Co-alignment of shape collections
Functional maps

Anastasia Dubrovina
Computer Science Dept.
Stanford University

Slides credits: L. Guibas, Q. Huang, V. Kim, J. Solomon
Mapping Between Data Sets

- Multiscale mappings
  - Point/pixel level
  - Part level

Maps capture what is the same or similar across two data sets
Why Do We Care About Maps and Alignments?

- To stitch data together
- To transfer information
- To compute distances and similarities
- To perform joint analysis
Estimate maps between pairs of shapes

- Given a pair of shapes, find corresponding points
Pairwise intrinsic shape matching

Mobius voting [Lipmann and Funkhouser 2009]

Distortion of $m_1$  Distortion of $m_2$  Distortion of $m_3$

Blended intrinsic maps [Kim et al., 2011]

One-point isometric matching [Ovsjanikov et al., 2010]

Functional maps [Ovsjanikov et al., 2012, 2013]
Today: Joint alignment in shape collections

Maps Join Data Together
Pairwise maps: problems and issues

Symmetry, ambiguity, scale, bad data
Non-Convex, Combinatorial Optimization

multiple minima

$n!$ permutations

Symmetry, ambiguity, scale, bad data
Combining Maps

Blended intrinsic maps
[Kim et al. 11]

Composition can correct correspondences
Individual Data Set Operations Can Have Errors

- The interpretation of a particular piece of geometric data is deeply influenced by our interpretation of other related data.

3D Segmentation
Lecture outline

• Unsupervised joint shape segmentation - optimization approach

• Joint matching in shape collections using cycle-consistency

• Joint matching in shape collections based on functional maps
JOINT SHAPE SEGMENTATION
Motivation

• Extraneous geometric clues

Single shape segmentation
[Chen et al. 09]

Joint shape segmentation
[Huang et al. 11]
Motivation

- Low saliency

Single shape segmentation [Chen et al. 09]

Joint shape segmentation [Huang et al. 11]
Motivation

- Articulated structures - vs. previous methods

Single shape segmentation
[Chen et al. 09]

Joint shape segmentation
[Huang et al. 11]
Pair-wise Joint Segmentation

Objective:

$$\max_{S_1, S_2} \text{score}(S_1) + \text{score}(S_2) + \text{consistency}(S_1, S_2)$$
Pair-wise Joint Segmentation

• Outline of the next slides
  • Segmentation parameterization
  • Segmentation score
  • Consistency score
  • 0-1 linear programming formulation
Segmentation Parameterization

Patches
[Golovinskiy and Funkhouser 08]

Super-pixels
[Ren and Malik 03]

Randomized Cuts

Initial Segments
Segmentation Parameterization

- Segmentations: subsets of initial segments obtained from randomized segmentations
Segmentation Parameterization

- Segmentations: subsets of initial segments obtained from randomized segmentations
- Segmentation constraints: each point is in exactly one segment

\[ |\text{cover}(p)| = 1, \quad \forall p \in W \]

The set of initial segments that cover point \( p \)

Randomized Cuts

Initial Segments
Segmentation Parameterization

- Segmentations: subsets of initial segments obtained from randomized segmentations
- Segmentation constraints: each point is in exactly one segment
- Segmentation score

\[
\text{score}(S) = \sum_{s \in S} \text{area}(s) r_s = \sum_{s \in S} \bar{w}_s
\]

Prevent tiny segments

Segment repetitions in randomized segmentations
Consistency Term

- Consistent segmentation = there exists a mapping between segments of two shapes
  - Many-to-one
  - Partial

Many-to-one correspondences  Partial similarity
Consistency Term

- Consistency defined in terms of mappings
  - Many-to-one
  - Partial
- Mapping score [Anguelov et al. 05]

\[
\text{score}(\mathcal{M}_{ij}) = \lambda \sum_{c \in \mathcal{M}_{ij}} \bar{w}_c + \mu \sum_{(c, c') \in \mathcal{A}_{ij}} \bar{w}(c, c')
\]

- Correspondence weight [Osoda et al. 02]
- Adjacent correspondence pair weight
Consistency Term

- Consistency defined in terms of mappings
  - Many-to-one
  - Partial
- Mapping score [Anguelov et al. 05]

\[
\text{score}(\mathcal{M}_{ij}) = \lambda \sum_{c \in \mathcal{M}_{ij}} \overline{w}_c + \mu \sum_{(c, c') \in A_{ij}} \overline{w}_{(c, c')}
\]

