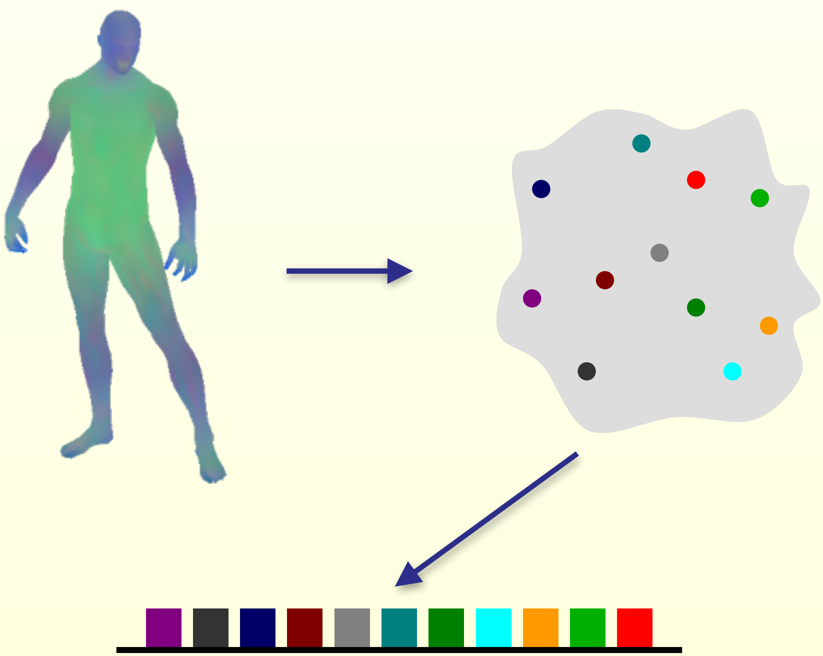
Shape retrieval
Intrinsic shape matching

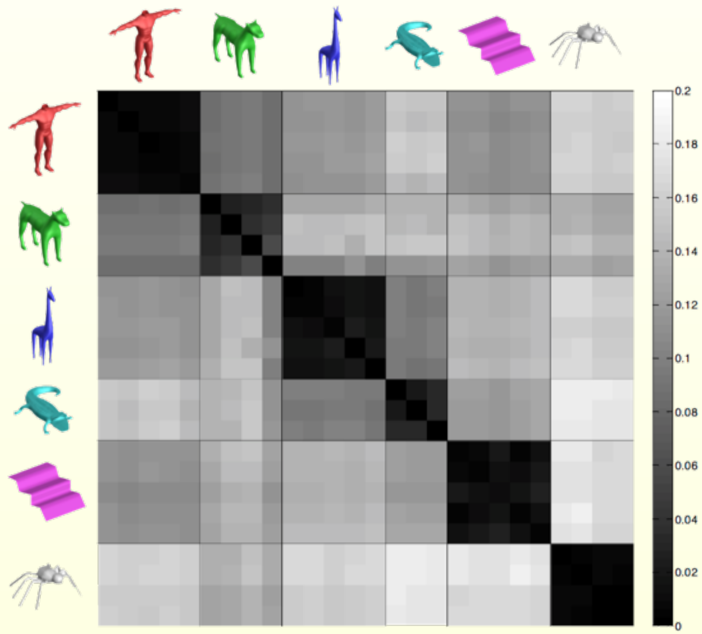


Anastasia Dubrovina
Computer Science Dept.
Stanford University

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.

Image from [Bronstein et al. 2006]

Applications: shape retrieval from large shape collection

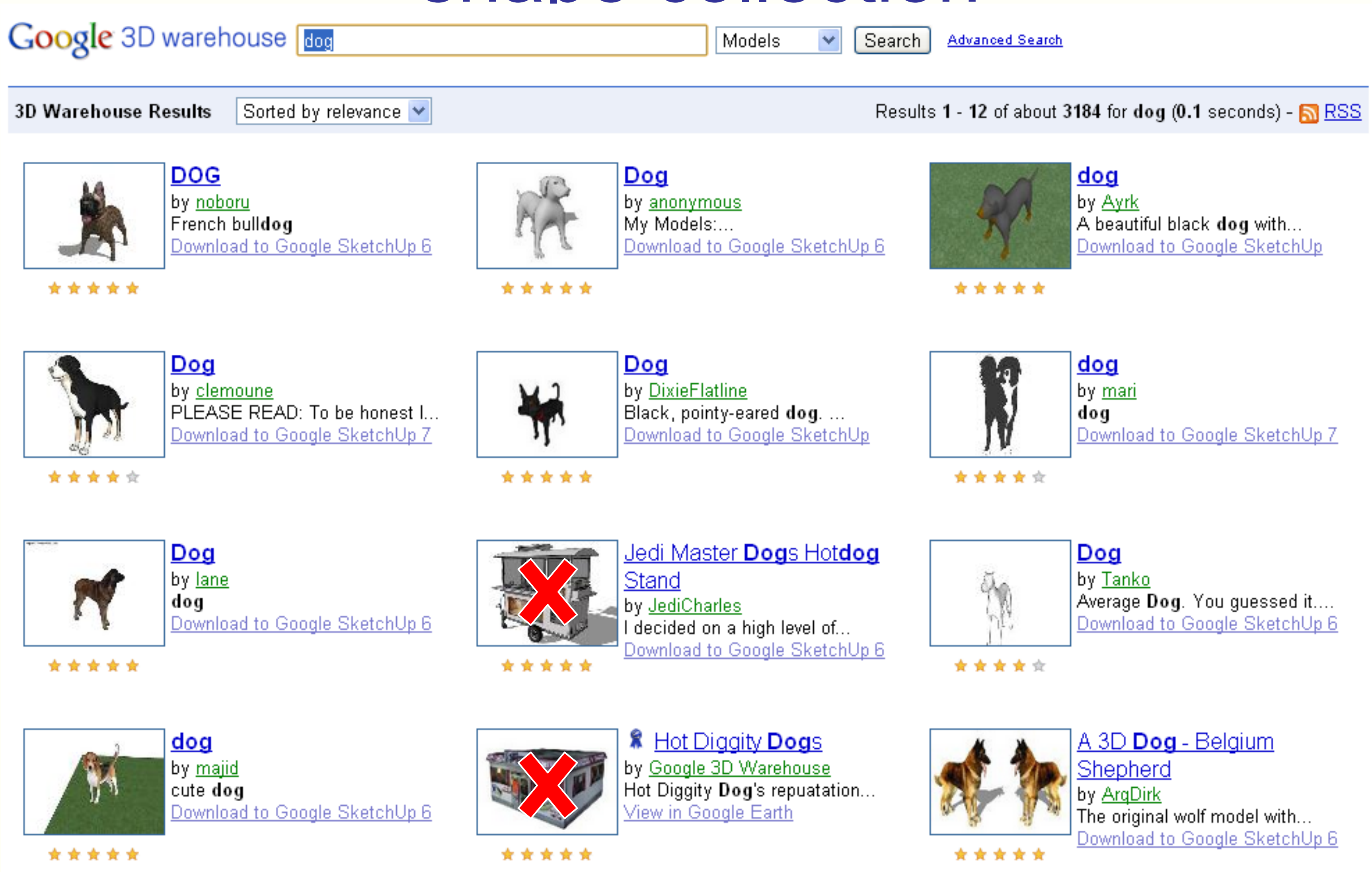


Image credit: M. Bronstein

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

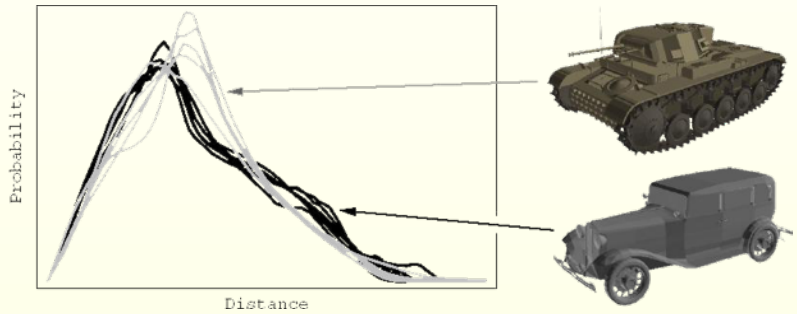
- Shape similarity and retrieval
    - Extrinsic shape similarity
    - Intrinsic shape similarity
    - Fine-grained similarity
    - Style similarity

- Deformable shape matching
    - If time permits

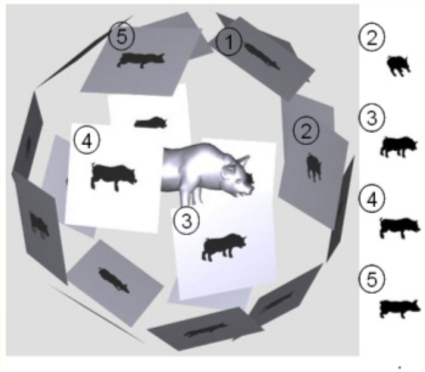
SHAPE SIMILARITY AND RETRIEVAL

Earlier work

- Descriptor-based similarity



Shape Distributions



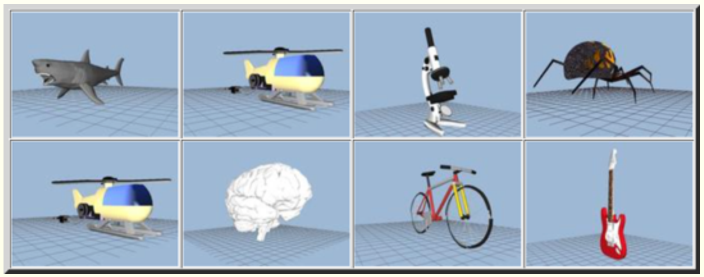
Light field descriptors

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

Images from **Princeton Shape Retrieval and Analysis Group** webpage

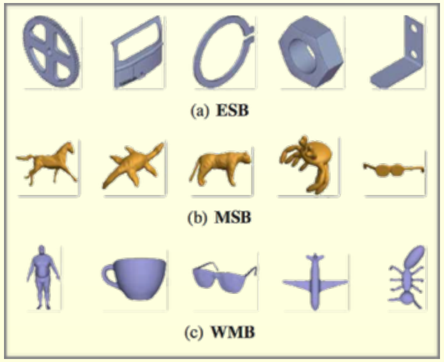
Datasets

- Examples



Princeton Shape Benchmark
1814 models, 90 classes

SHREK’11 Shape Retrieval Contest
1237 models

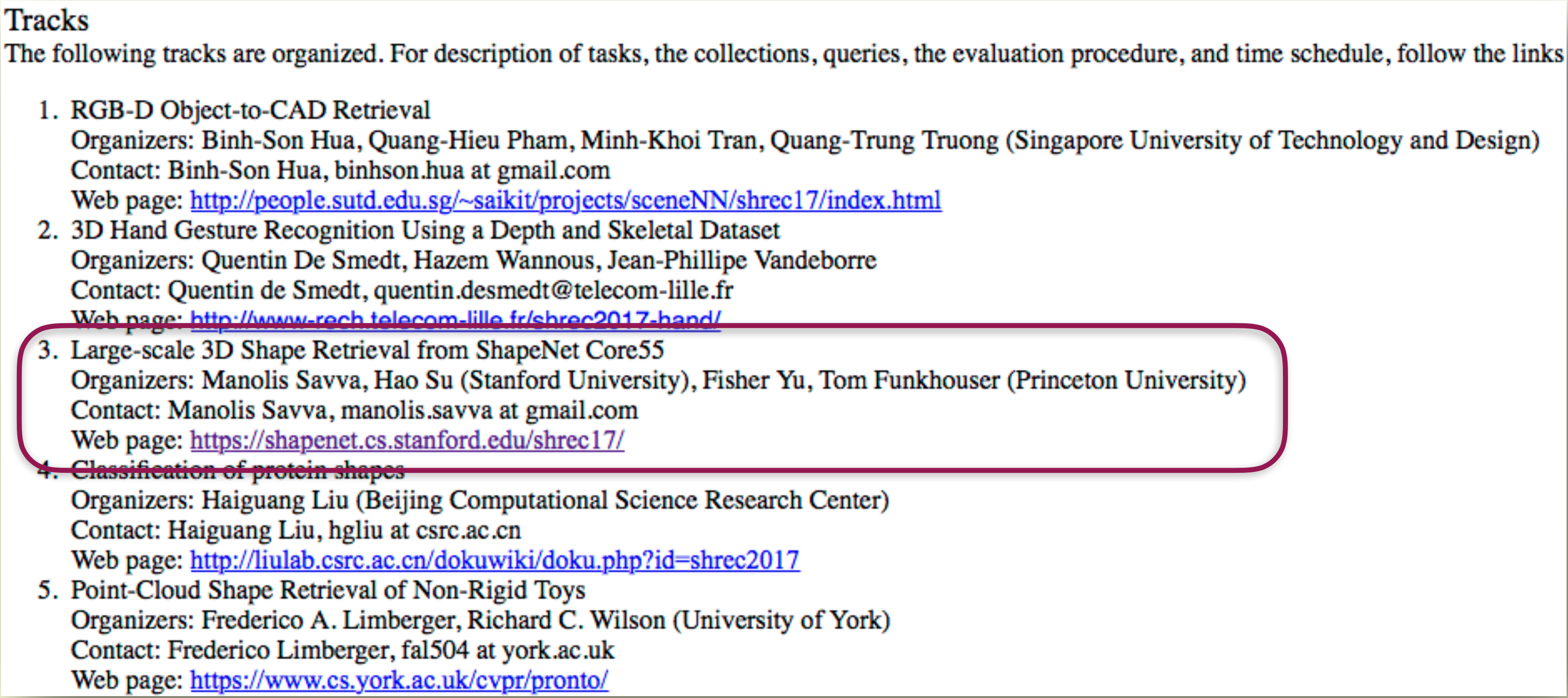


SHREK’14 Large Scale Retrieval Contest
8987 models, 171 classes

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
   Web page: http://people.sutd.edu.sg/~saikit/projects/sceneNN/shrec17/index.html
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
   Web page: http://www-rech.telecom-lille.fr/shrec2017-hand/
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/
4. Classification of protein shapes
   Organizers: Haiguang Liu (Beijing Computational Science Research Center)
   Contact: Haiguang Liu, hgliu at csrc.ac.cn
   Web page: http://liulab.csrc.ac.cn/dokuwiki/doku.php?id=shrec2017
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/

Large-scale retrieval contest using ShapeNet

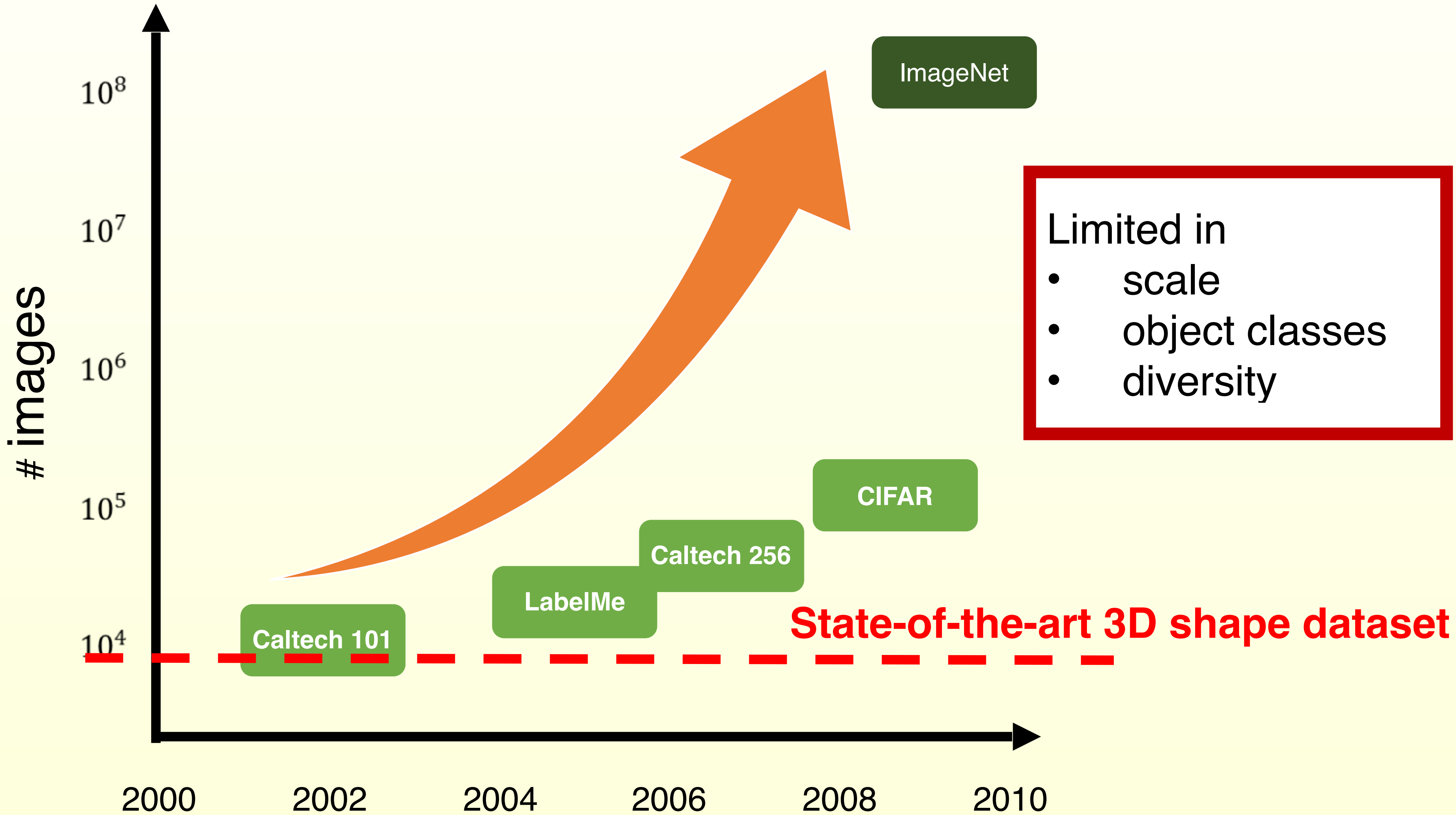
SHAPENET

Home Introduction News Dataset Procedure/Schedule Evaluation Results Team References

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.

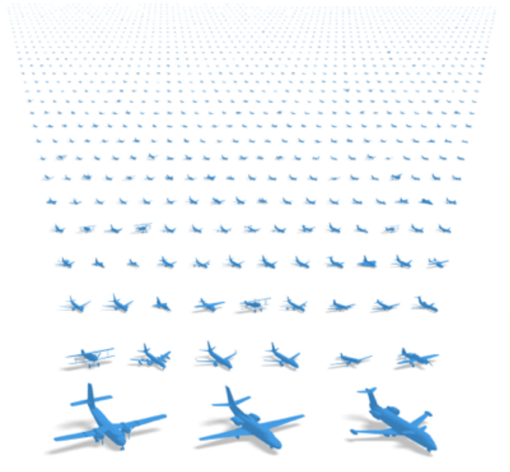
Image vs. 3D datasets



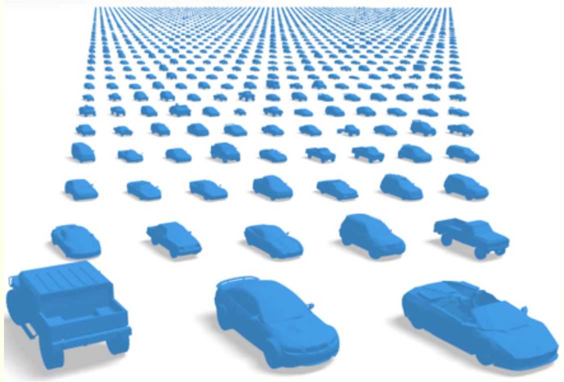
Limited in
- scale
- object classes
- diversity

Slide credit: H. Su

ShapeNet



...



~3 million models

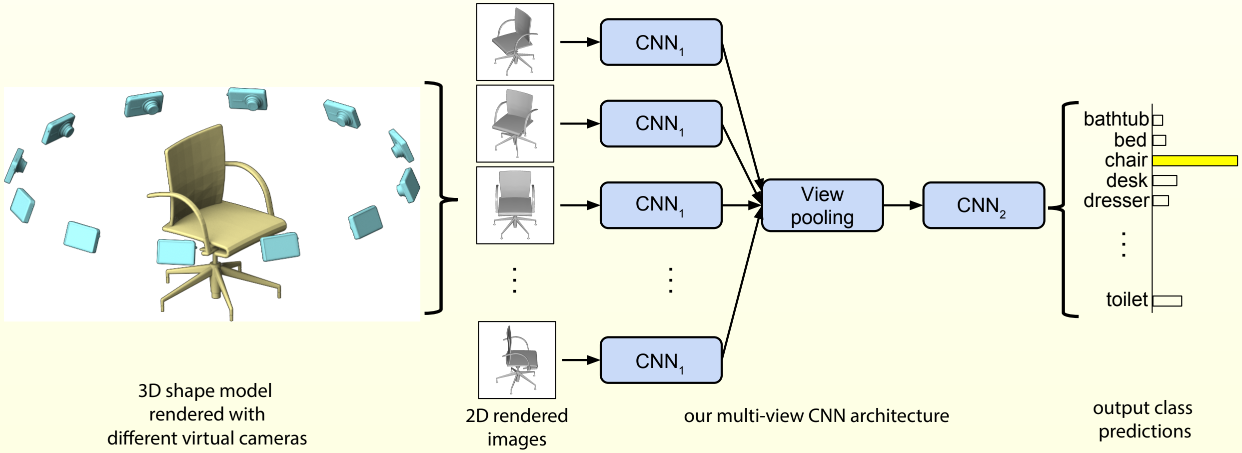
~2,000 classes

Rich annotations

Work in progress

Large-scale retrieval contest using ShapeNet

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



[Su et al. 2015]

- (to be covered in the second part of the course)
- All methods perform <u>extrinsic shape retrieval</u>

