MapTree: Recovering Multiple Solutions in the Space of Maps

Jing Ren, KAUST
Simone Melzi, Ecole Polytechnique
Maks Ovsjanikov, Ecole Polytechnique
Peter Wonka, KAUST
Map & Correspondences

Triangle meshes
Map & Correspondences

Source Shape

Target Shape

#3 of 30
• Point-based methods
  • [Bronstein et al. 2006],
  • [Huang et al. 2008]…

• Parametrization-based methods
  • [Lipman and Funkhouser 2009]
  • [Aigerman et al. 2017]…

• Optimal transport
  • [Solomon et al. 2016]
  • [Mandad et al. 2017]…

• Functional maps
  • [Ovsjanikov et al. 2012]
  • [Ezuz and Ben-Chen 2017]…
Problem Formulation

\[ \min_{T_{12}} E(T_{12}) \]

- **Objectives** \( E(\cdot) \)
  - geodesic distortion
  - Dirichlet energy
  - ...
- **Constraints**
  - bijective map
  - ...

How to compute a map?
Geodesic distortion

$$E(T_{12}) = \sum_{(v_i, v_j)} \left\| D_1(v_i, v_j) - D_2(T_{12}(v_i), T_{12}(v_j)) \right\|$$

• $D_k(v_i, v_j)$ stores the geodesic distance between the two vertices $v_i$ and $v_j$ on shape $S_k$
Geodesic distortion

\[ E(T_{12}) = \sum_{(v_i, v_j)} \left\| D_1(v_i, v_j) - D_2(T_{12}(v_i), T_{12}(v_j)) \right\| \]

Complicated map space

- Discrete
- Not differentiable w.r.t. \( T_{12} \)
- Complicated \( O(n^n) \)
- Multiple local-minima
- ……
Existing solutions

• [Sung and Kim 2013] “Finding the M-best consistent correspondences between 3D symmetric objects”
  • For each vertex, find multiple correspondences candidates to resolve global symmetry ambiguity

• [Sahillioğlu and Yemez 2013] ”Coarse-to-fine Isometric Shape Correspondence by tracking symmetric flips”
  • Avoid symmetry flip during map computation

Avoid undesired local minima?
Find all meaningful maps!

Landscape of the geodesic distortion

Source
Find all meaningful maps!

Landscape of the geodesic distortion

Source  symmetric  double flip  direct  back-to-front

#10 of 30
Observations

• Intrinsic objectives: not discriminative w.r.t. self-symmetries
• Multiple ground-truth can exist!

Inspirations

• Avoid symmetry flip during computation
• Find all meaningful maps and select later!
• Use Functional Map representation
Laplace-Beltrami Operator

Helmholtz Equation
\[ \Delta_S f = \lambda f \]

Shape \( S \)

\[ 0 = \lambda_1 \leq \lambda_2 \leq \lambda_3 \leq \lambda_4 \leq \ldots \]
Functional map

Source

Self-Map

\[ \phi_1 \]

\[ +1 \]

\[ \phi_2 \]

\[ +1 \]

\[ \phi_3 \]

\[ -1 \]
Functional map

Source

Self-Map

#14 of 30
Functional map

Source

Self-Map

$\phi_1 \quad \phi_2 \quad \phi_3$

$+1 \quad -1 \quad +1$
Observations

• Intrinsic (global) symmetry information is encoded in the spectral domain of the shapes

• Pointwise maps can be organized in the spectral domain along the frequencies

• Use F-norm between two functional maps to approximate the distance between two pointwise maps
Map Tree Exploration

Algorithms
1. Functional map expansion rules
2. Map tree exploration
Contributions

• **Progressive exploration of the map space**
  • Start from the smallest functional map
  • Expand the functional map along the frequency domain
  • Only keep good & sufficiently different maps at each iteration
Contributions

• Output *multiple* maps, e.g.,
  • 4 maps for human v.s. human, human v.s. gorilla
  • 2 maps for animal shape pairs
  • ≥4 maps for man-made objects (table, knots, glasses, cup…)
• Previous work: a *single* map
Contributions

