Shape retrieval
Intrinsic shape matching

Anastasia Dubrovina
Computer Science Dept.
Stanford University

Slides credits: L. Guibas, V. Kim, Y. Li, H. Su, A. Bronstein, M. Bronstein, R. Litman, V. Kalogerakis, T. Liu
Problem definition

• Goal: measure shape similarity

• Similarity: geometric, extrinsic vs. intrinsic, style similarity, etc.
• Tasks: classification, retrieval, etc.
Applications: shape retrieval from large shape collection

Image credit: M. Bronstein
Applications: fine-grained similarity for interactive shape modeling

Modeling by example
[Funkhouser et al., 2004]
Applications: suggesting objects to match scene style

[Lun et al. 2015]
Lecture outline

• Shape similarity and retrieval
  • Extrinsic shape similarity
  • Intrinsic shape similarity
  • Fine-grained similarity
  • Style similarity

• Deformable shape matching
  • If time permits
SHAPE SIMILARITY AND RETRIEVAL
Earlier work

- Descriptor-based similarity

  Shape Distributions
  Light field descriptors

- Requirements
  - Representative
  - Invariant (rigid transformations, small geometry changes, etc.)
  - Compact - for fast comparison
Datasets

• Examples

Princeton Shape Benchmark
1814 models, 90 classes

SHREK’11 Shape Retrieval Contest
1237 models

SHREK’14 Large Scale Retrieval Contest
8987 models, 171 classes

SHREC’14 - Non-Rigid 3D Human Models track
400 real and 300 synthetic models
SHREC - 3D Shape Retrieval Contests

- @Eurographics Workshop on 3D Object Retrieval
- E.g., this year (2017)

Tracks
The following tracks are organized. For description of tasks, the collections, queries, the evaluation procedure, and time schedule, follow the links:

1. RGB-D Object-to-CAD Retrieval
   Organizers: Binh-Son Hua, Quang-Hieu Pham, Minh-Khoi Tran, Quang-Trung Truong (Singapore University of Technology and Design)
   Contact: Binh-Son Hua, binhson.hua at gmail.com

2. 3D Hand Gesture Recognition Using a Depth and Skeletal Dataset
   Organizers: Quentin De Smedt, Hazem Wannous, Jean-Phillipe Vandeborre
   Contact: Quentin de Smedt, quentin.desmedt@telecom-lille.fr

3. Large-scale 3D Shape Retrieval from ShapeNet Core55
   Organizers: Manolis Savva, Hao Su (Stanford University), Fisher Yu, Tom Funkhouser (Princeton University)
   Contact: Manolis Savva, manolis.savva at gmail.com
   Web page: [https://shapenet.cs.stanford.edu/shrec17/](https://shapenet.cs.stanford.edu/shrec17/)

4. Classification of protein shapes
   Organizers: Haiguang Liu (Beijing Computational Science Research Center)
   Contact: Haiguang Liu, hgliu at csrc.ac.cn

5. Point-Cloud Shape Retrieval of Non-Rigid Toys
   Organizers: Frederico A. Limberger, Richard C. Wilson (University of York)
   Contact: Frederico Limberger, fal504 at york.ac.uk
   Web page: [https://www.cs.york.ac.uk/cvpr/pronto/](https://www.cs.york.ac.uk/cvpr/pronto/)
Large-scale retrieval contest using ShapeNet

LARGE-SCALE 3D SHAPE RETRIEVAL FROM SHAPENET CORE55

3D content is becoming increasingly prevalent and important to everyday life. With commodity depth sensors, everyone can easily scan 3D models from the real world. Better 3D modeling tools are allowing designers to produce 3D models more easily. And with the advent of virtual reality, the demand for high quality 3D models will only increase. The increasing availability of 3D models requires scalable and efficient algorithms to manage and analyze them. A key research problem is retrieval of relevant 3D models and the community has been actively working on this task for more than a decade. However, existing algorithms are usually evaluated on datasets with only thousands of models, even though millions of 3D models are now available on the Internet. Thanks to the efforts of the ShapeNet [1] team, we can now use a much bigger dataset of 3D models to develop and evaluate new algorithms. In this track, we aim to evaluate the performance of 3D shape retrieval methods on a subset of the ShapeNet dataset.
Image vs. 3D datasets