Today - intrinsic shape similarity

- Different from extrinsic, or rigid, similarity

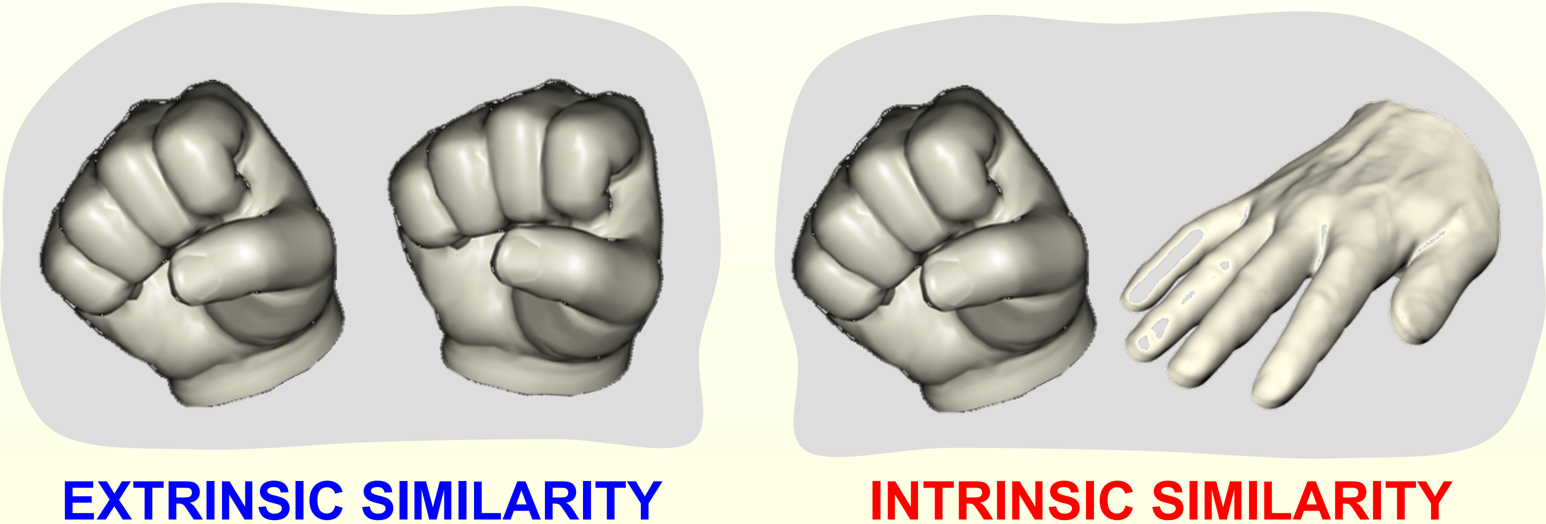
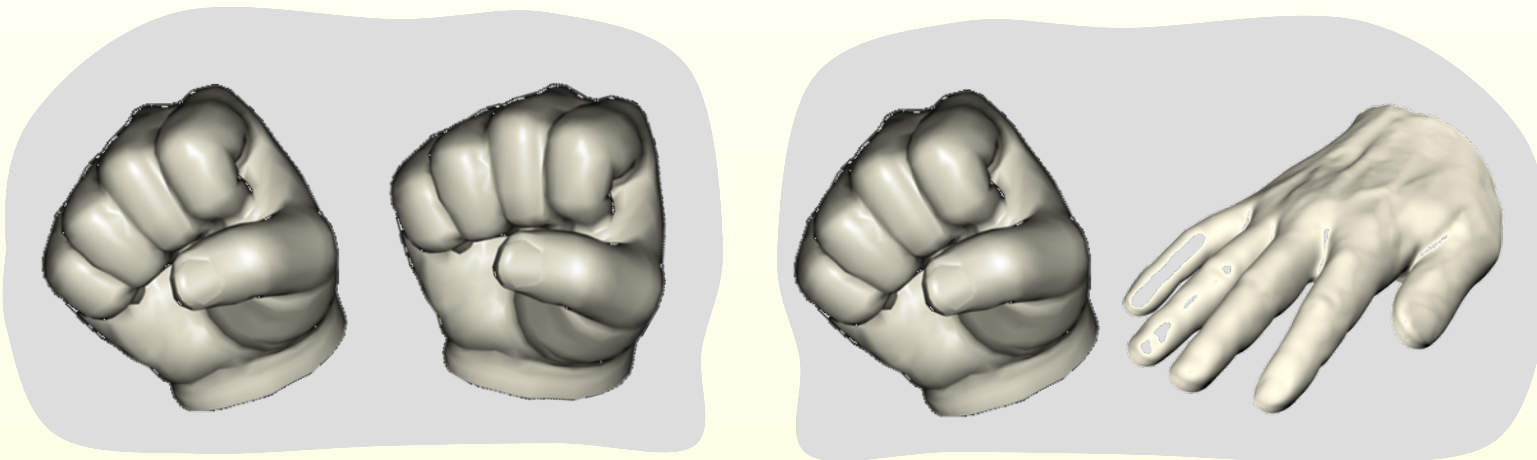


Image credit: A. Bronstein

Today - intrinsic shape similarity

- Different from extrinsic, or rigid, similarity



EXTRINSIC SIMILARITY **INTRINSIC SIMILARITY**

- Approaches we will discuss today
  - Shape Google [Bronstein et al. 2011]
  - Supervised Bag-of-features [Litman et al. 2014]

Image credit: A. Bronstein

Shape Google

Geometric words and expressions for shape retrieval

Alex Bronstein — Tel-Aviv University, Israel

Michael Bronstein — University of Lugano, Switzerland

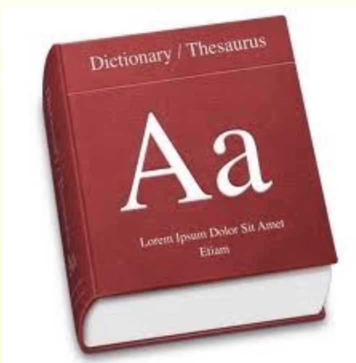
Maks Ovsjanikov, Leonidas Guibas — Stanford University, USA

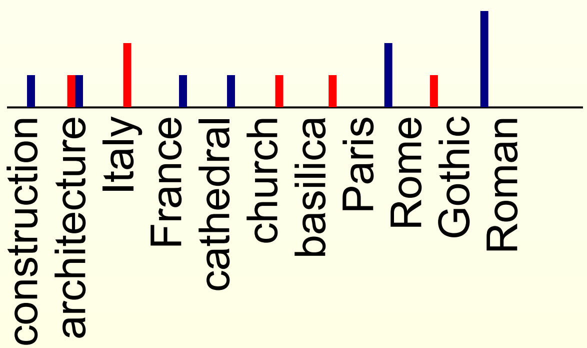
Università della Svizzera italiana | Faculty of Informatics | Institute of Computational Science

SIGGRAPH 2011, Vancouver, Canada

Bag of words

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.

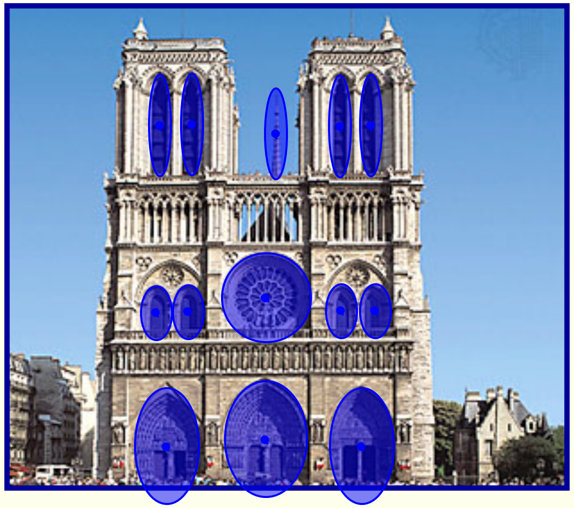


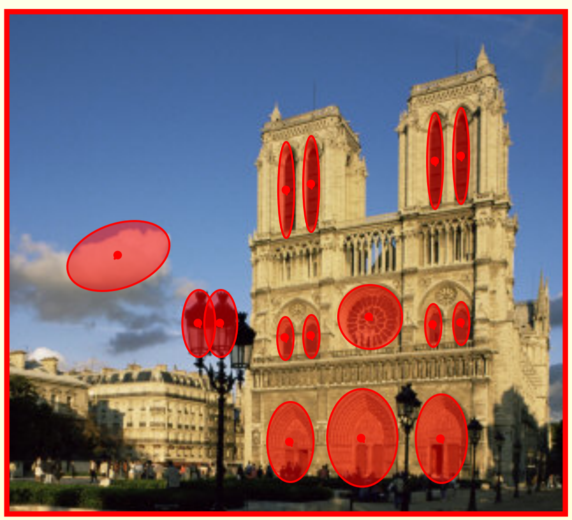


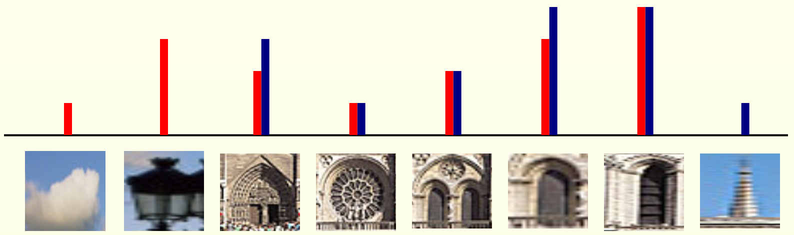
St. Peter's basilica is the largest church in world, located in Rome, Italy. As a work of architecture, it is regarded as the best building of its age in Italy.

Slide credit: M. Bronstein

Bags of visual features







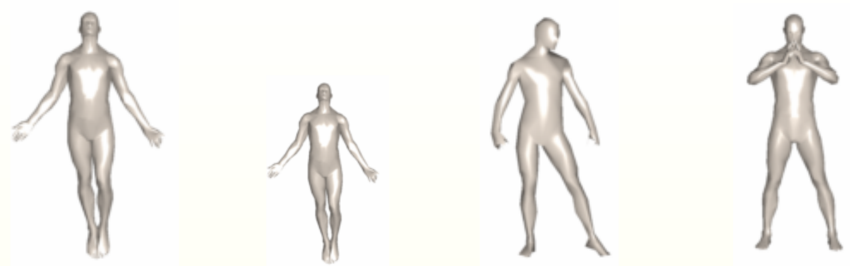
Visual vocabulary

Think of an image as a collection of primitive elements

Zisserman *et al.*

Slide credit: M. Bronstein

Local shape descriptors

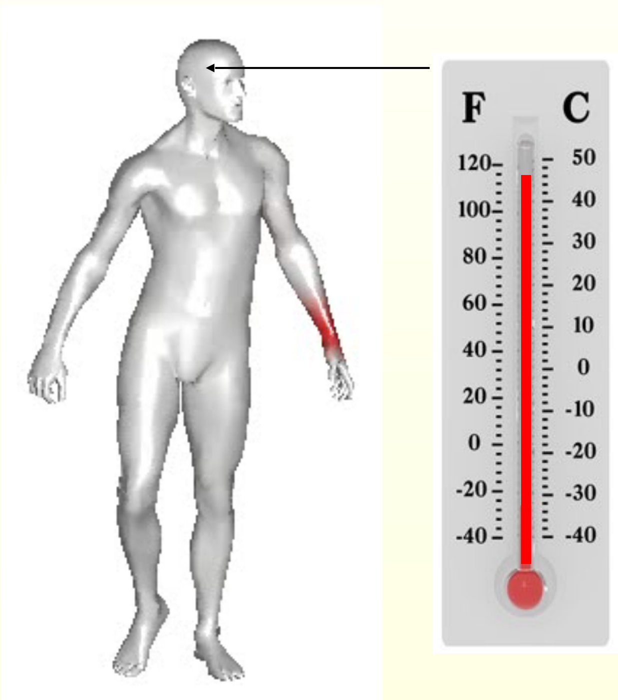


	Representation	Rigid	Scale	Bending	Topology
Curvature	Mesh	✓	✗	✓	✓
Spin image[2]	Mesh	✓	✗	✗	✗
Shape context[3]	Mesh	✓	✗	✓	✓
HKS[4]	Mesh	✓	✗	✓	✓
SI-HKS[5]	Mesh	✓	✓	✓	✓
Color HKS[6]	Mesh	✓	✓	✓	✓
vHKS[7]	Volume/Mesh	✓	✗	✓	✓
WKS[1]	Mesh	✓	✗	✓	✓

[1] Aubry *et al.* 2011; [2] Johnson, Hebert 1999; [3] Belongie *et al.* 2002; [4] Sun *et al.* 2009; Gebal *et al.* 2009
[5] B, Kokkinos 2010; [6] Kovnatsky, BB, Kimmel 2010; [7] Raviv, BB, Kimmel 2010

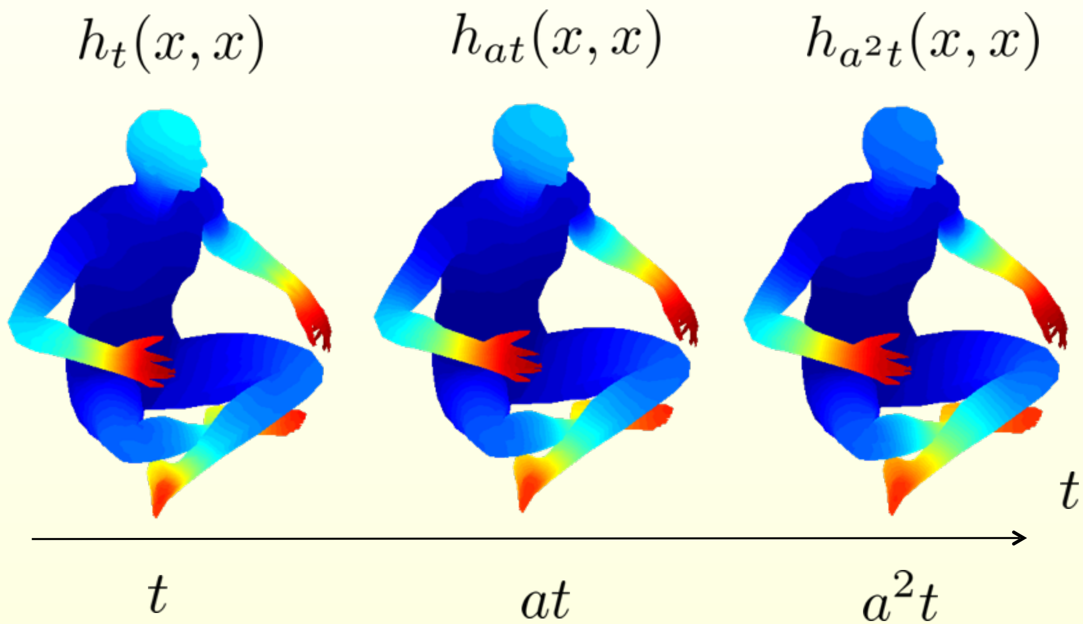
Slide credit: M. Bronstein

Heat kernel signature



Heat diffusion on a manifold

Diagonal of heat kernel $h_t(x,x)$

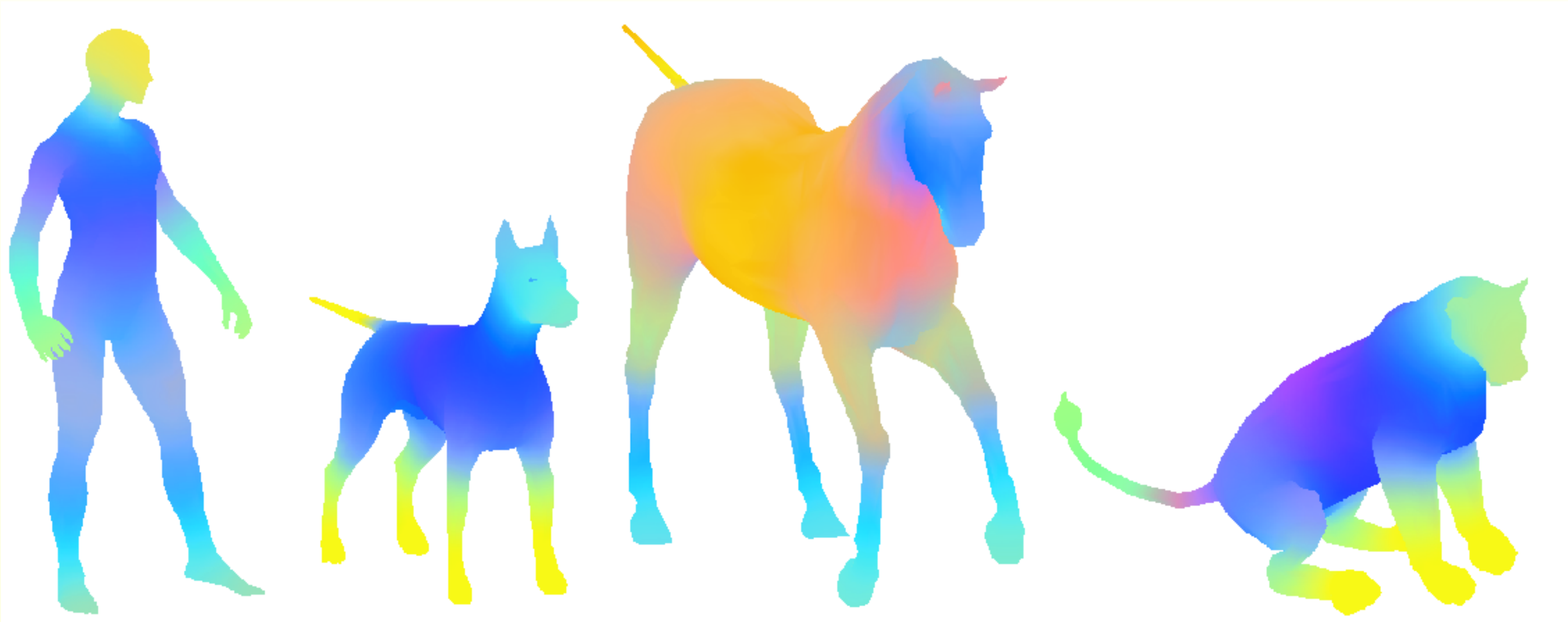


Multi-scale point descriptor

$$p(x) = (h_t(x,x), \ldots, h_{a^n t}(x,x))$$

Slide credit: M. Bronstein

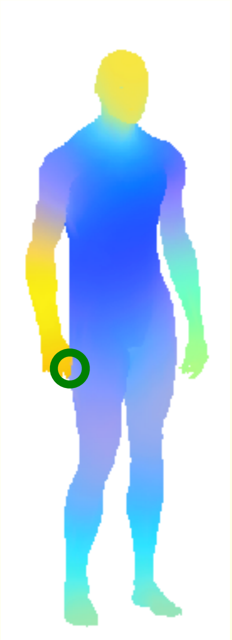
Heat kernel signature



Heat kernel signatures represented in RGB space

Heat kernel signature

Invariant to isometric deformations



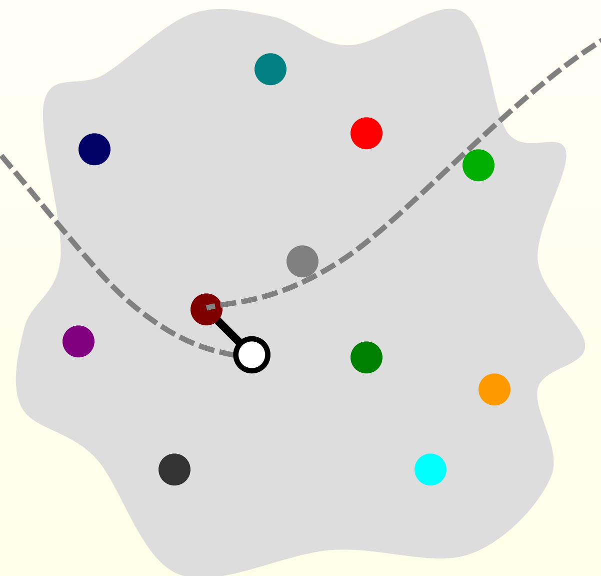
Localized sensitivity to topological noise

Not scale invariant

(scale-invariant HKS in [B&Kokkinos 2010])

Slide credit: M. Bronstein

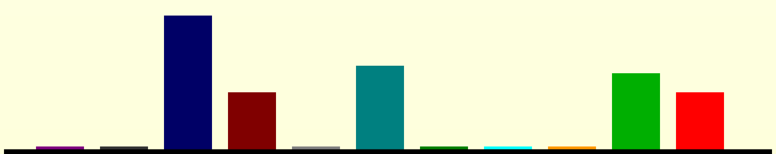
Shape Google



Geometric vocabulary $p_1, \ldots, p_V$

$$k^* = \underset{i=1,\ldots,V}{\operatorname{argmin}} \|p(x) - p_i\|$$

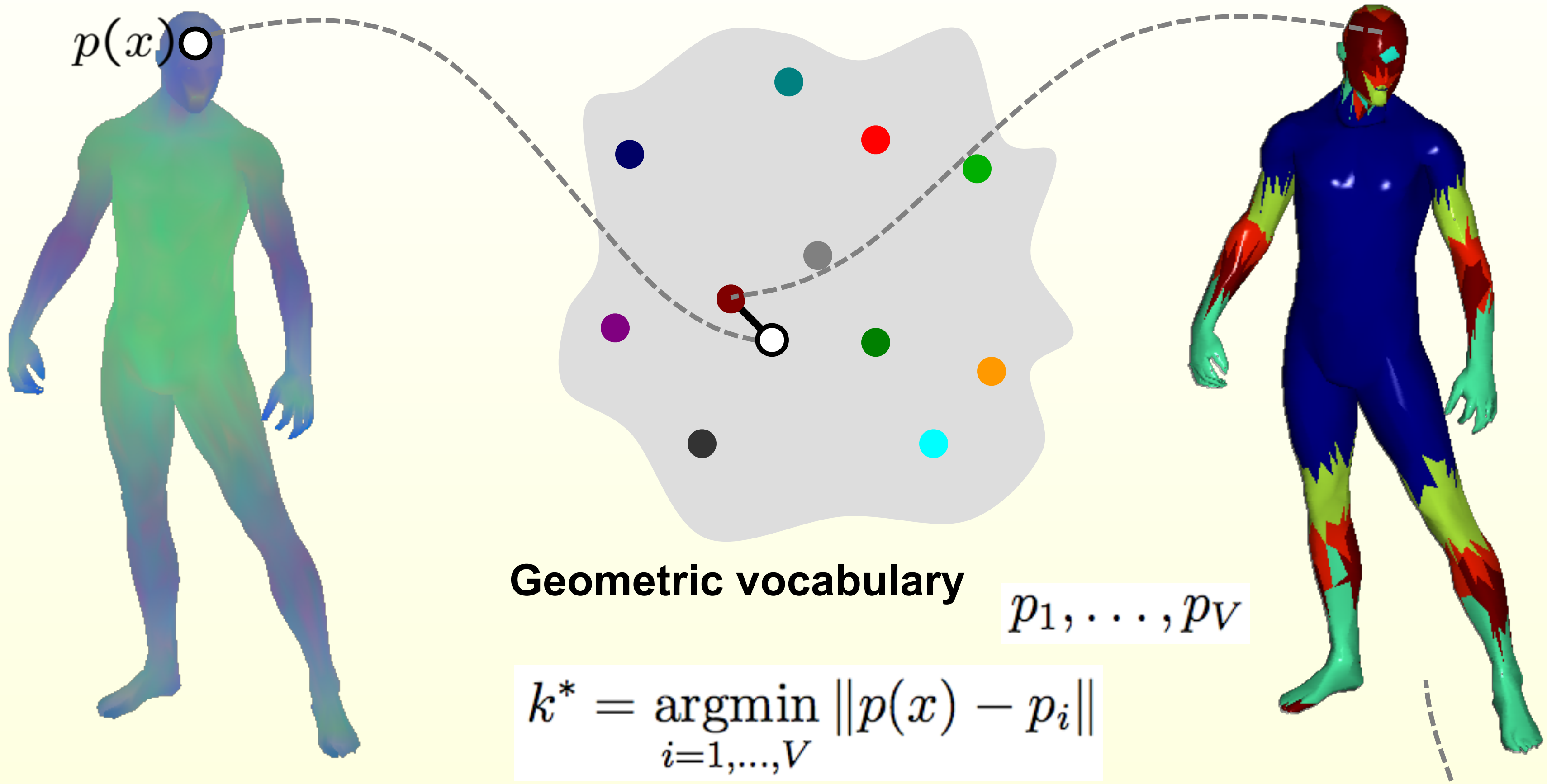
Think of a **shape** as a **collection of primitive elements**

