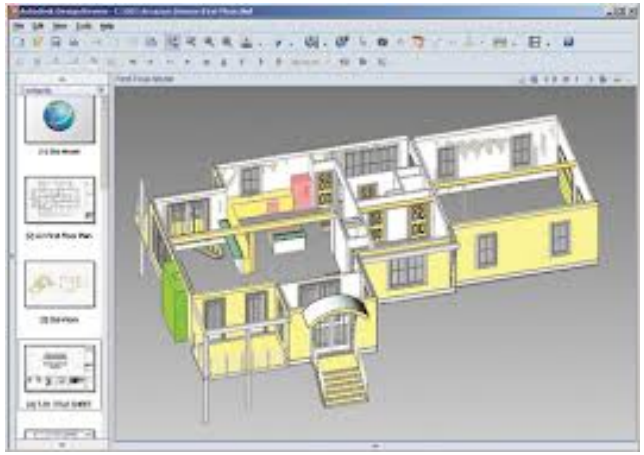
The Operator View of the World: Geometry Processing

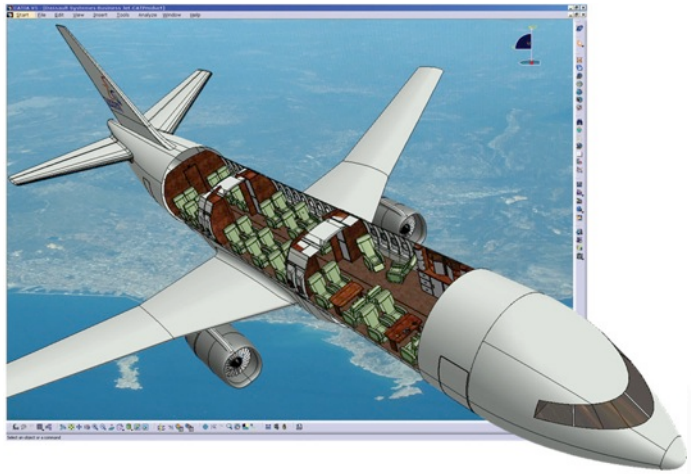
Maks Ovsjanikov

What is Geometry Processing

- Broad Goals:

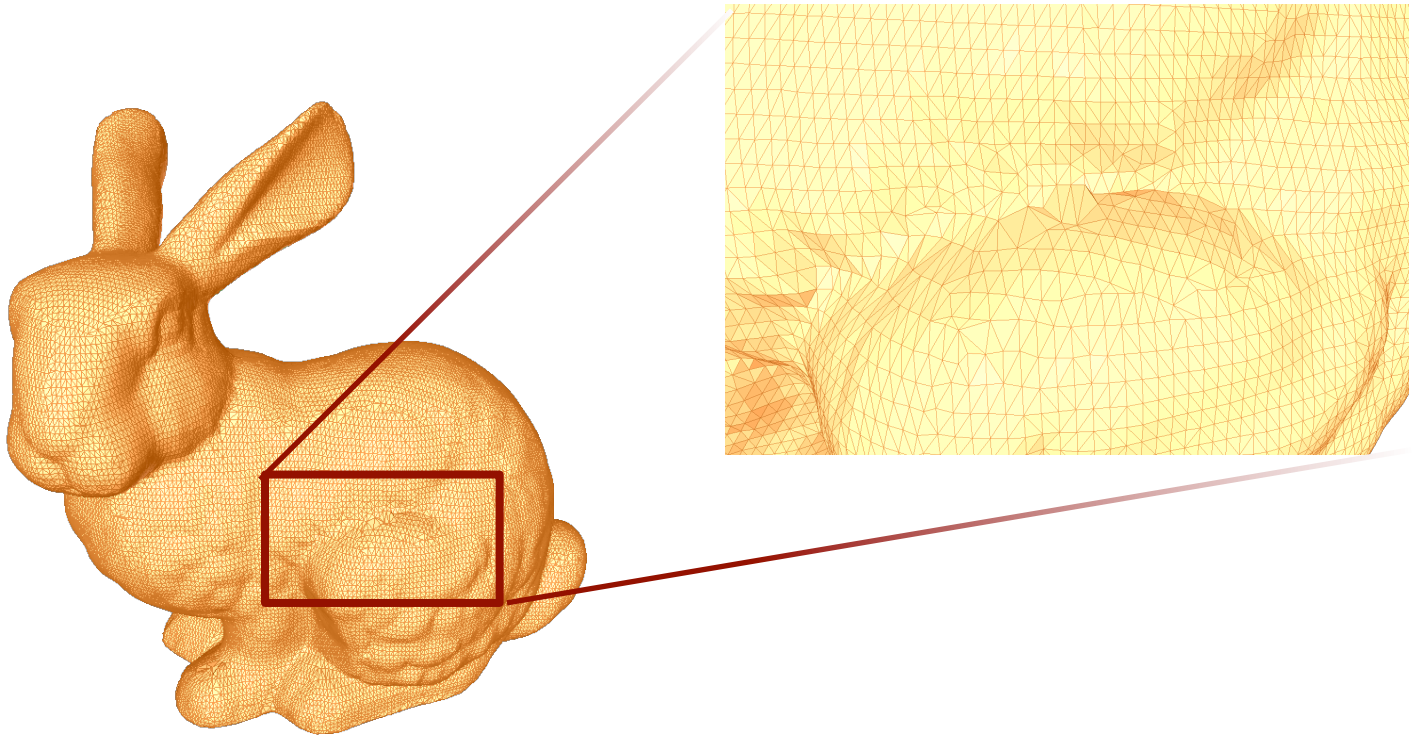
  To create mathematical models and practical tools for digital representation, manipulation and analysis of 3D shapes.





What is a Shape?

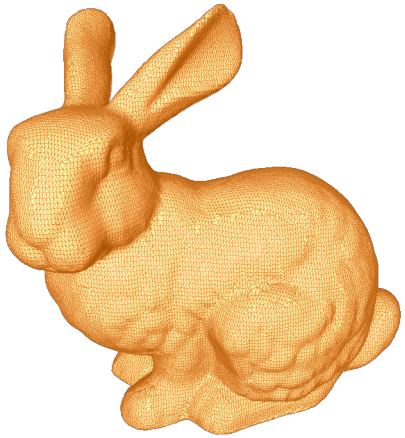
- A graph embedded in 3D: a triangle mesh.

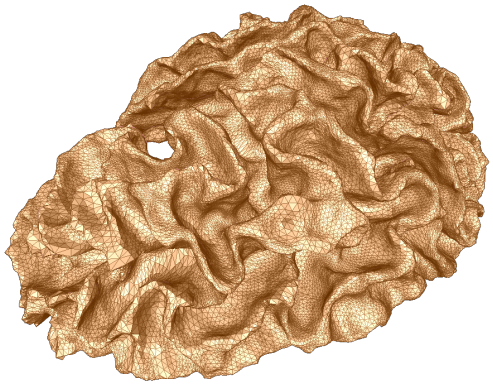


1k – 200k triangles

What is a Shape?

- A graph embedded in 3D: a triangle mesh.
  - Connected.
  - Manifold (each edge on at most 2 triangles).
  - Without boundary.





Brain data: Neurospin.

Overall Research Goals

- Understand **relations** between 3d shapes.
- Develop efficient methods to **compare** geometry
- Enable applications including
  - shape retrieval
  - shape comparison and exploration
  - shape matching
- Explore **non-rigid** relations.

Research Program (Today)

- Shape Matching.
- Vector field design and analysis.
- Shape exploration.

Research Program (Today)

- Shape Matching.
- Vector field design and analysis.
- Shape exploration.

Main observation:

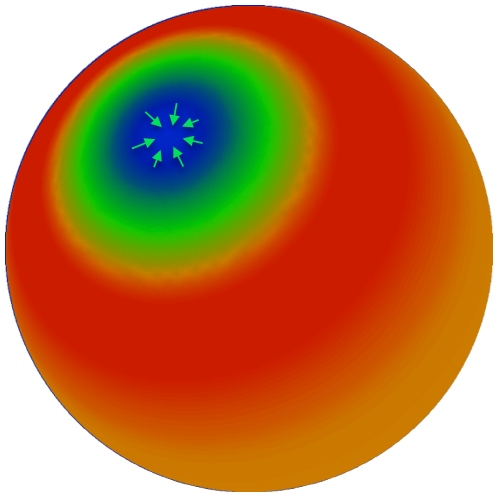
- Many tasks can be formulated through manipulation of linear operators defined on (L2) function spaces.
- Using a subspace of the functional space can help.

Laplace-Beltrami Operator

Given a compact Riemannian manifold M without boundary, the Laplace-Beltrami operator:

$$\Delta : C^\infty(M) \to C^\infty(M), \Delta f = \text{div } \nabla f$$

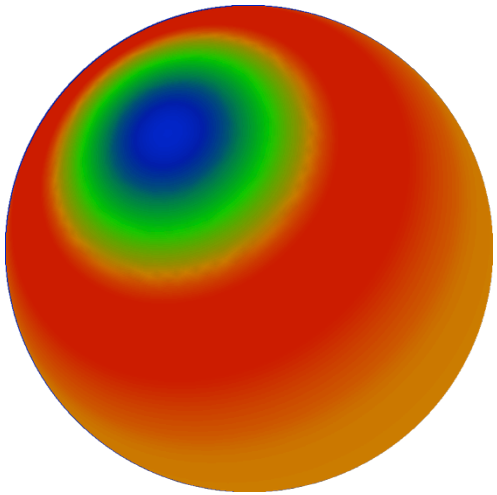
$\text{div}\nabla f$



Heat Equation on a Surface

Given a compact surface without the evolution of heat is given by $\dfrac{\partial f}{\partial t} = \Delta f = \operatorname{div} \nabla f.$

$$\frac{\partial f}{\partial t} = \Delta f$$

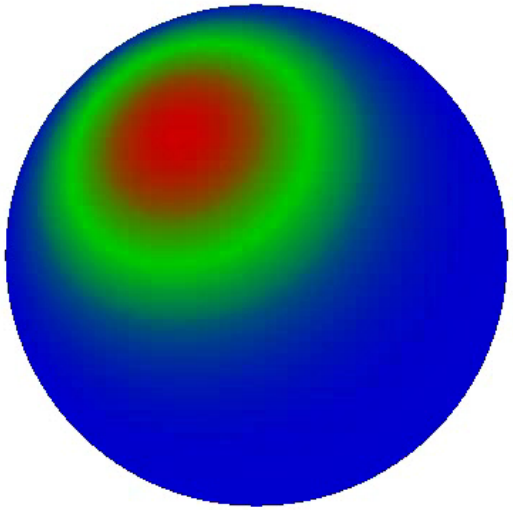


ÉCOLE POLYTECHNIQUE ParisTech

Heat Equation on a Surface

Given a compact surface without the evolution of heat is given by $\dfrac{\partial f}{\partial t} = \Delta f = \operatorname{div} \nabla f.$

$$\frac{\partial f}{\partial t} = \Delta f$$



Laplace-Beltrami Operator

Given a compact Riemannian manifold M without boundary, the Laplace-Beltrami operator Δ :

1. Is invariant under isometric deformations.

2. Characterizes the manifold completely.

3. Has a countable eigendecomposition:

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

that forms an orthonormal basis for $L^2(M)$.

Some background

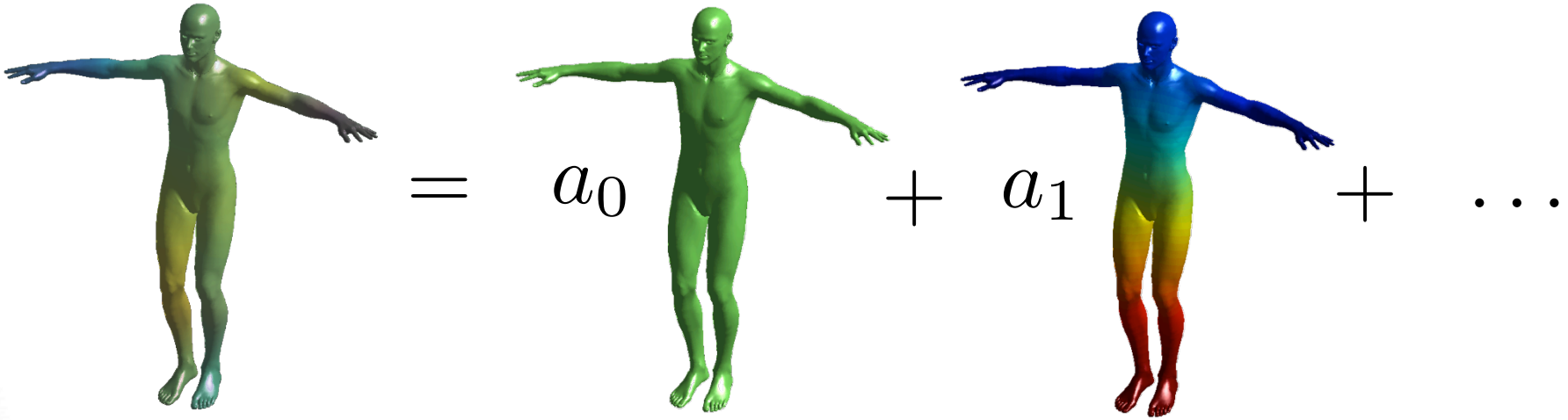
The Laplace-Beltrami operator Δ has an eigendecomposition: $\Delta\phi_i = \lambda_i\phi_i$

that forms an orthonormal basis for $L^2(M)$.

