Functional Shape Maps

A Flexible Representation of Maps Between Shapes

Maks Ovsjanikov\textsuperscript{\#}, Mirela Ben-Chen\textsuperscript{\|}, Justin Solomon\textsuperscript{\$}, Adrian Butscher\textsuperscript{\$} and Leonidas Guibas\textsuperscript{\$}

\textsuperscript{\#}LIX, École Polytechnique, \textsuperscript{\|}Technion \textsuperscript{\$}Stanford University

Jožef Stefan Institute
Tue, Feb 26, 2013
What is shape matching?

Given a pair of shapes, find a **correspondence** between them.
What is shape matching?

Given a pair of shapes, find a correspondence between them.
What is shape matching?

Given a pair of shapes, find a **correspondence** between them.
What is shape matching?

Given a pair of shapes, find a **correspondence** between them.

Finding the **best** alignment/map/correspondence.
What is shape matching?

Given a pair of geometric objects, find a **correspondence** between them.

Finding the **best** alignment/map/correspondence.
**Motivation**

**Why shape matching?**

Given a correspondence, we can *transfer*:

- texture and parametrization
- segmentation and labels
- deformation

Other applications: shape interpolation, reconstruction ...
Applications

Why shape matching?

- Manufacturing: one shape is a **model**, the other is a **scan**. Finding defects.
- Medicine: correspondences between 3D MRI scans.
- Deformable shape reconstruction.
- Shape interpolation.
- Statistical Shape Analysis.

*A Statistical Model of Human Pose and Body Shape*  
Hasler et al. Eurographics 09
Applications

Why shape matching?

- Manufacturing: one shape is a **model**, the other is a **scan**. Finding defects.
- Medicine: correspondences between 3D MRI scans.
- Deformable shape reconstruction.
- Shape interpolation.
- Statistical Shape Analysis.

*Female, 65kg*

*A Statistical Model of Human Pose and Body Shape*

Hasler et al. Eurographics 09
Applications

Why shape matching?

- Manufacturing: one shape is a **model**, the other is a **scan**. Finding defects.
- Medicine: correspondences between 3D MRI scans.
- Deformable shape reconstruction.
- Shape interpolation.
- Statistical Shape Analysis.
- Shape comparison.
- Parametrization.
- Exploration.
- Processing.
- etc., etc....
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
Method Taxonomy

Local vs. Global
refinement (e.g. ICP) vs. alignment (search)

Rigid vs. Deformable
rotation, translation vs. general deformation

Pair vs. Collection
two shapes vs. multiple shapes

Mostly Solved
Pairwise Rigid Correspondence

Iterative Closest Point:

1. For each $x_i \in M$ find its nearest neighbor $y_i \in N$.
2. Find the deformation $R, t$ minimizing:

$$\sum_{x_i \in M} \left\| Rx_i + t - y_i \right\|$$

Pairwise Rigid Correspondence

Iterative Closest Point:

For a given pair of point clouds $M$ and $N$. Iterate:

1. For each $x_i \in M$ find its nearest neighbor $y_i \in N$.
2. Find the deformation $R, t$ minimizing:

$$\sum_{x_i \in M} \| Rx_i + t - y_i \|.$$ 

Pairwise Rigid Correspondence

Iterative Closest Point:

1. For each $x_i \in M$ find its nearest neighbor $y_i \in N$.
2. Find the deformation $R, t$ minimizing:
   \[ \sum_{x_i \in M} \| Rx_i + t - y_i \| . \]

Pairwise Rigid Correspondence

Iterative Closest Point:

For a given pair of point clouds $M$ and $N$. Iterate:

1. For each $x_i \in M$ find its nearest neighbor $y_i \in N$.
2. Find the deformation $R, t$ minimizing:

$$\sum_{x_i \in M} \| Rx_i + t - y_i \|.$$ 

Pairwise Rigid Correspondence

Iterative Closest Point:

For a given pair of point clouds $M$ and $N$. Iterate:

1. For each $x_i \in M$ find its nearest neighbor $y_i \in N$.
2. Find the deformation $R, t$ minimizing:

$$\sum_{x_i \in M} \| Rx_i + t - y_i \|.$$

Pairwise Rigid Correspondence

Iterative Closest Point:

For a given pair of point clouds $M$ and $N$. Iterate:

1. For each $x_i \in M$ find its nearest neighbor $y_i \in N$.
2. Find the deformation $R, t$ minimizing:
   \[ \sum_{x_i \in M} \| R x_i + t - y_i \| \]