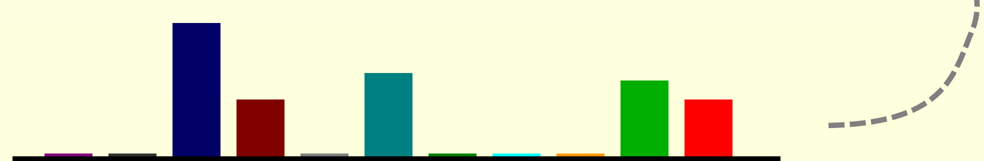


Bag of geometric words

Slide credit: M. Bronstein

Shape Google



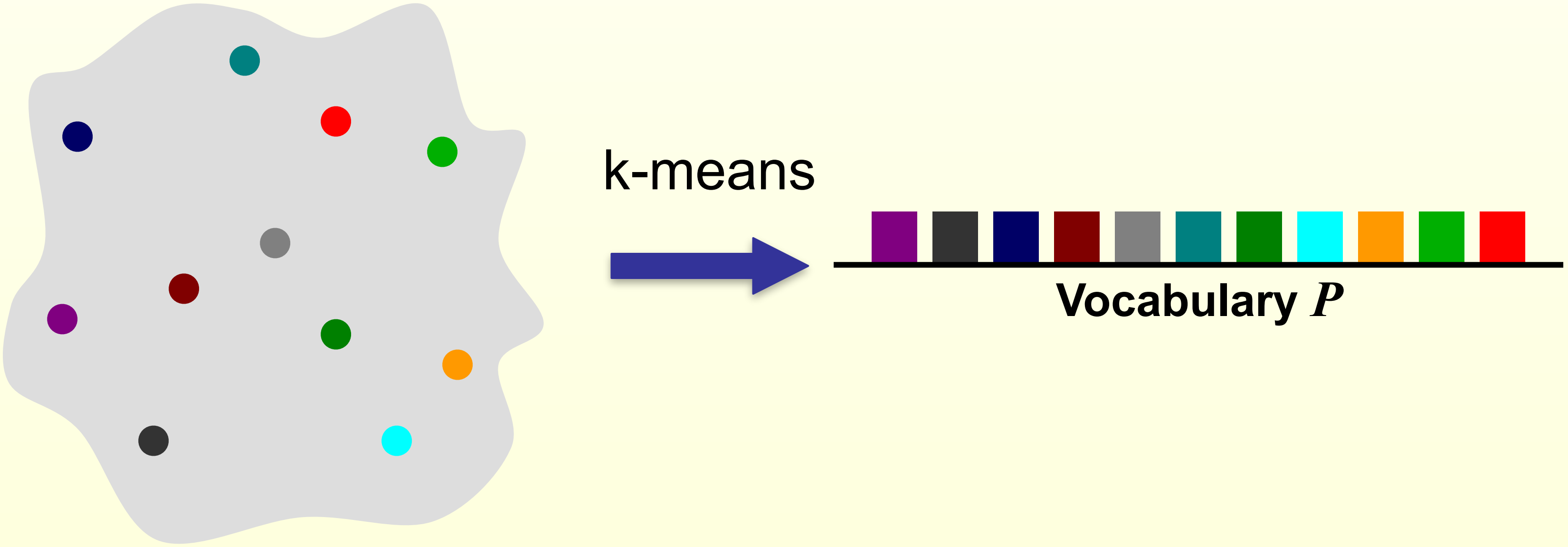


Shape descriptors can be computed at stable points, or at all points

Slide credit: M. Bronstein

BoGW - computation details

- A vocabulary $\mathcal{P} = \{\mathrm{p}_1, \ldots, \mathrm{p}_V\}$ 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



BoGW - computation details

- A vocabulary $\mathcal{P} = \{\mathrm{p}_1, \ldots, \mathrm{p}_V\}$ 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

$$\theta_i(x) = c(x) e^{-\frac{\|\mathrm{p}(x) - \mathrm{p}_i\|_2^2}{2\sigma^2}}$$

"probability of the point x to be associated with the descriptor pi"

- Integrate over the whole shape X

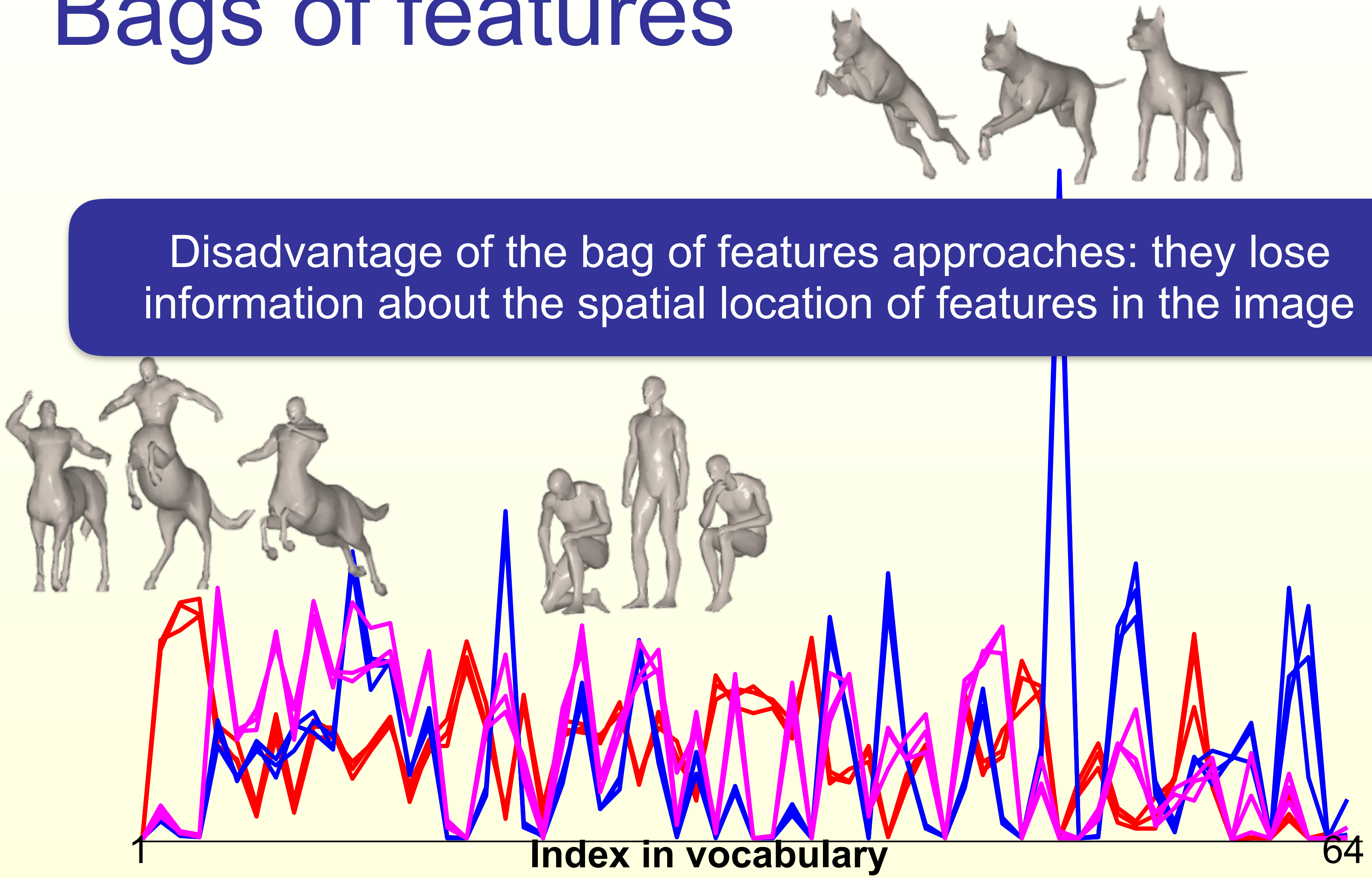
$$\mathrm{f}(X) = \int_X \theta(x) d\mu(x)$$

weighting by area element

Slide credit: M. Bronstein

Bags of features

Disadvantage of the bag of features approaches: they lose information about the spatial location of features in the image

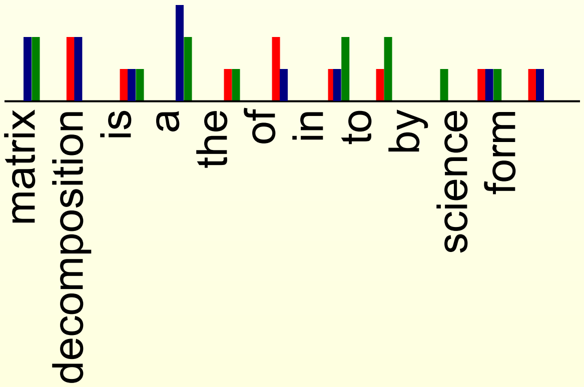


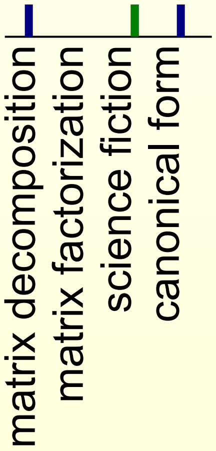
Expressions

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.

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 science fiction movie released in 1999. Matrix refers to a simulated reality created by machines in order to subdue the human population.

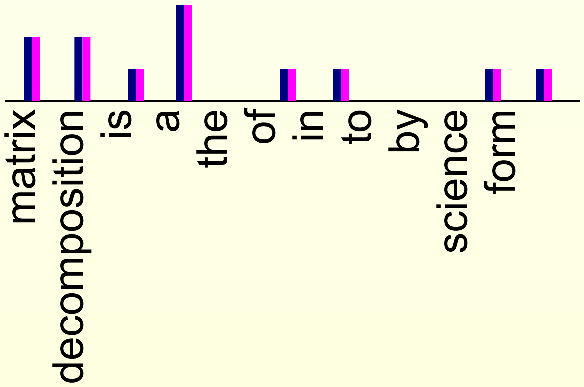


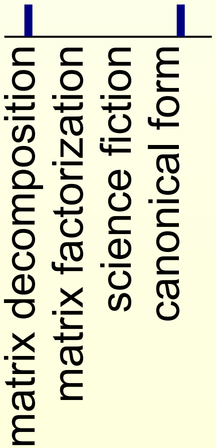


Expressions

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.

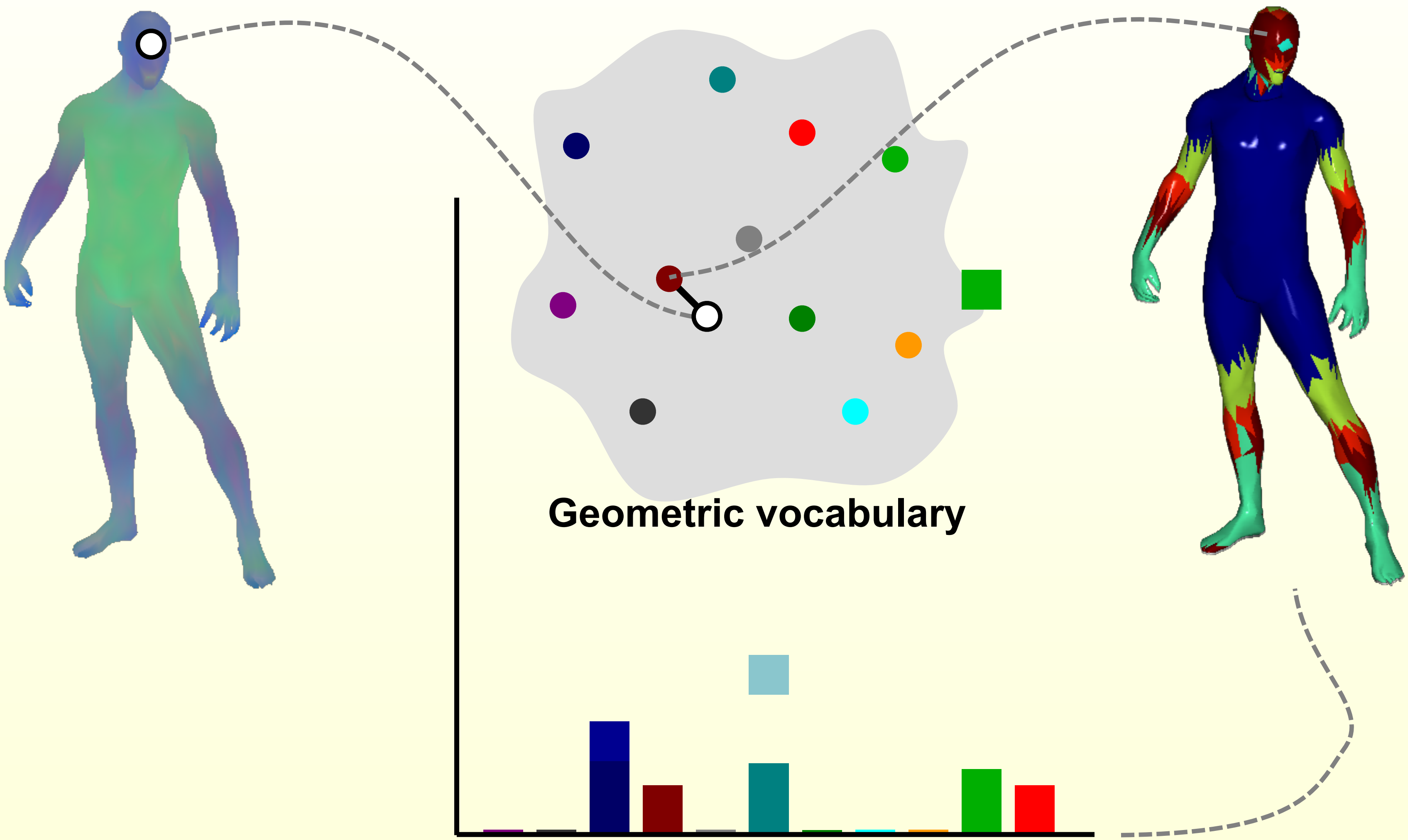
In particular matrix used type a some science, decomposition form a factorization of is canonical. matrix math decomposition is in a Each problem. into of





Slide credit: M. Bronstein

Bags of geometric expressions



Bag of geometric words (BoF)

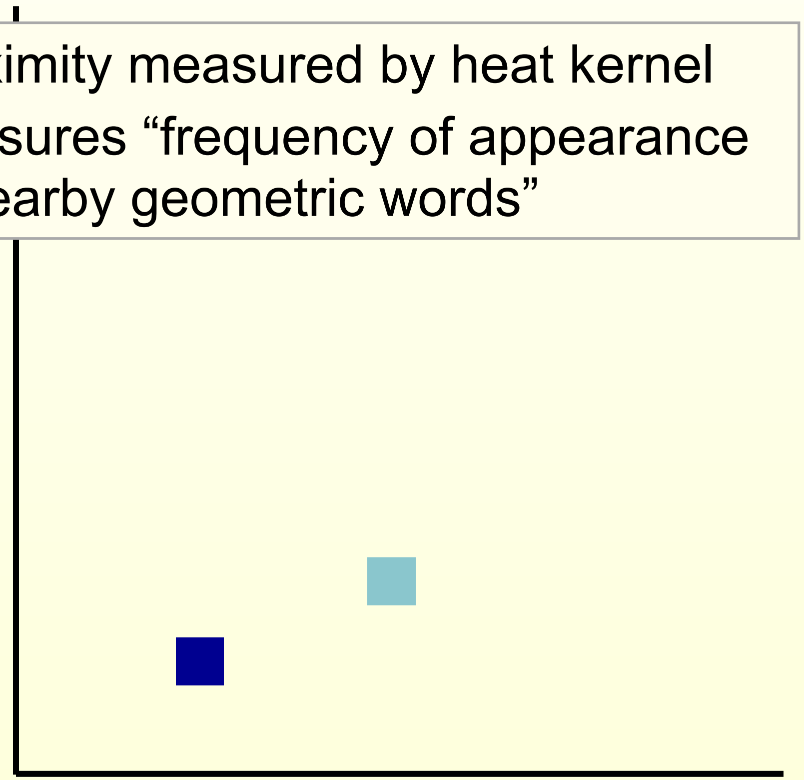
Bags of geometric expressions (SS-BoF)

Slide credit: M. Bronstein

Spatially Sensitive Bags of Features (SS-BoF)

$$\mathrm{F}(X) = \int_{X \times X} \theta(x)\theta^{\mathrm{T}}(y) K_t(x, y)\, d\mu(x)\, d\mu(y)$$

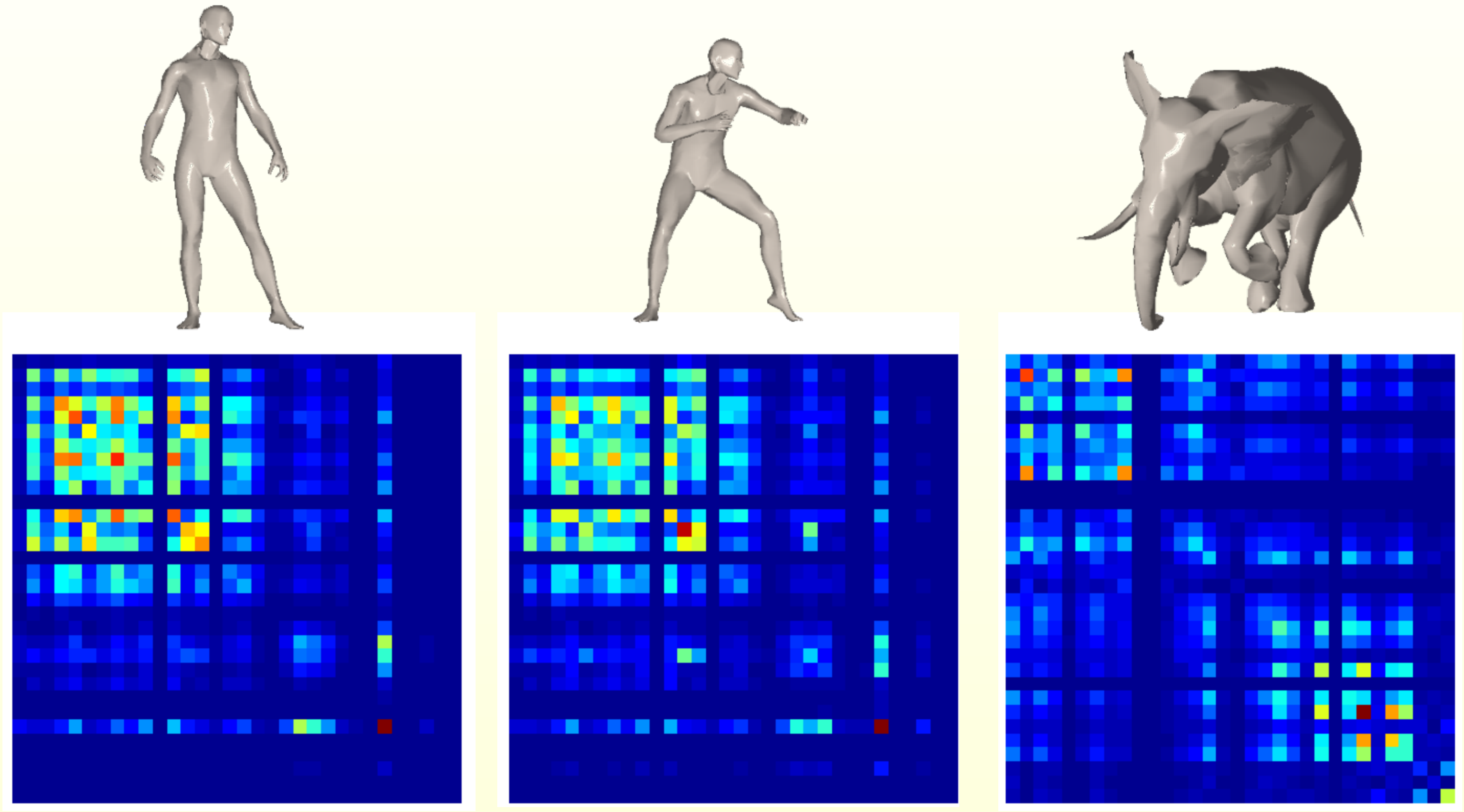
- Proximity measured by heat kernel
- Measures "frequency of appearance of nearby geometric words"

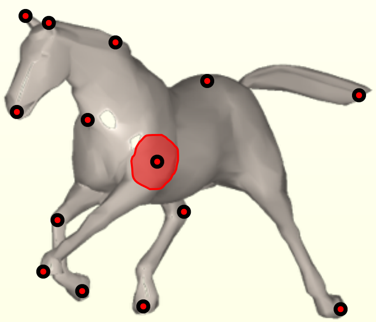


Bag of geometric expressions (SS-BoF)