- Consistency score

\[
\text{consistency}(S_1, S_2) = \sum_{ij \in \{12, 21\}} \max \text{ score}(\mathcal{M}_{ij})
\]
Consistency Term

• Consistency defined in terms of mappings
  • Many-to-one
  • Partial
• Mapping score [Anguelov et al.05]

\[
\text{score}(\mathcal{M}_{ij}) = \lambda \sum_{c \in \mathcal{M}_{ij}} w_c + \mu \sum_{(c,c') \in \mathcal{A}_{ij}} w(c,c')
\]

• Consistency score

\[
\text{consistency}(S_1, S_2) = \sum_{ij \in \{12, 21\}} \max \text{ score}(\mathcal{M}_{ij})
\]

• Mapping constraints

\[
|\text{cover}(p)| = 1, \quad \forall p \in \mathcal{P}_i, \quad 1 \leq i \leq 2,
\]
0-1 Linear Programming Formulation

- Introduce binary indicators

Segments

\[ x_s = \begin{cases} 
1 & s \in S_1 \cup S_2 \\
0 & \text{otherwise} 
\end{cases} \]
0-1 Linear Programming Formulation

• Introduce binary indicators

Segments

\[ x_s = \begin{cases} 
1 & s \in S_1 \cup S_2 \\
0 & \text{otherwise}
\end{cases} \]

Correspondences

\[ y_c = \begin{cases} 
1 & c \in M_{12} \cup M_{21} \\
0 & \text{otherwise}
\end{cases} \]
0-1 Linear Programming Formulation

- Introduce binary indicators

\[
x_s = \begin{cases} 
1 & s \in S_1 \cup S_2 \\
0 & \text{otherwise}
\end{cases}
\]

\[
y_c = \begin{cases} 
1 & c \in M_{12} \cup M_{21} \\
0 & \text{otherwise}
\end{cases}
\]

\[
z_{(c, c')} = \begin{cases} 
1 & (c, c') \in A_{12} \cup A_{21} \\
0 & \text{otherwise}
\end{cases}
\]
0-1 Linear Programming Formulation

- Introduce binary indicators

<table>
<thead>
<tr>
<th>Segments</th>
<th>Correspondences</th>
<th>Correspondence pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_s = \begin{cases} 1 &amp; s \in S_1 \cup S_2 \ 0 &amp; \text{otherwise} \end{cases} )</td>
<td>( y_c = \begin{cases} 1 &amp; c \in M_{12} \cup M_{21} \ 0 &amp; \text{otherwise} \end{cases} )</td>
<td>( z(c,c') = \begin{cases} 1 &amp; (c,c') \in A_{12} \cup A_{21} \ 0 &amp; \text{otherwise} \end{cases} )</td>
</tr>
</tbody>
</table>

Objective function:

\[
\max \sum_{i \in \{1,2\}} x_i^T w_i^\text{seg} + \sum_{i,j \in \{12,21\}} (\lambda y_{ij}^T w_{ij}^\text{corr} + \mu z_{ij}^T w_{ij}^\text{adj})
\]
0-1 Linear Programming Formulation

- Introduce binary indicators

<table>
<thead>
<tr>
<th>Segments</th>
<th>Correspondences</th>
<th>Correspondence pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_s = \begin{cases} 1 &amp; s \in S_1 \cup S_2 \ 0 &amp; \text{otherwise} \end{cases}$</td>
<td>$y_c = \begin{cases} 1 &amp; c \in M_{12} \cup M_{21} \ 0 &amp; \text{otherwise} \end{cases}$</td>
<td>$z(c,c') = \begin{cases} 1 &amp; (c,c') \in A_{12} \cup A_{21} \ 0 &amp; \text{otherwise} \end{cases}$</td>
</tr>
</tbody>
</table>

Segmentation constraints

$$\sum_{s \in \text{cover}(p)} x_s = 1, \quad \forall p \in P_i$$

Mapping constraints

- Selected correspondences form a valid mapping

Compatibility constraints

- Selected correspondences and correspondence pairs are consistent
0-1 Linear Programming Formulation

- Linear programming relaxation

\[
\begin{align*}
\max \quad & \sum_{i \in \{1,2\}} x_i^T w^i + \sum_{ij \in \{12,21\}} (\lambda y_{ij}^T w_{ij}^{corr} + \mu z_{ij}^T w_{ij}^{adj}) \\
\text{s.t.} \quad & A_1 x_1 = 1 \\
& B_{12} y_{12} \leq D_{12} x_1 \\
& B'_{12} y_{12} \leq D'_{12} x_2 \\
& E_{12} z_{12} \leq F_{12} y_{12} \\
\text{and} \quad & 0 \leq x \leq 1 \\
& A_2 x_2 = 1 \\
& B_{21} y_{21} \leq D_{21} x_2 \\
& B'_{21} y_{21} \leq D'_{21} x_1 \\
& E_{21} z_{21} \leq F_{21} y_{21} \\
& \forall x \in x_1, x_2, y_{12}, y_{21}, z_{12}, z_{21}
\end{align*}
\]
Multi-way joint segmentation

