Shape Co-analysis

Daniel Cohen-Or

Tel-Aviv University
High-level Shape analysis

Upright orientation

Illustrating assemblies

Shape abstraction

Symmetry hierarchy

Learning segmentation

Exploration of shape collections
Segmentation and Correspondence

Segmentation

Correspondence
Individual vs. Co-segmentation
Individual vs. Co-segmentation
Challenge

Similar geometries can be associated with different semantics
Challenge

Similar semantics can be represented by different geometries
Large set are more challenging

Methods do not give perfect results
Related works

• Supervised segmentation

[Kalogerakis et al.10, van Kaick et al. 11]
Related Work

• Geometry-based unsupervised co-segmentation

[Golovinskiy and Funkhouser 09]

[Xu et al.10]
Descriptor-based unsupervised co-segmentation

[Sidi et al. 11]
Constraints as Features

• Semi-Supervised (constrained) Clustering
• How to incorporate user-given constraints in order to improve the results of a clustering algorithm?
• Embed constraints into the data as “extra features”
Clustering (basic stuff)

• Clustering:
  – Takes a set of points:
Clustering

- Clustering:
  - Takes a set of points,
  - And groups them into several separate clusters:
Clustering

• Difficulties in Clustering:
  – Clean separation to groups not always possible
  – Must make “hard splitting” decisions
  – Number of groups not always known, or can be very difficult to determine from data
Clustering

• Example of difficulties:
  – Given a set of points
Clustering

• Example of difficulties:
  – Hard to determine number of clusters
Clustering

• Example of difficulties:
  – Hard to determine number of clusters
Clustering

• Example of difficulties:
  – Hard to decide where to split clusters
Clustering

• Example of difficulties:
  – Hard to decide where to split clusters
Clustering

• Two general types of input for Clustering:
  – Spatial Coordinates (points, feature space), or
  – Inter-object Distance matrix
Clustering

- Two general types of input for Clustering:
  - Spatial Coordinates (points, feature space), or
  - Inter-object Distance matrix
Clustering

• Easy to extract distance matrices from feature spaces:
  – Just calculate distances between points
Clustering

• Much harder to extract a feature space from a distance matrix:
  – Requires embedding method (like MDS)
Clustering

• Popular clustering algorithms:
  – K-Means, EM, Mean-Shift
Clustering

• Popular clustering algorithms:
  – Linkage, DBSCAN, Spectral Clustering
Clustering

• Underlying assumptions behind all clustering algorithms:
  – Points represent some kind of objects.
Clustering

- Underlying assumptions behind all clustering algorithms:
  - Neighboring points imply similar objects.
Clustering

• Underlying assumptions behind all clustering algorithms:
  – Distant points imply dissimilar objects.
Clustering

- When these assumptions hold, clustering algorithms are expected to work well:
  - Clustering makes sense, objects grouped properly
Clustering

- When assumptions fail, result is not useful:
  - Similar objects are distant in feature space
Clustering

- When assumptions fail, result is not useful:
  - Dissimilar objects are close in feature space
Clustering

• Assumptions might fail because:
  – Data is difficult to analyze
  – Similarity/Dissimilarity of data not well defined
  – Feature space is insufficient or distorted
Supervised Clustering

• Supervised clustering used to improve clustering results.
  – Takes training set of labeled data (pre-clustered)
Supervised Clustering

- Supervised clustering used to improve clustering results.
  - Infer from learned information to a test set of unlabelled data.
  - (Hopefully, overcoming possible failures of unsupervised clustering)
Supervised Clustering

• Some supervised clustering methods:
  – SVM (Support Vector Machine) and variants
  – Distance Metric Learning (many algorithms)
Semi-Supervised Clustering

• Supervision as pair-wise constraints:
  – Must Link and Cannot-Link
Semi-Supervised Clustering

• Cluster data while respecting constraints
Semi-Supervised Clustering

• Many Semi-Supervised methods:
  – Modifications to clustering methods
  – Distance Metric Learning
  – Data/Affinity Modification
Learning from labeled and unlabeled data
Supervised learning
Unsupervised learning
Semi-supervised learning
Constrained clustering
Active Co-Analysis of a Set of Shapes