Slide credit: M. Bronstein

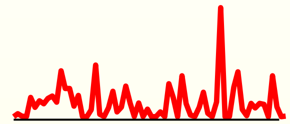
Bags of geometric expressions





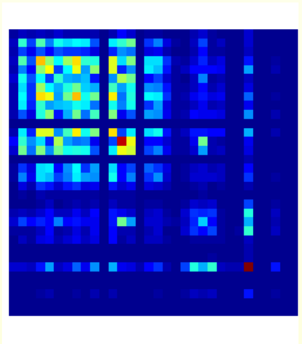
Feature descriptor

Geometric words



Bag of geometric words

Geometric expressions



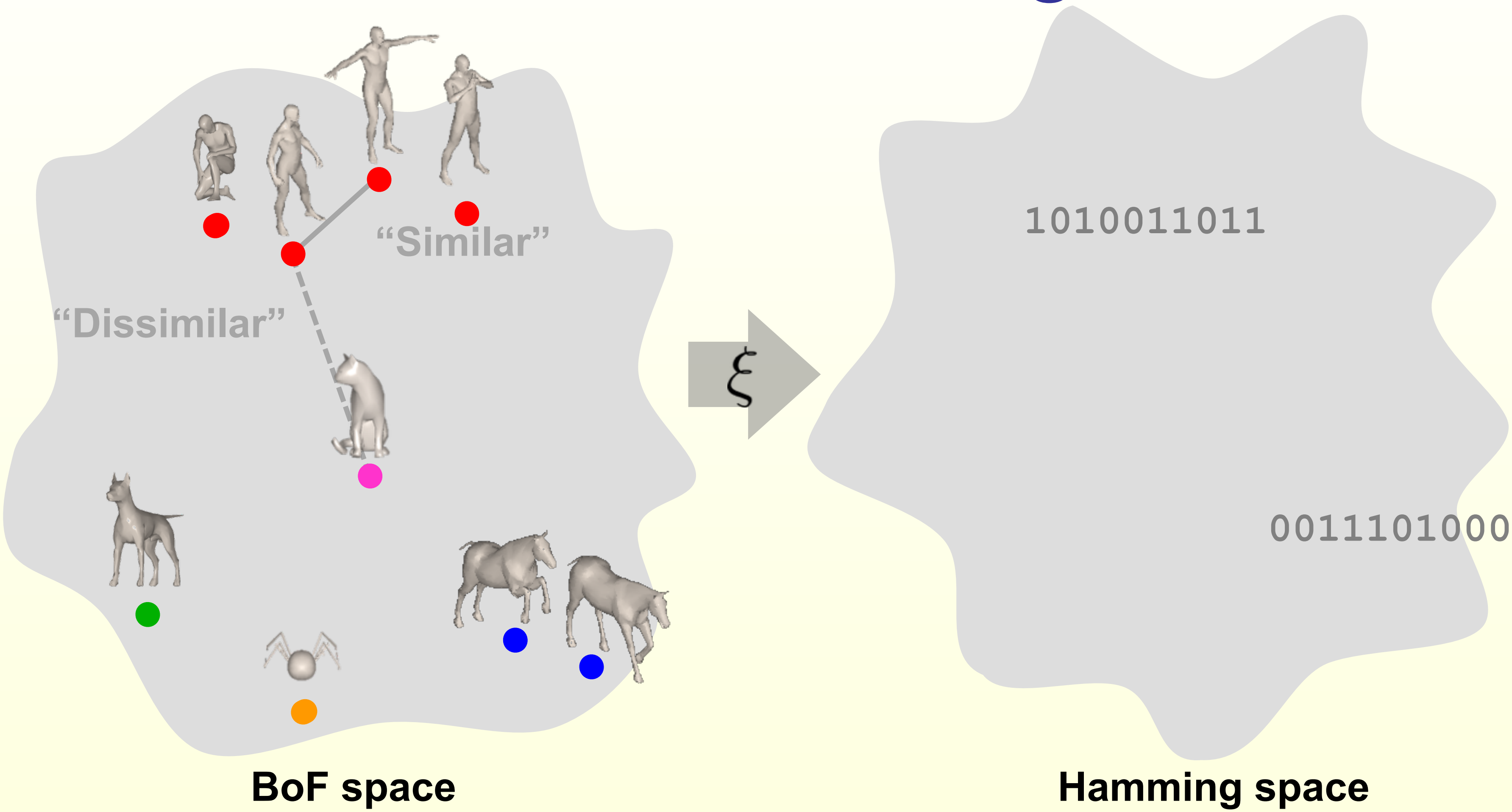
Spatially-sensitive bag of words

0100110110100 1

Shapes as binary codes: similarity-sensitive hashing

Slide credit: M. Bronstein

Metric learning

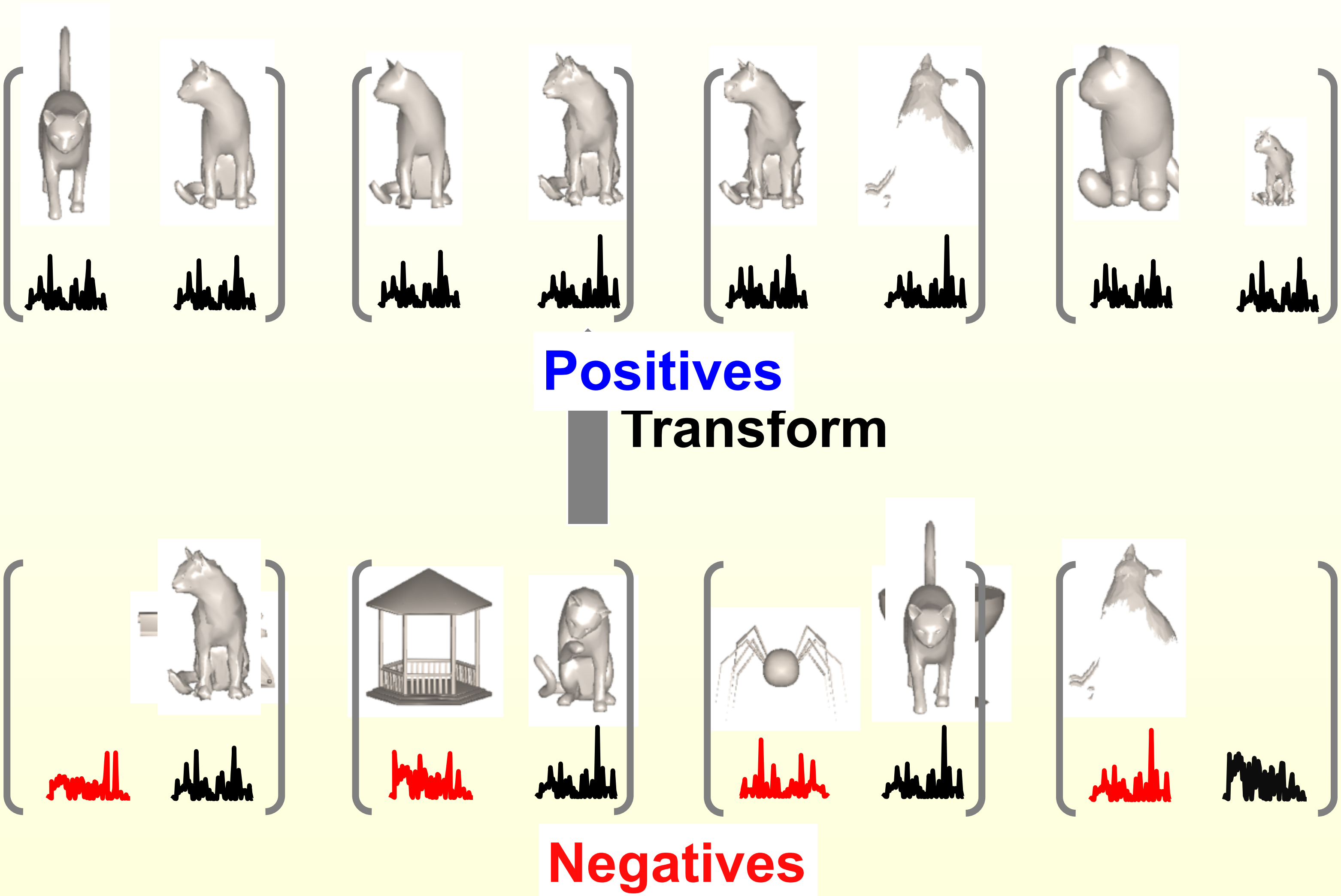


Shape hash: just 64 bits!

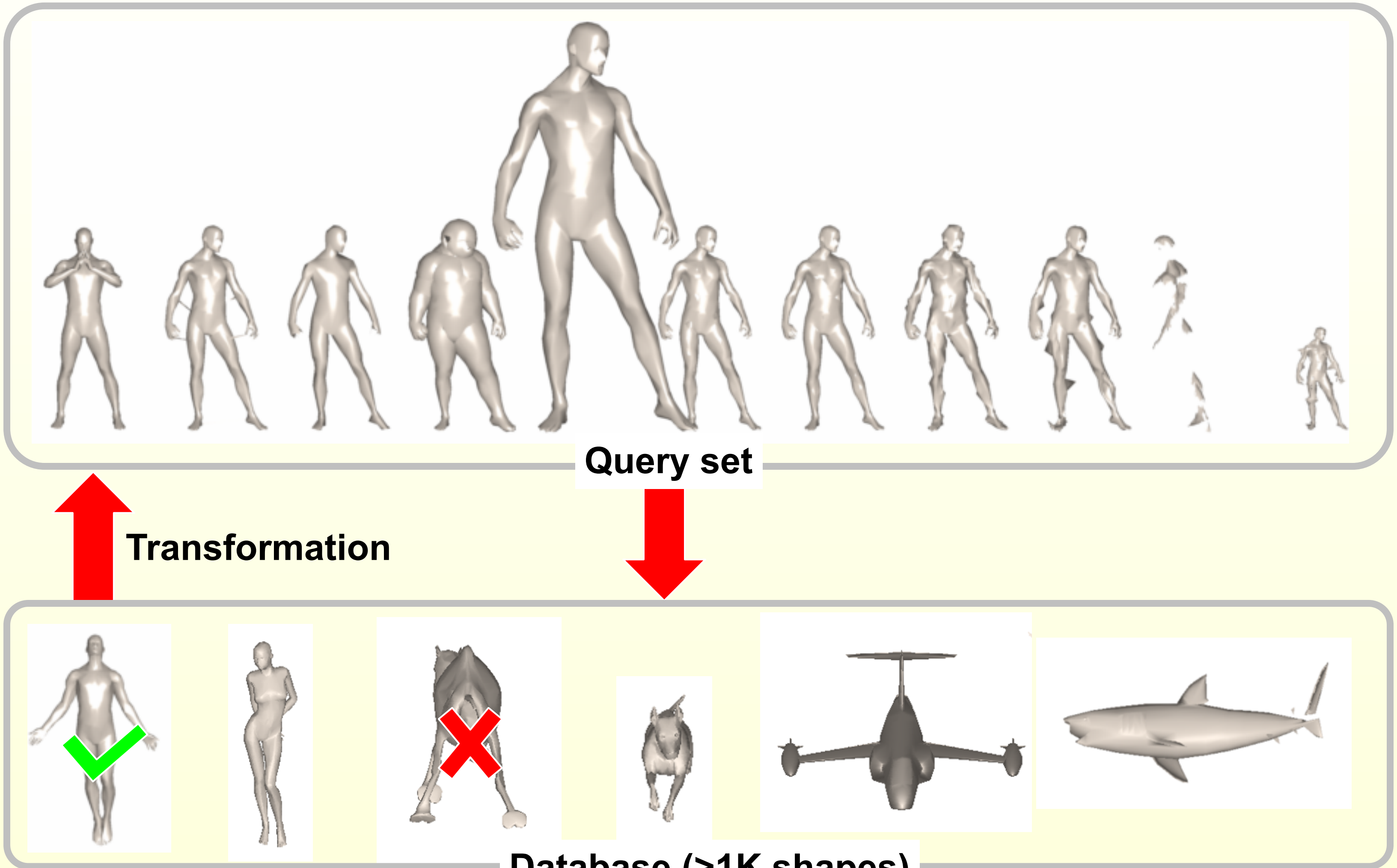
Shakhnarovich *et al.*; Grauman *et al.*; BB, Kimmel 2009; Ovsjanikov, BB, Guibas 2009; Strecha, BB, Fua 2010

Slide credit: M. Bronstein

Training



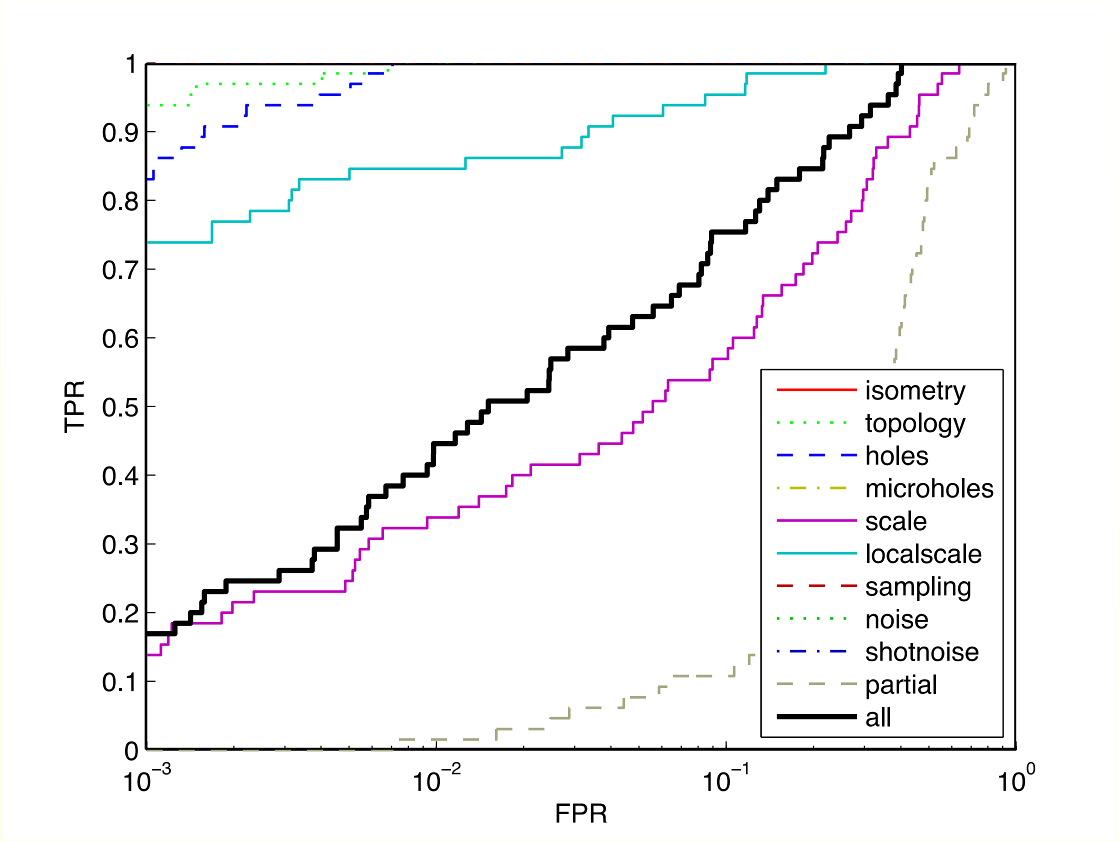
SHREC 2010: Robust shape retrieval benchmark



TOSCA dataset + Princeton dataset (Funkhouser *et al.*)

Slide credit: M. Bronstein

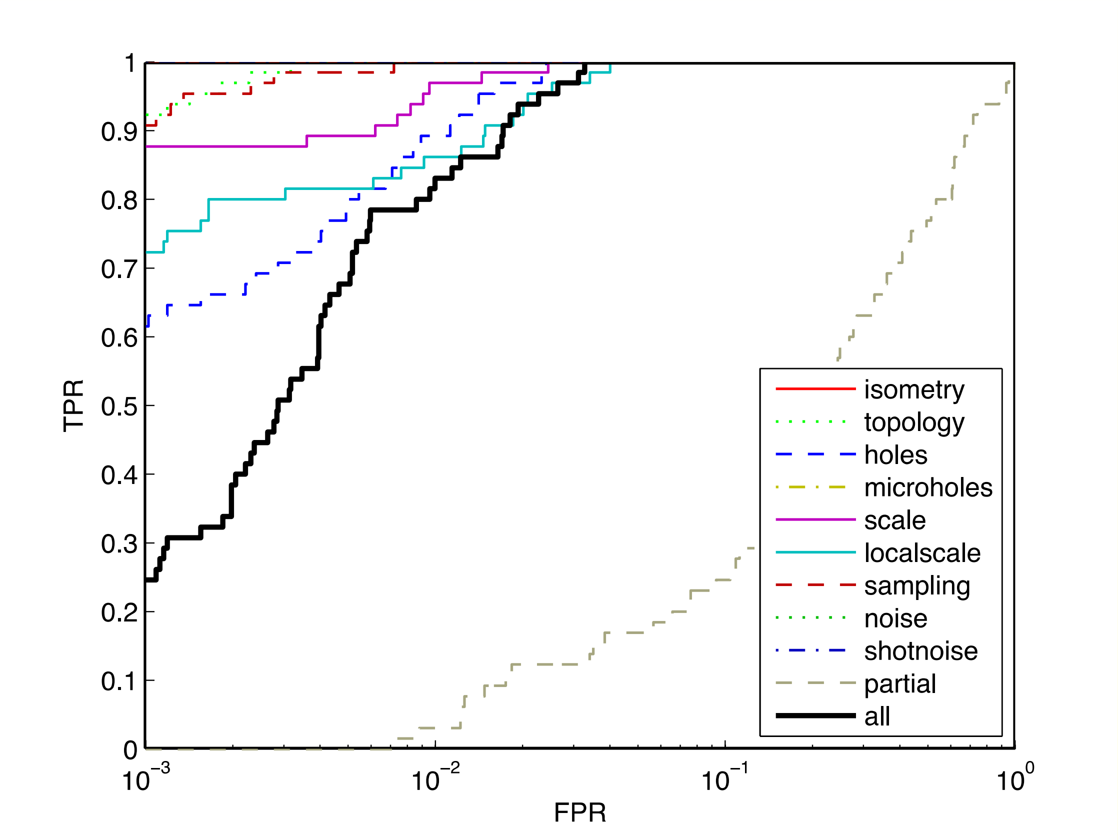
Results



Shape Google (HKS)

Slide credit: M. Bronstein

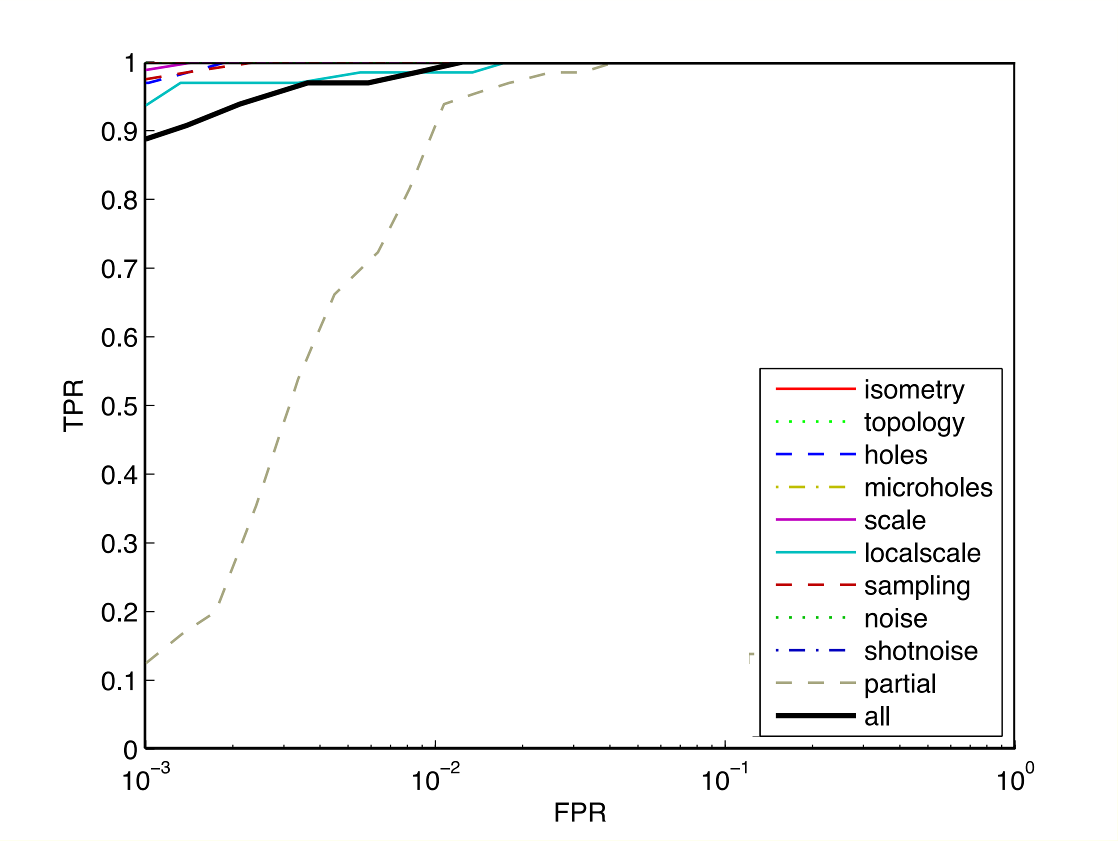
Results



Shape Google (Scale-invariant HKS)

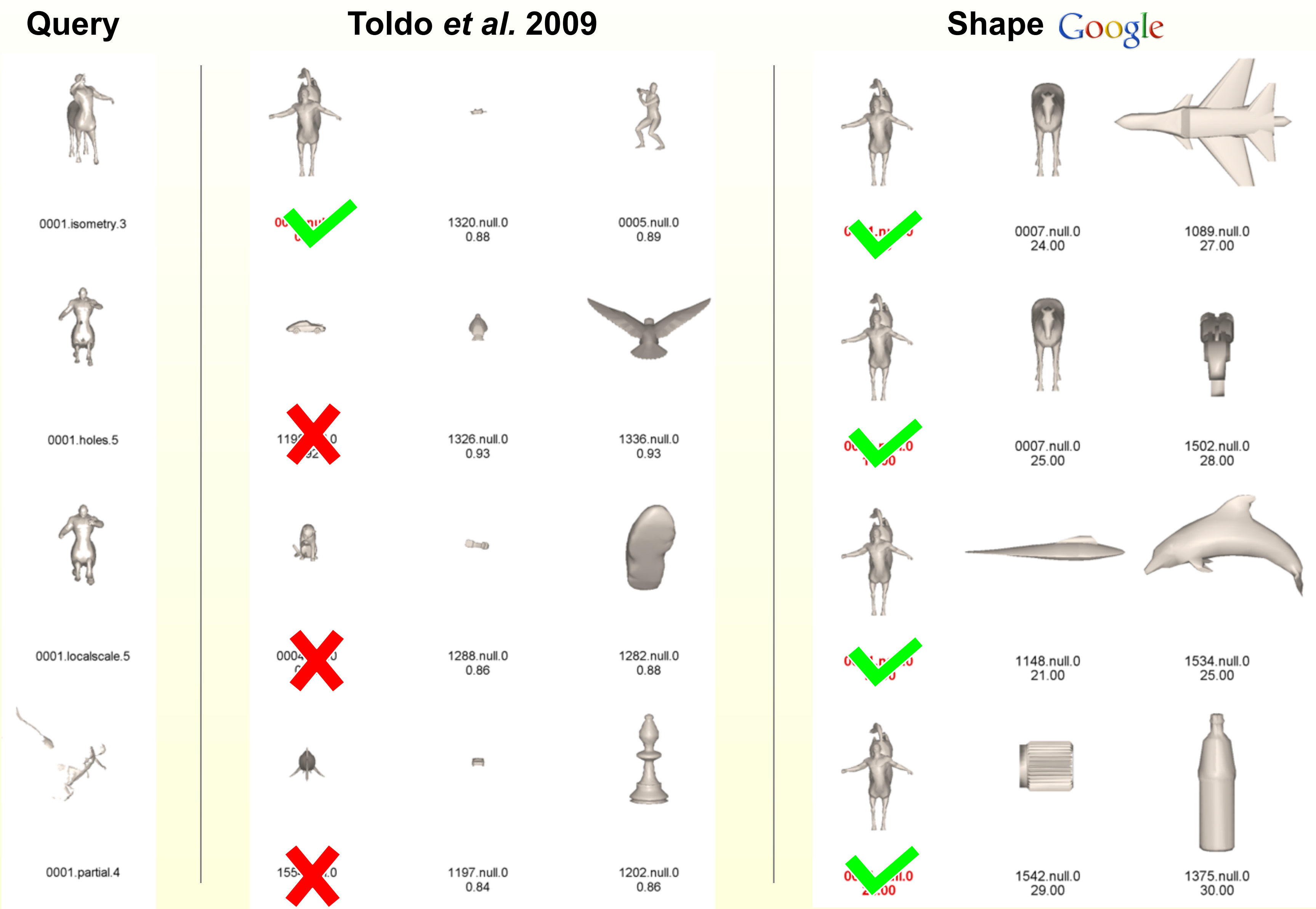
Slide credit: M. Bronstein

Results



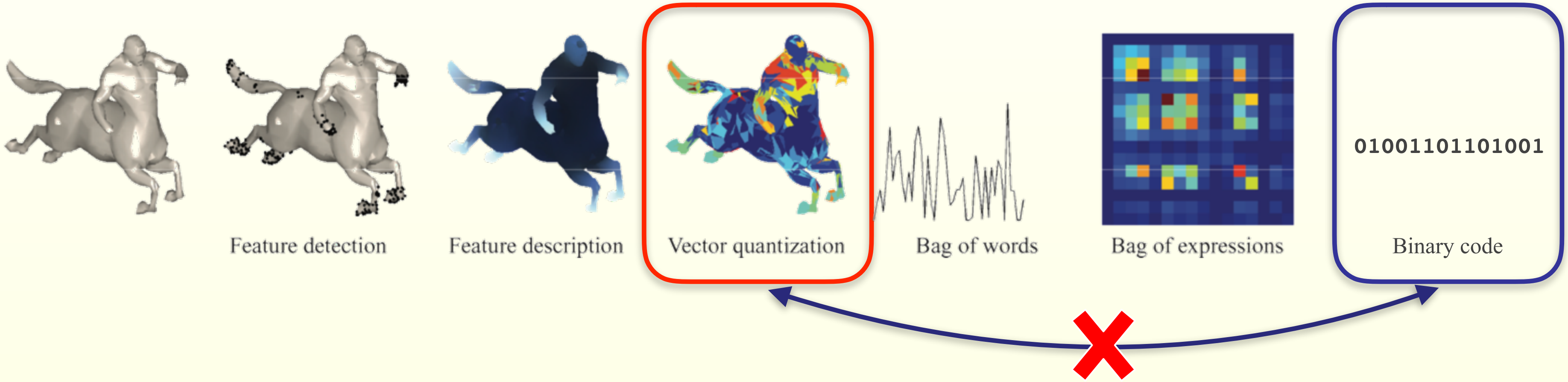
Shape Google+Metric learning

Slide credit: M. Bronstein



Slide credit: M. Bronstein

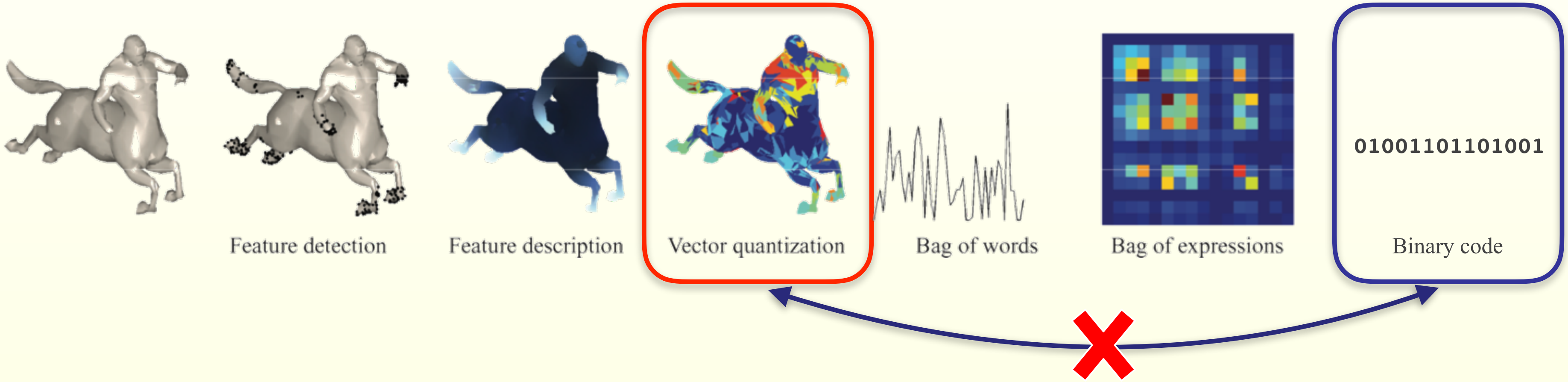
Drawback of the standard BoF construction