- **State-of-the-art 3D shape dataset**

- Limited in:
  - scale
  - object classes
  - diversity

- Datasets:
  - Caltech 101
  - Caltech 256
  - LabelMe
  - CIFAR
  - ImageNet

Slide credit: H. Su
ShapeNet

~3 million models  ~2,000 classes  Rich annotations

Work in progress

Slide credit: H. Su
Large-scale retrieval contest using ShapeNet

- In 2016, all methods used deep learning
- Best-performing method

![Diagram of multi-view CNN architecture](image)

[Su et al. 2015]

- (to be covered in the second part of the course)
- All methods perform extrinsic shape retrieval
Today - intrinsic shape similarity

- Different from extrinsic, or rigid, similarity

Image credit: A. Bronstein
Today - intrinsic shape similarity

- Different from extrinsic, or rigid, similarity

- Approaches we will discuss today
  - Shape Google [Bronstein et al. 2011]
  - Supervised Bag-of-features [Litman et al. 2014]
<table>
<thead>
<tr>
<th>Name</th>
<th>Institution</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alex Bronstein</td>
<td>Tel-Aviv University</td>
<td>Israel</td>
</tr>
<tr>
<td>Michael Bronstein</td>
<td>University of Lugano</td>
<td>Switzerland</td>
</tr>
<tr>
<td>Maks Ovsjanikov</td>
<td>Stanford University</td>
<td>USA</td>
</tr>
<tr>
<td>Leonidas Guibas</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Notre Dame de Paris is a **Gothic cathedral** in the fourth quarter of **Paris, France**. It was the first **Gothic architecture cathedral**, and its **construction** spanned the **Gothic period**.

St. Peter’s **basilica** is the largest **church** in world, located in **Rome, Italy**. As a work of **architecture**, it is regarded as the best building of its age in **Italy**.
Bags of visual features

Think of an image as a collection of primitive elements

Visual vocabulary

Zisserman et al.
Slide credit: M. Bronstein
## Local shape descriptors

<table>
<thead>
<tr>
<th>Representation</th>
<th>Rigid</th>
<th>Scale</th>
<th>Bending</th>
<th>Topology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Curvature</td>
<td>Mesh</td>
<td>✔</td>
<td>❌</td>
<td>✔</td>
</tr>
<tr>
<td>Spin image(^2)</td>
<td>Mesh</td>
<td>✔</td>
<td>❌</td>
<td>❌</td>
</tr>
<tr>
<td>Shape context(^3)</td>
<td>Mesh</td>
<td>✔</td>
<td>❌</td>
<td>✔</td>
</tr>
<tr>
<td>HKS(^4)</td>
<td>Mesh</td>
<td>✔</td>
<td>❌</td>
<td>✔</td>
</tr>
<tr>
<td>SI-HKS(^5)</td>
<td>Mesh</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Color HKS(^6)</td>
<td>Mesh</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>vHKS(^7)</td>
<td>Volume/Mesh</td>
<td>✔</td>
<td>❌</td>
<td>✔</td>
</tr>
<tr>
<td>WKS(^1)</td>
<td>Mesh</td>
<td>✔</td>
<td>❌</td>
<td>✔</td>
</tr>
</tbody>
</table>


Slide credit: M. Bronstein
Heat kernel signature

Heat diffusion on a manifold

Diagonal of heat kernel

$h_t(x, x)$

$h_{at}(x, x)$

$h_{a^2t}(x, x)$

Distance of heat kernel

$p(x) = (h_t(x, x), \ldots, h_{at}(x, x))$

Multi-scale point descriptor

Slide credit: M. Bronstein
Heat kernel signature

Heat kernel signatures represented in RGB space

Slide credit: M. Bronstein
Heat kernel signature

Invariant to isometric deformations

Localized sensitivity to topological noise

Not scale invariant

(scale-invariant HKS in [B&Kokkinos 2010])
Think of a **shape** as a collection of primitive elements.

Let \( p(x) \) be the shape.

**Geometric vocabulary**

\[
k^* = \arg\min_{i=1,...,V} \| p(x) - p_i \|
\]

\( p_1, \ldots, p_V \)

**Bag of geometric words**
Shape descriptors can be computed at stable points, or at all points.

Geometric vocabulary

\[ k^* = \arg \min_{i=1,\ldots,V} \| p(x) - p_i \| \]
BoGW - computation details

- A vocabulary \( \mathcal{P} = \{p_1, \ldots, p_V\} \) of size \( V \) is a set of representative vectors in the descriptor space.
- It is obtained using vector quantization through k-means in the HKS descriptor space.