- Input shapes
  - Different objects
  - Different categories
Multi-way joint segmentation

- Perform all pair-wise joint segmentation to determine pairs of similar shapes
Multi-way joint segmentation

- Objective function

\[
\sum_{i=1}^{n} \text{score}(S_i) + \sum_{(S_i, S_j) \in \mathcal{E}} \text{consistency}(S_i, S_j)
\]
Alternating Optimization

Shape-wise optimizations:

\[
\max_{x_i} \quad x_i^T w_i^{\text{seg}} - \frac{n}{|\mathcal{E}|} \sum_{\{j|(i,j)\in\mathcal{E}\}} \gamma \|x_i - x_{ij}\|^2 \\
\text{s.t.} \quad A_i x_i = 1, \quad 0 \leq x_i \leq 1
\]

Pair-wise optimizations:

\[
\max_{x_{ij}, y_{ij}, z_{ij}} \quad \lambda y_{ij}^T w_i^{\text{corr}} + \mu z_{ij}^T w_i^{\text{adj}} - \gamma \|x_i - x_{ij}\|^2 \\
\text{s.t.} \quad B_{ij} y_{ij} \leq D_{ij} x_{ij}, \quad B'_{ij} y_{ij} \leq D'_{ij} x_{ij}, \quad E_{ij} z_{ij} \leq F_{ij} y_{ij}, \quad A_i x_{ij} = 1, \quad 0 \leq x_{ij} \leq 1, \quad 0 \leq y_{ij} \leq 1, \quad 0 \leq z_{ij} \leq 1
\]
Princeton Segmentation Benchmark

- 380 shapes in 19 categories
- Manual segmentations for each shape (4300 in total)
Rand Index Scores on PSB

When shape variation of the input is big

Top: Joint

Bottom: JointAll

<table>
<thead>
<tr>
<th></th>
<th>SD</th>
<th>RC</th>
<th>Supervised</th>
<th>Joint</th>
<th>JointAll</th>
<th>Human</th>
</tr>
</thead>
<tbody>
<tr>
<td>Armadillo</td>
<td>8.9</td>
<td>9.2</td>
<td>8.4</td>
<td>7.4</td>
<td>7.4</td>
<td>8.3</td>
</tr>
</tbody>
</table>
Rand Index Scores on PSB

When shape variation of the input is small

<table>
<thead>
<tr>
<th></th>
<th>SD</th>
<th>RC</th>
<th>Supervised</th>
<th>Joint</th>
<th>JointAll</th>
<th>Human</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airplane</td>
<td>9.3</td>
<td>13.4</td>
<td>8.2</td>
<td>12.9</td>
<td>10.2</td>
<td>9.2</td>
</tr>
</tbody>
</table>
Joint versus JointAll

- Joint = per-class segmentation
- JointAll = segment all classes together
Summary

• Joint-shape segmentation
  • Takes a set of unorganized shapes as input
  • Enforce structural similarity and segment invariance between segmentations of pairs of similar shapes
  • Significantly better than single-shape segmentations
  • Competitive with supervised methods

• Formulated as 0-1 linear program

• Optimization Strategy
  • Linear programming relaxation
  • Alternating optimization
JOINT MAP PROCESSING
Cycle Consistency

- Maps are consistent along cycles
Cycle Consistency

- Maps are consistent along cycles

Composite maps along cycles are identity maps
Cycle Consistency

Inconsistent
Cycle Consistency

Blended intrinsic maps [Kim et al. 11]

Inconsistent

Composition
Cycle Consistency Can Help

Direct

Consistent

Composition
Cycle consistency in computer vision

Disambiguating visual relations using loop constraints, Zach et al., CVPR’10
Cycle consistency for joint mapping

- Idea: given a collection of pairwise maps, compute a set of alternative maps, compositions of those given, which are consistent, and individually at times much better than the original
  - Dependent on the quality of input maps
  - Computationally expensive

An Optimization Approach to Improving Collections of Shape Maps, Nguyen et al., SGP’11
Spectral techniques