(Yunhai Wang, Oliver van Kaick, Baoquan Chen, Hao Zhang, Shmuel Asafi, Danny Cohen-Or)
Active Co-analysis of a Set of Shapes
SIGGRAPH ASIA 2012
Active Co-Analysis

• A semi-supervised method for co-segmentation with \textit{minimal} user input
Automatically suggest the user which constraints can be effective
Initial co-segmentation

- Over-segmentation mapped to a descriptor space (geodesic distance, shape diameter function, normal histogram)
Initial co-segmentation

- Diffusion map clustering [Sidi et al. 2011]
Constrained clustering

Initial Co-segmentation

Constrained Clustering

Final result

Active Learning
Constrained Clustering
Constrained Clustering

Spring embedding

K-mean clustering
Constrained Clustering

• Employ a **Spring System** to modify data according to Must-Link & Cannot-Link constraints.
Spring System

- A spring system is used to re-embed all the points in the feature space, so the result of clustering will satisfy constraints.
Spring System

• Result of clustering after re-embedding (mistakes marked with circle):
The Spring System models a set of springs attached to the nodes in the feature space.

When a spring is stretched or squeezed it exerts force on the nodes it’s attached to.

When it’s relaxed – it exerts no force.

Hooke’s Law
Constrained clustering & Co-segmentation
Defining Constraints
Active Learning

Initial Co-segmentation

Constrained Clustering

Final result

Active Learning
Uncertain points

- “Uncertain” points are located using the Silhouette Index:

Darker points have lower confidence
Silhouette Index

- Silhouette Index of node $x$:

$$S(x) = \frac{b(x) - a(x)}{\max[b(x), a(x)]}$$
Constraint Suggestion

- Pick super-faces with lowest confidence
- Pick the highest confidence super-faces
- Ask the user to add constraints between such pairs
Results

- Eleven sets of shapes

<table>
<thead>
<tr>
<th>Set</th>
<th>#shapes</th>
<th>#super-fcs.</th>
<th>#constr.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Candelabra</td>
<td>28</td>
<td>164</td>
<td>24</td>
</tr>
<tr>
<td>Chairs</td>
<td>20</td>
<td>236</td>
<td>36</td>
</tr>
<tr>
<td>Four-legged</td>
<td>20</td>
<td>264</td>
<td>69</td>
</tr>
<tr>
<td>Goblets</td>
<td>12</td>
<td>49</td>
<td>4</td>
</tr>
<tr>
<td>Guitars</td>
<td>44</td>
<td>330</td>
<td>6</td>
</tr>
<tr>
<td>Lamps</td>
<td>20</td>
<td>97</td>
<td>2</td>
</tr>
<tr>
<td>Vases</td>
<td>28</td>
<td>169</td>
<td>34</td>
</tr>
<tr>
<td>Irons</td>
<td>18</td>
<td>138</td>
<td>26</td>
</tr>
<tr>
<td>Large chairs</td>
<td>400</td>
<td>2,832</td>
<td>162</td>
</tr>
<tr>
<td>Large tele-aliens</td>
<td>200</td>
<td>1,869</td>
<td>106</td>
</tr>
<tr>
<td>Large vases</td>
<td>300</td>
<td>1,527</td>
<td>44</td>
</tr>
</tbody>
</table>
Candelabra: 28 shapes, 164 super-faces, 24 constraints
Fourleg: 20 shapes, 264 super-faces, 69 constraints
Tele-alien: 200 shapes, 1869 super-faces, 106 constraints
Vase: 300 shapes, 1527 super-faces, 44 constraints