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

Some background

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



$$f = \sum_{i=0}^{\infty} a_i \phi_i \qquad a_i = \int_M f(x)\phi_i(x)d\mu$$

Some background

- The basis is *multiscale*.

$$\|\nabla\phi_i\|_2^2 = \int_{x\in M} \|\nabla\phi_i(x)\|_2^2 d\mu = \lambda_i^2$$

- Conversely, for any function

$$f = \sum_{i=0}^{N} a_i\phi_i \quad \Rightarrow \quad \frac{\|\nabla f\|_2^2}{\|f\|_2^2} \leq \lambda_N$$

$$\|f\|_2^2 = \int_{x\in M} f^2(x)d\mu(x) = \sum_i a_i^2$$

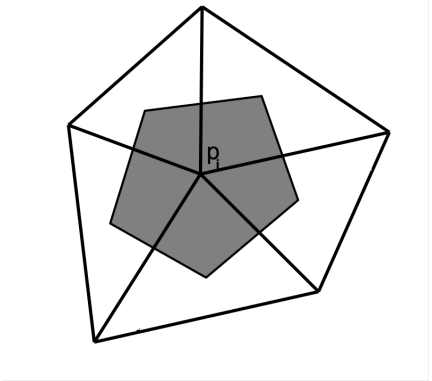
In the Discrete World

- Functions are defined at vertices of the mesh.

- Integration is defined with respect to a discrete volume measure:

$$\|f\|_2^2 = f^T A f$$

A - diagonal matrix of area weights.

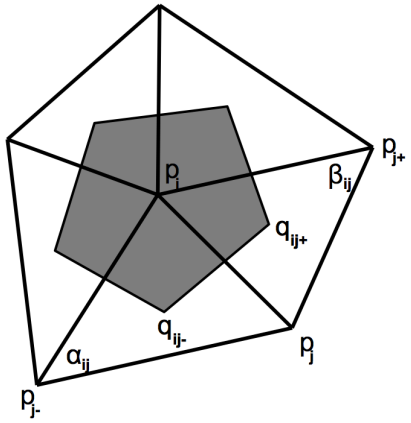


In the Discrete World

- Laplacian is discretized as a matrix $L = A^{-1}W$

$$W_{ij} = \frac{\cot \alpha_{ij} + \cot \beta_{ij}}{2}$$

so that $$\Delta f(p_i) \approx \frac{1}{s_i} \sum_{j \in \mathcal{N}(i)} \frac{\cot \alpha_{ij} + \cot \beta_{ij}}{2} [f(p_j) - f(p_i)]$$



- Can be derived from 1st order FEM.

In the Discrete World

- Computing the eigenfunctions of the Laplacian reduces to solving the eigenvalue problem:

$$L\phi = \lambda\phi \quad \Leftrightarrow \quad W\phi = \lambda A\phi$$

- Both A and W are sparse positive semidefinite.

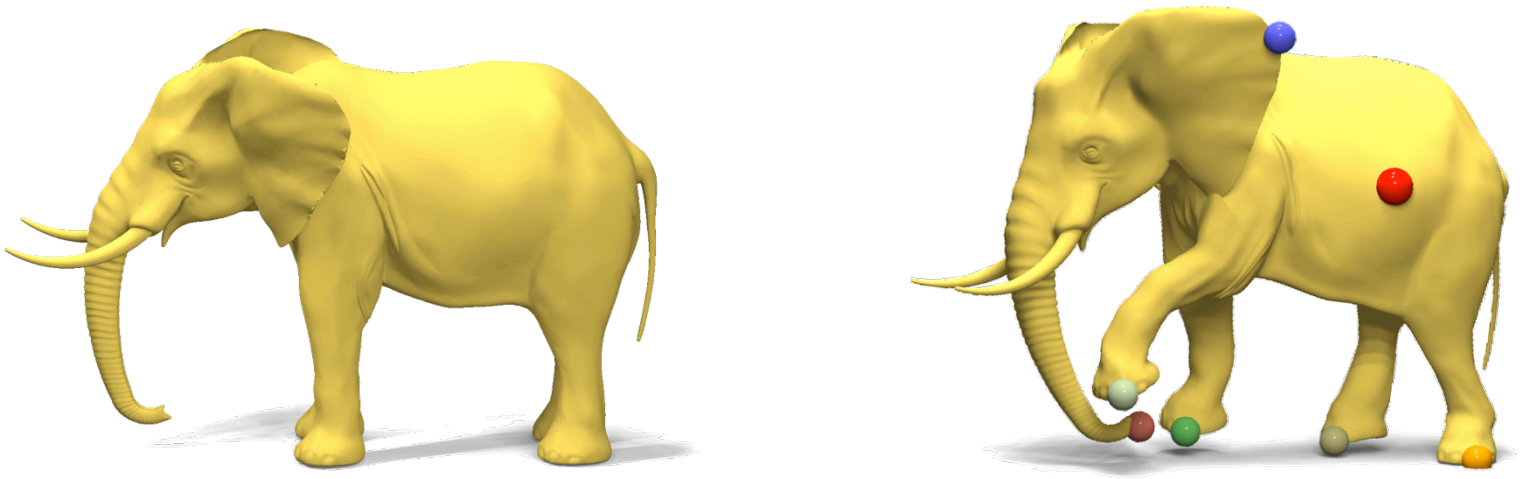
Number of triangles	**Computation time** (in s)
5000	0.65
25000	2.32
50868	3.6
105032	10

ÉCOLE POLYTECHNIQUE ParisTech

Shape Matching

Problem:

Given a pair of shapes, find corresponding points.



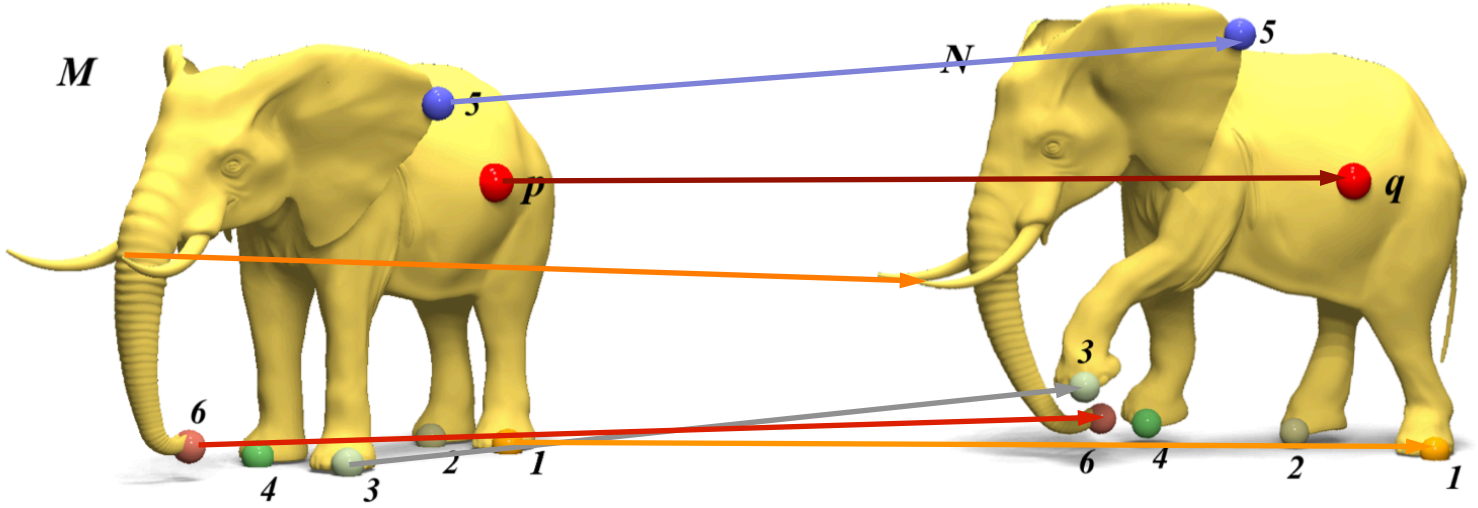
Functional Maps: A Flexible Representation of Maps Between Shapes
O., Ben-Chen, Solomon, Butscher, Guibas SIGGRAPH 2012

ÉCOLE POLYTECHNIQUE ParisTech

Shape Matching

Problem:

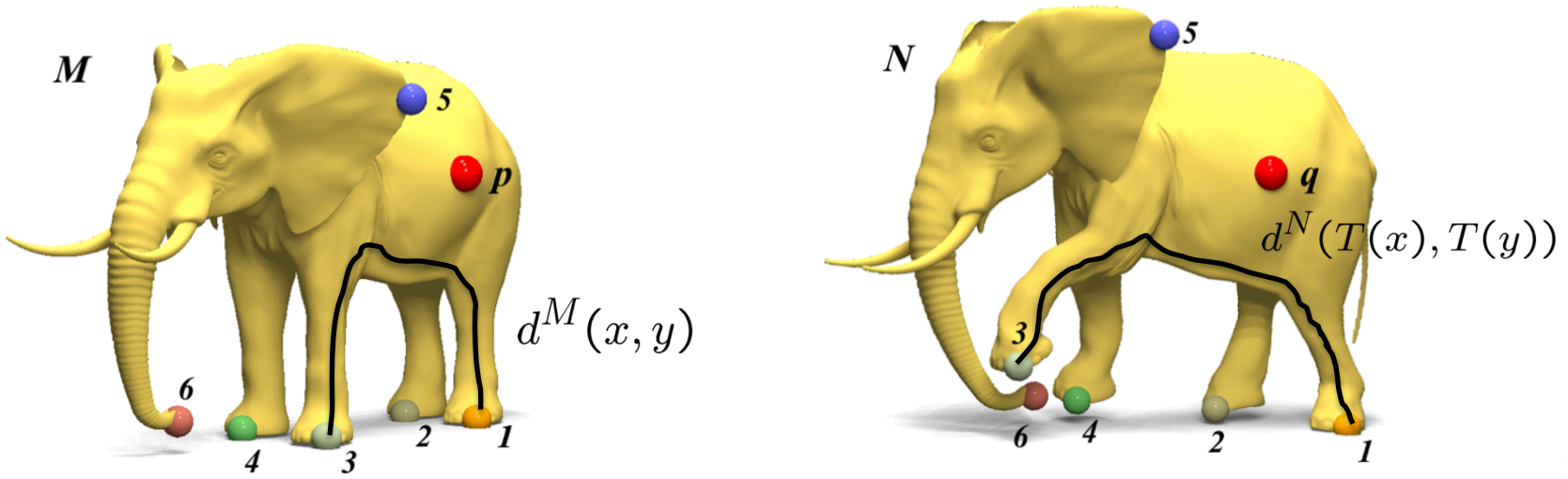
Given a pair of shapes, find corresponding points.



Non-Rigid Shape Matching

Problem:

Given a pair of shapes, find corresponding points.

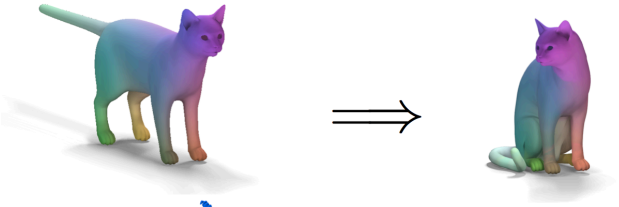


Find a correspondence that preserves intrinsic (geodesic) distances on the shapes.

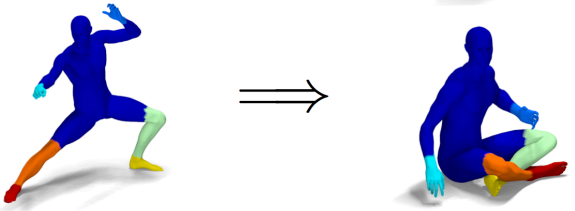
Why Shape Matching

Given a correspondence, we can transfer:

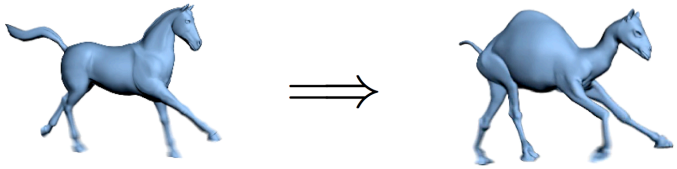
texture and parametrization



segmentation and labels



deformation



Other applications: shape interpolation, reconstruction ...