![Vocabulary visualization](image)

k-means

Vocabulary \( P \)

Slide credit: M. Bronstein
BoGW - computation details

- A vocabulary \( \mathcal{P} = \{ p_1, \ldots, p_V \} \) of size \( V \) is a set of representative vectors in the descriptor space.
- It is obtained using vector quantization through k-means in the HKS descriptor space.
- Given a point \( x \) with a descriptor \( p(x) \), compute
  \[
  \theta_i(x) = c(x) e^{-\frac{||p(x) - p_i||^2}{2\sigma^2}}
  \]
  “probability of the point \( x \) to be associated with the descriptor \( p_i \)”
- Integrate over the whole shape \( X \)
  \[
  f(X) = \int_X \theta(x) d\mu(x)
  \]
  weighting by area element
Bags of features

Disadvantage of the bag of features approaches: they lose information about the spatial location of features in the image.
In math science, \textit{matrix decomposition} is a factorization of a matrix into some \textit{canonical form}. Each type of decomposition is used in a particular problem.

In biological science, decomposition is the process of organisms to break down into simpler form of matter. Usually, decomposition occurs after death.

Matrix is a \textit{science fiction} movie released in 1999. \textit{Matrix} refers to a simulated reality created by machines in order to subdue the human population.
In math science, **matrix decomposition** is a factorization of a matrix into some **canonical form**. Each type of decomposition is used in a particular problem.
Bags of geometric expressions

Geometric vocabulary

Bag of geometric expressions (BoF)
Spatially Sensitive Bags of Features (SS-BoF)

\[ F(X) = \int_{X \times X} \theta(x)\theta^T(y)K_1(x, y)d\mu(x)d\mu(y) \]

- Proximity measured by heat kernel
- Measures “frequency of appearance of nearby geometric words”

Bag of geometric expressions (SS-BoF)

Slide credit: M. Bronstein
Bags of geometric expressions
Feature descriptor

Geometric words

Bag of geometric words

Shapes as binary codes: similarity-sensitive hashing

Spatially-sensitive bag of words

Slide credit: M. Bronstein
Metric learning

“Similar”

“Dissimilar”

BoF space

Hamming space

Shape hash: just 64 bits!

Shakhnarovich et al.; Grauman et al.; BB, Kimmel 2009; Ovsjanikov, BB, Guibas 2009; Strecha, BB, Fua 2010
SHREC 2010: Robust shape retrieval benchmark

Transformation

Query set

Database (>1K shapes)

TOSCA dataset + Princeton dataset (Funkhouser et al.)

Slide credit: M. Bronstein
Results

Shape Google (HKS)
Results

Shape Google (Scale-invariant HKS)
Results

Shape Google+Metric learning

Slide credit: M. Bronstein
Query

Toldo et al. 2009

Shape
Drawback of the standard BoF construction

- The dictionary is constructed in an unsupervised manner using clustering, unaware of the following learning stage.
Drawback of the standard BoF construction

- The dictionary is constructed in an unsupervised manner using clustering, unaware of the following learning stage.
- Suggested improvement: add **supervision** to the BoF training

“Supervised learning of bag-of-features shape descriptors using sparse coding” [Litman et al. 2014]
BoF computation flow - simplified

- Compute local descriptors - e.g., HKS
BoF computation flow - simplified

- Compute local descriptors - e.g., HKS
- Get a dictionary (= vocabulary) by vector quantization (VQ)
BoF computation flow - simplified

- Compute local descriptors - e.g., HKS
- Get a dictionary (= vocabulary) by vector quantization (VQ)
- Replace each descriptor by a binary indicator vector

Image credit: R. Litman, M. Bronstein
BoF computation flow - simplified

- Compute local descriptors - e.g., HKS
- Get a dictionary (= vocabulary) by vector quantization (VQ)
- Replace each descriptor by a binary indicator vector
- Sum up all indicator vector to obtain the BoF
Suggested improvements

