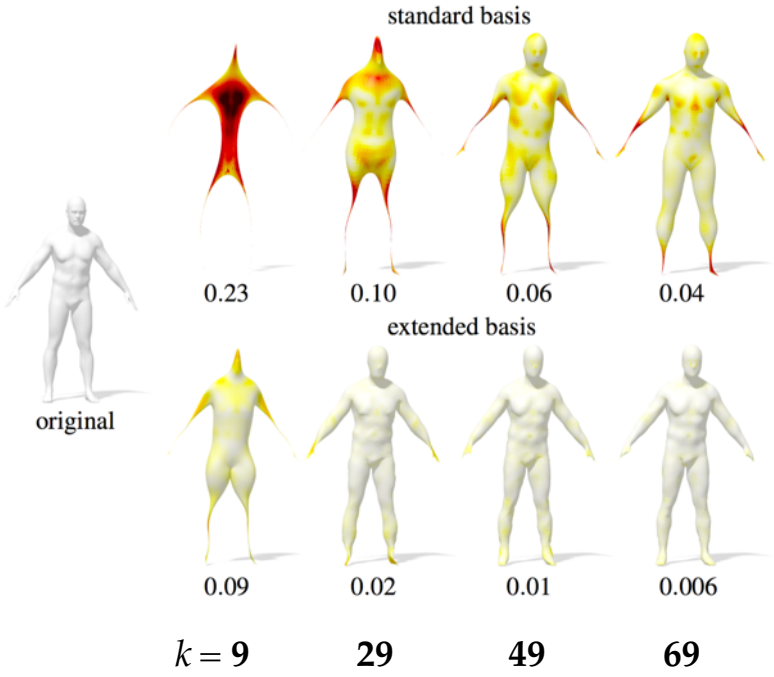
One (slight) challenge: extended basis is not orthonormal. Can use SVD/basis pursuit.

Improved Functional Mappings via Product Preservation, Nogneng, Melzi, Rodolà, Castellani, Bronstein, O. Eurographics 2018

Function transfer without p2p

Example 1:

Approximating the coordinate (XYZ) functions:



Improved Functional Mappings via Product Preservation, Nogneng, Melzi, Rodolà, Castellani, Bronstein, O. Eurographics 2018

Function transfer without p2p

Theorem

For a fixed C, there is *a linear transformation* between extended bases, given in closed-form expression:

$$\widetilde{\mathbf{C}} = \begin{bmatrix} \mathbf{C} & \mathcal{R}(\mathbf{C}) \\ 0 & \mathbf{C}(1:k,1:k) \otimes \mathbf{C}(1:k,1:k) \end{bmatrix}$$

with

$$\mathcal{R}(\mathbf{C}) = \begin{bmatrix} \phi_0 \mathbf{C}(0,1:k) \otimes \mathbf{C}(0:k,1:k) \\ +\phi_0 \begin{bmatrix} 0 \dots 0 \\ \mathbf{C}(1:k,1:k) \end{bmatrix} \otimes \mathbf{C}(0,1:k) \end{bmatrix}$$

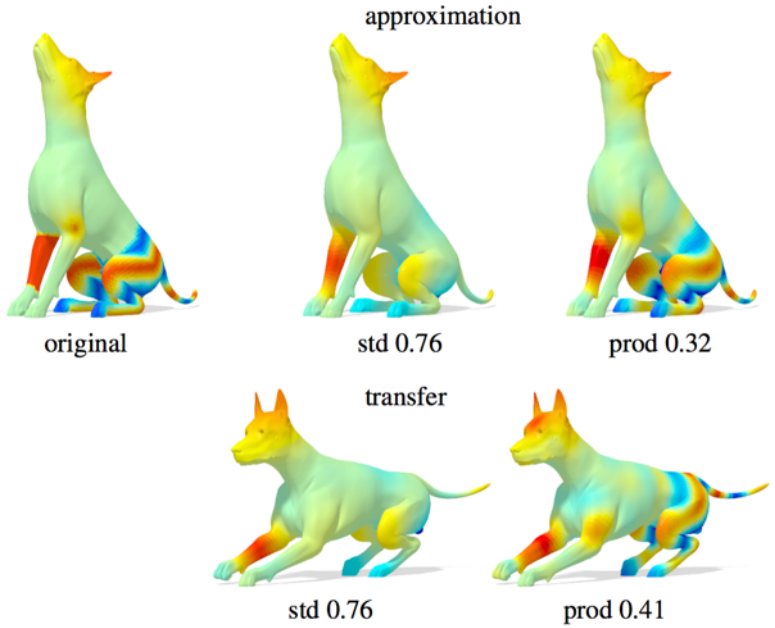
Note that $\widetilde{\mathbf{C}}$ is not linear in $\mathbf{C}$ but is a *linear map across extended bases.*

Improved Functional Mappings via Product Preservation, Nogneng, Melzi, Rodolà, Castellani, Bronstein, O. Eurographics 2018

Function transfer without p2p

Example 2:

Approximating *and transferring* a smooth function:



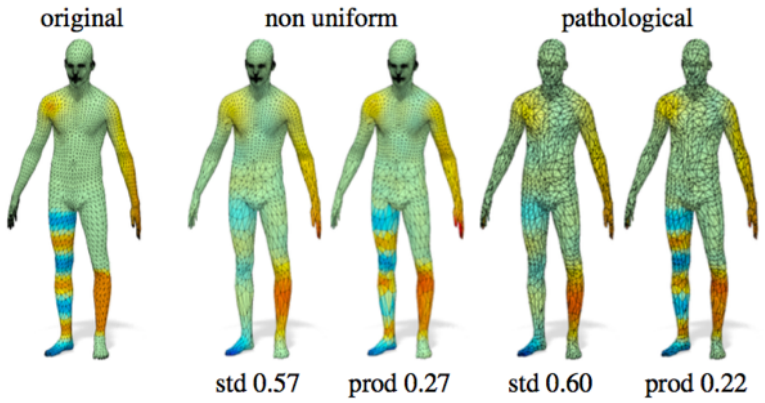
Improved Functional Mappings via Product Preservation, Nogneng, Melzi, Rodolà, Castellani, Bronstein, O. Eurographics 2018

Function transfer without p2p

Evaluations:

Transfer of various functions using computed (approximate) functional maps:

computed	approx			transfer		
function	**std**	**our A**	**our B**	**std**	**our A**	**our B**
hk k	**0.00**	0.03	0.01	0.13	0.13	0.13
hk K	0.82	0.54	**0.49**	0.82	**0.57**	0.58
HKS	0.55	0.14	**0.00**	0.55	**0.22**	**0.22**
WKS	0.14	0.03	**0.00**	0.14	0.07	**0.06**
Random	0.48	0.04	**0.01**	0.49	**0.15**	**0.15**
XYZ	0.12	0.09	**0.06**	0.13	**0.11**	**0.11**
Indicator	0.28	0.19	**0.17**	0.28	0.20	**0.19**
SHOT	0.81	0.76	**0.74**	0.81	**0.77**	0.78
AWFT	0.24	0.19	**0.17**	0.25	**0.20**	**0.20**

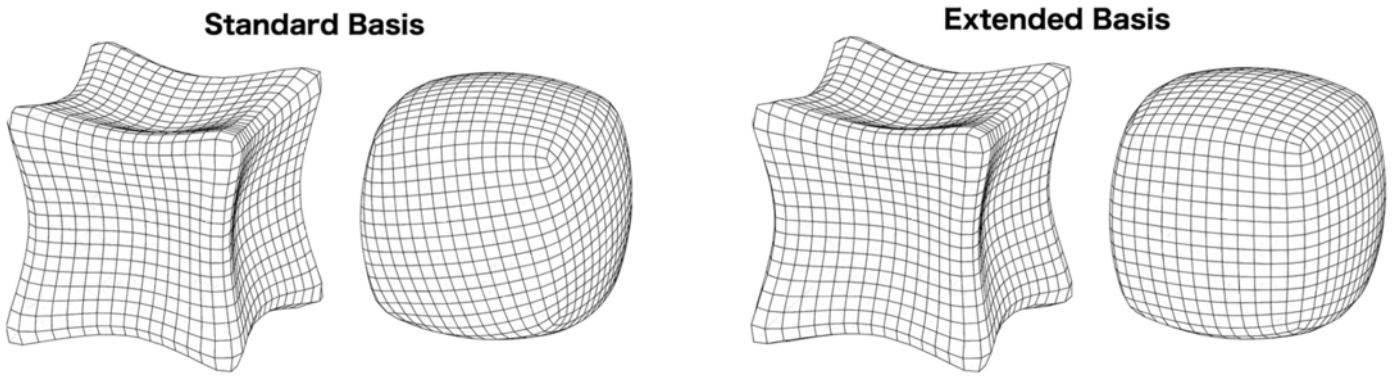


Improved Functional Mappings via Product Preservation, Nogneng, Melzi, Rodolà, Castellani, Bronstein, O. Eurographics 2018

Function transfer without p2p

Application:

Joint quadrangulation of triangle meshes



Based on: *Consistent Functional Cross Field Design for Mesh Quadrangulation,* Azencot, Corman, Ben-Chen, O. SIGGRAPH 2017

Improved Functional Mappings via Product Preservation, Nogneng, Melzi, Rodolà, Castellani, Bronstein, O. Eurographics 2018

Conclusions & Follow-up

Functional maps provide a convenient representation for computing correspondences.

In addition to the vector space structure, can exploit the *functional algebra* for:

- Descriptor preservation
- Function transfer

SIGGRAPH Course Website:

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

or http://bit.do/fmaps2017

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

Thank you!

Questions?

Acknowledgements:

D. Nogneng, Simone Melzi, Emanuele Rodolà, Umberto Castellani, Michael Bronstein… Work supported by chair a Google Focused Research Award, ERC Starting Grant No. 758800 (EXPROTEA), ERC Consolidator Grant No. 724228 (LEMAN), and Rudolf Diesel fellow- ship from the TUM Institute for Advanced Study.

Reconstructing from LB basis

Map reconstruction error using a fixed size matrix.

27.9k vertices