Pairwise Rigid Correspondence

Iterative Closest Point:

For a given pair of point clouds $M$ and $N$. Iterate:

1. For each $x_i \in M$ find its nearest neighbor $y_i \in N$.
2. Find the deformation $R, t$ minimizing:

$$\sum_{x_i \in M} \| Rx_i + t - y_i \|.$$ 

Pairwise Rigid Correspondence

Iterative Closest Point:

For a given pair of point clouds $M$ and $N$. Iterate:

1. For each $x_i \in M$ find its nearest neighbor $y_i \in N$.
2. Find the deformation $R, t$ minimizing:

$$\sum_{x_i \in M} \| Rx_i + t - y_i \|.$$

Pairwise Rigid Correspondence

Iterative Closest Point:

For a given pair of point clouds $M$ and $N$. Iterate:

1. For each $x_i \in M$ find its nearest neighbor $y_i \in N$.
2. Find the deformation $R, t$ minimizing:

$$\sum_{x_i \in M} \|Rx_i + t - y_i\|.$$ 

Pairwise Rigid Correspondence

Iterative Closest Point:

For a given pair of point clouds $M$ and $N$. **Iterate:**

1. For each $x_i \in M$ find its nearest neighbor $y_i \in N$.
2. Find the deformation $R, t$ minimizing:

$$\sum_{x_i \in M} \|Rx_i + t - y_i\|$$

Both steps can be done efficiently and the solution will converge given a rough initial alignment.
Global non-rigid shape matching remains very challenging.

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

Kim et al., *Blended Intrinsic Maps*, 2011

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

A recipe for non-rigid shape matching.

1. Sample a small set of landmark points on shape $M$.
2. Find matches for landmark points
   E.g. by preserving signatures & distances
   
   \[
   T_{\text{opt}} = \arg \min_T \sum_i \| S(p_i) - S(T(p_i)) \| + \sum_{ij} \| d^M(p_i, p_j) - d^N(T(p_i), T(p_j)) \| .
   \]
3. Find dense correspondences from sparse ones.

![Diagram of shape matching]
Motivation

A recipe for non-rigid shape matching.

1. Sample a small set of landmark points on shape $M$.
2. Find matches for landmark points
   E.g. by preserving signatures & distances
   \[
   T_{opt} = \arg\min_{T} \sum_i \| S(p_i) - S(T(p_i)) \| + \sum_{ij} \| d^M(p_i, p_j) - d^N(T(p_i), T(p_j)) \|.
   \]
3. Find dense correspondences from sparse ones.
Motivation

Although sub-sampling reduces complexity, the solution space is still discrete, making it difficult to:

1. Optimize given quadratic energy functions.
2. Enforce global and continuity constraints.
3. Incorporate uncertainty into prior data.
4. Return meaningful results in the presence of ambiguities.

Rather than limitations of the methods these are limitations of the representation.
Embed the shapes into a domain where matching is easier.
Spectral embeddings (e.g. Jain et al. ’06, Mateus et al. ’08, etc.): use LB eigenfunctions. Major challenge: embedding is unstable under tiny perturbations.
Möbius voting (Lipman et al. ’09, Kim et al. ’11): use Conformal Flattening (Uniformization) to embed each shape onto a 2d domain. Correspondence between embeddings computed in closed form using 3 point pairs.
Canonical Embeddings

1. Spectral embeddings (e.g. Jain et al. '06, Mateus et al. '08, etc.): use LB eigenfunctions. Major challenge: embedding is unstable under tiny perturbations.

2. Möbius voting (Lipman et al. '09, Kim et al. '11): use Conformal Flattening (Uniformization) to embed each shape onto a 2d domain. Correspondence between embeddings computed in closed form using 3 point pairs.

3. Heat kernel map (O. et al. '09): use the heat kernel to a fixed point to define a canonical embedding, with correspondence between embeddings from 1 point pair.
Canonical Embeddings

Existing embeddings are either (extremely) unstable or still require some number of **exact** fixed point correspondences.

Often these can only be established approximately.
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.
Given a pair of shapes and a pointwise bijection $T : M \rightarrow N$

The induced functional correspondence:

$$T_F(f) = g, \text{ s.t. } g(y) = f(T^{-1}(y)), y \in N.$$
Background

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

\[ T_F(f) = g : N \rightarrow \mathbb{R} \]

The induced functional correspondence:

\[ T_F(f) = g, \text{ s.t. } g(y) = f(T^{-1}(y)), y \in N. \]
Background

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

The induced functional correspondence:

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

The induced functional correspondence:

$$T_F(f) = g, \text{ s.t. } g(y) = f(T^{-1}(y)), y \in N.$$
Background

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

The induced functional correspondence:

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

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

The induced functional correspondence:

$$T_F(f) = g, \text{ s.t. } g(y) = f(T^{-1}(y)), y \in N.$$
Given a pair of shapes and a pointwise bijection $T: M \rightarrow N$

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

$T_F(f) = g$, s.t. $g(y) = f(T^{-1}(y))$, $y \in N$.