- The dictionary is constructed in an unsupervised manner using clustering, unaware of the following learning stage

Image credit: M. Bronstein

Drawback of the standard BoF construction

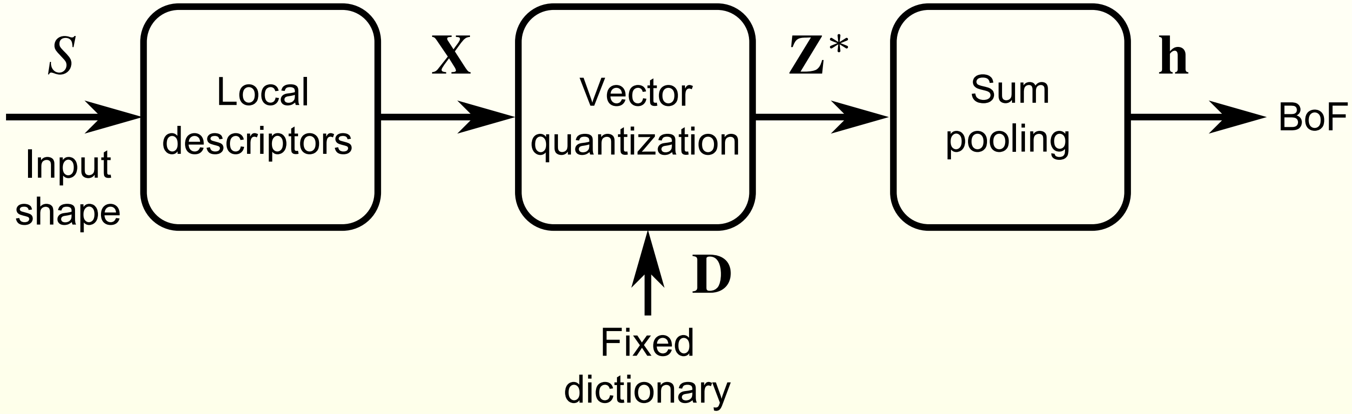


- The dictionary is constructed in an unsupervised manner using clustering, unaware of the following learning stage
- Suggested improvement: add **supervision** to the BoF training

  "Supervised learning of bag-of-features shape descriptors using sparse coding" [Litman et al. 2014]

Image credit: M. Bronstein

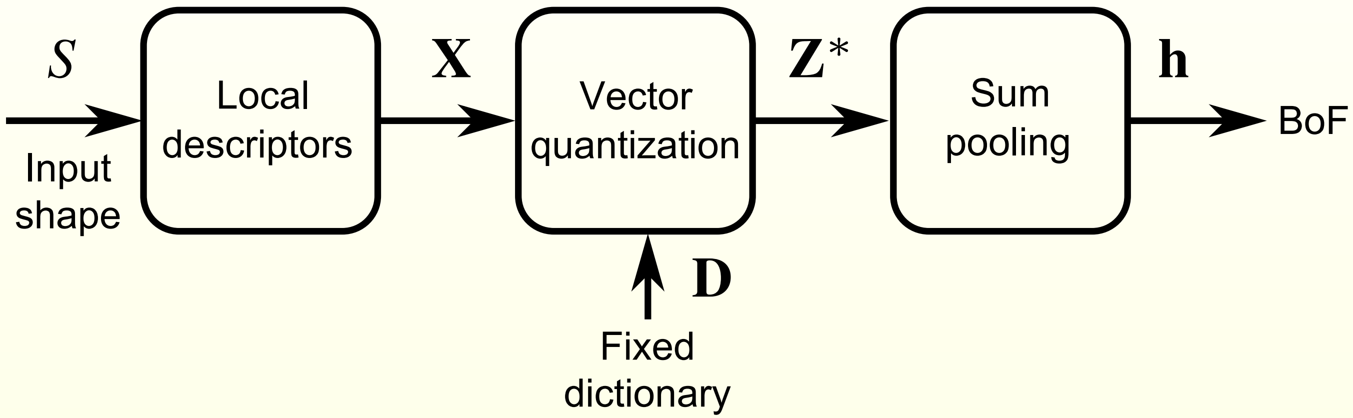
BoF computation flow - simplified



- Compute local descriptors - e.g., HKS

Image credit: R. Litman

BoF computation flow - simplified



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

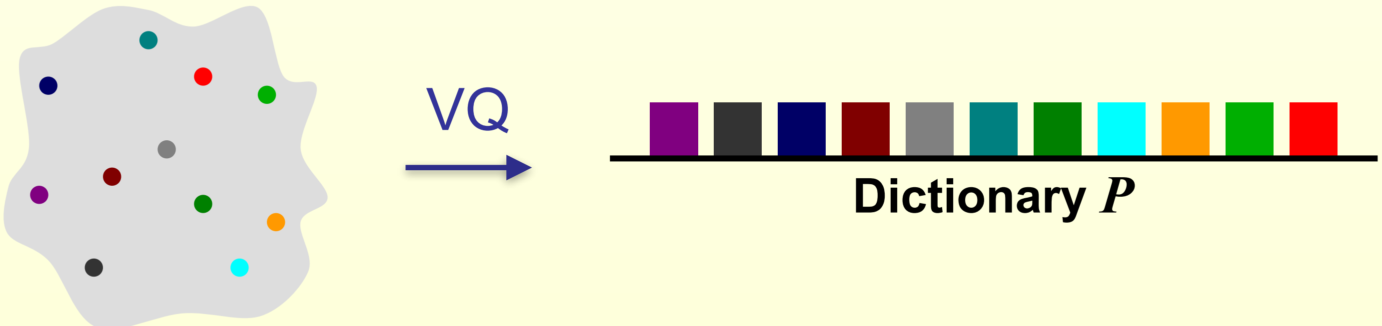
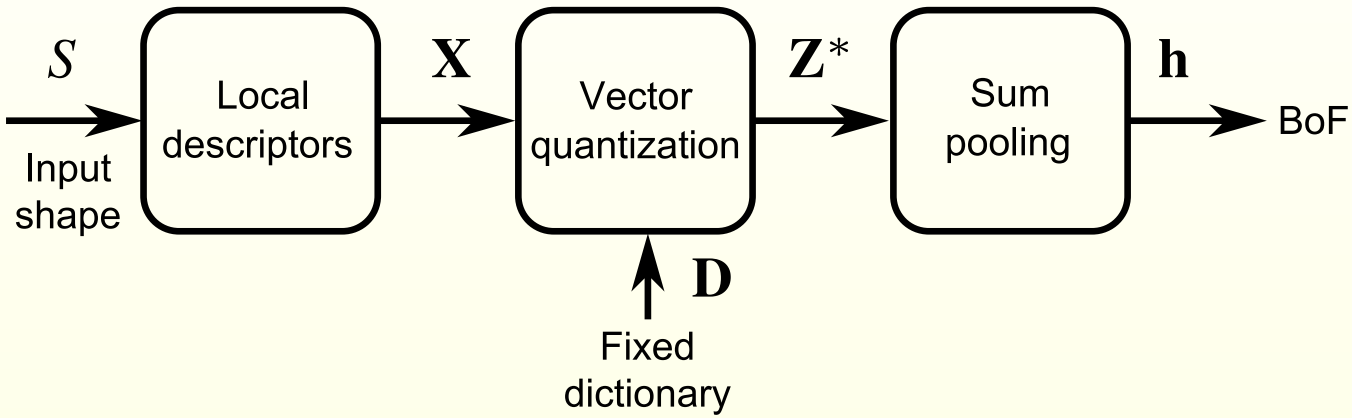


Image credit: R. Litman, M. Bronstein

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

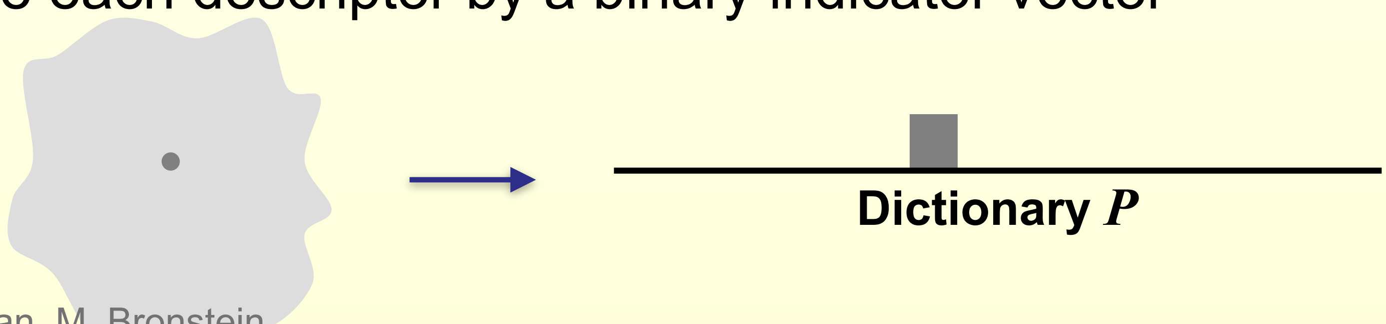
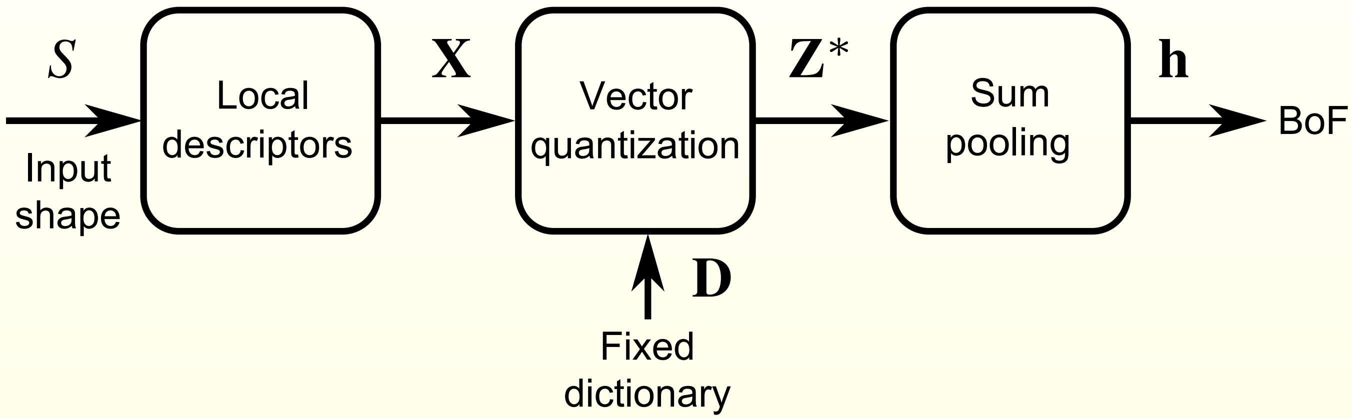


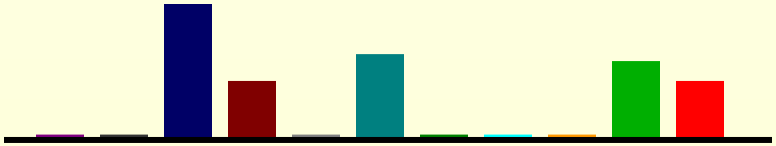
Image credit: R. Litman, M. Bronstein

46

BoF computation flow - simplified



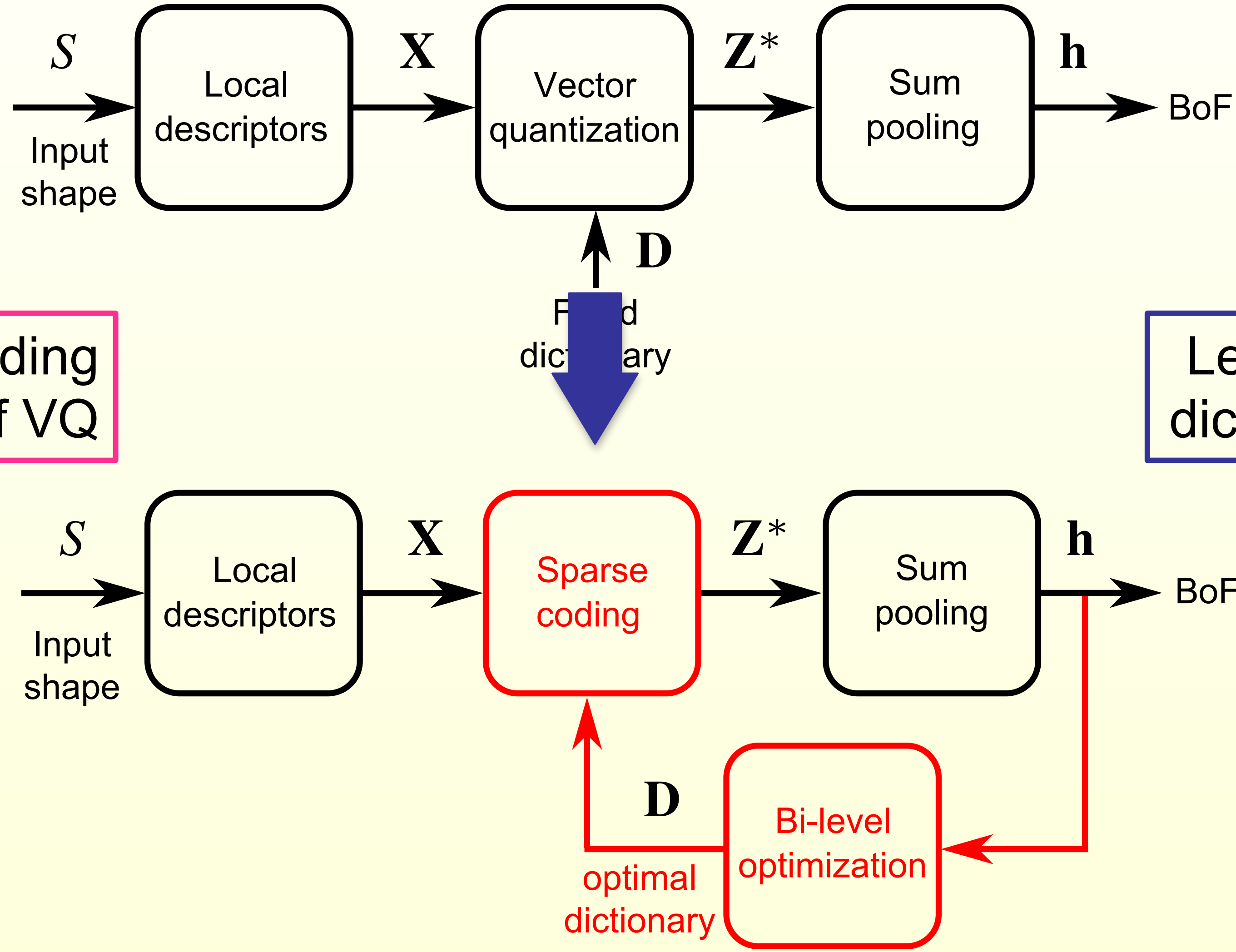
- Compute local descriptors - e.g., HKS
- Get a dictionary (= vocabulary) by vector quantization (VQ)
- Replace each descriptor by a binary indicator vector
- Sum up all indicator vector to obtain the BoF