Space coding instead of VQ

 Learned dictionary

Bi-level optimization

Optimal dictionary
Sparse coding

- Represent data $X$ as a **sparse linear combination** of atoms of dictionary $D$
- Dates back to [Olshausen and Field 1996]

---

Slide credit: R. Litman
Sparse coding

- Very successful when **dictionary D is learned from data**
- State-of-the-art in many applications.
Sparse coding for BoF - example

Query – $S$

Positive – $S_+$

Negative – $S_-$

Slide credit: R. Litman
Sparse coding for BoF - example

Query – $S$
Positive – $S_+$
Negative – $S_-$

See the paper for implementation details

Slide credit: R. Litman
Descriptor pooling example

Pooled descriptors example: $h(Z^*)$, $h(Z^*_+)$, $h(Z^*_-)$

VQ (unsupervised)

Sparse coding + Unsupervised DL

Slide credit: R. Litman
Dictionary learning

- Dictionary learning is task-driven - specified by a loss function
  - Input: labeled set of training shapes $S$
  - Each shape has an attached BoF $h(S)$
  - Optimize over dictionary $D$ to minimize loss of training set

$$\min_D \sum_{S \in S} \ell(h)$$

Slide credit: R. Litman
Dictionary learning using *triplet* loss

Make $\|h(Z) - h(Z_+)\|$ small and $\|h(Z) - h(Z_-)\|$ larger (in comparison) by minimizing

$$\ell = \alpha \ell_+ + (1 - \alpha) \ell_-$$

$$\ell_+(Z, Z_+) = \|h(Z) - h(Z_+)\|_1$$
$$\ell_-(Z, Z_+, Z_-) = \max\{0, \mu + \|h(Z) - h(Z_+)\|_1 - \|h(Z) - h(Z_-)\|_1\}$$

[Weinberger and Saul 2009]
BoF after dictionary learning

Pooled descriptors example: $h(Z^*)$, $h(Z^*_+)$, $h(Z^*_\text{-})$

Slide credit: R. Litman
BoF after dictionary learning

Distance ratios \( \frac{\|h(Z^*) - h(Z_\perp^*)\|_1}{\|h(Z^*) - h(Z^*)\|_1} \)

6.26  \( \uparrow \)  3.53  \( \uparrow \)  0.98  \( \uparrow \)

VQ (unsupervised)  Sparse coding + Unsupervised DL  Sparse coding + Supervised DL

Slide credit: R. Litman
SHREC’14 Dataset

- Goal: given a human model, detect this model in other poses
SHREC’14 results

- Goal: given a human model, detect this model in other poses
Query example - nearest neighbor

Query

ShapeGoogle (Bronstein et al. 2011)

Unsupervised DL

Slide credit: R. Litman
Questions so far?
Fine-grained shape classification

- Global shape descriptors - work well for shapes from different classes
- Next: a method for fine-grained sub-class classification from a sparse and noisy set of labeled shapes
Fine-grained shape classification

- Global shape descriptors - work well for shapes from different classes
- Next: a method for fine-grained sub-class classification from a sparse and noisy set of labeled shapes

“Fine grained semi supervised labeling of large shape collections” [Huang et al. 2013]
Problem definition

- Large shape collection
  E.g., 5850 chairs, 26 classes

- Sparse and noisy labels for each class
  
  \[
  \text{label} = \text{shape class}
  \]

- Subtle geometric differences

- Goal: produce labels for all shapes in collection

Image credit: H. Su
Approach overview

Input Shapes
1: With-arms
2: Side
3: Windsor
4: Rex

Shape Matching
Global phase
Local phase

Distance Learning
with-arms
side
windsor
rex
Distance field
Spin images

Graph-Based Classification
With-arms
Side
Windsor
Rex

Image credit: H. Su
Shape matching

- **Global phase**: global affine shape alignment

\[ T_i : (x, y, z) \in S_i \rightarrow (x', y', z') \]

\[
\begin{pmatrix}
    x' \\
    y'
\end{pmatrix}
= \begin{pmatrix}
    s_i^x & 0 \\
    0 & s_i^y
\end{pmatrix}
R(\theta_i)
\begin{pmatrix}
    x \\
    y
\end{pmatrix}
+ \begin{pmatrix}
    t_i^x \\
    t_i^y
\end{pmatrix}, \quad z' = s_i^z z.
\]