• **Bijective ZoomOut** for map refinement
  • ZoomOut [Melzi et al 2019]: spectral refinement method that enforces the **orthogonality** of a functional map
  • We propose Bijective ZoomOut that enforces both
    • **Orthogonality** of a single functional map
    • **Bijectivity** of the functional maps from both sides
  • **Similar** computation complexity
  • **Better** accuracy
Results

• Multi-solution shape matching
• Self-symmetry detection
• Non-rigid shape matching
Multi-solution shape matching

Source shape

- Direct map
- Symmetric
- Back-to-front
- Left-to-right & Back-to-front
Multi-solution shape matching

Source shape

Left-to-right

Upside-down

Left-to-right & Upside-down
Multi-solution shape matching

Source

OrientRev

OrientRev + ZoomOut

Ours
Non-rigid shape matching

Symmetry ambiguity!

Source Ours #26 of 30
Non-rigid shape matching

SHREC’19 challenge: state-of-the-art accuracy

<table>
<thead>
<tr>
<th>Methods \ Measurement</th>
<th>Accuracy ($\times 10^{-3}$)</th>
<th>GeoDist ($\times 10^2$)</th>
<th>Dirichlet Energy</th>
<th>Conformal Distortion</th>
<th>Runtime (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIM</td>
<td>83.69</td>
<td>1.418</td>
<td>3.278</td>
<td>1.970</td>
<td>164</td>
</tr>
<tr>
<td>GroupRep</td>
<td>311.1</td>
<td>5.254</td>
<td>13.41</td>
<td>7.787</td>
<td>3.95</td>
</tr>
<tr>
<td>IntSymm</td>
<td><strong>62.50</strong></td>
<td>1.945</td>
<td>12.17</td>
<td>7.123</td>
<td>2.05</td>
</tr>
<tr>
<td>OrientRev (Ini)</td>
<td>137.2</td>
<td>4.682</td>
<td>22.07</td>
<td>12.69</td>
<td>0.52</td>
</tr>
<tr>
<td>Ini + ICP</td>
<td>108.9</td>
<td>3.604</td>
<td>10.49</td>
<td>6.235</td>
<td>8.33</td>
</tr>
<tr>
<td>Ini + PMF</td>
<td>119.4</td>
<td>2.444</td>
<td>15.98</td>
<td>9.605</td>
<td>425</td>
</tr>
<tr>
<td>Ini + RHM</td>
<td>118.7</td>
<td>4.166</td>
<td>7.352</td>
<td>4.369</td>
<td>28.7</td>
</tr>
<tr>
<td>Ini + BCICP</td>
<td>96.72</td>
<td>2.466</td>
<td>5.633</td>
<td>3.741</td>
<td>157</td>
</tr>
<tr>
<td>Ini + ZoomOut</td>
<td>80.30</td>
<td>2.858</td>
<td>6.601</td>
<td>3.838</td>
<td>6.58</td>
</tr>
<tr>
<td>MapTree - GT</td>
<td><strong>39.62</strong></td>
<td>1.512</td>
<td>3.763</td>
<td>0.949</td>
<td>65.2</td>
</tr>
<tr>
<td>MapTree - Auto</td>
<td><strong>47.48</strong></td>
<td>1.507</td>
<td>3.833</td>
<td><strong>0.929</strong></td>
<td>65.2</td>
</tr>
</tbody>
</table>

We also propose an auto-selection algorithm via cycle-consistency!
Self-symmetry detection

Source shape

Obtained map

Highlight intrinsic partial self-symmetry
Robustness w.r.t. decimation

- $n = 500$
- $n = 100$
- $n = 50$
- $n = 1K$
- $n = 7K$
- $n = 5K$
MapTree: Recovering Multiple Solutions in the Space of Maps

Jing Ren, KAUST
Simone Melzi, Ecole Polytechnique
Maks Ovsjanikov, Ecole Polytechnique
Peter Wonka, KAUST

Code: https://github.com/llorz/SGA20_mapExplor