BoF

Image credit: R. Litman, M. Bronstein

Suggested improvements

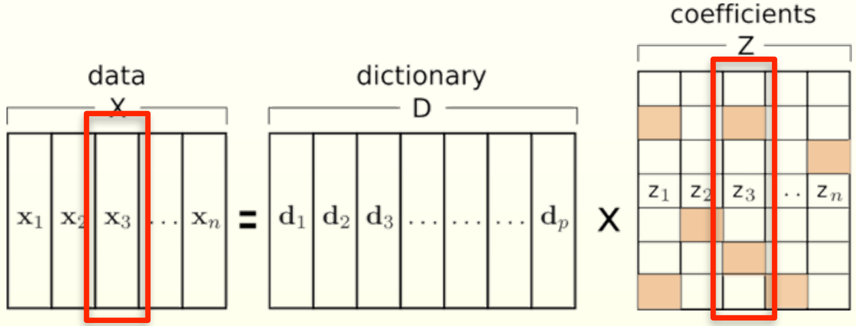


Space coding instead of VQ

Learned dictionary

Image credit: R. Litman

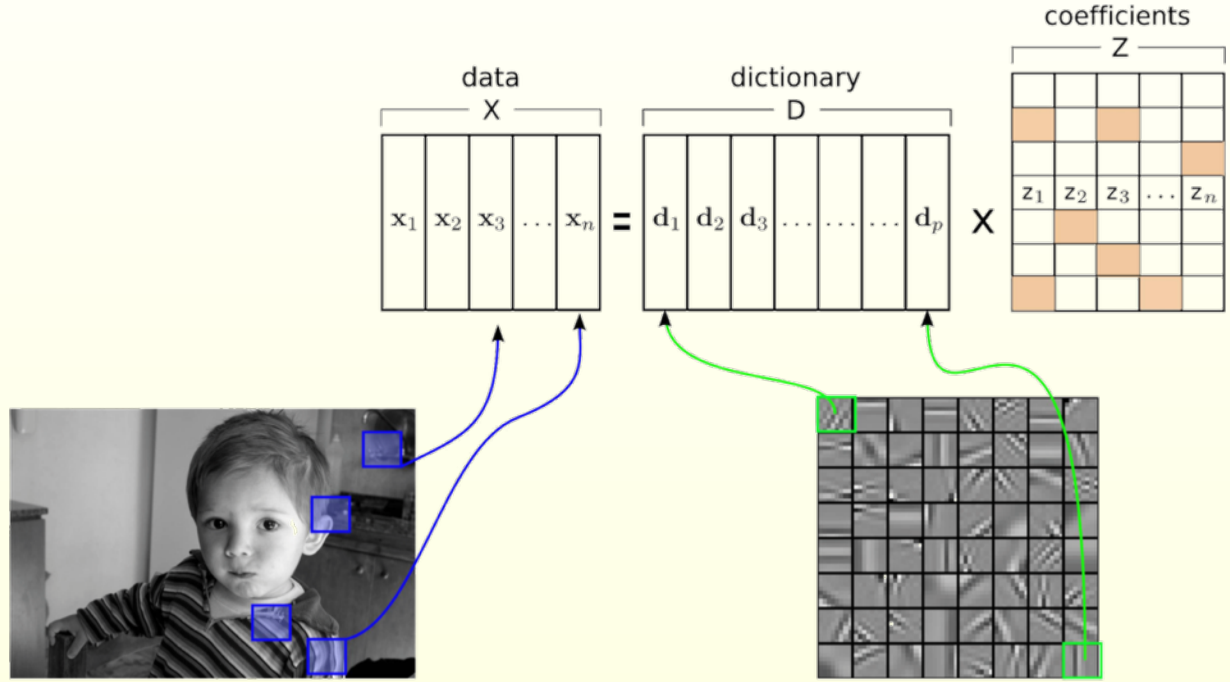
Sparse coding



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

Slide credit: R. Litman

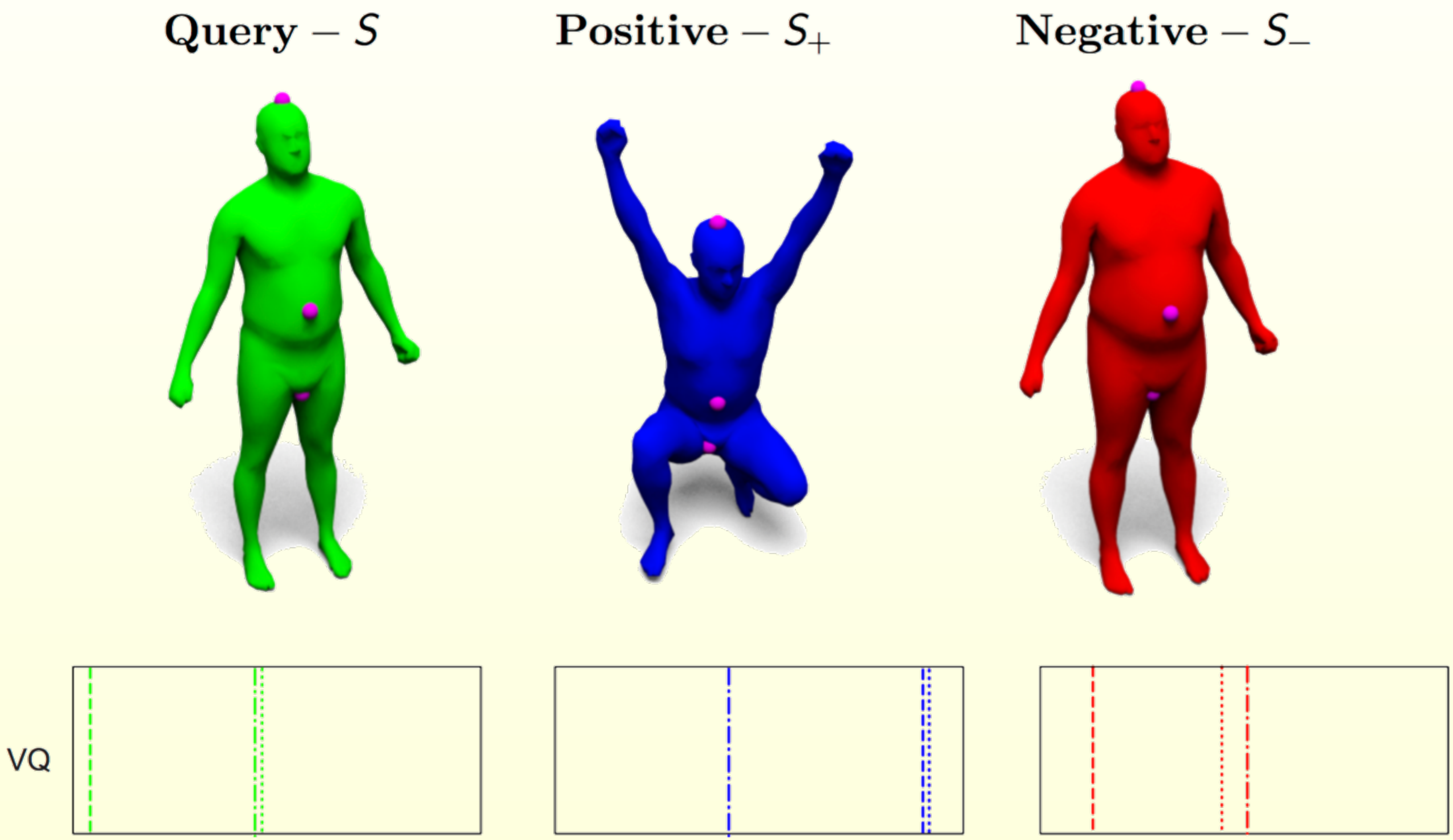
Sparse coding



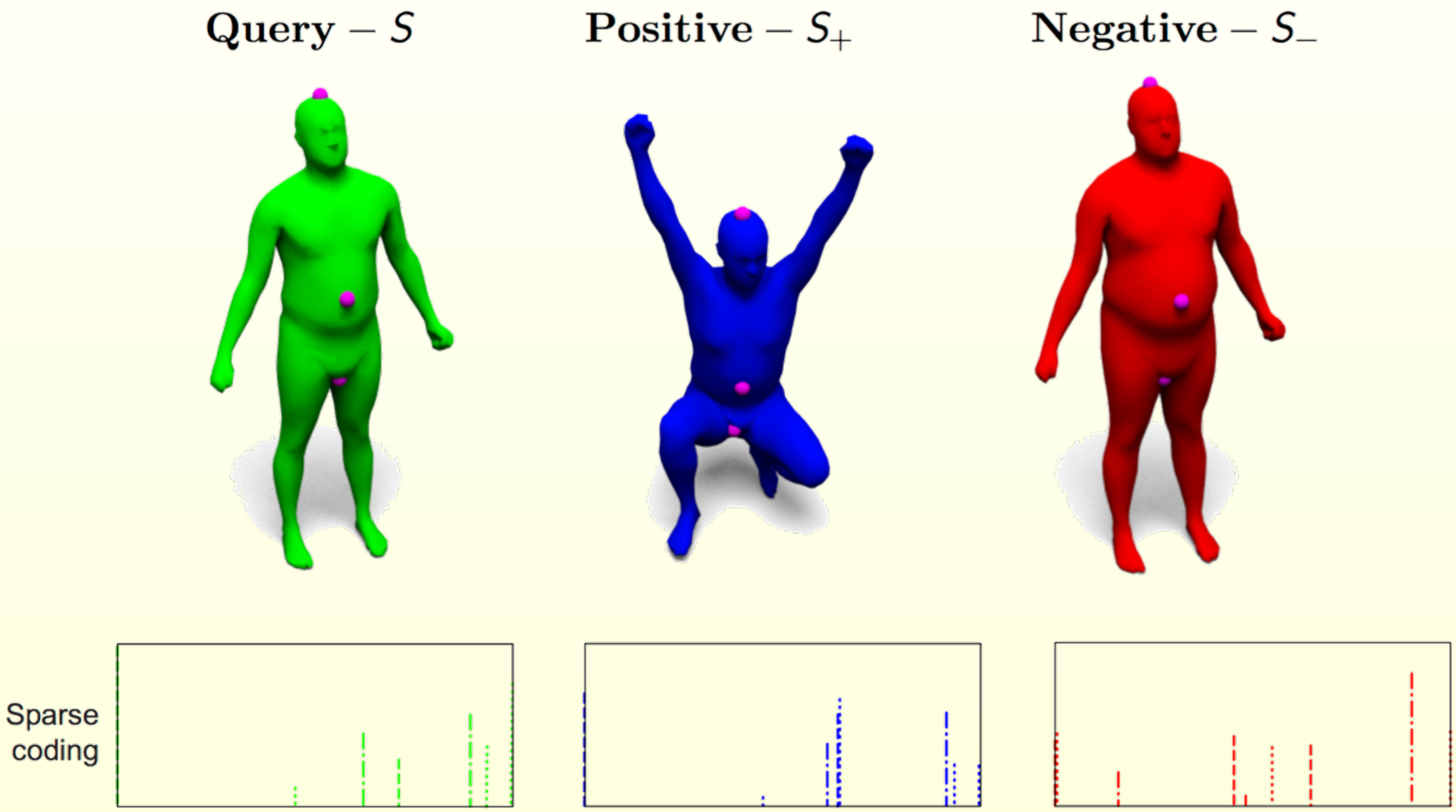
- Very successful when **dictionary D is learned from data**
- State-of-the-art in many applications.

Slide credit: R. Litman

Sparse coding for BoF - example

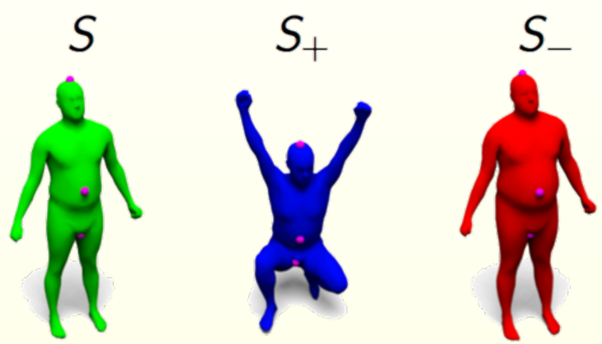


Sparse coding for BoF - example

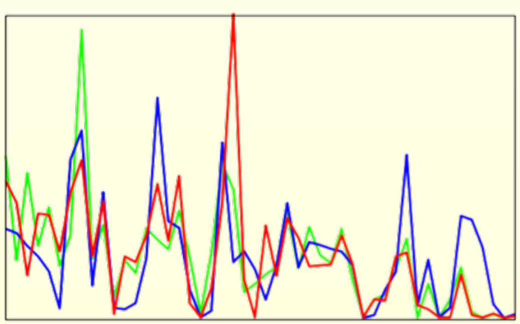


See the paper for implementation details

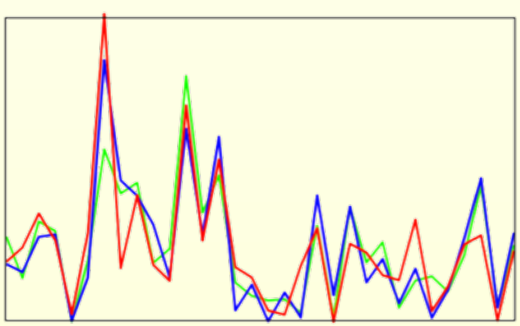
Descriptor pooling example



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



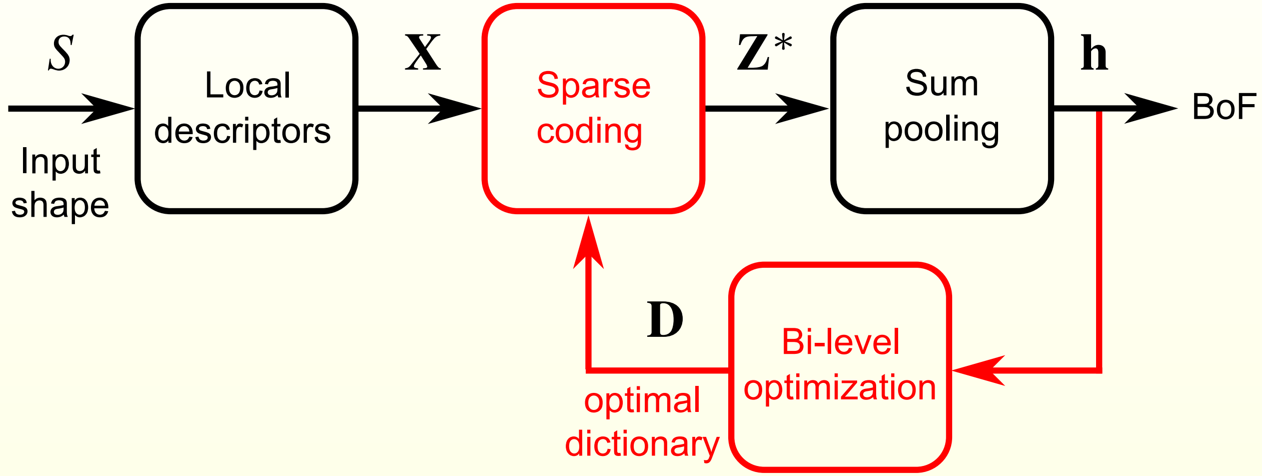
VQ
(unsupervised)



Sparse coding +
Unsupervised DL

Slide credit: R. Litman

Dictionary learning

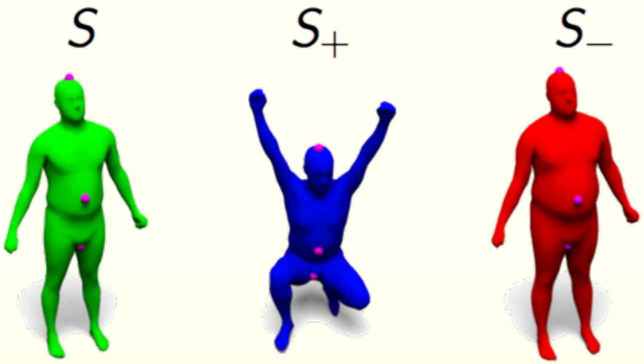


- Dictionary learning is task-driven - specified by a loss function
  - Input: labeled set of training shapes **S**
  - Each shape has an attached BoF *h(S)*
  - Optimize over dictionary **D** to minimize loss of training set

$$\min_{\mathbf{D}} \sum_{S \in \mathcal{S}} \ell(\mathbf{h})$$

Slide credit: R. Litman

Dictionary learning using *triplet* loss



Make $\|\mathbf{h}(\mathbf{Z}) - \mathbf{h}(\mathbf{Z}_+)\|$ small and $\|\mathbf{h}(\mathbf{Z}) - \mathbf{h}(\mathbf{Z}_-)\|$ larger (in comarison) by minimizing

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

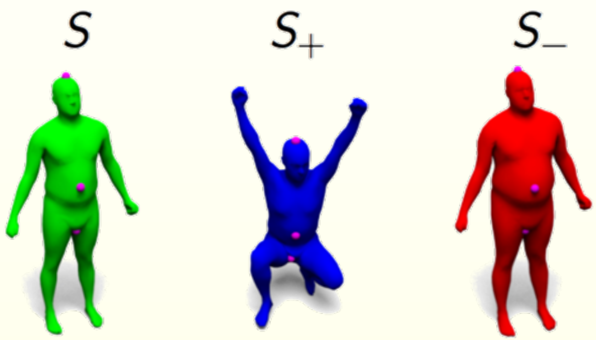
$$\ell_+(\mathbf{Z}, \mathbf{Z}_+) = \|\mathbf{h}(\mathbf{Z}) - \mathbf{h}(\mathbf{Z}_+)\|_1$$

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

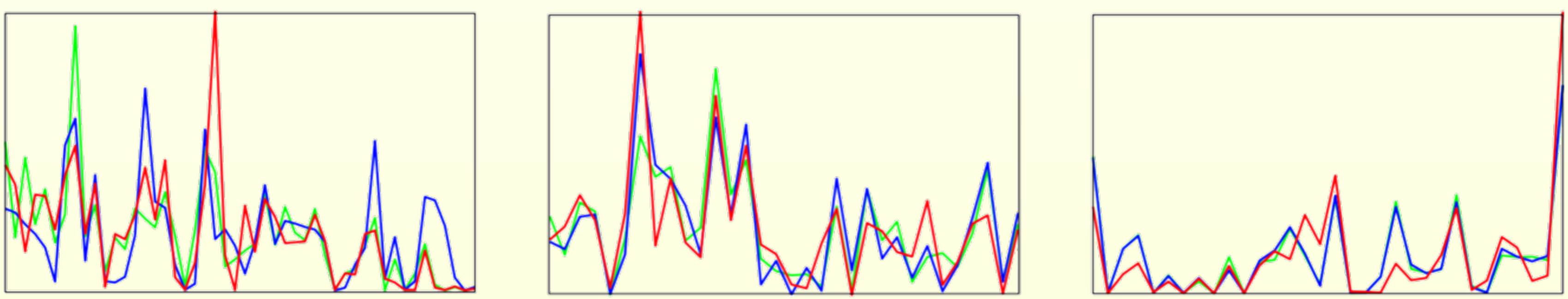
[Weinberger and Saul 2009]

Slide credit: R. Litman

BoF after dictionary learning



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



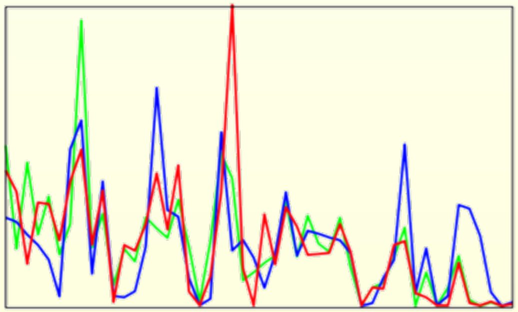
VQ (unsupervised) | Sparse coding + Unsupervised DL | Sparse coding + Supervised DL

Slide credit: R. Litman

BoF after dictionary learning

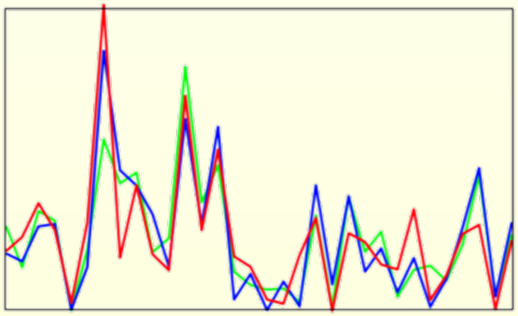
Distance ratios $\frac{\|\mathbf{h}(\mathbf{Z}^*)-\mathbf{h}(\mathbf{Z}^*_+)\|_1}{\|\mathbf{h}(\mathbf{Z}^*)-\mathbf{h}(\mathbf{Z}^*_-)\|_1}$

6.26 $\updownarrow$



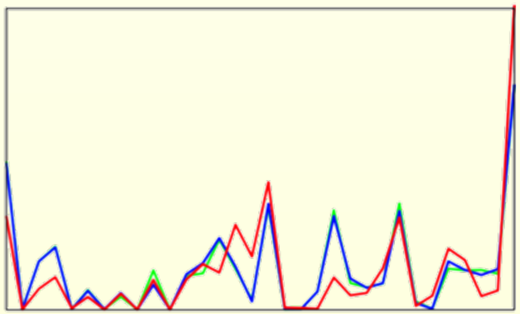
VQ
(unsupervised)

3.53 $\updownarrow$



Sparse coding +
Unsupervised DL

0.98 $\updownarrow$

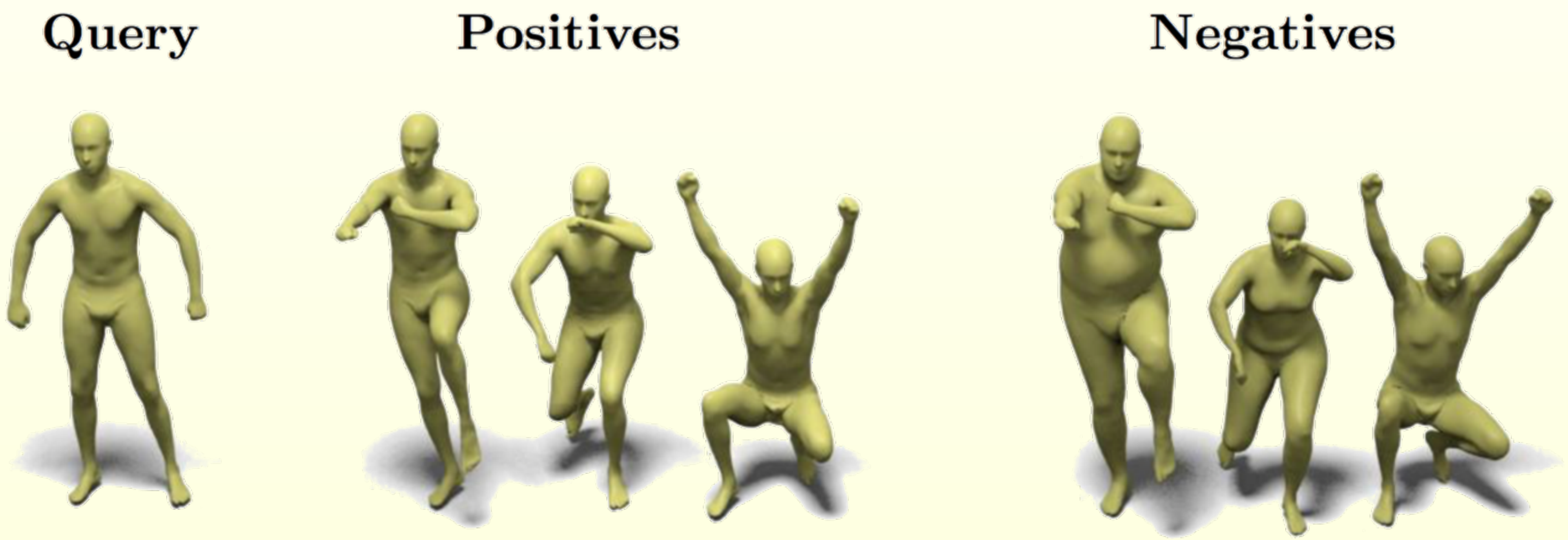


Sparse coding +
Supervised DL

Slide credit: R. Litman

SHREC'14 Dataset

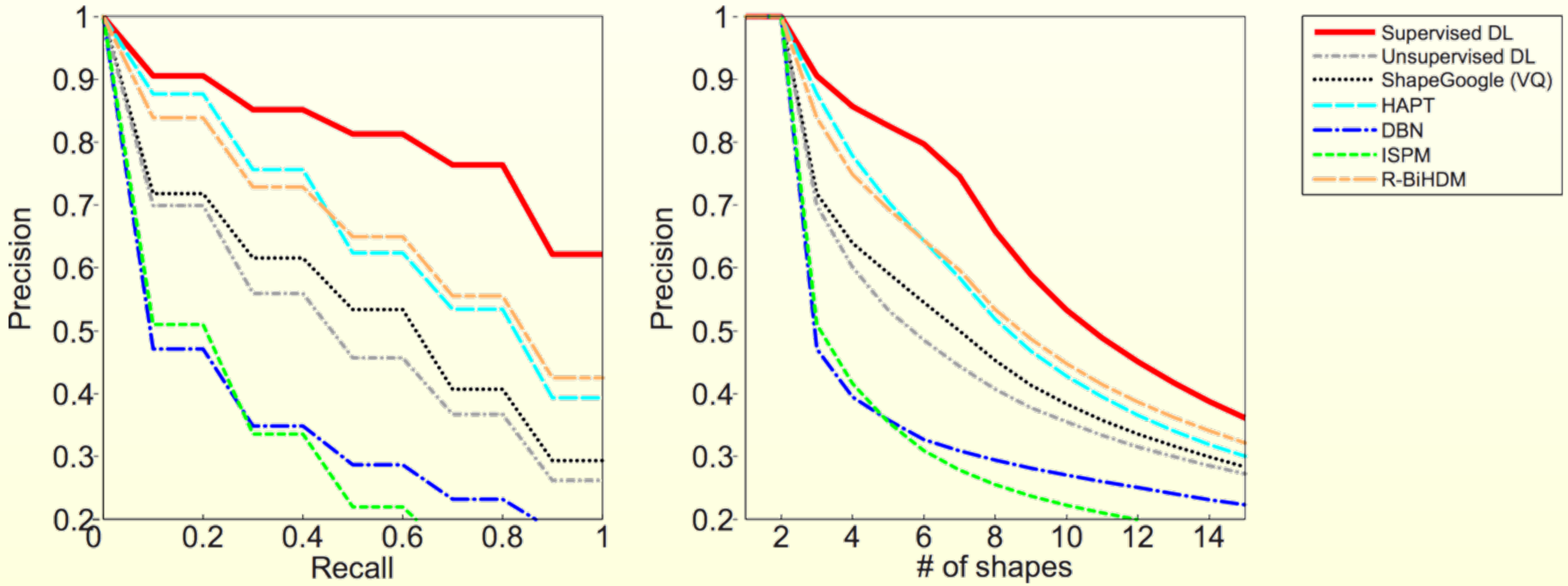
- Goal: given a human model, detect this model in other poses



Slide credit: R. Litman

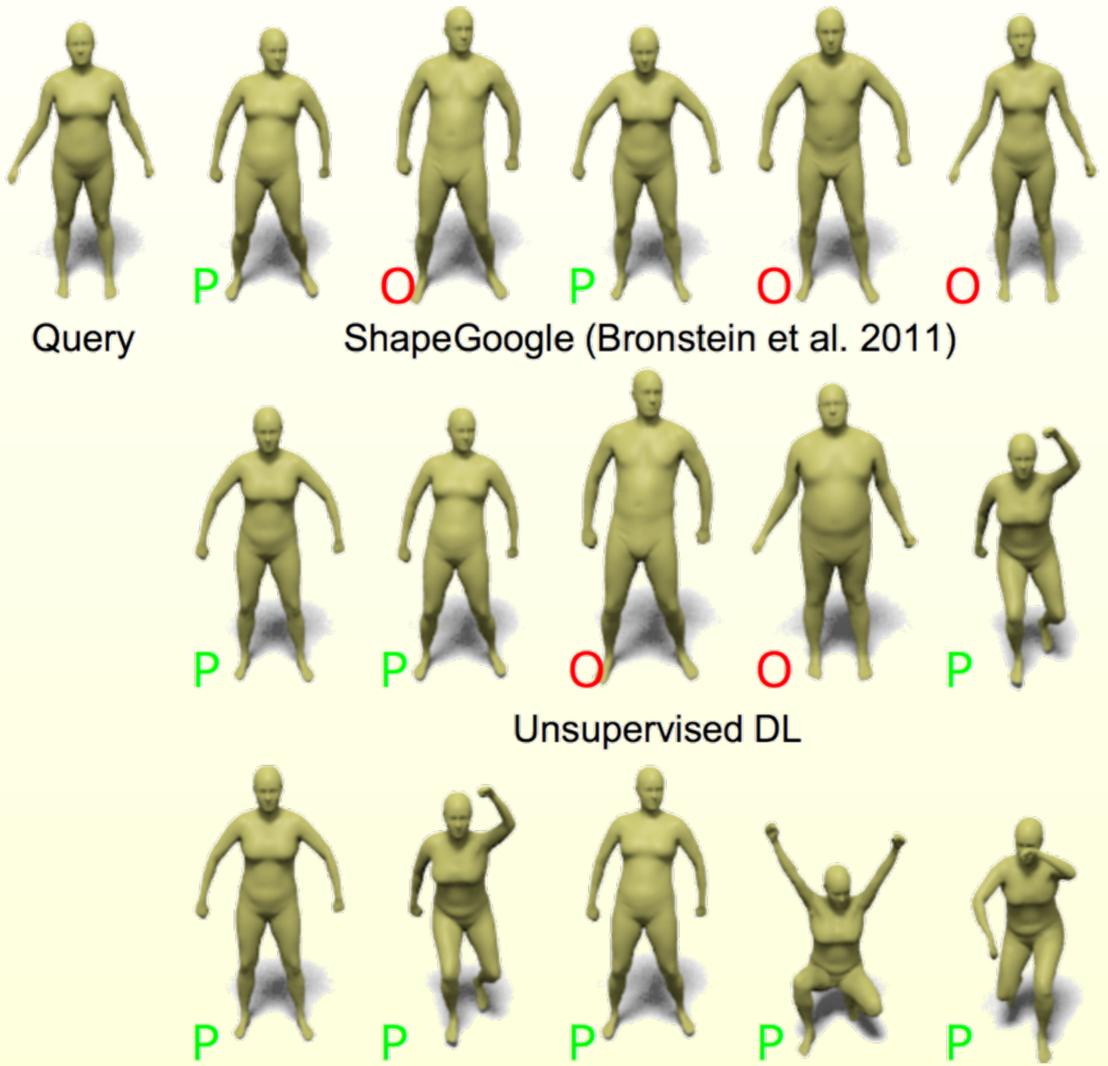
SHREC'14 results

- Goal: given a human model, detect this model in other poses



Slide credit: R. Litman

Query example - nearest neighbor



Slide credit: R. Litman

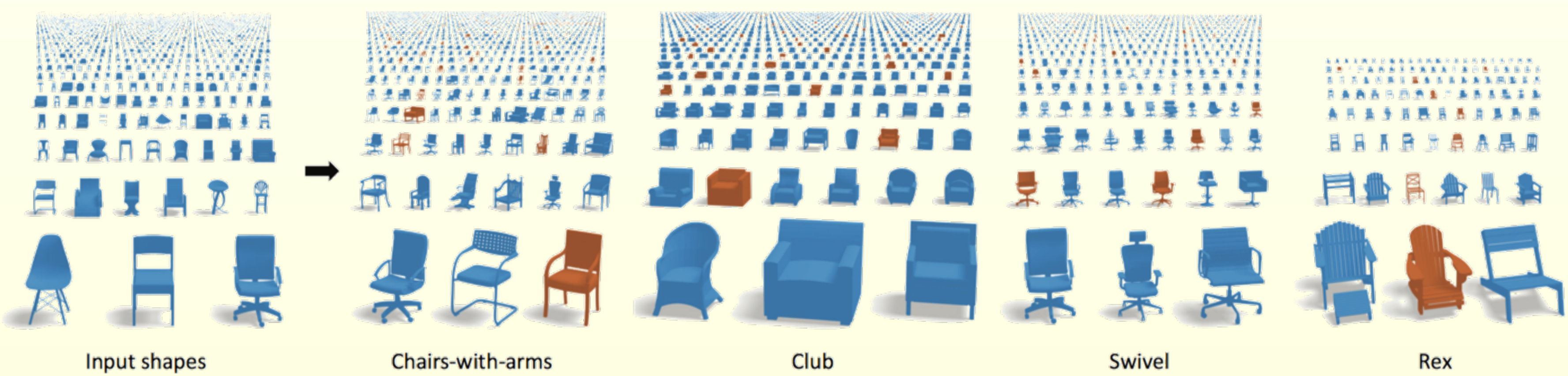
Questions so far?