Method Taxonomy

Local vs. Global

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

Rigid vs. Deformable

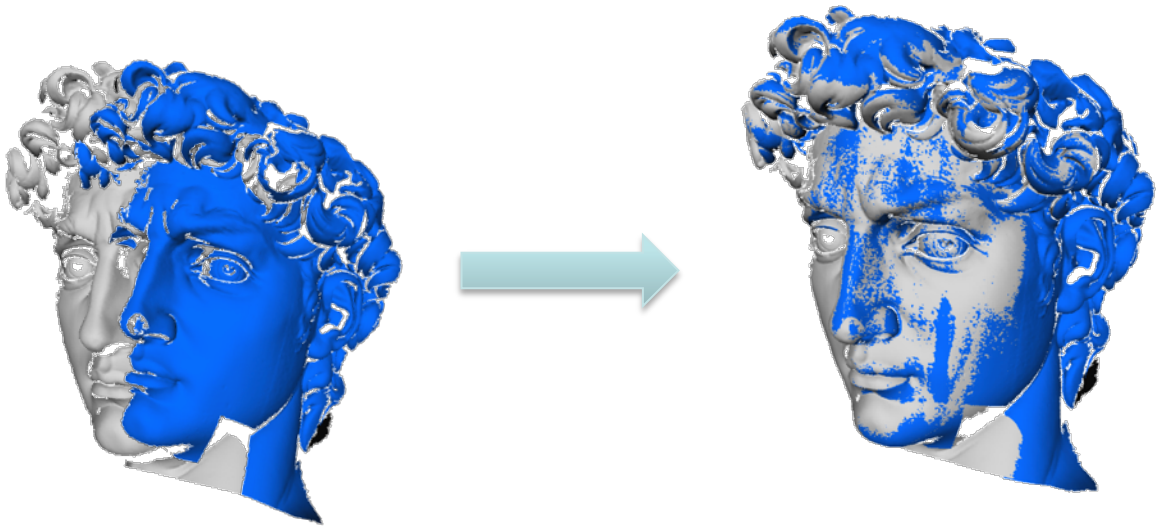
rotation, translation | general deformation

Pair vs. Collection

two shapes | multiple shapes

Local Alignment

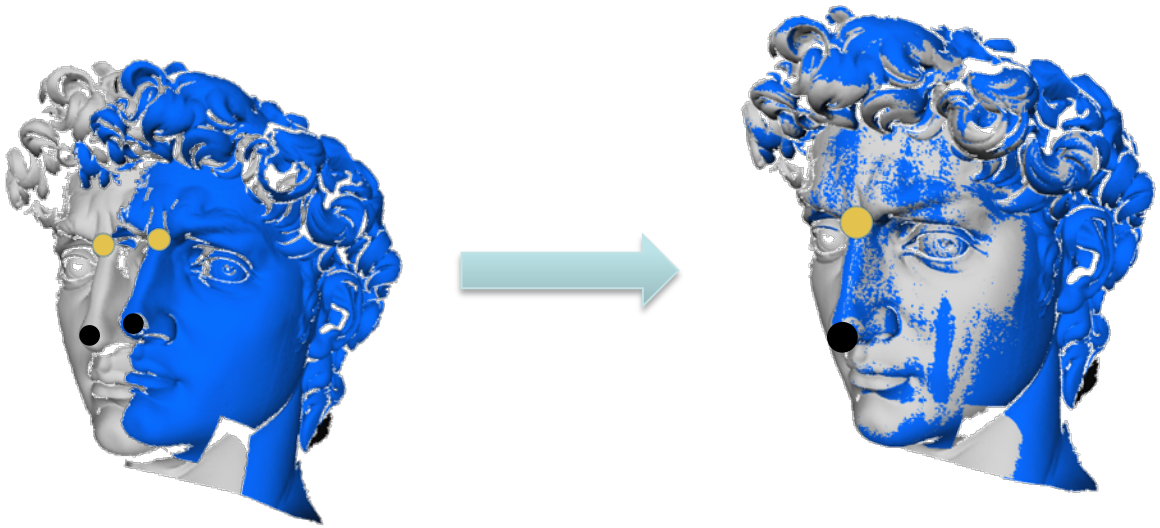
- Simplest instance of the registration problem



Given two shapes that are **approximately aligned** (e.g. by a human) we want to find the optimal tranformation.

Local Alignment

- What does it mean for an alignment to be **good**?

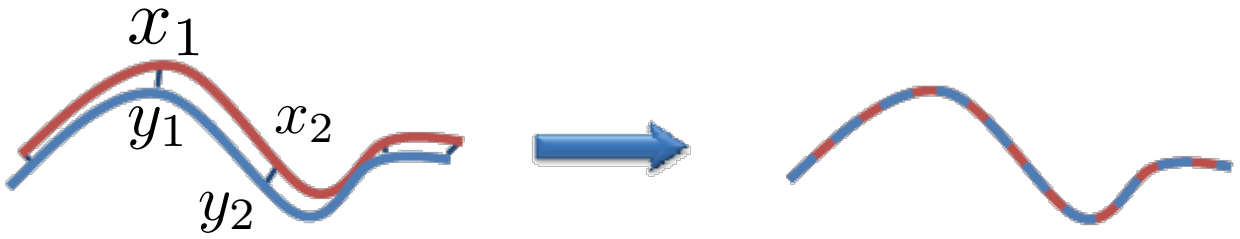


Intuition: want corresponding points to be close after transformation.

Problems
1. We don't know what points correspond.
2. We don't know the optimal alignment.

Iterative Closest Point (ICP)

- Approach: iterate between finding correspondences and finding the transformation:



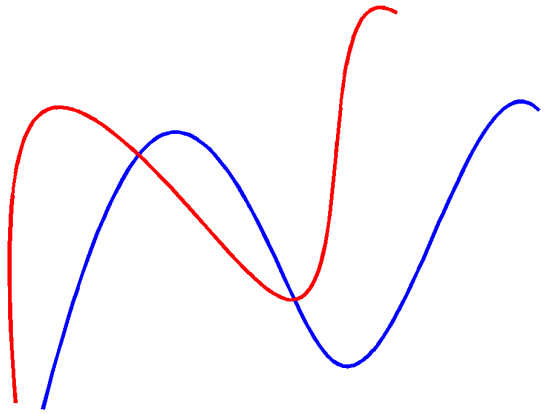
Given a pair of shapes, X and Y, iterate:

1. For each $x_i \in X$ find **nearest** neighbor $y_i \in Y$.
2. Find deformation $\mathbf{R}, t$ minimizing:

$$\sum_{i=1}^{N} \|\mathbf{R}x_i + t - y_i\|_2^2$$

Iterative Closest Point (ICP)

- Approach: iterate between finding correspondences and finding the transformation:

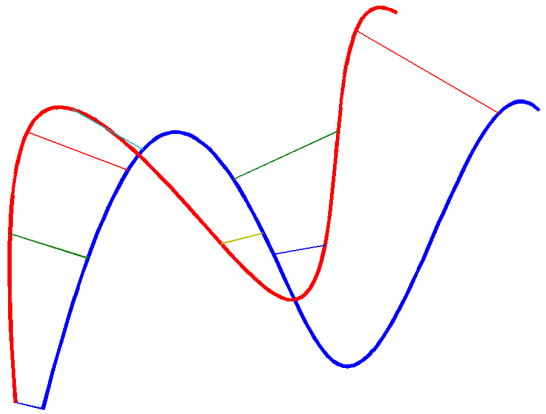


Given a pair of shapes, X and Y, iterate:

1. For each $x_i \in X$ find **nearest** neighbor $y_i \in Y$.
2. Find deformation $\mathbf{R}, t$ minimizing: $\sum_{i=1}^{N} \|\mathbf{R}x_i + t - y_i\|_2^2$

Iterative Closest Point

- Approach: iterate between finding correspondences and finding the transformation:

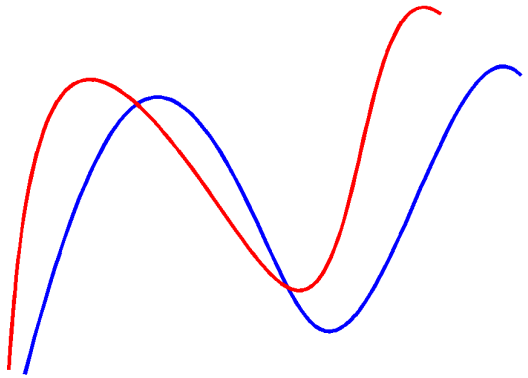


Given a pair of shapes, X and Y, iterate:

1. For each $x_i \in X$ find **nearest** neighbor $y_i \in Y$.
2. Find deformation $\mathbf{R}, t$ minimizing: $\sum_{i=1}^{N} \|\mathbf{R}x_i + t - y_i\|_2^2$

Iterative Closest Point

- Approach: iterate between finding correspondences and finding the transformation:

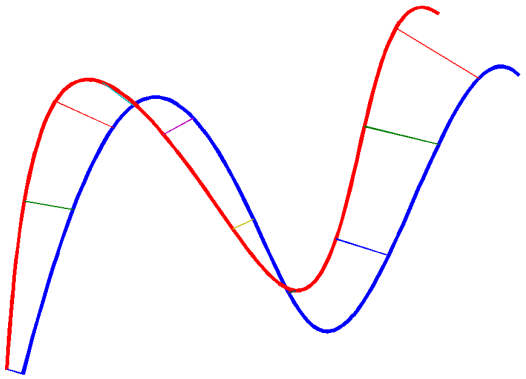


Given a pair of shapes, X and Y, iterate:

1. For each $x_i \in X$ find **nearest** neighbor $y_i \in Y$.
2. Find deformation $\mathbf{R}, t$ minimizing: $\sum_{i=1}^{N} \|\mathbf{R}x_i + t - y_i\|_2^2$

Iterative Closest Point

- Approach: iterate between finding correspondences and finding the transformation:

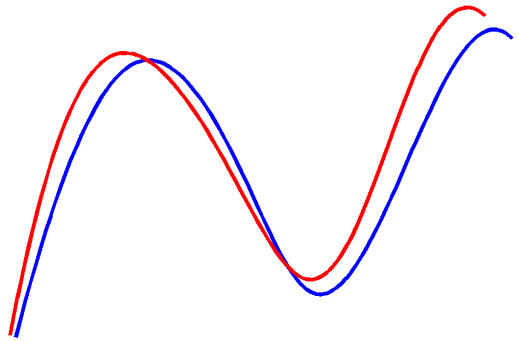


Given a pair of shapes, X and Y, iterate:

1. For each $x_i \in X$ find **nearest** neighbor $y_i \in Y$.
2. Find deformation $\mathbf{R}, t$ minimizing: $\sum_{i=1}^{N} \|\mathbf{R}x_i + t - y_i\|_2^2$

Iterative Closest Point

- Approach: iterate between finding correspondences and finding the transformation:

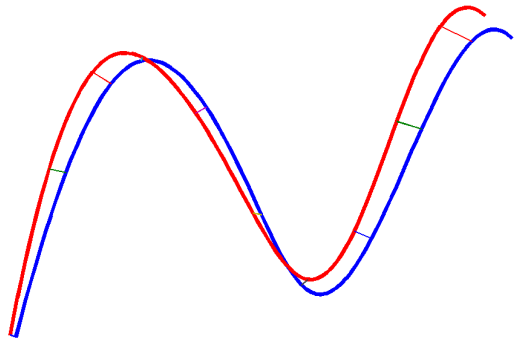


Given a pair of shapes, X and Y, iterate:

1. For each $x_i \in X$ find **nearest** neighbor $y_i \in Y$.
2. Find deformation $\mathbf{R}, t$ minimizing: $\sum_{i=1}^{N} \|\mathbf{R}x_i + t - y_i\|_2^2$

Iterative Closest Point

- Approach: iterate between finding correspondences and finding the transformation:

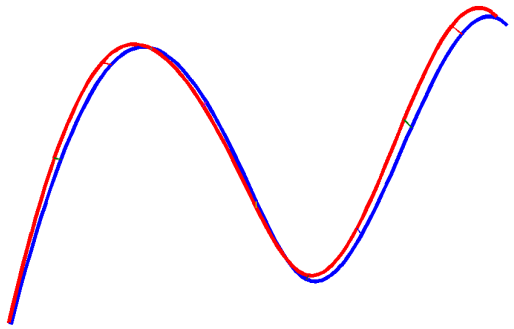


Given a pair of shapes, X and Y, iterate:

1. For each $x_i \in X$ find **nearest** neighbor $y_i \in Y$.
2. Find deformation $\mathbf{R}, t$ minimizing: $\sum_{i=1}^{N} \|\mathbf{R}x_i + t - y_i\|_2^2$

Iterative Closest Point

- Approach: iterate between finding correspondences and finding the transformation:

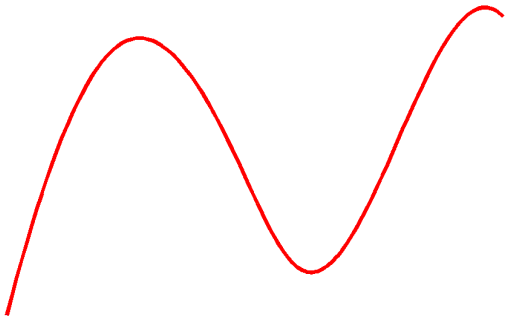


Given a pair of shapes, X and Y, iterate:

1. For each $x_i \in X$ find **nearest** neighbor $y_i \in Y$.
2. Find deformation $\mathbf{R}, t$ minimizing: $\sum_{i=1}^{N} \|\mathbf{R}x_i + t - y_i\|_2^2$

Iterative Closest Point

- Approach: iterate between finding correspondences and finding the transformation:



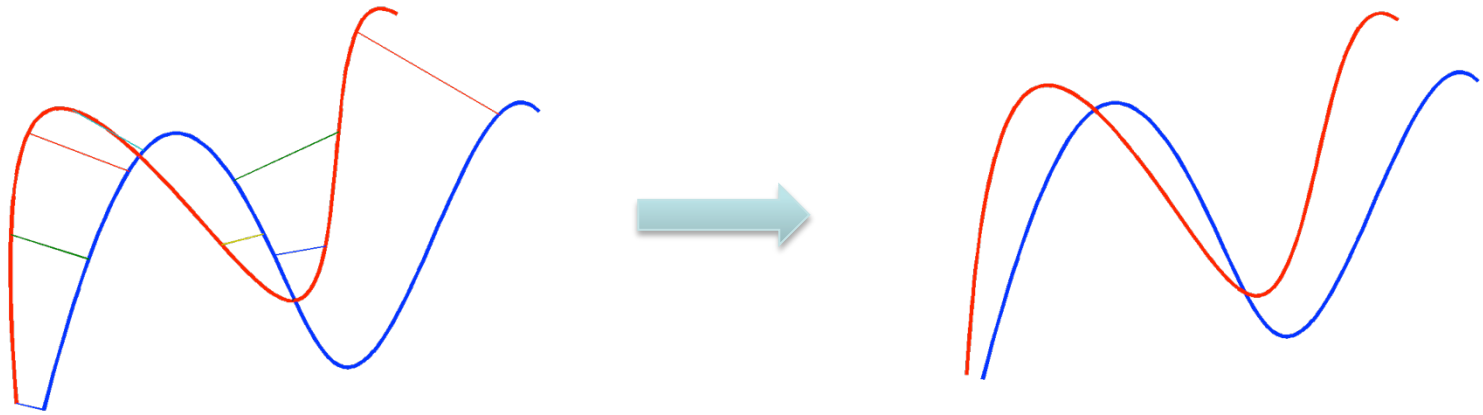
Given a pair of shapes, X and Y, iterate:

1. For each $x_i \in X$ find **nearest** neighbor $y_i \in Y$.
2. Find deformation $\mathbf{R}, t$ minimizing: $\sum_{i=1}^{N} \|\mathbf{R}x_i + t - y_i\|_2^2$

Iterative Closest Point

- Requires two main computations:

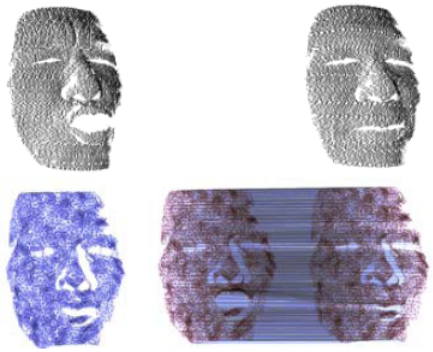
1. Computing nearest neighbors – Voronoi Diagram.
2. Computing the optimal transformation – Closed form.



