Continuous and Orientation-preserving Correspondences via Functional Maps

Jing Ren, KAUST
Adrien Poulenard, École Polytechnique
Peter Wonka, KAUST
Maks Ovsjanikov, École Polytechnique
Shape matching

Source

Target
Corresponding descriptors

Gaussian curvatures
Joint segmentation
Wave kernel signature
Problems

Left-right ambiguity

Leg-arm ambiguity
How to break the **symmetry** ambiguity?

Our **solution**: define **orientation–preserving** operators in the **functional map** framework.

**Q1**: what is the standard **functional map pipeline** to solve the shape matching problem?

**Q2**: how to define the **orientation–preserving** operator? Why does it work?
Q1: **Functional map pipeline**

Laplacian–Beltrami operator

**Shape**

\( \phi_1^S \)

\( \phi_2^S \)

\( \phi_3^S \)

\( \phi_i^S \)

\( \phi_{\kappa s}^S \)
Q1: **Functional map pipeline**  

**Laplacian–Beltrami operator**

\[ f \approx a_1 + a_2 + a_3 + \ldots + a_i + a_{k_S} \]
Q1: **Functional map pipeline**

Source

\[ f \approx \Phi^S a \]

Target

\[ g \approx \Phi^T b \]

\[ \{f_i\}_{i=1}^k \xrightarrow{\Phi^S} \{a_i\}_{i=1}^k \]

\[ Ca_i = b_i \quad \forall i = 1, \ldots, k \]

\[ \{g_i\}_{i=1}^k \xrightarrow{\Phi^T} \{b_i\}_{i=1}^k \]
Q1: Functional map pipeline

\[ a = (\Phi^S)^{-1} f \]

\[ k_s = 50 \]

\[ k_f = 50 \]

Functional Map \( C \)

\[ a \]

\[ b \]

\[ \hat{g} \triangleq \Phi^T b \]

\[ g \approx \]

\[ f \]

\[ g \]
Given a pair of shapes $S_1, S_2$, with Laplacian operators $\Delta_{S_1}, \Delta_{S_2}$

- Compute the **functional bases** on the two shapes. Store them in matrices $\Phi^{S_1}, \Phi^{S_2}$

- Project the corresponding **descriptors** $\{f_i, g_i\}_{i=1}^k$ into the functional space. Store the **coefficients** in matrices $A = [a_1 \cdots a_k], B = [b_1 \cdots b_k]$

- Solve $C^* = \arg\min_{C \in \mathbb{R}^{k_T \times k_S}} E(C) = \|CA - B\|^2 + \|C\Delta_{S_1} - \Delta_{S_2}C\|^2$

- **Orientation-preservation** term $\sum_{i=1}^k \|C\Omega_{f_i} - \Omega_{g_i}C\|^2$
Q2: **Orientation-preserving operators**

A smooth map $T$ is orientation preserving if and only if $dT$ is orientation preserving.
Q2: Orientation-preserving operators

Map differential
\[ dT: T_p(S_1) \to T_q(S_2) \]

Frame at \( p \): \((w, v, n_p)\)
Frame at \( q = T(p) \): \((dT(w), dT(v), n_{T(p)})\)

\((w \times v)^T n_p\) should have the same sign as \((dT(w) \times dT(v))^T n_{T(p)}\)
Q2: Orientation-preserving operators

\[ f_1 \]
\[ \nabla f_1 \]
\[ f_2 \]
\[ \nabla f_2 \]
\[ g_1 \]
\[ \nabla g_1 \]
\[ g_2 \]
\[ \nabla g_2 \]
Q2: Orientation-preserving operators

\[ \nabla f_1 \]

\[ \nabla f_2 \]

\[ (\nabla f_1 \times \nabla f_2)^T n_{S_1} \]

\[ \nabla g_1 \]

\[ \nabla g_2 \]

\[ (\nabla g_1 \times \nabla g_2)^T n_{S_2} \]
Q2: **Orientation-preserving operators**

\[
(f \times \nabla h)^T n_{S_1}
\]

\[
(\nabla f \times \nabla C h)^T n_{S_2}
\]

- **Non-linear**
- **Linear!** - $\Omega(\cdot)$
Q2: Orientation–preserving operators

Shape

A descriptor function $f$
With symmetry ambiguity

$\Omega(f)$
antisymmetric

$\Omega(\cdot)$
Q2: Orientation-preserving operators

Shape

A descriptor function $f$
With symmetry ambiguity

$\Omega(f)$
antisymmetric
Q2: Orientation–preserving operators

- Shape

- A descriptor function $f$
  With symmetry ambiguity

- $\Omega(f)$
  antisymmetric
Orientation-preserving operators – defects

**Intrinsic** – no guarantee

**Induced** – depends on the descriptors
Refinement! – iterative closest point (ICP)?

Problems of ICP:
- **discontinuous**
- Only in the spectral domain – no **spatial** info

A better refinement to improve the **quality**?