- **Joint alignment** - via MRF optimization

- **Local phase**: local non-rigid registration using free-form deformation
Distance learning

- Learn distance metric using the aligned labeled shapes - per class

- Distance between pair of shapes = parameterized using fixed-size voxels

\[ \text{dist}(\cdot, \cdot) = \sum_{k \in \text{Voxels}} x^T \text{dist}(k) \]

- Learning formulated to
  - Minimize distances between shape pairs in the similar sets
  - Maximize distances between shapes from dissimilar sets
Learned metric - illustration

Windsor

Swivel

Rocking

Cantilever

Folding

Morris

Image credit: H. Su
Graph-based classification

- Per class: create similarity graph using k-NN of each shape

- Assign labels via graph partitioning using graph diffusion distances

Image credit: H. Su
Labeling results

Propeller planes

Image credit: H. Su
Comparison to linear classifier result

Propeller planes

Image credit: H. Su
Questions?
Style similarity

- Two papers presented in Siggraph 2015

Lun et al. 2015

Stylistically incompatible

Stylistically compatible

Liu et al. 2015
Style similarity

- Two papers presented in Siggraph 2015

Lun et al. 2015

No assumption about input shapes

Liu et al. 2015

Stylistically compatible

Stylistically incompatible

Furniture models
Style compatibility for furniture models

Stylistically incompatible

Stylistically compatible

Liu et al. 2015
Style compatibility for furniture models

- Crowdsource compatibility between pairs of models

---

Slide credit: T. Liu
Crowdsourcing compatibility preferences

Design of user study [Wilber et al. 2014]

Please select the two most compatible pairs
Crowdsourcing compatibility preferences

Rater’s selection
Converted into 8 triplets

and 4 more triplets ...
Crowdsourcing compatibility preferences

Living room

Dining room

Collected 63,800 triplets for living room and 20,200 for dining room

Slide credit: T. Liu
Part-aware geometric features

Step 1: Consistent segmentation [Kim et al. 2013] (next lecture)

Armrest  Back  Legs  Seat

Slide credit: T. Liu
Part-aware geometric features

Step 2: Computing geometry features for each part

Back

- Curvature histogram
- Shape diameter histogram
- Bounding box dimensions
- Normalized surface area

Slide credit: T. Liu
Part-aware geometric features

Step 3: Concatenating features of all parts

\[ x = [x_{\text{back}}, x_{\text{legs}}, \ldots] \]
Learning object-class specific embeddings
Style-aware shape retrieval

Query model

Dining chair

Most incompatible chairs

Slide credit: T. Liu
Learning perceptual style similarity

Lun et al. 2015
Learning perceptual style similarity

Which of the two shapes (B or C) is more similar **style-wise** to shape A?

Slide credit: Z. Lun
Learn measure parameters via crowdsourcing

Which of the two objects on the bottom (B or C) is more similar style-wise to the object on the top (A)?

(i) B
(ii) C
(iii) Both
(iv) Neither

Slide credit: Z. Lun
Geometric criteria for element similarity

- Style-related elements are frequently designed to be distinct
Algorithm for measuring style similarity

Input: a pair of shapes
Output: a measure of style dissimilarity (distance)

\[ D(\cdot, \cdot) = ? \]

Slide credit: Z. Lun
Extraction of matching elements

- **Multi-scale** segmentation
- Patches as *initial seeds* to detect elements
Extraction of matching elements

Align with **affine transformation**, measure patch

**stylistic similarity:**

\[
\text{distance}(\quad,\quad) = w_1 \times d_1(\quad,\quad) + w_2 \times d_2(\quad,\quad) + w_3 \times d_3(\quad,\quad) + \ldots
\]

- surface point-to-point distance
- distance between feature curves
- distance between curvature histograms

Slide credit: Z. Lun
Extraction of matching elements

Group patches into matching elements

Slide credit: Z. Lun
Algorithm for measuring style similarity

input shapes  matching elements

Slide credit: Z. Lun
Algorithm for measuring style similarity

input shapes | matching elements | distance components

- Computed for each element using geometric cues
- Percentage of the area on both models not covered by any matched elements, weighted by their saliency
- Same distance we used to match elements
Algorithm for measuring style similarity