Initial
Second Work

Constraints as Features
Cannot-Link Springs
Constraints as Features

- **Goal:** Modify data so distances fit constraints
- **Basic idea:**
  - Convert constraints into extra-features that are added to the data (augmentation)
  - Recalculate the distances
  - Unconstrained clustering of the modified data
  - Clustering result more likely to satisfy constraints
- Apply this idea to Cannot-Link constraints
- Must-Link constraints handled differently
Modify distance to small value, and restore triangle-inequality by updating all other distances to make all distances consistent.
Cannot-link Constraints

• Points should be distant.
• What value should be given: \( D(c_1, c_2) = X \)?
  – Should relate to \( \max(D(x, y)) \), but how?
• If modified, how to restore triangle-inequality?
Constraints as Features

- **Solution:**
  - Add extra-dimension, where Cannot-Link pair is far away ($\pm 1$):
Constraints as Features

• **Solution:**
  – Add extra-dimension, where Cannot-Link pair is far away (±1):
  – **What values should other points be given?**
Constraints as Features

• Values of other points:
  – Points closer to $c_1$ should have values closer to $+1$,
  – Points closer to $c_2$ should have values closer to $-1$

• Formulation:
  \[
  v_i = \frac{(\varphi(i, c2) - \varphi(i, c1))}{(\varphi(i, c2) + \varphi(i, c1))}
  \]

• Simple distance $\varphi(i, c1)$ does not convey real closeness.
Constraints as Features

- Point A should be “closer” to $c_1$, despite smaller Euclidean distance.
Constraints as Features

- Use a Diffusion Map, where this holds true.
Constraints as Features

• Diffusion Maps related to random walk process on a graph

• Affinity Matrix:
  \[ A_{i,j} = e^{-\frac{D_{i,j}^2}{\sigma^2}} \]

• Eigen-Analysis of normalized A forms a Diffusion Map:
  \[ \Psi_t(x) = (\lambda_1^t \psi_1(x), \lambda_2^t \psi_2(x), \ldots, \lambda_K^t \psi_K(x)) \]
Constraints as Features

• Use Diffusion Map distances:

\[ \varphi(x, y) = |\Psi_t(x) - \Psi_t(y)| \]

• Calculate value of each point in new dimension:

\[ v_i = \frac{(\varphi(i, c2) - \varphi(i, c1))}{(\varphi(i, c2) + \varphi(i, c1))} \]
Constraints as Features

• Create new distance matrix, of distances in the new extra dimension:

\[ D_{i,j}^{(c)} = |v_i - v_j| \]

• Add distance matrix per Cannot-Link:

\[ \tilde{D}_{i,j}^p = \hat{D}_{i,j}^p + \sum_{c \in [1,N]} (\alpha \cdot D_{i,j}^{(c)})^p \]

• Cluster data by modified distance matrix \( \tilde{D} \)
Constraints as Features

Original

Springs

Features
Constraints as Features!!!

Unconstrained clustering of the modified data
Results – UCI (CVPR 2013)

Ionosphere

Wine

Iris

Hepatitis

- Affinity
- CSP
- Karnvar 2003
- Features
- ITML
Results – Image Segmentation

• We tested our method on ALL the 117 images from the Berkeley Segmentation Dataset (BSD) that have 4 or less separate segments.
• Each image over-segmented to 300 super-pixels.
• Feature space is naïve and simple:
  – 5 dimensions
  – \( (X, Y, R, G, B) \)
  – Average super-pixel color
Results – Image Segmentation

- Original With Constraints
- Super-Pixels
- Groundtruth Segmentation
- Feature Space with Constraints

Feature Constraints
Kamvar 2003
Affinity Propagation
CSP
Results – Image Segmentation
Results – Image Segmentation

- Affinity
- Kamvar 2003
- Features
- ITML
Summary

• A new semi-supervised clustering method.
• Constraints are embedded into the data, reducing the problem to an unconstrained setting.
• Constraints are not guaranteed to be satisfied.
• Optimal weighting of constraints relative to original data remains an open problem.
Limitations

• Convergence

• The active learning is limited to silhouette indices

• Prior errors introduced by the selected descriptors
Future Work

• Constrained clustering
• Design of new types of constraints
• Constraints suggestion
Thank you!