Deformable Shape Matching

No shape is completely deformable. Every deformable shape matching method uses *some* deformation model.

- Local Deformable Matching:



Wand et al. *SGP '07*

Li et al. *SGP '08*

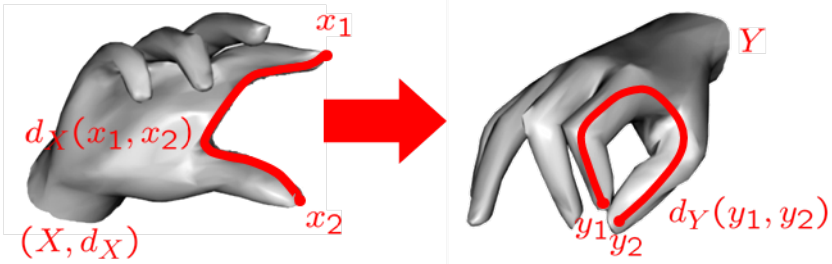
…

Each coordinate becomes an unknown. Use *smoothness*, elasticity constraints. Resulting problem is always *non-linear*.

Global Isometric Matching

- Optimization-based techniques:

GMDS



$$\min_{\{y_1,\dots y_n\}\subset Y} \|d_X(x_i, x_j) - d_Y(y_i, y_j)\|$$

Bronstein et al. PNAS '06

1) Sample a few landmark correspondences
2) Guess an initial correspondence
3) Locally modify the correspondence to minimize the energy.
4) Extend them to a dense map.

ÉCOLE POLYTECHNIQUE ParisTech

Global Isometric Matching

Recipe for non-rigid shape matching.

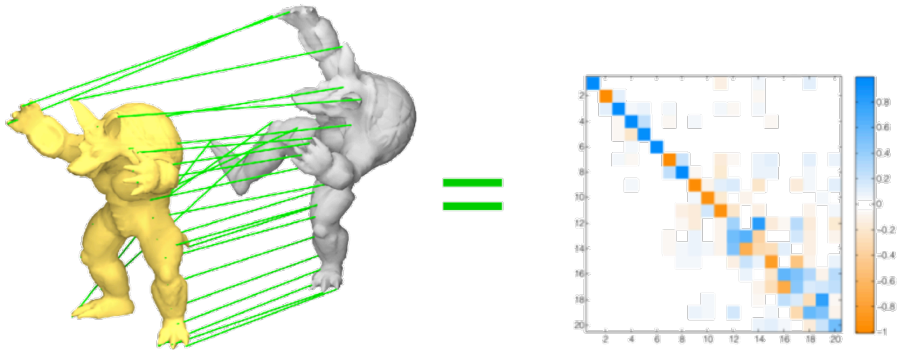
- Directly optimizing the correspondences leads to non-linear non-convex problems.
- Difficult to incorporate uncertainty into prior data.
- Difficult to return meaningful results in the presence of ambiguities.

Rather than limitations of the methods these are limitations of the representation.

Our Goals

Our main motivation is to define a representation of shape maps that is more amenable to direct optimization.

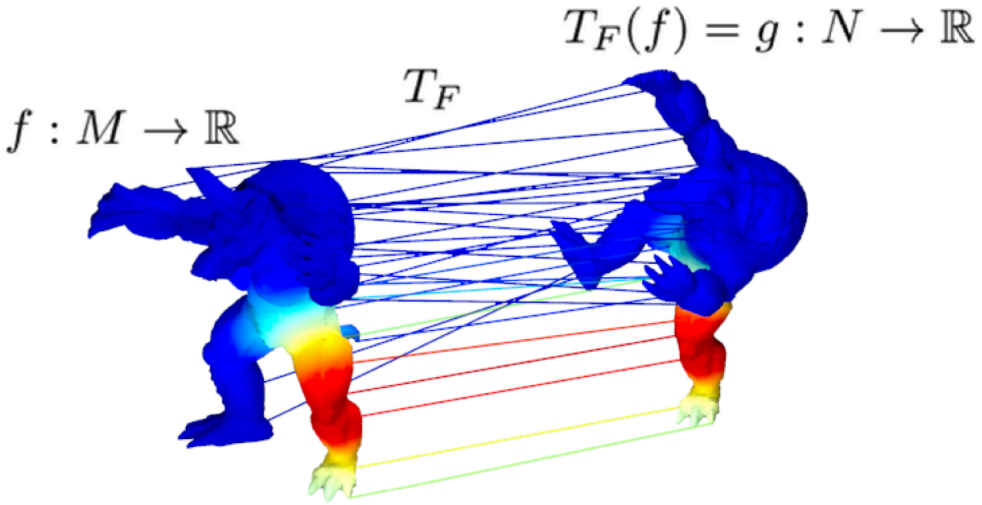
1. A compact representation for “natural” maps.
2. Inherently global and multi-scale.
3. Handles uncertainty and ambiguity gracefully.
4. Allows efficient manipulations (averaging, composition).
5. Leads to simple (linear) optimization problems.



ÉCOLE POLYTECHNIQUE ParisTech

Approach

Given a pair of shapes and a pointwise bijection $T : M \to N$



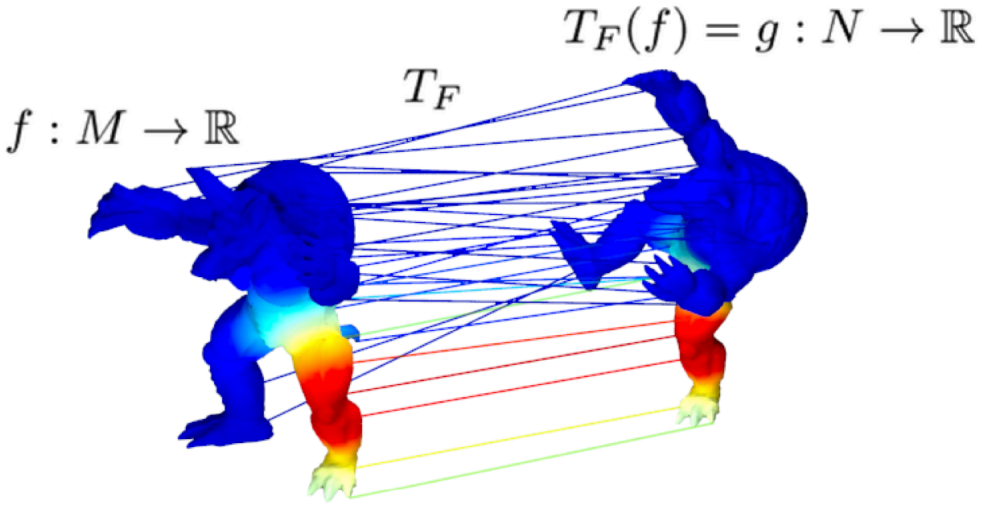
The induced functional correspondence:

$$T_F(f) = g, \quad g = f \circ T^{-1}$$

ÉCOLE POLYTECHNIQUE ParisTech

Approach

Given a pair of shapes and a pointwise bijection $T : M \to N$



$$T_F(f) = g, \quad g = f \circ T^{-1}$$

Note that $T_F(f)$ is:

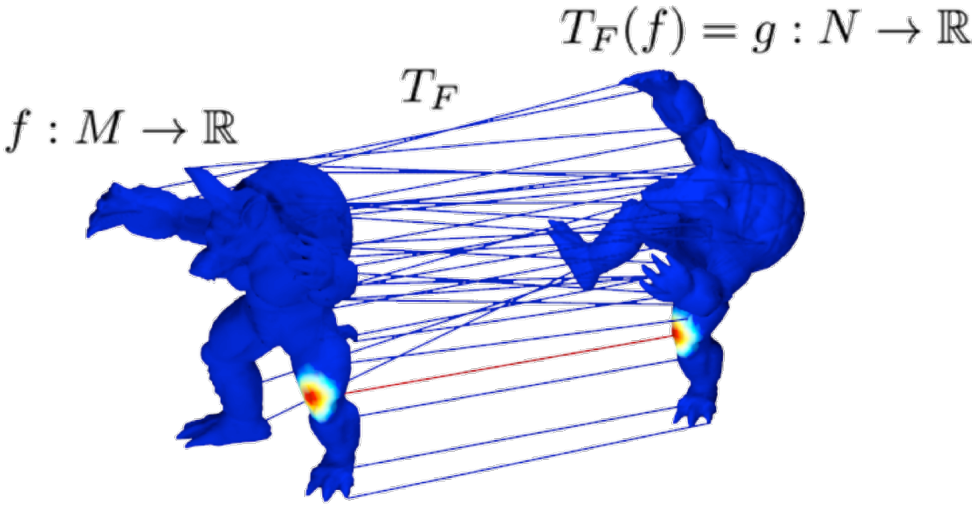
1) Linear $T_F(\alpha_1 f_1 + \alpha_1 f_1) = \alpha_1 T_F(f_1) + \alpha_2 T_F(f_2)$

2) Complete (recover T from indicator functions)

ÉCOLE POLYTECHNIQUE ParisTech

Approach

Given a pair of shapes and a pointwise bijection $T : M \to N$



$$T_F(f) = g, \quad g = f \circ T^{-1}$$

Note that $T_F(f)$ is:

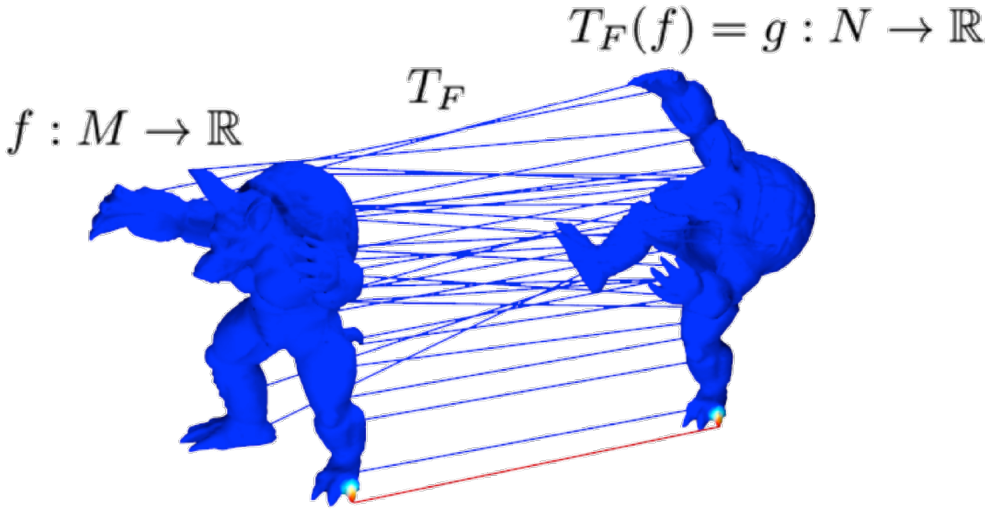
1) Linear $T_F(\alpha_1 f_1 + \alpha_1 f_1) = \alpha_1 T_F(f_1) + \alpha_2 T_F(f_2)$

2) Complete (recover T from indicator functions)

ÉCOLE POLYTECHNIQUE ParisTech

Approach

Given a pair of shapes and a pointwise bijection $T : M \to N$



$$T_F(f) = g, \quad g = f \circ T^{-1}$$

Note that $T_F(f)$ is:

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

2) Complete (recover T from indicator functions)