**Without ground-truth correspondences**, how do we measure the quality of a map?
Refinement! – improve **smoothness**

Source

Target: map 01

Target: map 02

**smoother** – better!
Refinement! – remove outliers

Source

Target: map 01

Target: map 02

smoother (no outliers) – better!
Refinement! – improve **coverage**

**Source**

Black region: **not covered** (no correspondence)

Coverage: **#vertex (or surface area) % covered by the map**

**Target: map 01**

Coverage: **48.9%**

**Target: map 02**

Coverage: **81.4%**

More vertices **covered** – better!
Refinement! – improve **bijectivity**

$(x, y)$ corresponds to each other in **both** $T_{12}$ and $T_{21}$
Bijective and Continuous ICP (BCICP)

- **Bijectivity**
  - Soft constraints: $T_{21} \circ T_{12}$ and $T_{12} \circ T_{21}$ are close to identity

- **Continuity**
  - Smooth the displacement vector field

- **Coverage**
  - Find the nearest neighbor with the largest preimage size

- **Outliers**
  - Detected by edge distortion and fixed by the nearest neighbor classified as an inlier
Benchmark datasets

**FAUST dataset [Bogo et al. 2014]**
- 10 different humans in 10 different poses
- We tested on
  - 200 isometric pairs
  - 400 non-isometric pairs

**TOSCA dataset [Bronstein et al 2008]**
- 80 different humans and animals in 9 categories
- We tested on
  - 568 isometric pairs
  - 190 non-isometric pairs
Measurement – Accuracy

Point-wise map to measure $T$

- $d_1$: direct error
- $d_2$: symmetric error
- $\min(d_1, d_2)$: per-vertex error

Ground-truth correspondence

Ground-truth symmetric correspondence
Results – **FAUST Isometric dataset (200 pairs)**

- **Solid** lines: our methods (with different descriptors)
- **Dashed** lines: state-of-the-art methods
Results – TOSCA non-Isometric dataset (190 pairs)

Solid lines: our methods (with different descriptors)
Dashed lines: state-of-the-art methods
Summary

• Introduce **orientation-preserving operator** into functional map framework

• Propose a refinement technique, **BCICP**, which improves the **bijectivity**, **continuity**, and **coverage**

• Verify the **usefulness** of the orientation-preserving operator and the BCICP refinement on **large datasets**, w.r.t. **different measurements**
Thanks for your attention 😊

Continuous and Orientation-preserving Correspondences via Functional Maps

Jing Ren, KAUST
Adrien Poulenard, École Polytechnique
Peter Wonka, KAUST
Maks Ovsjanikov, École Polytechnique
## Results – Summary

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Dataset (pairs)</th>
<th>FAUST</th>
<th>TOSCA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Isometric 200</td>
<td>Non-isometric 400</td>
<td>Isometric 586</td>
</tr>
<tr>
<td>Per-vertex</td>
<td>17.8%</td>
<td>24.1%</td>
<td>31.4%</td>
</tr>
<tr>
<td>Per-map</td>
<td>17.5%</td>
<td>18.4%</td>
<td>14.6%</td>
</tr>
<tr>
<td>Direct</td>
<td>17.5%</td>
<td>18.4%</td>
<td>38.8%</td>
</tr>
</tbody>
</table>

Relative improvement of our method (**directOp + BCICP**) over the best baseline methods w.r.t. the average error of three measurements.
Q2: Orientation-preserving operators