Note that $T_F$ is:

1. **Linear:** $T_F(\alpha_1 f_1 + \alpha_2 f_2) = \alpha_1 T_F(f_1) + \alpha_2 T(f_2)$.
2. **Complete:** recover $T$ from $T_F$ through indicator functions.
Given a pair of shapes and a pointwise bijection $T : M \rightarrow N$

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

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

$$T_F(f) = g, \text{ s.t. } g(y) = f(T^{-1}(y)), y \in N.$$ 

Note that $T_F$ is:

1. Linear: $T_F(\alpha_1 f_1 + \alpha_2 f_2) = \alpha_1 T_F(f_1) + \alpha_2 T(f_2)$.
2. Complete: recover $T$ from $T_F$ through indicator functions.
Given a pair of shapes and a pointwise bijection $T : M \rightarrow N$

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

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

$T_F$ \hspace{1cm} $T_F(f) = g$, s.t. $g(y) = f(T^{-1}(y))$, $y \in N$.

Note that $T_F$ is:

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

2. Complete: recover $T$ from $T_F$ through indicator functions.
Background

Function spaces are very easy to manipulate:

- The set of \((L^2)\) functions \(M \rightarrow \mathbb{R}\) forms a vector space.
- Function spaces have natural multi-scale bases.
- Function spaces are stable under deformations.

If \(\mathcal{F}_M\) and \(\mathcal{F}_N\) are both endowed with bases, then a linear map \(\mathcal{F}_M \rightarrow \mathcal{F}_N\) can be represented simply as a matrix.
Background

**Definition**

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

\[
\mathbf{b} = C \mathbf{b}
\]

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

\[
C_{ij} = \langle T_F(\phi^M_i), \phi^N_j \rangle.
\]
Example

For a pair of shapes with Laplace-Beltrami eigenfunctions and three maps:

source  direct  symmetric  head to tail

Note that the first two “natural” maps are much closer to being diagonal.
The functional representation is *flexible* since it allows different choices of bases for the function spaces. We use the Laplace-Beltrami eigenfunctions which are well suited for isometries and are:

1. Multi-scale and sparse.
2. Smooth.
3. Easy to compute.
Reconstruction Error

Multiscale:

Problem Formulation
Functional Maps
Applications
Conclusion & Future work

Reconstruction Error

Number of eigenvalues in the representation

Average Error

Sparsity:

source
Cat10 Cat1 Cat2 Cat6

0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1

0
10
20
30
40
50
60
70
80
90
100

120
130
140
150
160
170
180
190
200
210
220
230
240
250
260
270
280

Reconstruction Error

source  Cat10  Cat1  Cat2  Cat6

Sparsity:

\[
\text{nz = 4835} \quad \text{Cat0} \leftarrow \text{Cat6} \\
\text{nz = 4457} \quad \text{Cat0} \leftarrow \text{Cat10}
\]
Main Properties

The functional representation is inherently *continuous*. Also

1. Map composition becomes matrix multiplication.
3. Algebraic operations on functional maps are possible.

E.g. interpolating between two maps with

\[ C = \alpha C_1 + (1 - \alpha) C_2. \]
Map Constraints

Suppose we do not 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.

\[ C a \approx b. \]

where $f = \sum_i a_i \phi_i^M$, $g = \sum_i b_i \phi_i^N$. 
Map Constraints

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

$$Ca \approx b.$$

where $f = \sum_i a_i \phi_i^M$, $g = \sum_i b_i \phi_i^N$.

Given enough $a, b$ pairs, we can recover $C$ through a least squares system.
Suppose we do not 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.

$$Ca \approx b.$$

where $f = \sum_i a_i \phi_i^M$, $g = \sum_i b_i \phi_i^N$.

Given enough $a, b$ pairs, we can recover $C$ through a least squares system.
Map Constraints

Suppose we do not know \( C \). However, we expect a pair of functions \( f : M \rightarrow \mathbb{R} \) and \( g : N \rightarrow \mathbb{R} \), to correspond. Then \( C \) must be s.t.

\[
Ca \approx b.
\]

where \( f = \sum_i a_i \phi_i^M \), \( g = \sum_i b_i \phi_i^N \).