ÉCOLE POLYTECHNIQUE ParisTech

Functional Maps

Definition

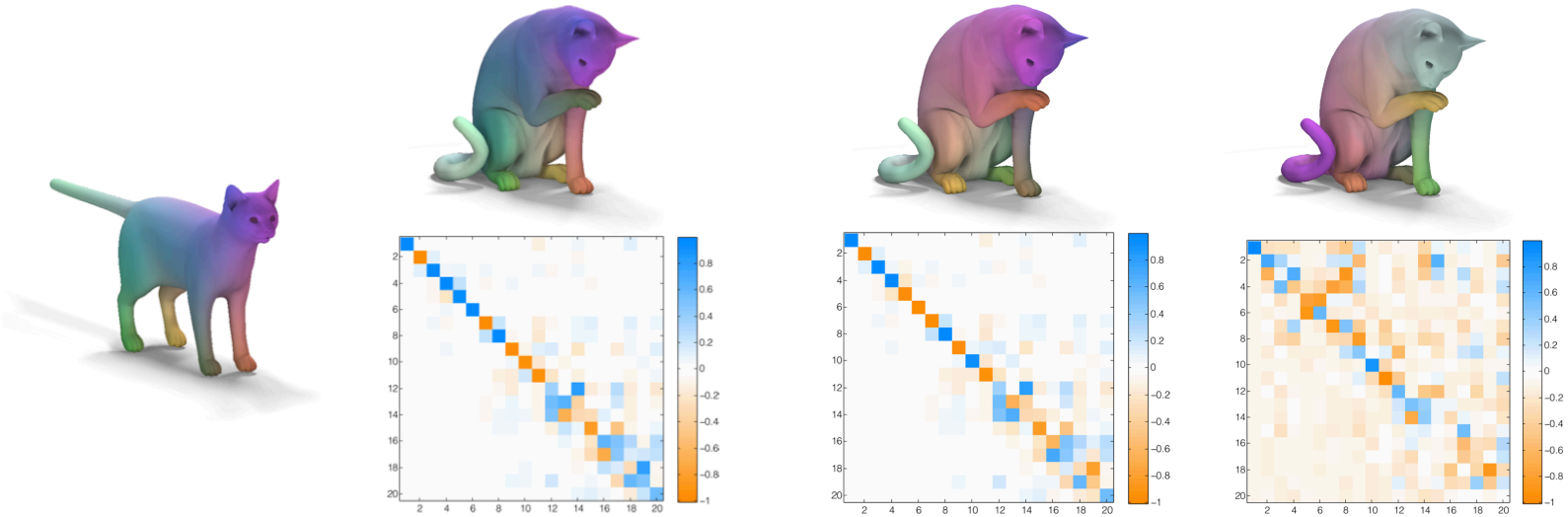
For a fixed choice of basis functions $\{\phi^M\}$ and $\{\phi^N\}$, and a bijection $T : M \to N$, define its **functional representation** as a matrix C, s.t. for all $f = \sum_i a_i \phi_i^M$, if $T_F(f) = \sum_i b_i \phi_i^N$ then:

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

If $\{\phi^M\}$ and $\{\phi^N\}$ are both orthonormal w.r.t. some inner product, then

$$C_{ij} = \left\langle T_F(\phi_i^M), \phi_j^N \right\rangle .$$

Encoding using LB basis

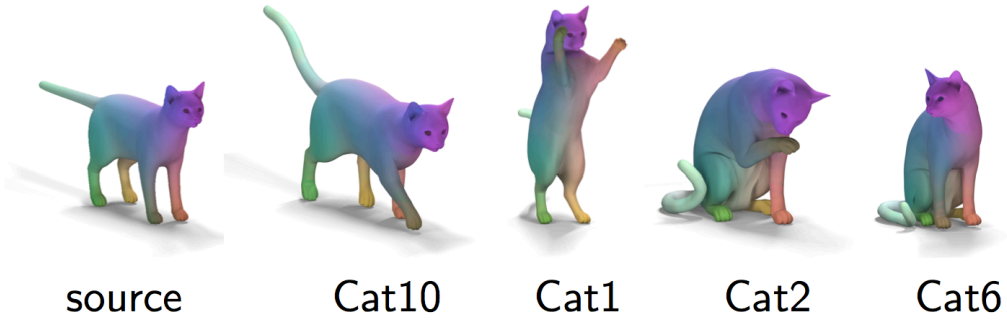


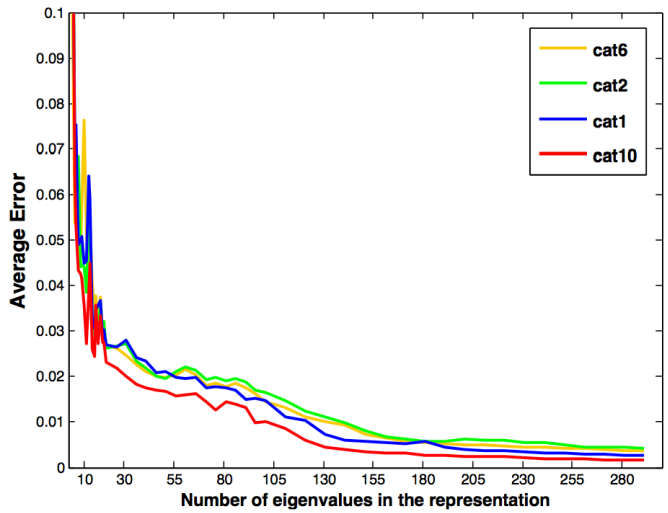
(a) *source* **(b)** *ground-truth map* **(c)** *left to right map* **(d)** *head to tail map*

ÉCOLE POLYTECHNIQUE ParisTech

Reconstructing from LB basis

Reconstruction error using a fixed size matrix.





ÉCOLE POLYTECHNIQUE ParisTech

Functional Map algebra

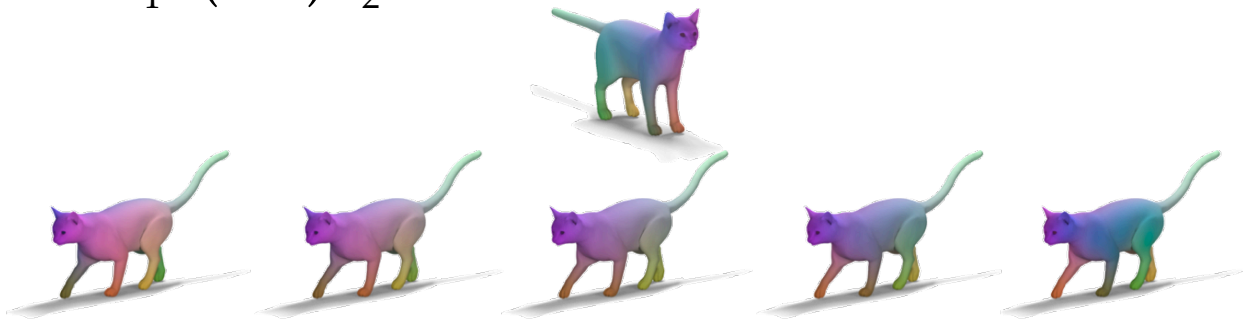
1. Map composition becomes matrix multiplication.

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

3. Algebraic operations on functional maps are possible.

E.g. interpolating between two maps with

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



(a) $\alpha = 0$ (b) $\alpha = 0.25$ (c) $\alpha = 0.5$ (d) $\alpha = 0.75$ (e) $\alpha = 1$

ÉCOLE POLYTECHNIQUE ParisTech

Map Constraints

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

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

where $f = \sum_i \mathbf{a_i}\phi_i^M, \quad g = \sum_i \mathbf{b}_i\phi_i^N$

Map Constraints

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

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

where $f = \sum_i \mathbf{a_i}\phi_i^M, \quad g = \sum_i \mathbf{b}_i\phi_i^N$

Given enough $\{\mathbf{a_i}, \mathbf{b}_i\}$ 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 $f : M \to \mathbb{R}$ and $g : N \to \mathbb{R}$ to correspond. Then, C must be s.t.

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

Function preservation constraint is general and includes:

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

Commutativity Constraints

In addition, we can phrase operator commutativity constraint, given two operators $S_1 : \mathcal{F}(M, \mathbb{R}) \to \mathcal{F}(M, \mathbb{R})$ and $S_2 : \mathcal{F}(N, \mathbb{R}) \to \mathcal{F}(N, \mathbb{R})$.

$$\begin{array}{ccc} \mathcal{F}(M, \mathbb{R}) & \xrightarrow{C} & \mathcal{F}(N, \mathbb{R}) \\ \downarrow{\scriptstyle S_1} & & \downarrow{\scriptstyle S_2} \\ \mathcal{F}(M, \mathbb{R}) & \xrightarrow{C} & \mathcal{F}(N, \mathbb{R}) \end{array}$$

Thus: $CS_1 = S_2 C$ or $\|CS_1 - S_2 C\|$

Note: this is a linear constraint on C. S_1 and S_2 could be symmetry operators or e.g. Laplace-Beltrami or Heat operators.

Regularization

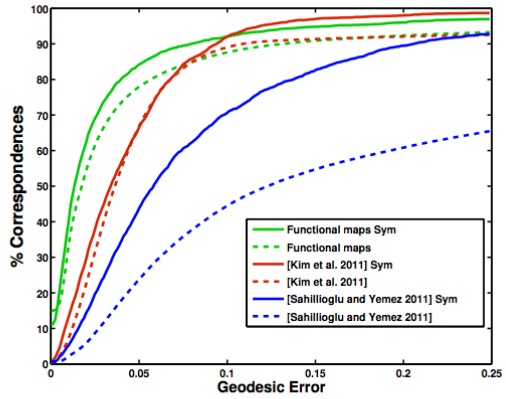
Lemma:

If the mapping is *locally volume preserving*, then the functional map matrix must be *orthonormal*.

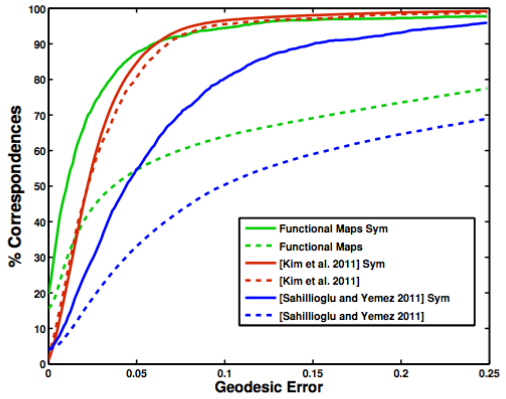
Together with this regularization, we get a very efficient shape matching method.

Results

A very simple method that puts together many constraints and uses 100 basis functions outperforms state-of-the-art:



SCAPE

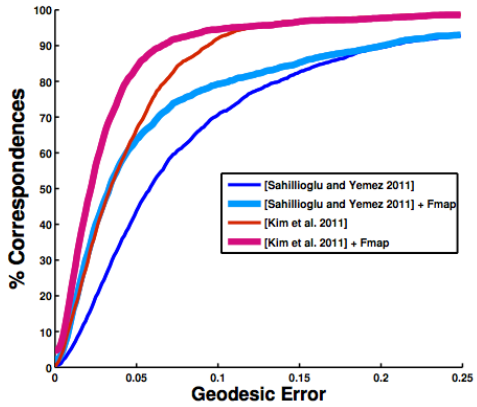


TOSCA

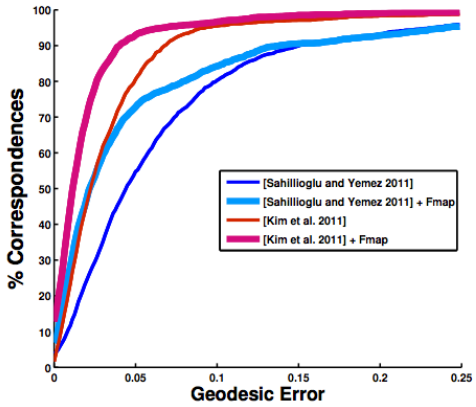
ÉCOLE POLYTECHNIQUE ParisTech

Results

Our representation is helpful for other methods too. If we treat their correspondences as constraints for a functional map (and do this iteratively), we improve their results.



SCAPE

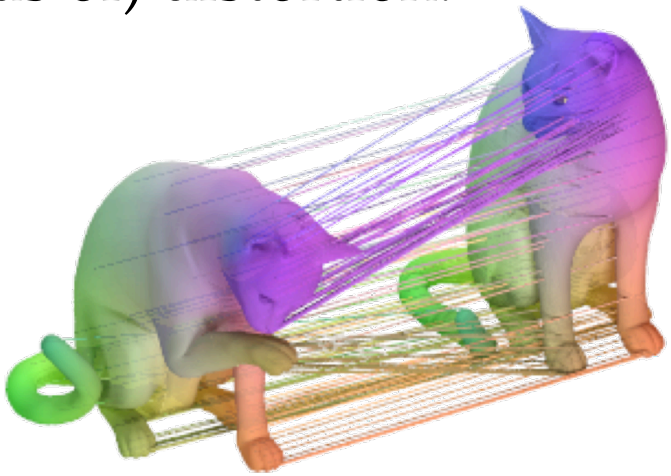


TOSCA

ÉCOLE POLYTECHNIQUE ParisTech

Motivation

- Given a pair of shapes (and a map between them), identify (areas of) distortion.