Encode into functional map framework:
• Given a functional map $C$ maps function on $S_1$ to function on $S_2$
• For every pair of corresponding descriptor $(f, g), f \in \mathcal{F}(S_1), g \in \mathcal{F}(S_2)$
• We can add the orientation-preserving constrain $C \left( (\nabla f \times \nabla h)^T n_{S_1} \right) = (\nabla g \times \nabla C(h))^T n_{S_2}$
  • $\nabla f, \nabla g$: vector field on the source and target mesh resp.
  • $h, C(h)$: functions defined on the source and target resp.
  • $\nabla h, \nabla C(h)$: vector field
  • $(\nabla f \times \nabla h)^T n_{S_1}$ defines a function on the source
  • $(\nabla g \times \nabla C(h))^T n_{S_2}$ defines a function on the target
  • they should correspond to each other! – use functional map to transport the function
• $C \left( (\nabla f \times \nabla h)^T n_{S_1} \right) = (\nabla g \times \nabla C(h))^T n_{S_2}$
• OrientationPreserving($C$) for every pair of corresponding descriptors $(f, g)$
Q2: Orientation-preserving operators

- Define \( \omega(w, v, n) = (w \times v, n) \)
- If \( T \) is orientation preserving, we have \( \text{sign} \left( \omega(w, v, n_p) \right) = \text{sign} \left( \omega(dT(w), dT(v), n_{T(p)}) \right) \)
  for any pair of \( w, v \in \mathcal{T}_p(S_1) \), and for any vertex \( p \)

Encode into functional map framework:
- Given a pair of corresponding descriptor \( (f, g) \), \( f \in \mathcal{F}(S_1), g \in \mathcal{F}(S_2) \)
- \( \forall h \in \mathcal{F}(S_1) \), it is mapped to \( C(h) \in \mathcal{F}(S_2) \) via a functional map \( C \)
- Orientation-preserving:
  - For any point \( p \in S_1 \) (corresponds to \( q \in S_2 \) via \( C \))
  - \( \omega(\nabla f(p), \nabla h(p), n_p) \approx \omega(\nabla g(q), \nabla (C(h))(q), n_q) \)
  - Define a function \( \Omega(\cdot, \cdot, \cdot) \in \mathcal{F}(S_1) \) such that \( \Omega(p) = \omega(\cdot_p, \cdot_p, \cdot_p) \)
  - \( C \left( \Omega(\nabla f, \nabla h, n_{S_1}) \right) = \Omega(\nabla g, \nabla C(h), n_{S_2}) \)
  - OrientationPreserving\( C \) for every pair of corresponding descriptors \( (f, g) \)
BCICP refinement – improve bijectivity

Notation:
• Point-wise map (vector!) $T_{12}: i$–th vertex on $S_1$ is mapped to $T_{12}(i)$–th vertex on $S_2$
• Shape $S_i$ has Laplacian–Beltrami Basis $\Phi_i$
• Functional map (matrix!) $C_{12}$: maps the functional space $\mathcal{F}(S_1|\Phi_1)$ to $\mathcal{F}(S_2|\Phi_2)$

Recall ICP:
• $C_{12}$ is associated with $T_{21}$
  • Arbitrary function $f \in \mathcal{F}(S_1)$ can be transported to $S_2$ in two ways:
    • Using the point-wise map directly, i.e., $g = f(T_{21}) \in \mathcal{F}(S_2)$
    • Use the functional map, i.e., $g = \Phi_2(C_{12}(\Phi_1^f))$
• Two transportations should give similar result – for any $f$
• Minimize $\|\Phi_2 C_{12} - \Phi_1(T_{21},:)\|^2$
  • ICP: alternatively solve for $C_{12}$ and $T_{21}$
BCICP refinement – improve bijectivity

Bijective in the spectral domain:
- $\| \Phi_2 C_{12} - \Phi_1 (T_{21}, :) \|^2 + \| \Phi_2 C_{21} - \Phi_1 (T_{12}, :) \|^2$

Bijective in the spatial domain:
- $T_{21} \circ T_{12}$: maps $S_1$ to itself ($S_1$)
- We can add a similar term $\| \Phi_1 C_{11} - \Phi_1 (T_{21} \circ T_{12}, :) \|^2$
  - where $C_{11}$ is an auxiliary variable, maps the functional space $\mathcal{F}(S_1 | \Phi_1)$ to itself
- We can similarly define the energy for $T_{12} \circ T_{21}$

New energy
- $\lambda_1 \| \Phi_2 C_{12} - \Phi_1 (T_{21}, :) \|^2$
- $\lambda_2 \| \Phi_2 C_{21} - \Phi_1 (T_{12}, :) \|^2$
- $\lambda_3 \| \Phi_1 C_{11} - \Phi_1 (T_{21} \circ T_{12}, :) \|^2$
- $\lambda_4 \| \Phi_2 C_{22} - \Phi_2 (T_{12} \circ T_{21}, :) \|^2$
BCICP refinement – improve bijectivity