Fine-grained shape classification

- Global shape descriptors - work well for shapes from different classes
- Next: a method for fine-grained <u>sub-class classification</u> 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

  label = shape class

- Subtle geometric differences

- Goal: produce labels for all shapes in collection

Swivel

Image credit: H. Su

Approach overview

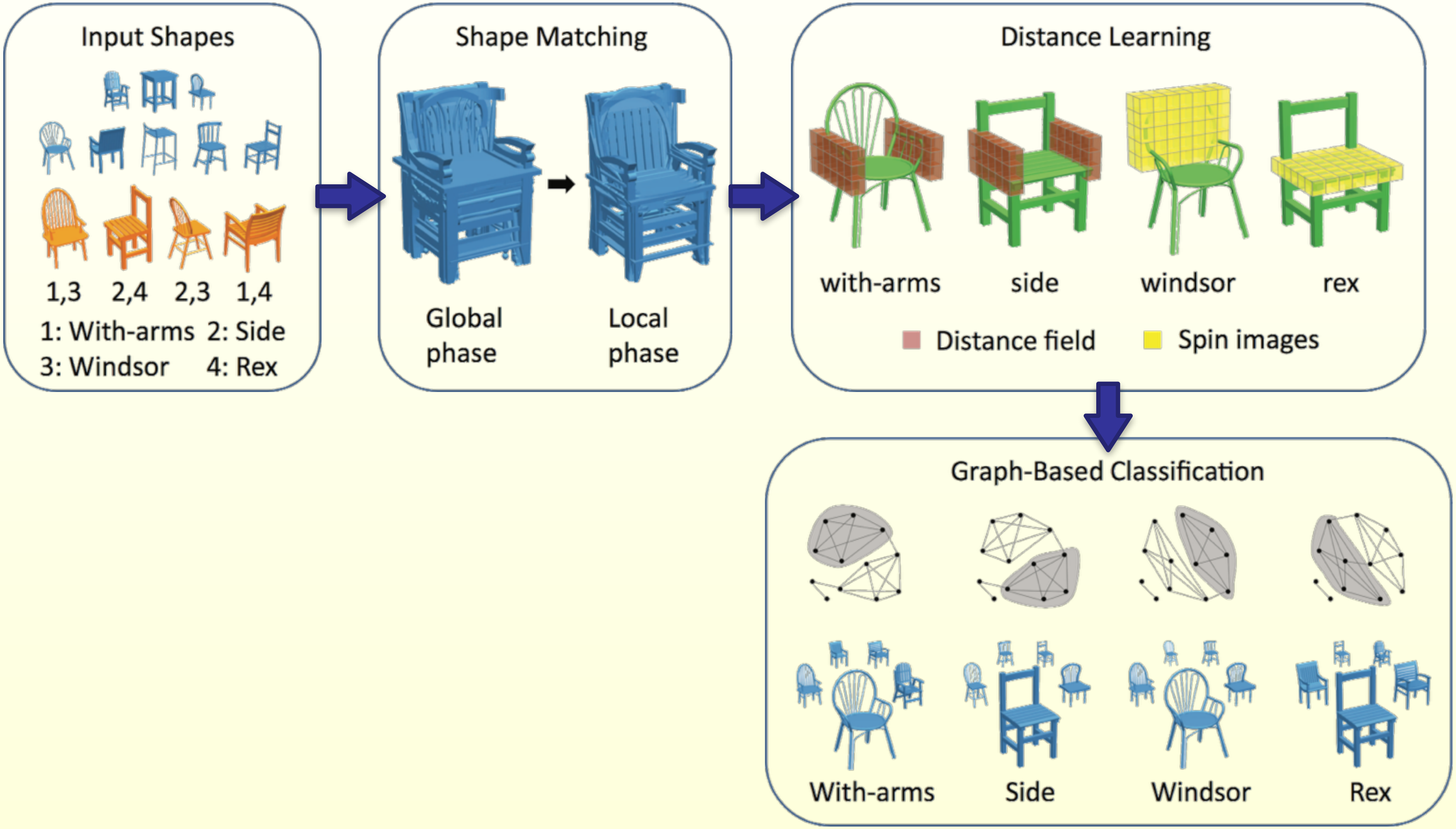


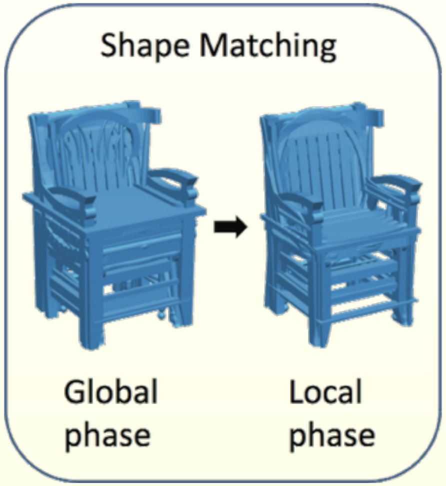
Image credit: H. Su

Shape matching

- <u>Global phase</u>: 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

- <u>Local phase</u>: local non-rigid registration using free-form deformation

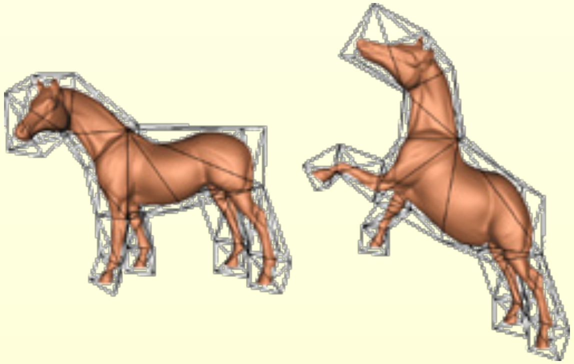
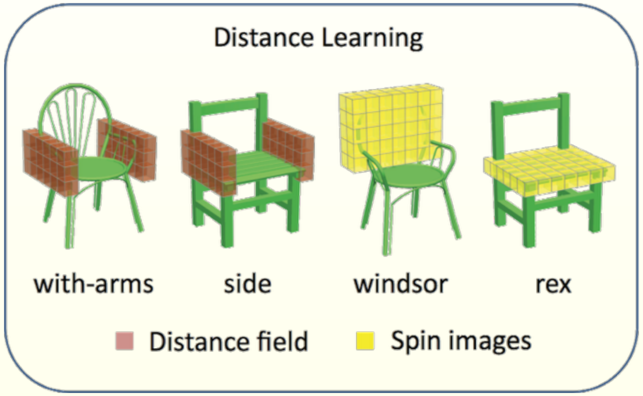


Image credit: H. Su

Distance learning



- Learn distance metric using the aligned labeled shapes - per class
- Distance between pair of shapes = parameterized using fixed-size voxels

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

Distance per voxel

Learned coefficients

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

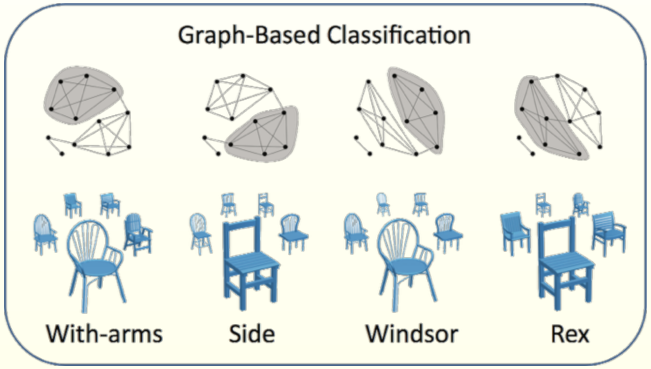
Image credit: H. Su

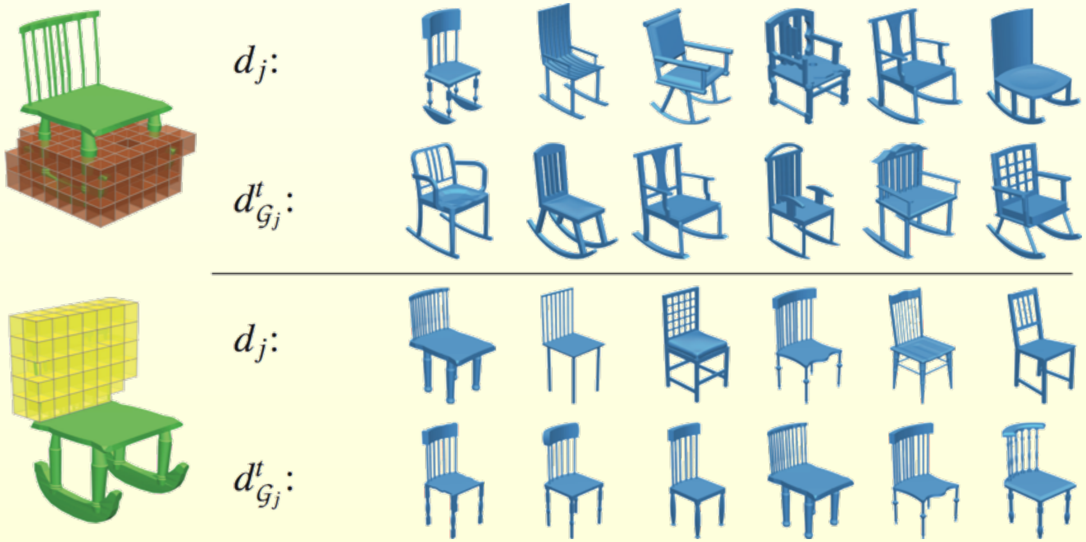
Learned metric - illustration

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





Learned metric

Diffusion distance

Learned metric

Diffusion distance

Image credit: H. Su

Labeling results

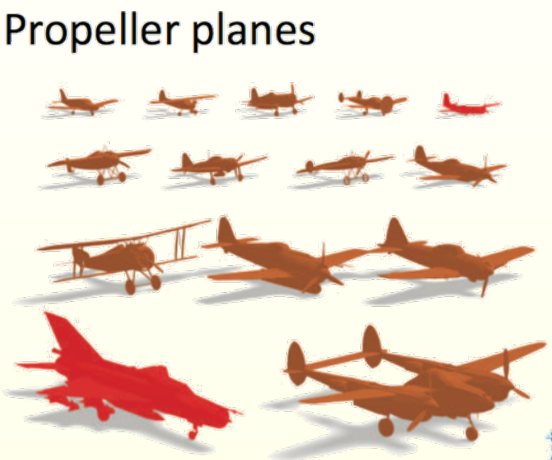


Image credit: H. Su

Comparison to linear classifier result

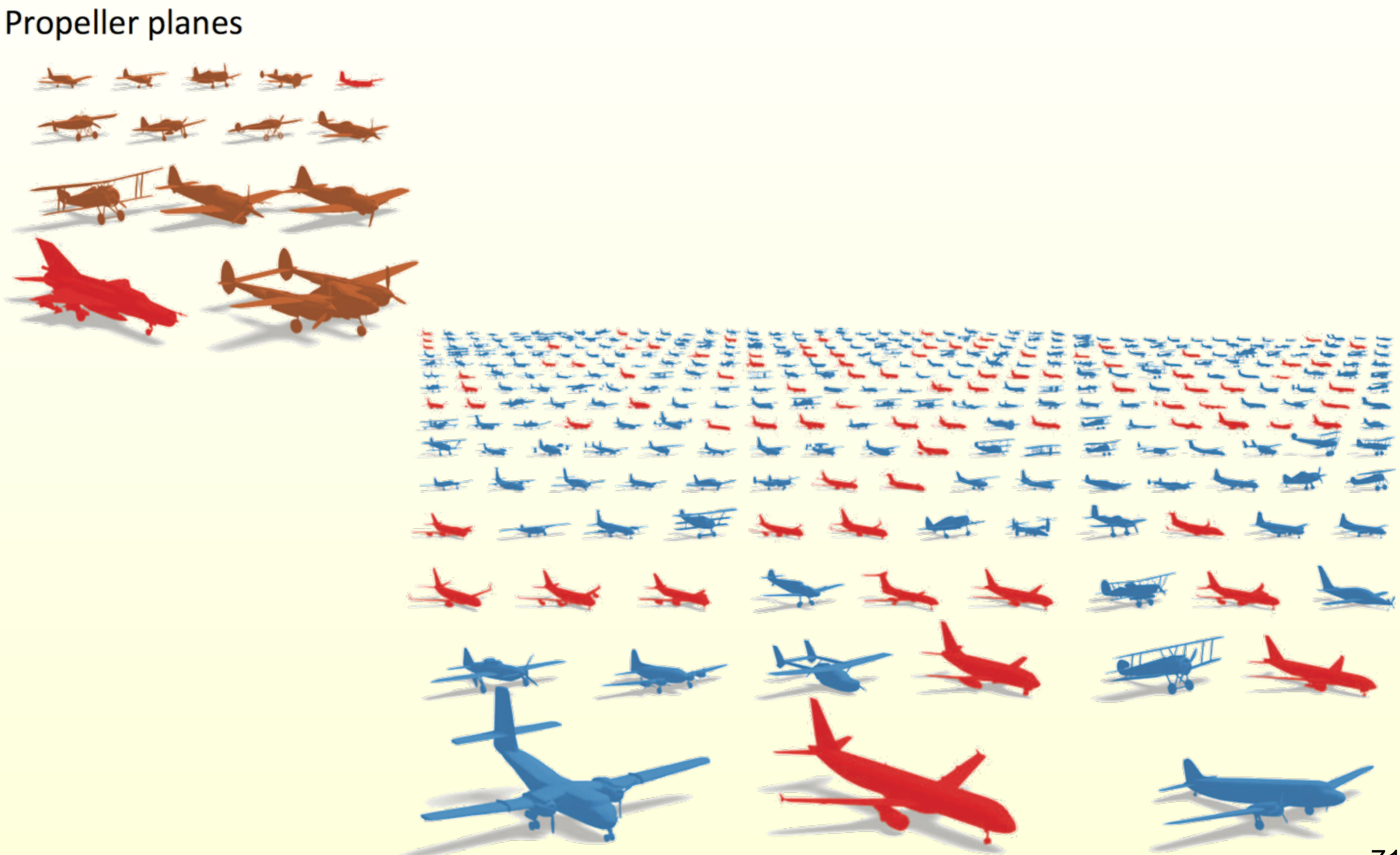
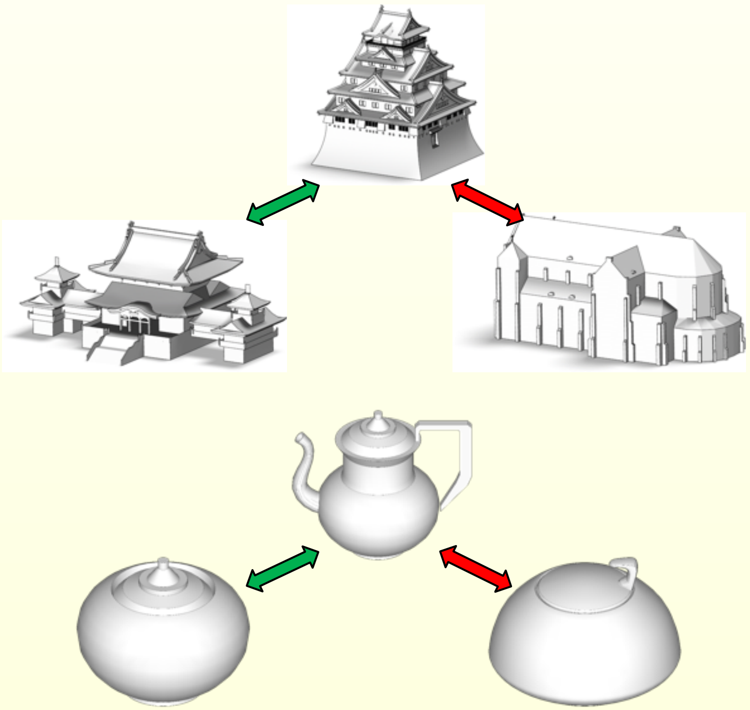


Image credit: H. Su

Questions?