N

M

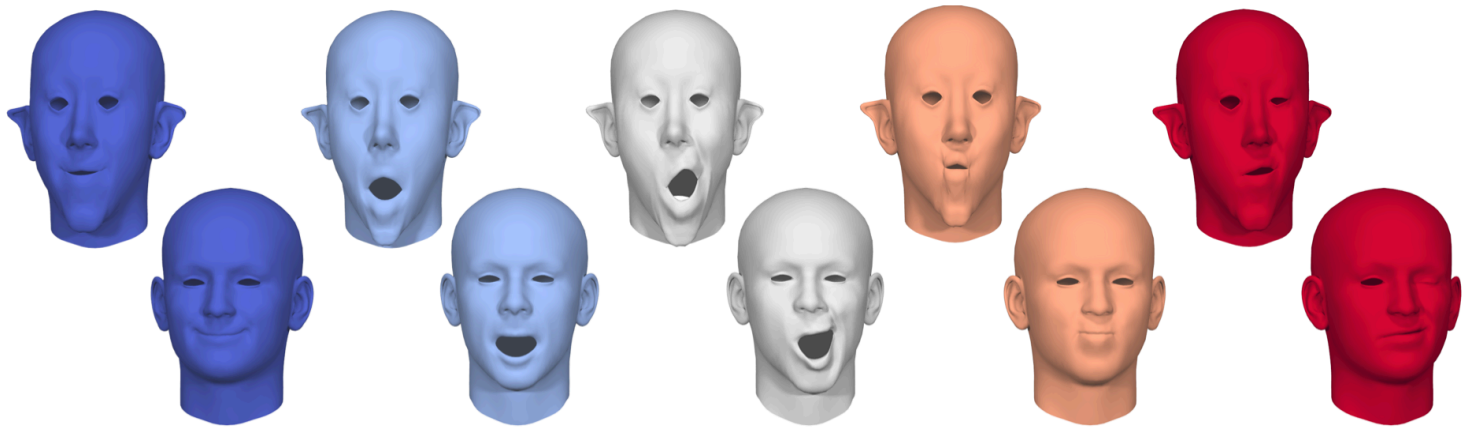
Large number of previously proposed metrics:

Global – Gromov-Hausdorff, Bag of Features, etc.

Local – Signature comparison, etc.

Shape Differences

- Fully characterize the distortion using two linear functional operators.
- Can compute areas of maximal distortion through eigendecomposition.
- Can *compare* distortion of different pairs A->B, vs C->D.



Map-Based Exploration of Intrinsic Shape Differences and Variability, *Rustamov, O., Azencot, Ben-Chen, Chazal, Guibas,* SIGGRAPH 2013

ÉCOLE POLYTECHNIQUE ParisTech

Shape Differences Overview

Given a function $f : M \to \mathbb{R}$

$$f = \sum_{i=0}^{\infty} a_i \phi_i$$

$$\|f\|^2 = \int_{x \in M} f^2(x) d\mu(x) = \sum_i a_i^2$$

Given a functional map C associated with a map T.

Then: $\dfrac{\|C\mathbf{a}\|_2}{\|\mathbf{a}\|_2}$ Measures the *distortion* of a function by T.

ÉCOLE POLYTECHNIQUE ParisTech

Shape Differences Overview

Given a functional map C associated with a map T.

Lemma 1:

$C^T C = I$ if and only if T is area preserving.

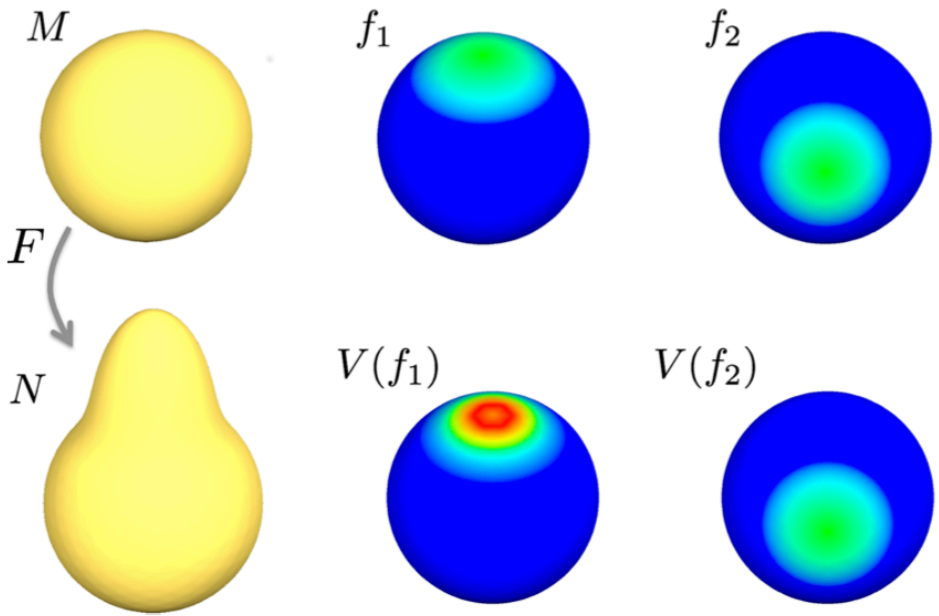
Lemma 2:

$L_1^{-1} C^T L_2 C = I$ if and only if T is area conformal.

Shape Differences Overview

Given a functional map C associated with a map T.

$$V = C^T C : \mathcal{F}(M) \rightarrow \mathcal{F}(M)$$



ÉCOLE POLYTECHNIQUE
ParisTech

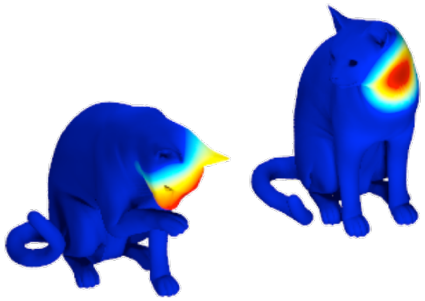
Shape Differences Overview

Given a functional map C associated with a map T.

$$V = C^T C : \mathcal{F}(M) \to \mathcal{F}(M)$$
$$R = L_1^{-1} C^T L_2 C : \mathcal{F}(M) \to \mathcal{F}(M)$$

The map T is an isometry if and only if V and R are both identity (linear condition).

Eigenfunctions of V and R capture areas of highest distortion.



ÉCOLE POLYTECHNIQUE ParisTech

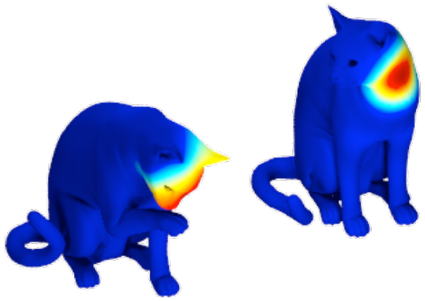
Shape Differences Overview

Given a functional map C associated with a map T.

$$V = C^T C : \mathcal{F}(M) \to \mathcal{F}(M)$$
$$R = L_1^{-1} C^T L_2 C : \mathcal{F}(M) \to \mathcal{F}(M)$$

The map T is an isometry if and only if V and R are both identity (linear condition).

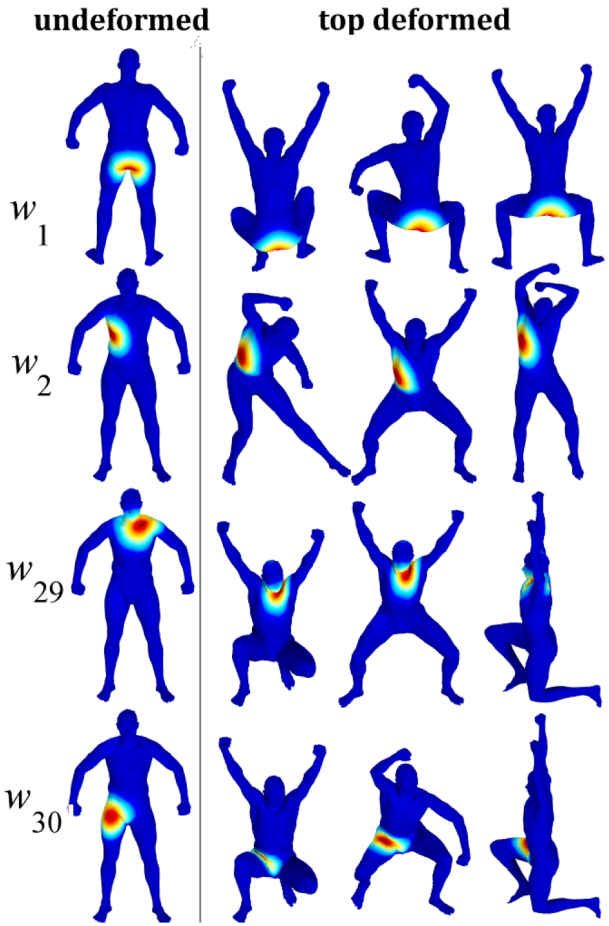
Eigenfunctions of V and R capture areas of highest distortion.



ÉCOLE POLYTECHNIQUE ParisTech

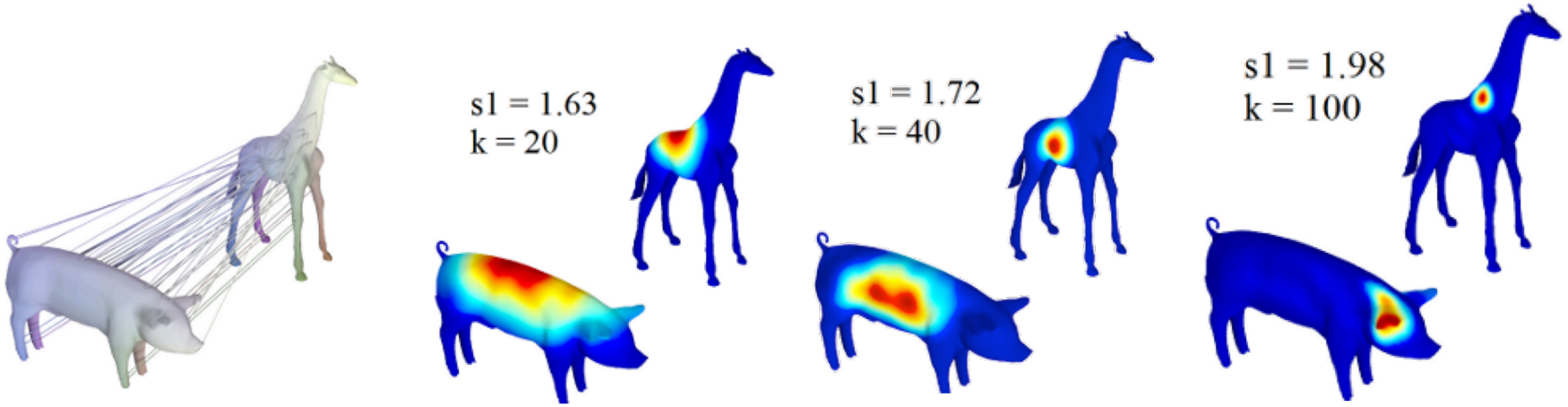
Our Approach – Multiple Shapes

- With same method, can visualize maps to *multiple* shapes.



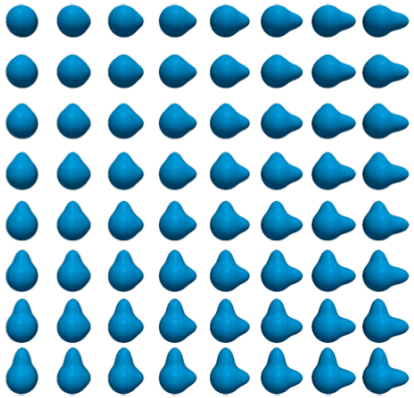
Our Approach

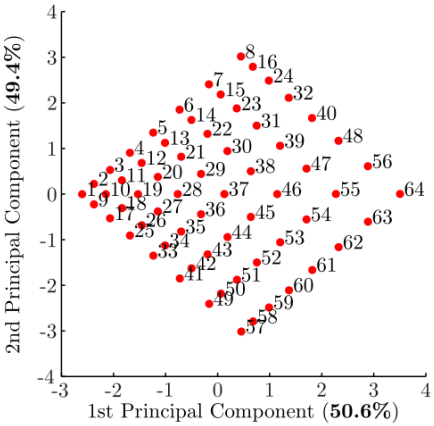
- By controlling the basis (i.e. which functions are allowed among the argmax), we can control the *scale*.



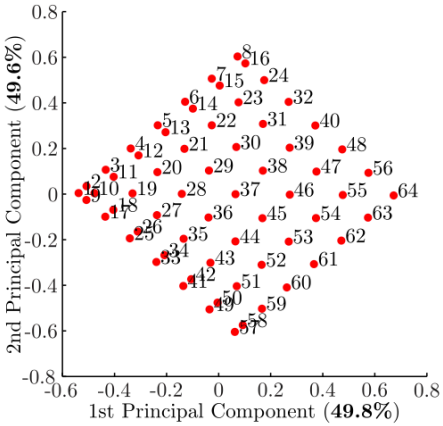
ÉCOLE POLYTECHNIQUE
ParisTech

Shape Differences in Collections

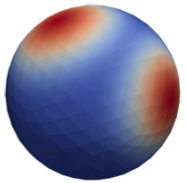




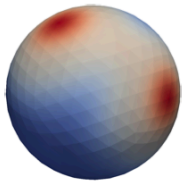
PCA on area-based shape differences



PCA on conformal shape differences



Variability localization for area

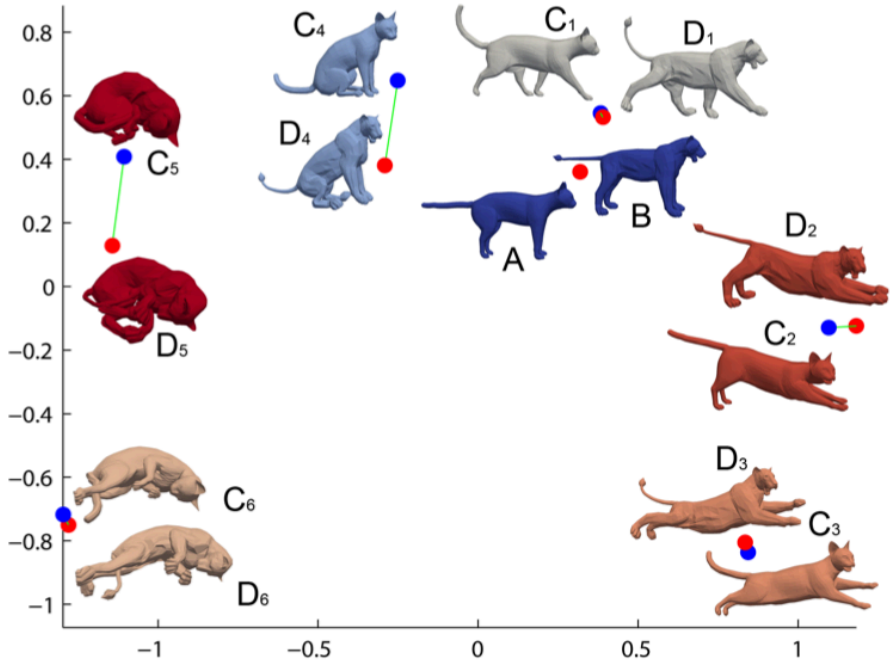


Variability localization for conformal

ÉCOLE POLYTECHNIQUE ParisTech

Comparing Shape Differences

Find a shape D_i, such that the difference between shapes B and D_i is as close as possible to the difference between A and C_i.