ICP energy
\[ \| \Phi_2 C_{12} - \Phi_1 (T_{21}, : ) \|^2 \]

New energy
\[ \lambda_1 \| \Phi_2 C_{12} - \Phi_1 (T_{21}, : ) \|^2 \\
+ \lambda_2 \| \Phi_2 C_{21} - \Phi_1 (T_{12}, : ) \|^2 \\
+ \lambda_3 \| \Phi_1 C_{11} - \Phi_1 (T_{21} \circ T_{12}, : ) \|^2 \\
+ \lambda_4 \| \Phi_2 C_{22} - \Phi_2 (T_{12} \circ T_{21}, : ) \|^2 \]
BCICP refinement – improve smoothness

Source
VtxPos $X$

Target
VtxPos $Y$

Displacement
$t_{xi} = y_{T(i)} - x_i$

$T(i)$
BCICP refinement – improve smoothness

Source $VtxPos_X$  \[ \rightarrow \] Target $VtxPos_Y$

Average displacement $\bar{\ell}$

Nearest-neigh in $Y$ to $x_i + \bar{\ell}$

$T(i)$

Average displacement $\bar{\ell}$

$\bar{u}$

[Papazov and Burschka 2011]
BCICP refinement – improve smoothness

Source VtxPos $X$

Target VtxPos $Y$

Average displacement $\bar{\tau}$

Nearest-neigh in $Y$ to $x_i + \bar{\tau}$

Smoothed map
BCICP refinement – remove outliers

Source

Target

Edges with large distortion

Trustworthy vertices

Outlier vertices

Final map

Compute the edge distortion $r = d_2 / d_1$

Remove these edges from the adjacency matrix of the mesh

Find the outlier region

Outlier – find its NN in the “trustworthy” vertex set

Find the largest connected component
BCICP refinement – improve coverage

Shape $S_1$

Shape $S_2$
Measurement – Bijectivity

- $d$: geodesic distance between $p$ and $T_{21}(T_{12}(p))$
- Measure the difference between $T_{21}(T_{12}(\cdot))$ and the identity map.
Measurement – Smoothness

• $\frac{d_2}{d_1}$: edge distortion ratio to measure the smoothness
Refinement

Heuristic measurement
• High bijectivity
• High smoothness
  • No outliers
  • High coverage

Our solution – bijective and continuous ICP (BCICP)
• Refine the map simultaneously in the spectral and the spatial domain
• Improve the bijectivity, continuity, and coverage
Q1: Functional map pipeline

Source

\[ f \approx a_1 + a_2 + a_3 + \ldots + a_i \]

Target

\[ g \approx b_1 + b_2 + b_3 + \ldots + b_j \]

Descriptors \( f/g \)

\[ \phi_1^S \]
\[ \phi_2^S \]
\[ \phi_3^S \]
\[ \phi_i^S \]
\[ \phi_{kS}^S \]

\[ \phi_1^T \]
\[ \phi_2^T \]
\[ \phi_3^T \]
\[ \phi_j^T \]
\[ \phi_{kT}^T \]

48 of 30
Q1: **Functional map pipeline**

Source

\( \phi_1^S \)
\( \phi_2^S \)
\( \phi_3^S \)

Target

\( \phi_1^T \)
\( \phi_2^T \)
\( \phi_3^T \)

Laplacian–Beltrami basis

\( \phi_{1s}^S \)
\( \phi_{2s}^S \)
\( \phi_{3s}^S \)

... ...

\( \phi_{1t}^T \)
\( \phi_{2t}^T \)
\( \phi_{3t}^T \)

... ...

49 of 30
Q2: Orientation-preserving operators

Map differential
\( dT: \mathcal{T}_p(S_1) \to \mathcal{T}_q(S_2) \)

Frame at \( p \): \((w, v, n_p)\)
Frame at \( q = T(p) \): \((dT(w), dT(v), n_{T(p)})\)

\((w \times v)^T n_p\) should have the same sign as \((dT(w) \times dT(v))^T n_{T(p)}\)
Q1: Functional map pipeline

• First introduced by Ovsjanikov et al in 2012: “Functional Maps: A Flexible Representation of Maps between Shapes”

• Map a function defined on the source shape to another function on the target

• The functions defined on the source/target shape are represented in a compressed form using the Laplacian–Beltrami basis