- Compute fuzzy correspondence based on diffusion distance in graph represented by initial correspondences.
CONSISTENT SHAPE MAPS VIA SEMIDEFINITE PROGRAMMING
Cycle-Consistency $\equiv$ Low-Rank

• In a map network, commutativity, path-invariance, or cycle-consistency are equivalent to a low rank or semidefiniteness condition on a big mapping matrix

$$X = \begin{pmatrix}
I_m & X_{1,2} & \cdots & X_{1,n} \\
X_{1,2} & I_m & \cdots & \cdots \\
\vdots & \vdots & I_m & X_{(n-1),n} \\
X_{n,1} & \vdots & X_{n,(n-1)} & I_m 
\end{pmatrix}.$$

• Conversely, such a low-rank condition can be used to
  • regularize and clean up primal or functional maps
  • extract shared structure

Consistent Shape Maps via Semidefinite Programming, Huang and Guibas, GSP’13
Basic Primal Setting

$n$ objects, each object has $m$ points (keypoints)
Matrix Representation of Primal Maps

$X_{12} = \begin{bmatrix}
0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
\end{bmatrix}$

Diagonal blocks are identity matrices

Off diagonal blocks are permutation matrices

Symmetric

$X = \begin{bmatrix}
I_m & X_{12} & \cdots & \cdots & X_{1n} \\
X_{12}^T & I_m & \cdots & \cdots & \vdots \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
X_{1n}^T & X_{(n-1),n}^T & \cdots & I_m \\
\end{bmatrix}$
Cycle Consistency Constraint

\[ X_{ij} = X^T_{j1} X_{i1} \quad \iff \quad X = \begin{bmatrix} I_m & \cdots & X_{n1} \\ \vdots & \ddots & \vdots \\ X^T_{n1} & \cdots & I_m \end{bmatrix} \]

(Positive) semidefiniteness

Low rank
Map Representation

\[ X = \begin{pmatrix}
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 
\end{pmatrix} \]

\[ X_{ii} = I_m, \quad 1 \leq i \leq n \]
\[ X_{ij}^T 1 = 1, X_{ij}^T 1 = 1, \quad 1 \leq i < j \leq n \]
\[ X \succeq 0 \]
\[ X \in \{0, 1\}^{nm} \]
Relaxation / Convexification

\[ X = \begin{bmatrix} & & \\ & \mathbf{1} & \\ & \mathbf{1} & \end{bmatrix} \]

\[ X_{ii} = I_m, \quad 1 \leq i \leq n \]

\[ X_{ij} = 1, X_{ij}^T = 1, \quad 1 \leq i < j \leq n \]

\[ X \succeq 0 \]

\[ 1 \geq X \geq 0 \]
Convex Program: SDP Formulation

minimize \[ \sum_{(i,j) \in \mathcal{E}} \|X_{ij}^{input} - X_{ij}\|_1 \]

subject to
\[ X_{ii} = I_m, \quad 1 \leq i \leq n \]
\[ X_{ij} 1 = 1, X_{ij}^T 1 = 1, \quad 1 \leq i < j \leq n \]
\[ X \succeq 0 \]
\[ 1 \succeq X \succeq 0 \]

ADMM [Boyd et al.11]
Map Correction
Exact Recovery Conditions

# fraction incorrect corres. per point

\[ \text{\textless} \text{algebraic-connectivity}(G)/4n \]

\[ \lambda_2(L_G) \]

Algorithm error tolerance
For Complete Graph

- 25% incorrect correspondences
- Worst-case scenario
  - Two clusters of objects of equal size
  - Wrong correspondences between objects of different clusters only (50%)

- Can recover even with 50% wrong correspondences in random case
CMU Hotel Sequence

Input:
102 images (30 points per image)
RANSAC [Fisher 81]
Each image connects with 10 random images
SDP Running time: 6m19s (3.2GHZ, single core)

<table>
<thead>
<tr>
<th></th>
<th>SDP</th>
<th>RPCA</th>
<th>Leordeanu et al. 12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>64.1%</td>
<td>100%</td>
<td>90.1%</td>
</tr>
</tbody>
</table>

13
Chair Data Set

16 images
Clustered SIFT features + RANSAC (60-120 points per image)

SDP Running time: 2m19s (3.2GHZ, single core)
JOINT MATCHING USING FUNCTIONAL MAPS
Pairwise maps: problems and issues

Symmetry, ambiguity, scale, bad data
Non-Convex, Combinatorial Optimization

multiple minima

$n!$ permutations

Symmetry, ambiguity, scale, bad data
Find alternative representation more amenable to optimization.
Function Spaces and Functional Maps
Functional Maps

• Map functions rather than points
Functional Maps

• Map functions rather than points

\[ T_\phi : L^2(N) \rightarrow L^2(M) \]
Functional Maps

• How to represent functions on surfaces?
Functional Maps

• How to represent functions on surfaces?
Functional Maps

- How to represent functions on surfaces?

\[ f(x) = a_1 \cdot + a_2 \cdot + a_3 \cdot + \cdots \]

Laplace-Beltrami
Functional Maps

• How to represent functional maps?