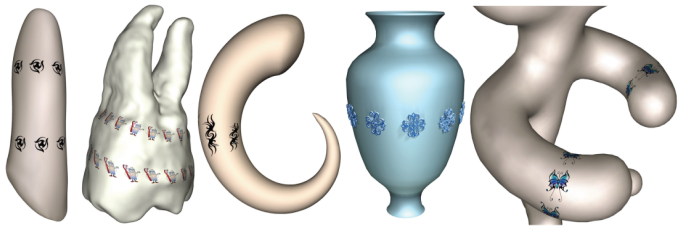


ÉCOLE POLYTECHNIQUE ParisTech

Motivation

- Many geometry processing operations require design and manipulation of tangent vector fields.

Pattern Generation

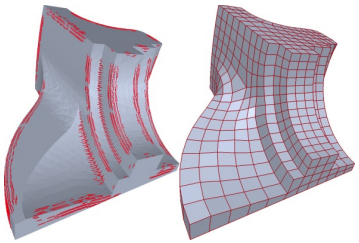


[Ben-Chen et al. 10]

Texture Synthesis

[Fisher et al. 07]

Quad Remeshing

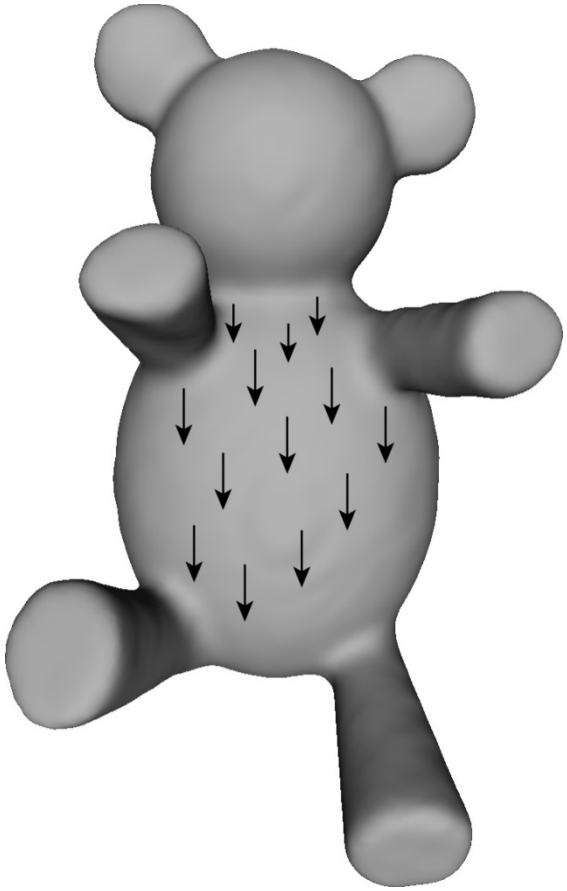


[Bommes et al. 09]

ÉCOLE POLYTECHNIQUE ParisTech

What is a Vector Field

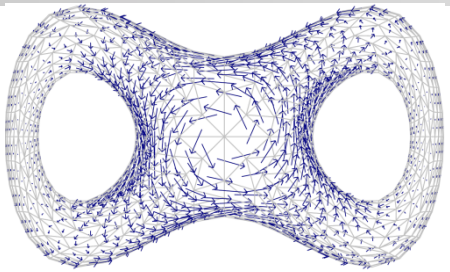
Vector Field

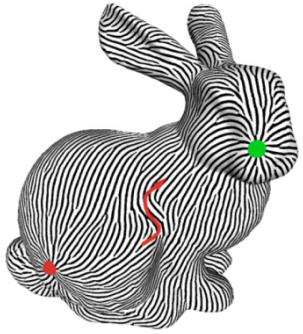


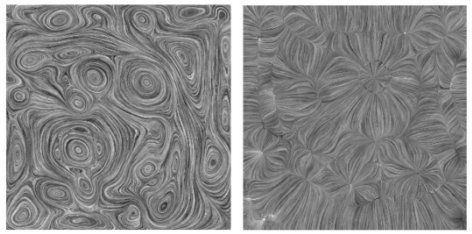
Flow

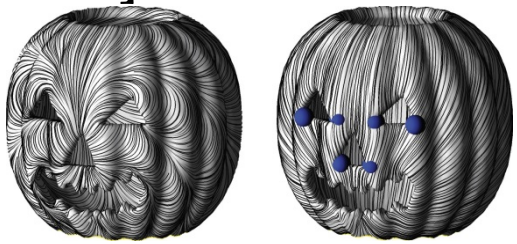
What is a Vector Field

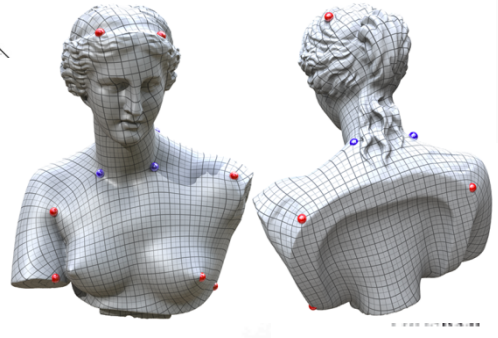
- Tangent vector per simplex
  - [Polthier et al. 03]
  - [Tong et al. 03]
- DEC
  - [Fisher et al. 07]
- *N*-RoSy fields
  - [Palacios et al. 07]
  - [Ray et al. 09]
  - [Crane et al. 10]





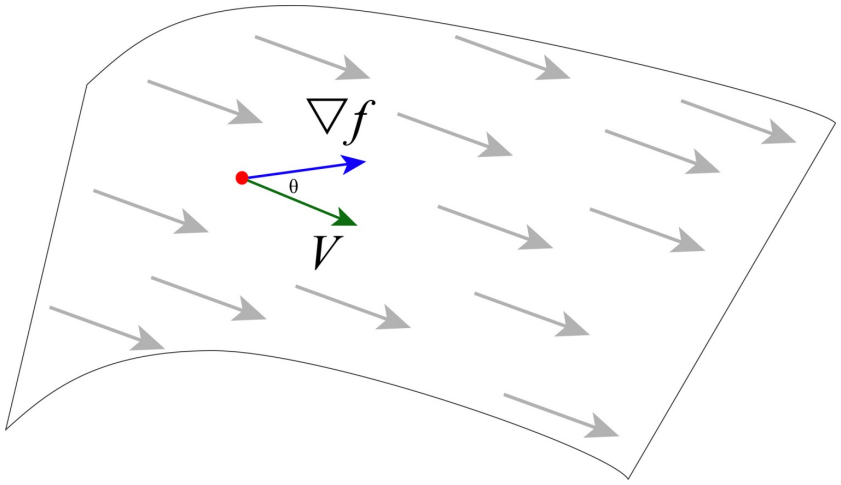






Our Approach

- Represent VFs using operators: $V \leftrightarrow D_V$
- D_V acts on smooth functions defined on M.
- A common view in differential geometry

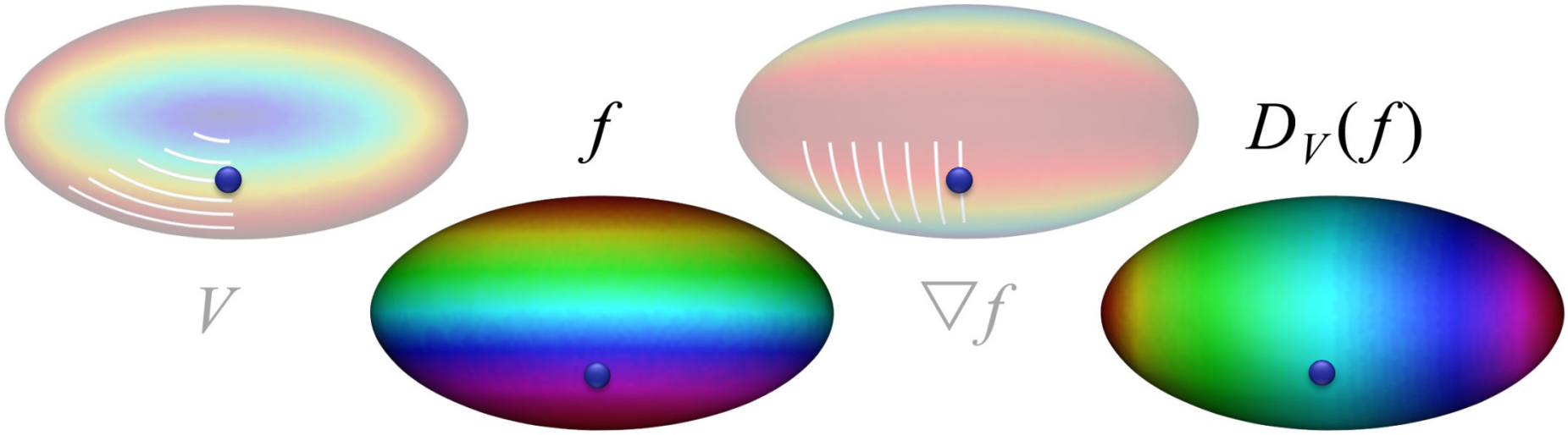


ÉCOLE POLYTECHNIQUE
ParisTech

Our Approach

For a given vector field V, define:

$$D_V(f) = \langle V, \nabla f \rangle$$



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

ÉCOLE POLYTECHNIQUE ParisTech

Our Approach

A functional operator D_V corresponds to some vector field if and only if it is:

- Linear

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

- Satisfies the product rule

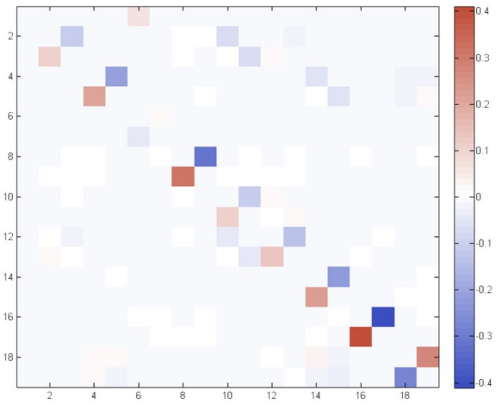
$$D_V(f_1 f_2) = f_1 D_V(f_2) + f_2 D_V(f_1)$$

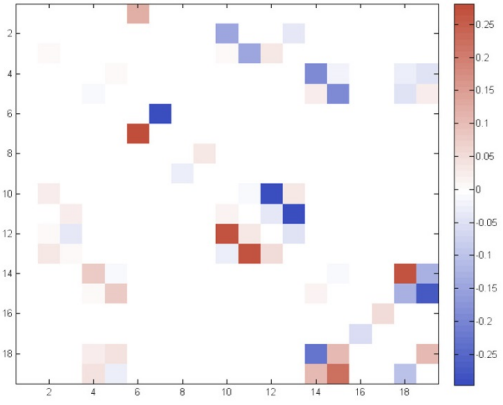
- If so, V can be reconstructed from D_V

Also called the covariant derivative.

Representing VFs

Using the LB eigenbasis.