<table>
<thead>
<tr>
<th>input shapes</th>
<th>matching elements</th>
<th>distance components</th>
<th>output distance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0.82 x 0.16 + 0.65 x 0.45 + 0.08 x 0.37</td>
<td>D(A,B) = 0.29</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.01 x 0.30 + 0.27</td>
<td>D(C,D) = 0.59</td>
</tr>
</tbody>
</table>

Slide credit: Z. Lun
Parameter learning

Learn parameters from training triplets:
• element-similarity weights ($w$)
• saliency weights ($v$)
• prevalence penalty ($t$)

that maximize likelihood function & regularizer to promote sparsity:

$$L(w, v, t) = \sum_{\text{triplet } \{A,B,C\}} \text{confidence}(B) \cdot \log P(B \text{ is more similar to } A \text{ than } C)$$

$$+ \sum_{\text{triplet } \{A,B,C\}} \text{confidence}(C) \cdot \log P(C \text{ is more similar to } A \text{ than } B)$$

$$+ \text{regularizer}(w, v, t)$$

Slide credit: Z. Lun
Validation

Does it work?

Slide credit: Z. Lun
Our result

(i) B - 90%
(ii) C - 0%
(iii) Both - 0%
(iv) Neither - 10%
Failure case

(i) B - 0%
(ii) C - 70%
(iii) Both - 0%
(iv) Neither - 30%
Application:
Style-based shape tagging
Asian
Questions?
Shape similarity and retrieval - another flavor

Joint shape and image embedding

[Li et al. 2015]
Shape based Image Retrieval

<table>
<thead>
<tr>
<th>Query</th>
<th>Top 5 Neighbors</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Chair" /></td>
<td><img src="image2" alt="Chair" /></td>
</tr>
<tr>
<td><img src="image3" alt="Chair" /></td>
<td><img src="image4" alt="Chair" /></td>
</tr>
<tr>
<td><img src="image5" alt="Chair" /></td>
<td><img src="image6" alt="Chair" /></td>
</tr>
<tr>
<td><img src="image7" alt="Airplane" /></td>
<td><img src="image8" alt="Airplane" /></td>
</tr>
<tr>
<td><img src="image9" alt="Airplane" /></td>
<td><img src="image10" alt="Airplane" /></td>
</tr>
<tr>
<td><img src="image11" alt="Car" /></td>
<td><img src="image12" alt="Car" /></td>
</tr>
<tr>
<td><img src="image13" alt="Car" /></td>
<td><img src="image14" alt="Car" /></td>
</tr>
<tr>
<td><img src="image15" alt="Car" /></td>
<td><img src="image16" alt="Car" /></td>
</tr>
</tbody>
</table>

[Li et al. 2015]
Image based Shape Retrieval

Multi-modal embeddings - in the second part of the course

[Li et al. 2015]
DEFORMABLE SHAPE MATCHING
Mapping Between Data Sets

- Multiscale mappings
  - Point/pixel level
  - Part level

Maps capture what is the same or similar across two data sets

Slide credit: L. Guibas
Why Do We Care About Maps and Alignments?

- To stitch data together
- To transfer information
- To compute distances and similarities
- To perform joint analysis
Problem definition

• Given a pair of shapes, find corresponding points

Image credit: L. Guibas, M. Ovsjanikov
Problem definition

- Given a pair of shapes, find corresponding points
- When shapes differ by rigid transformation - 6 degrees of freedom
- Use rigid alignment algorithm
Problem definition

- Given a pair of shapes, find corresponding points
- When shapes differ by non-rigid transformation - degrees of freedom can grow rapidly
- If transformation is isometric, we can use isometry-invariant shape properties to find correspondence
Problem definition

- Given a pair of shapes, find corresponding points
- When shape differ by elastic / topological transformation

SHREC’14 - Non-Rigid 3D Human Models track

[Huang et al 2011]
Matching algorithm: desired properties

• Given two (or more) shapes, find a map that is
  • Automatic
  • Fast to compute
  • Bijective (if we expect to have a global correspondence)
  • Low-distortion
  • Confirm to cycle-consistency constraints - in shape collections
Why this is important?

- Supervised machine learning algorithms require having shape collections with consistent annotations
- Some applications require having consistent alignment
- Co-alignment in shape collections
  - Harder than pairwise alignment
  - Can produce better results than pairwise alignment
- More in the following lecture
- For overview of shape alignment methods take cs233