\( \phi : \)
Application of Basis

\[ T_\phi[f](x) = T_\phi[a_1 \cdot \text{image} + a_2 \cdot \text{image} + a_3 \cdot \text{image} + \cdots ] \]

\[ = a_1 T_\phi[\text{image}] + a_2 T_\phi[\text{image}] + a_3 T_\phi[\text{image}] + \cdots \]

Enough to know these
Functional Map Matrix

Functional Map Matrix = change of basis
Example Maps

(c) left to right map
Functional Map Computation

• Simple Algorithm
  • Compute some geometric functions to be preserved: A, B
  • Solve in least-squares sense for C, \( B = C A \)

• Additional Considerations
  • Favor commutativity
  • Favor orthonormality (if shapes are isometric)
  • Efficiently getting point-to-point correspondences
Application: Segmentation Transfer
Consistent Shape Segmentation Via Low Rank Recovery

Functional Map Networks for Analyzing and Exploring Large Shape Collections,
Huang et al., TOG, 2014
First Build a Network

Use the D2 shape descriptor and connect each shape to its nearest neighbors.

\[ G = (F, E) \]
Start From Noisy Shape Descriptor Correspondences

Lift to functional form

\[ C_i X_{ij} \approx D_j \]

\[ C_i \cdot \cdot \cdot D_i \]
Cycle Consistency Under Partial Similarity
The Pipeline

Original shapes with noisy maps → Cleaned up maps → Consistent basis functions extracted
Joint Map Optimization

◆ Step 1: Convex low-rank recovery using robust PCA – we minimize over all $X$

$$X^* = \lambda \|X\|_\star + \min_X \sum_{(i,j) \in G} \|X_{ij}C_{ij} - D_{ij}\|_{2,1}$$

trace norm $\|X\|_\star = \sum_i \sigma_i(X)$

convex!

Dual ADMM

◆ Step 2: Perturb the above $X$ to force the factorization

$$\sum_{1 \leq i,j \leq N} \|X_{ij}^* - Y_j^+Y_i\|_F^2 + \mu \sum_{i=1}^{N} \sum_{1 \leq k < l \leq L} (y_{ik}^T y_{il})^2$$

Non-linear least squares

Gauss-Newton descent

The $Y_i$ give us the desired latent spaces
Low-Rank Matrix Factorization

\[ X := \begin{pmatrix} X_{11} & \cdots & X_{N1} \\ \vdots & \ddots & \vdots \\ X_{1N} & \cdots & X_{NN} \end{pmatrix} = \begin{pmatrix} Y_1^+ \\ \vdots \\ Y_N^+ \end{pmatrix} \begin{pmatrix} Y_1 & \cdots & Y_N \end{pmatrix} \]

- **Robust computation** --- recover a low-rank matrix from noisy measurements of its entries (initial functional correspondences)
- **Structure recovery** --- \( Y \) matrices encode shared structures across the shape collection
- **Efficient encoding** --- We just need to store the \( Y \) matrices
Consistent Shape Segmentation

Via 2\textsuperscript{nd} order MRF on each shape independently
Large-Scale Data

8K shapes
Hierarchical Scaling

Multiple abstraction levels

Route maps via the abstractions
Application: Entity Extraction in Images (aka Co-Segmentation)

• Task: jointly segment a set of related images
  • same object, different viewpoints/scales:

    ![Car Images]

  • similar objects of the same class:

    ![Cow Images]

• Benefits and challenges:
  • Images can provide weak supervision for each other
  • But exactly how should they help each other? How to deal with clutter and irrelevant content?
MSRC: 5 images per class are shown
MSRC: 5 images per class are shown
Cheetah + Safari (red: cheetah; yellow: lion; magenta: monkey.)

Cow + pasture (red: black cow; green: brown cow; blue: man in blue.)

Dog + park (red: black dog; green: brown dog; blue: white dog.)

Dolphin + aquarium (red: killer whale; green: dolphin.)
THAT’S IT FOR TODAY