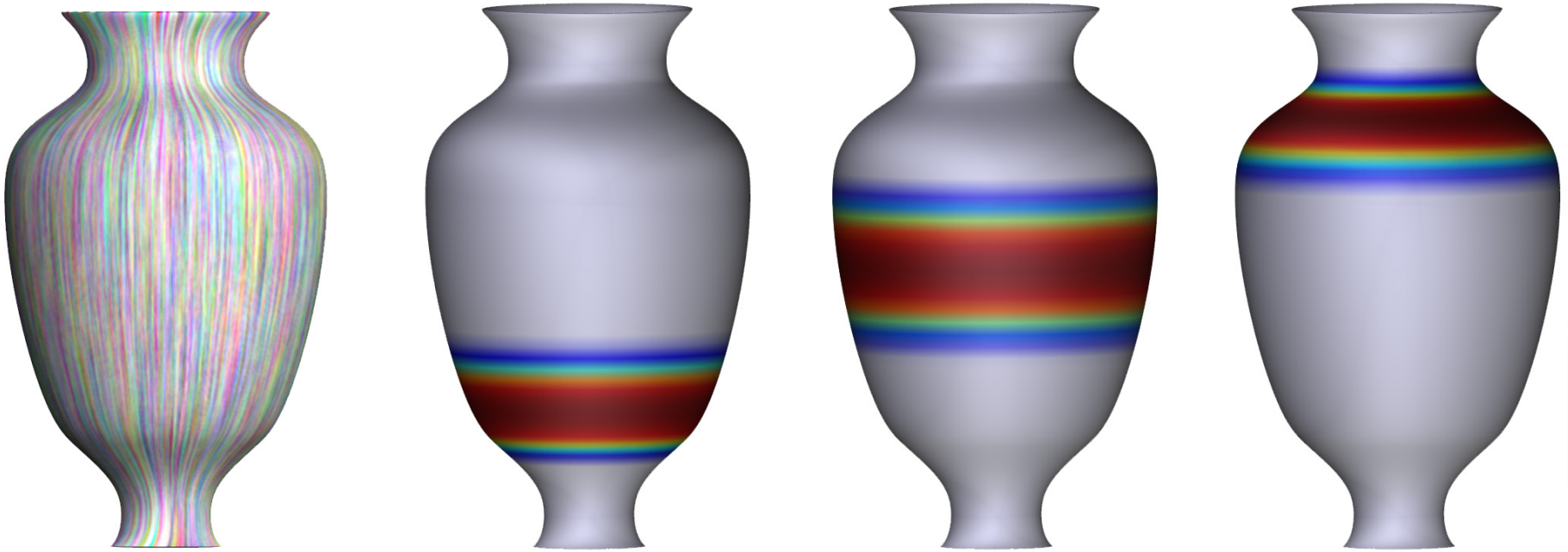




Relation to Functional Maps

The flow T_V^t is a self-map and the corresponding functional map

$$T_F^t = \exp(tD_V)$$

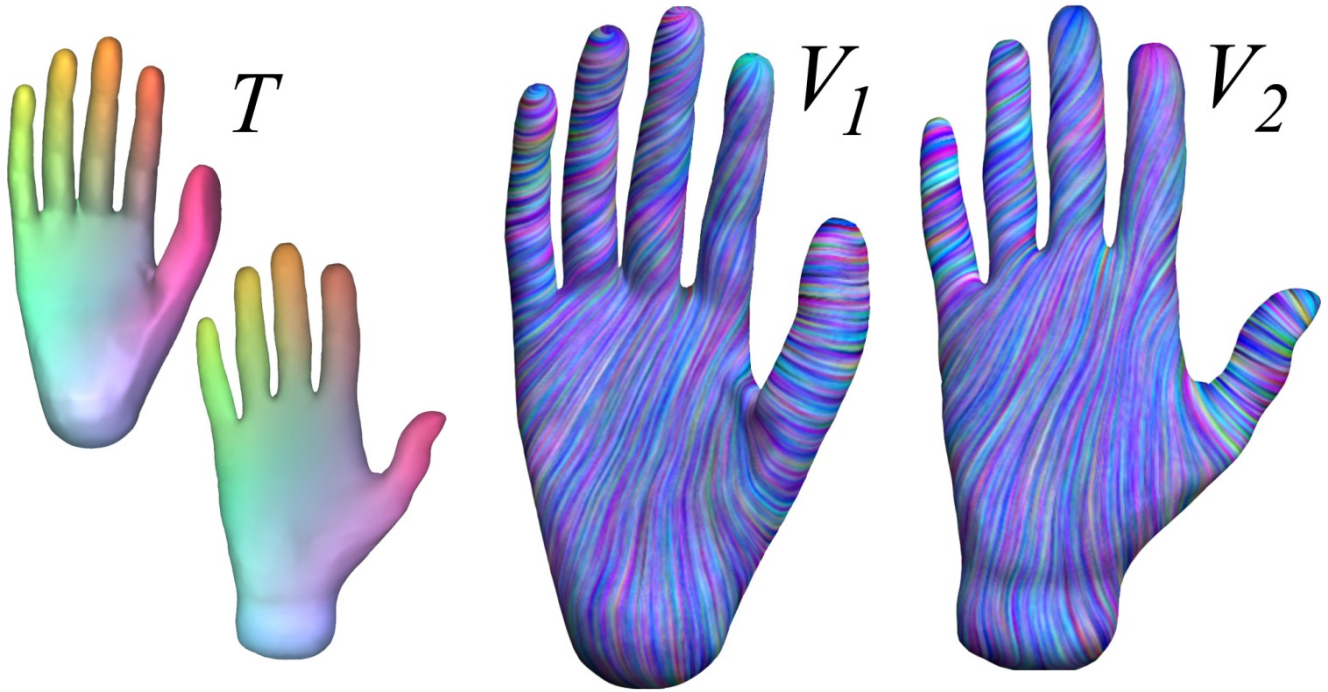


V

Relation to Functional Maps

Can transport vector fields by composing with a functional map

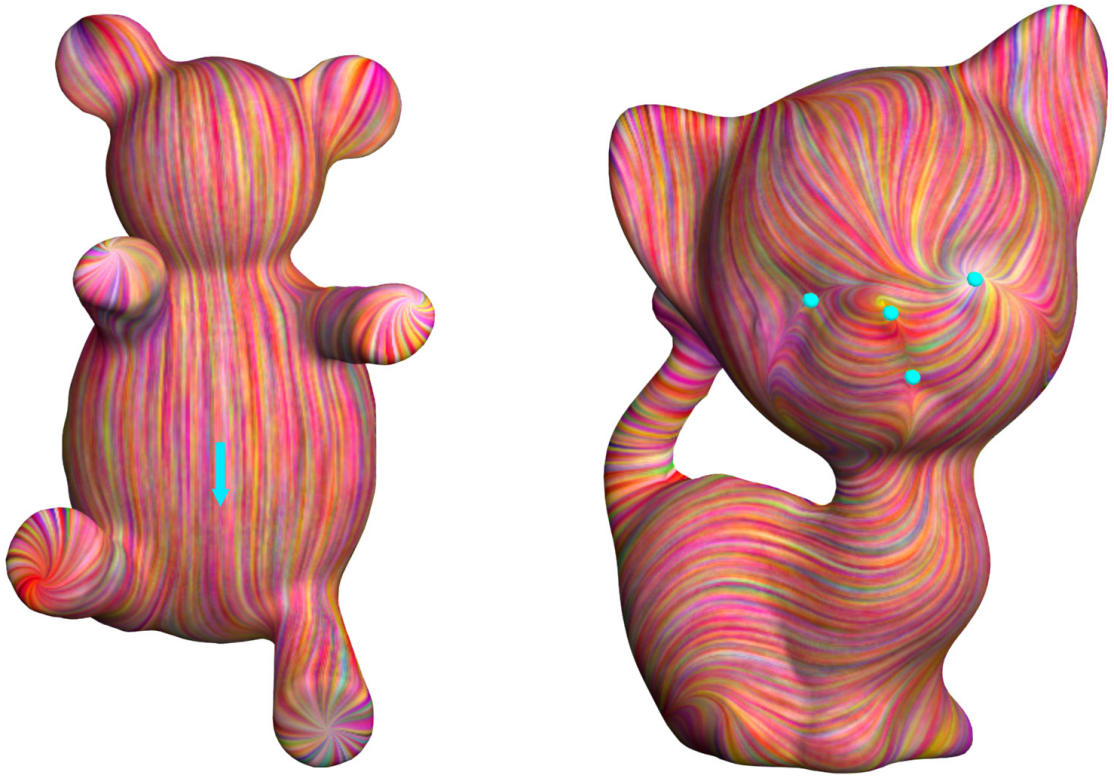
$$D_{V_2} = T_F^{-1} \circ D_V \circ T_F$$



ÉCOLE POLYTECHNIQUE ParisTech

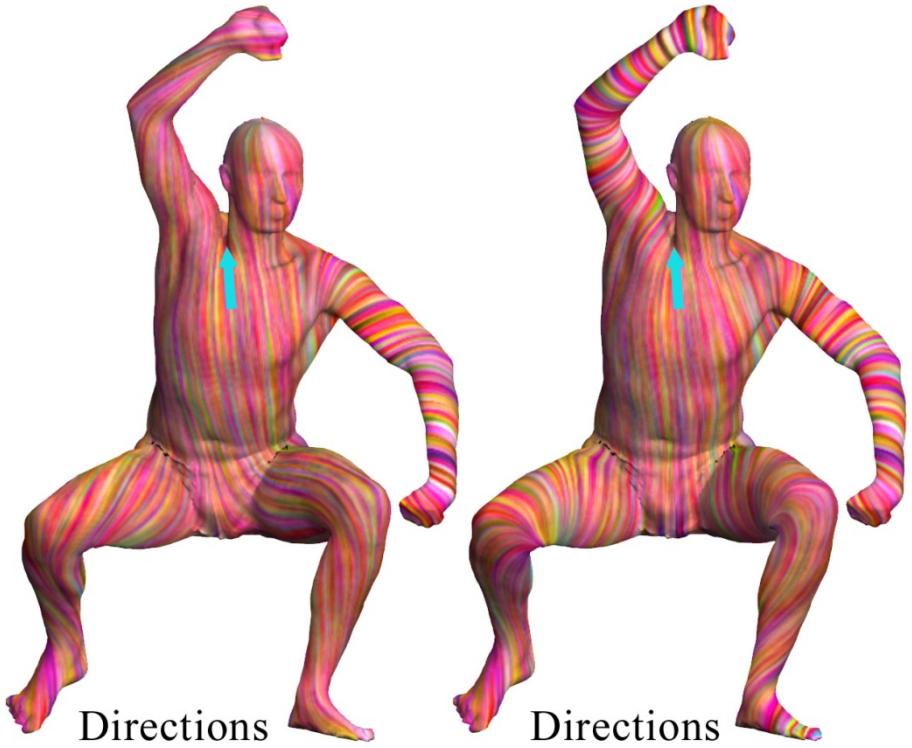
Generating Vector fields

Directional and singularity constraints are linear on the operator:



Generating Vector fields

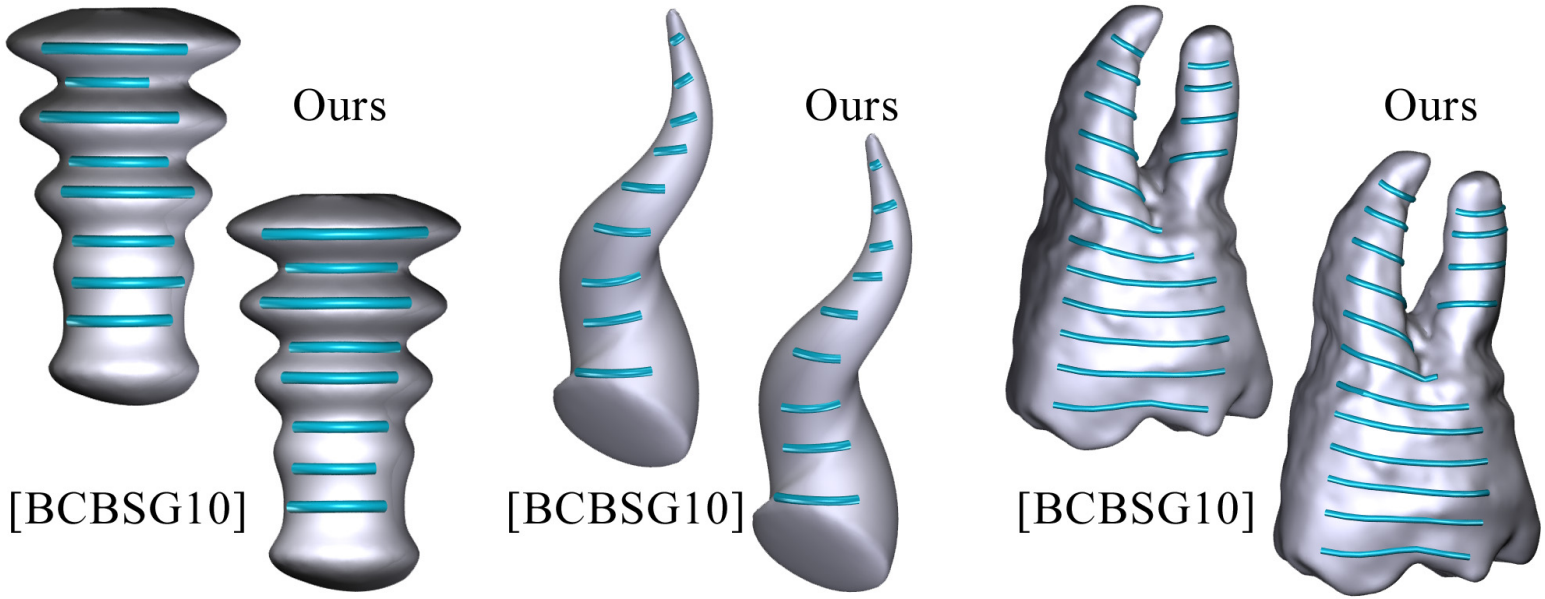
Symmetry constraints are linear



Generating Vector fields

Isometry constraints are linear:

$$\Delta \circ D_V = D_V \circ \Delta$$



ÉCOLE POLYTECHNIQUE ParisTech

Conclusion

- Many geometry processing tasks are best viewed as linear operators on functional spaces.
- Operator composition, inversion and *inference* all lead to simple algebraic operations
- Using multiscale bases can improve compactness.
- Performing spectral analysis on the operators can reveal the structure in a way that is easy to visualize.

Future Work

- A coherent mathematical framework to describe the different operators and functional spaces.
- Applications in other domains (images, point clouds, volumetric data).
- Analysis of second-order (composition) constraints on operators.
- Joint analysis of heterogeneous data.