Given enough \( a, b \) pairs, we can recover \( C \) through a least squares system.
Map Constraints

Linearity of constraints: given a pair of functions $f : M \rightarrow \mathbb{R}$, $f = \sum_i a_i \phi_i^M$ and $g : N \rightarrow \mathbb{R}$, $g = \sum_i b_i \phi_i^N$ that are known to correspond, a constraint on $C$ becomes:

$$Ca \approx b.$$ 

Function preservation constraint is general and includes:

1. Texture preservation.
2. Descriptor preservation (e.g. Gauss curvature).
3. Landmark correspondences (e.g. distance to the point).
4. Segment correspondences (e.g. indicator function).
In addition, we can phrase operator commutativity constraint, given two operator $S_1 : \mathcal{F}(M, \mathbb{R}) \to \mathcal{F}(M, \mathbb{R})$ and $S_2 : \mathcal{F}(N, \mathbb{R}) \to \mathcal{F}(N, \mathbb{R})$.

\[
\begin{align*}
\mathcal{F}(M, \mathbb{R}) & \xrightarrow{C} \mathcal{F}(N, \mathbb{R}) \\
\mathcal{F}(M, \mathbb{R}) & \xrightarrow{C} \mathcal{F}(N, \mathbb{R})
\end{align*}
\]

Thus:

\[CS_1 = S_2 C \quad \text{or} \quad \|CS_1 - S_2 C\|.
\]

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.
What if the solution we get is not associated with a point-to-point mapping? Functional maps are strictly more general than point bijections. One thing we know:

**Theorem**

*If the underlying map $T$ (discrete or continuous) is locally volume preserving or the basis functions are discrete and orthonormal with respect to the standard inner product, the functional representation of $T$ must be orthonormal.*

Together with this regularization, we get a very efficient shape matching method.
A very simple method that puts together many constraints and uses 100 basis functions outperforms state-of-the-art:
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
Segmentation Transfer

To transfer functions we do not need to convert functional maps to bijections, since we can do it directly.

We can also transfer segmentations: for each segment, transfer its indicator function, and for each point pick the segment that gave the highest value.
Multishape Analysis

Given a collection of maps of related shapes, create a large matrix with a block for each pairwise maps. Iteratively taking powers of this matrix, we can “rewrite” maps as averages of other maps and removing bad outlier maps.
Even given a map $T : M \rightarrow N$, it is often hard to visualize it.

Common visualizations:
- Connecting (some) points by lines
- Plotting a function $f$ on $N$ and $f \circ T$ on $M$. 

Question: how to pick a “good” function $f$. 
Map Visualization

If $f$ is an indicator of a region $\Omega$, then:

$$\|f\|^2_2 = \sum_x f(x)^2 A(x) = \text{Area}(\Omega),$$
$$\|f \circ T\|^2_2 = \text{Area}(T^{-1}(\Omega)).$$

Finding the functions maximizing (minimizing) the ratio $\frac{\|f\|^2_2}{\|f \circ T\|^2_2}$ can identify regions most distorted by $T$. 

![Diagram showing $T^{-1}(\Omega)$ and $\Omega$]
Map Visualization

In the functional representation and LB basis:

\[
\frac{\|f\|_2^2}{\|f \circ T\|_2^2} = \frac{\|a\|_2^2}{\|Ca\|_2^2}
\]

and optimal functions are simply the *singular vectors* of \( C \).
Map Visualization

Singular vectors of the functional representation $C$ of $T$ identify most distorted regions in a multi-scale way.
Map Visualization

Can show that singular vectors of the functional representation $C$ of $T$ identify most distorted regions in a multi-scale way.
Conclusions:

- Introduced a representation for maps between shapes.
- Compact global and leads to efficient optimization.
- Constraints become linear in function space.
- Map processing and analysis enabled via linear algebra.
Future Work:

- Try other bases and *design* better bases.
- Consider optimal combinations of constraints.
- Other representations based on coupling vector spaces.
  - Higher order forms (e.g. vector spaces).
  - Fundamental groups (homology bases).
- Map analysis, processing and manipulation through their functional representation.
Thank you

Questions & Suggestions?

Acknowledgements: NSF grants FODAVA 0808515, IIS 0914833 and CCF 1011228, a research grant from Google, Inc., and the support of the Max Planck Center for Visual Computing and Communications.