Style similarity

- Two papers presented in Siggraph 2015



Lun et al. 2015

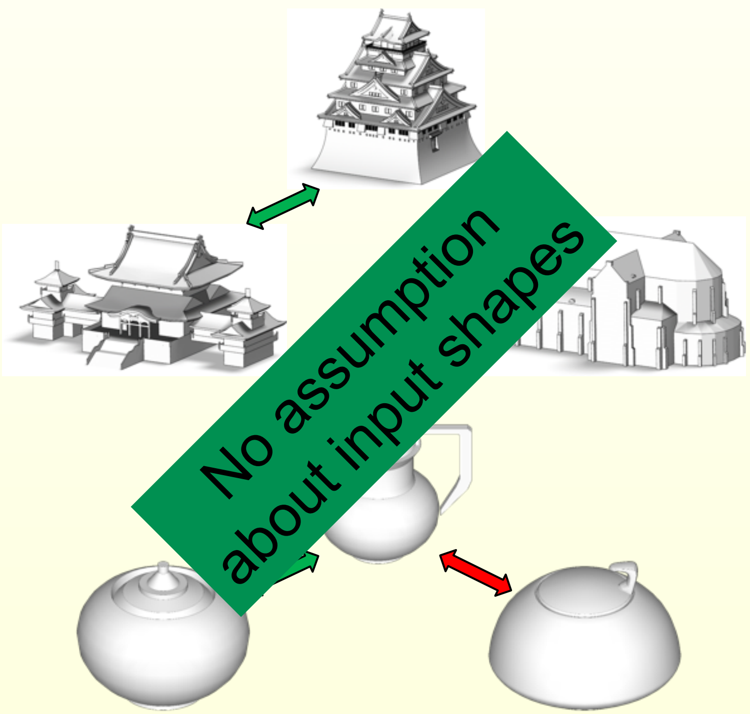
Stylistically incompatible

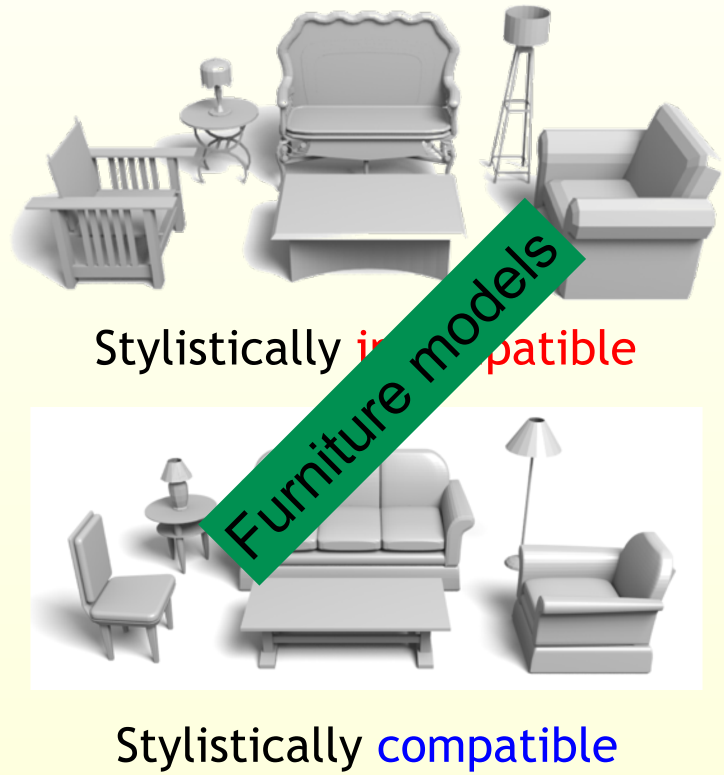
Stylistically compatible

Liu et al. 2015

Style similarity

- Two papers presented in Siggraph 2015





Lun et al. 2015

Liu et al. 2015

Style compatibility for furniture models

Stylistically incompatible

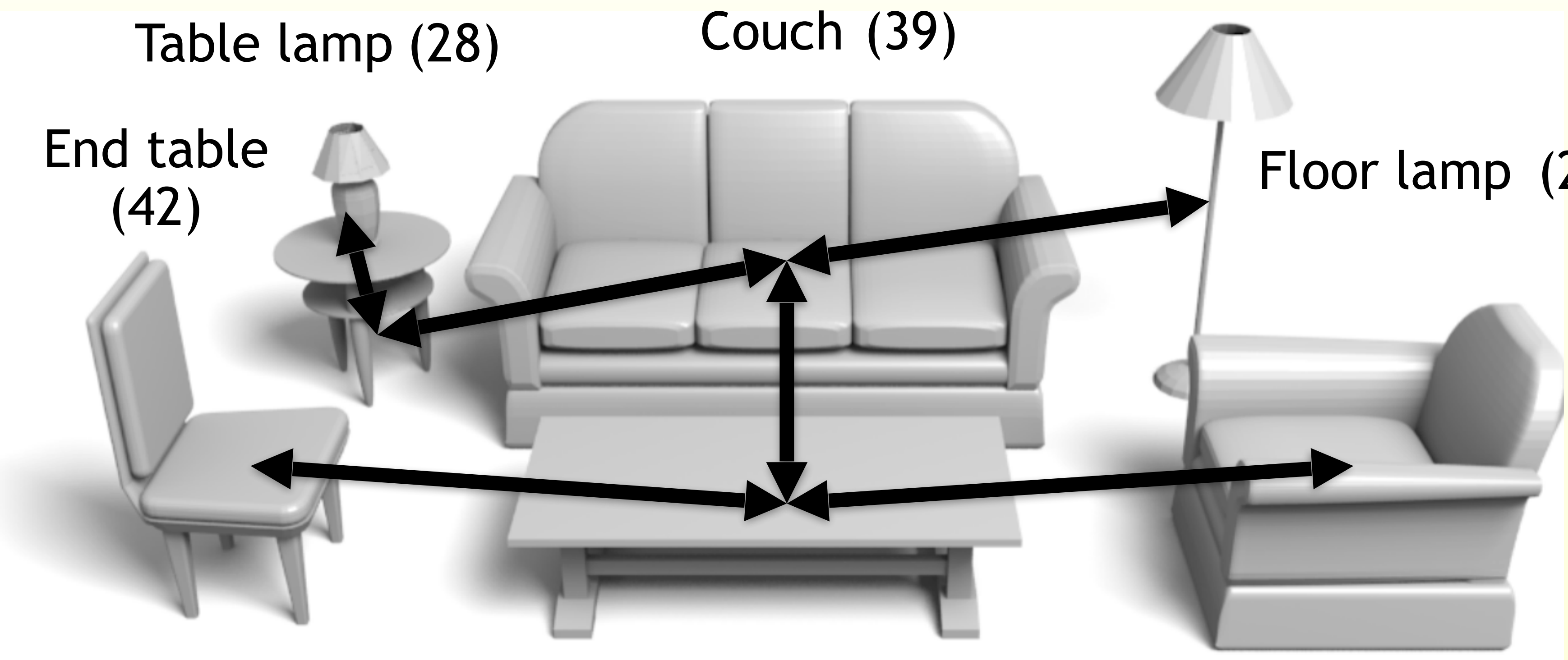
Stylistically compatible

Liu et al. 2015

Slide credit: T. Liu

Style compatibility for furniture models

- Crowdsource compatibility between <u>pairs</u> of models

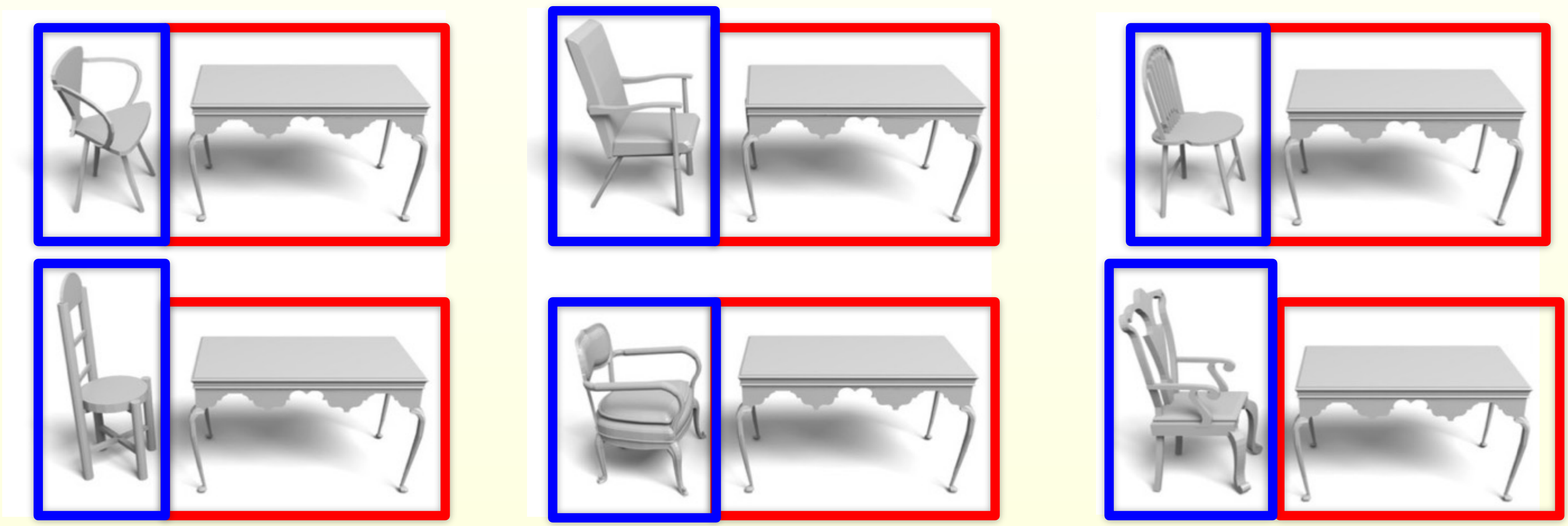


Living room

Slide credit: T. Liu

Crowdsourcing compatibility preferences

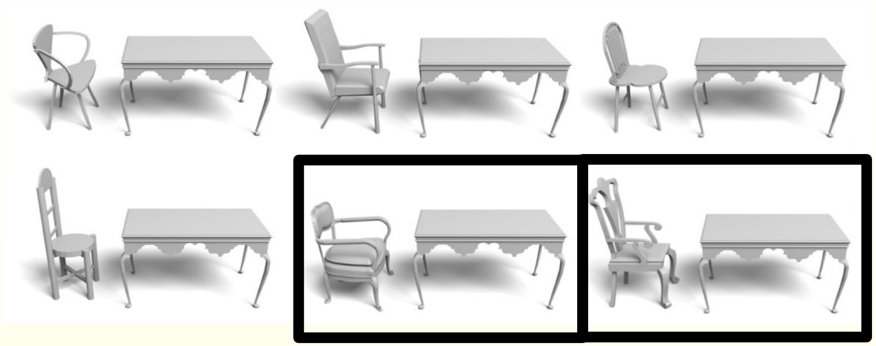
Design of user study [Wilber et al. 2014]



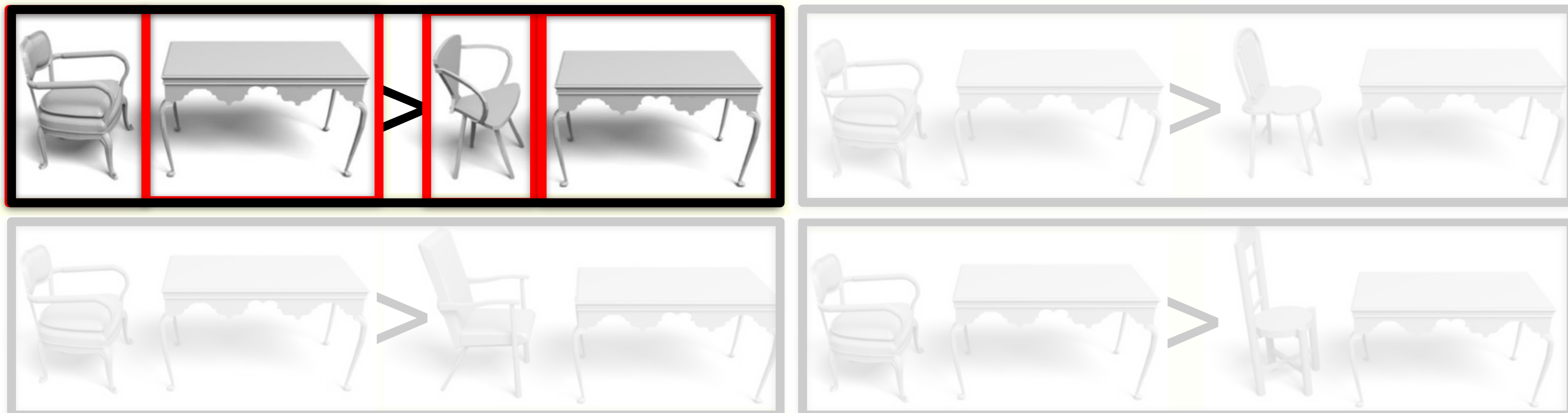
Please select the two most compatible pairs

Crowdsourcing compatibility preferences

Rater's selection



Converted into 8 triplets



and 4 more triplets …

Crowdsourcing compatibility preferences

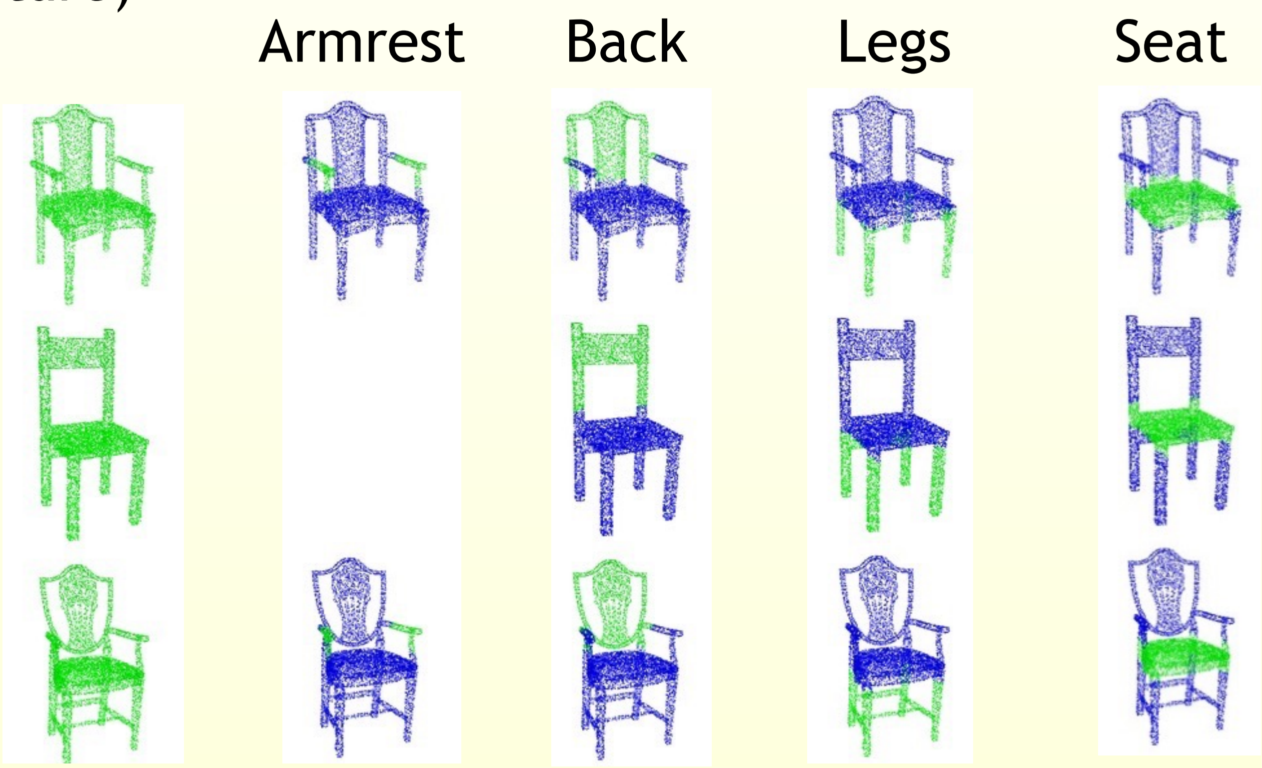
Living room

Dining room

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

Part-aware geometric features

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



Slide credit: T. Liu

Part-aware geometric features

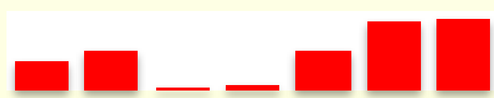
Step 2: Computing geometry features for each part

Back

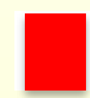
Curvature histogram

Bounding box dimensions

Shape diameter histogram



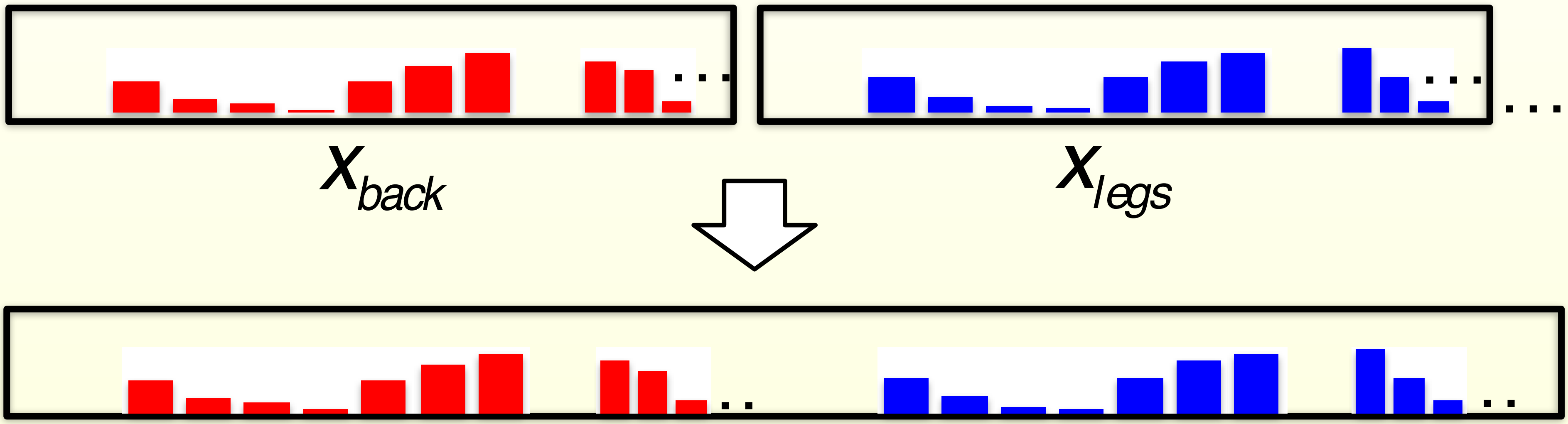
Normalized surface area



Slide credit: T. Liu

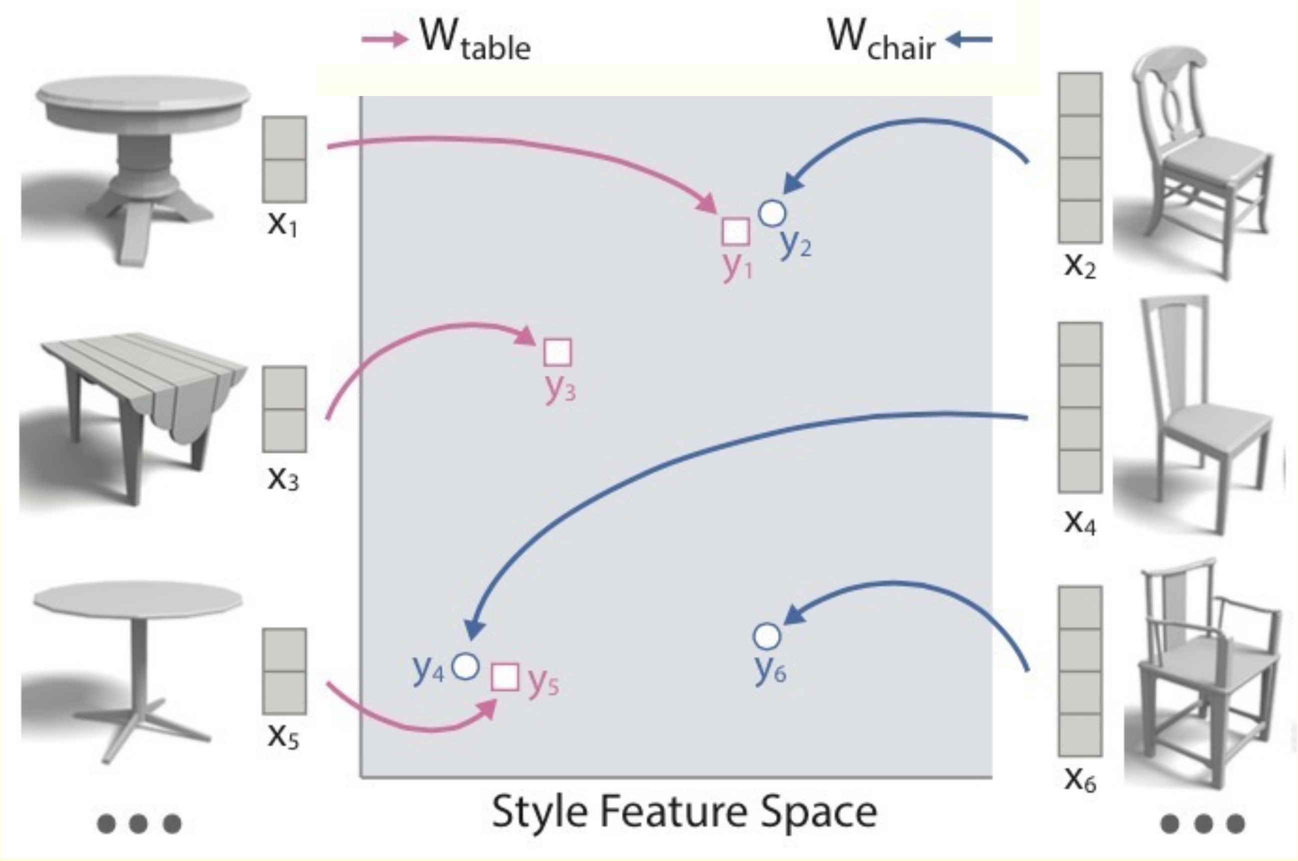
Part-aware geometric features

Step 3: Concatenating features of all parts



$$X = [X_{back}, X_{legs}, \ldots]$$

Learning object-class specific embeddings



Slide credit: T. Liu

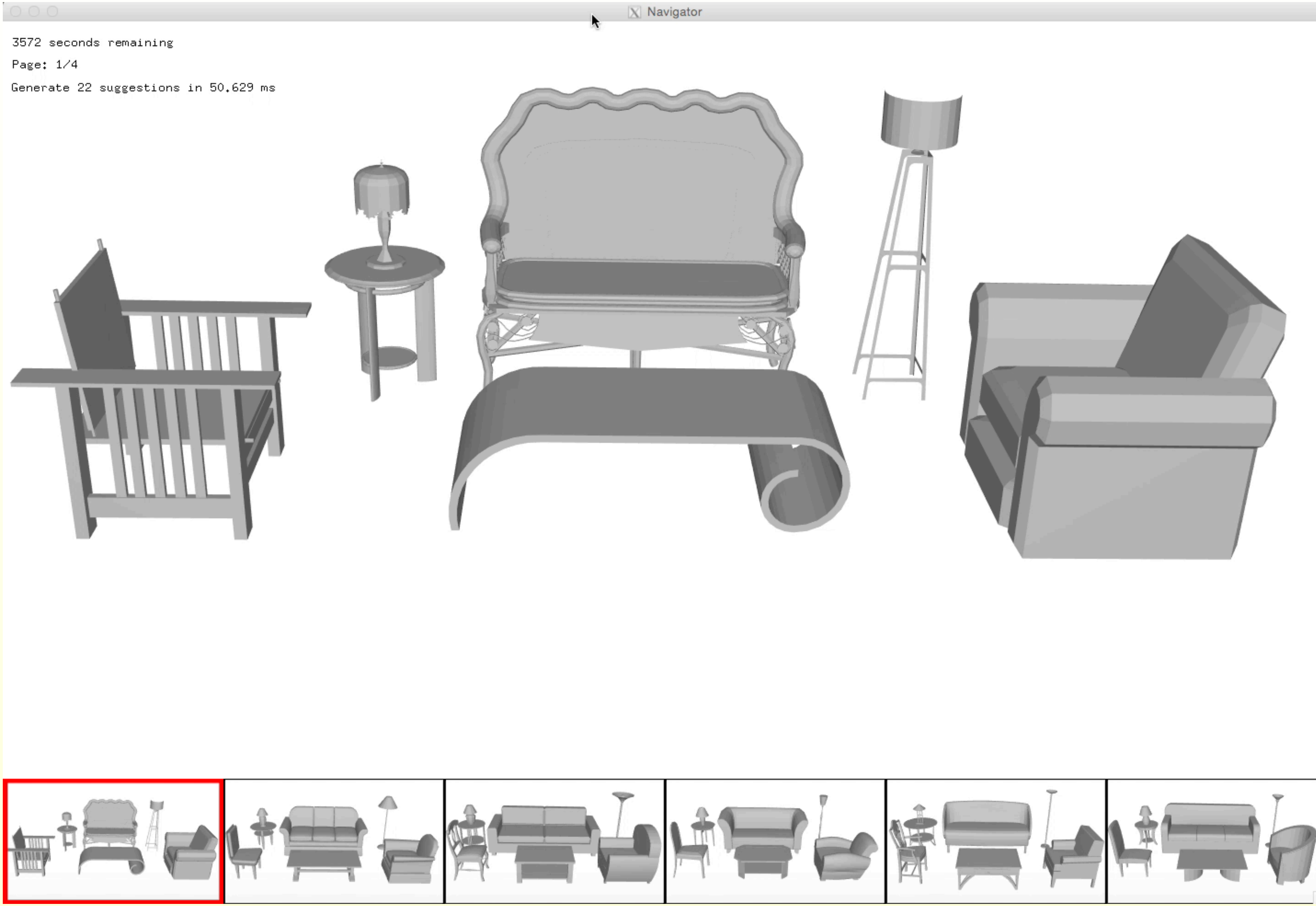
Style-aware shape retrieval

Query model

Dining chair

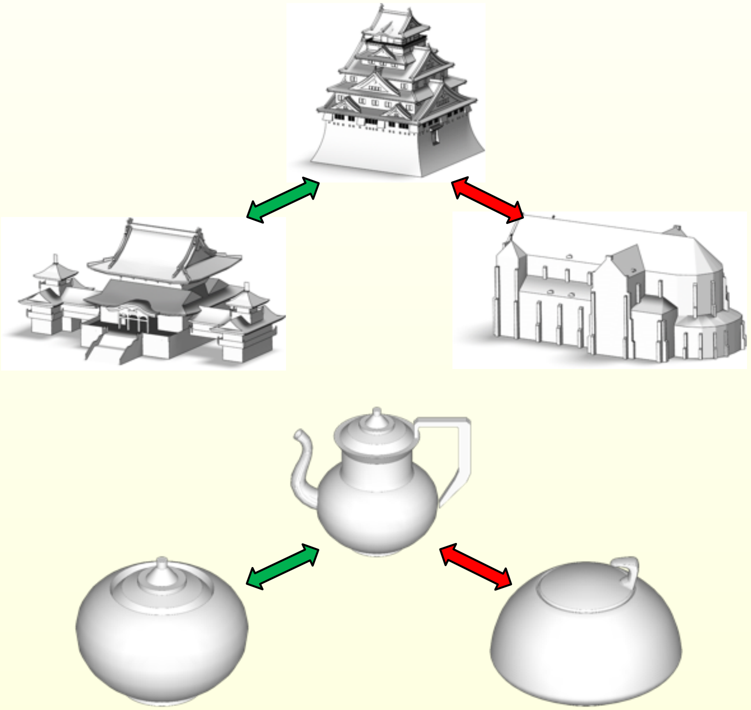
Most incompatible chairs

Slide credit: T. Liu



Slide credit: T. Liu

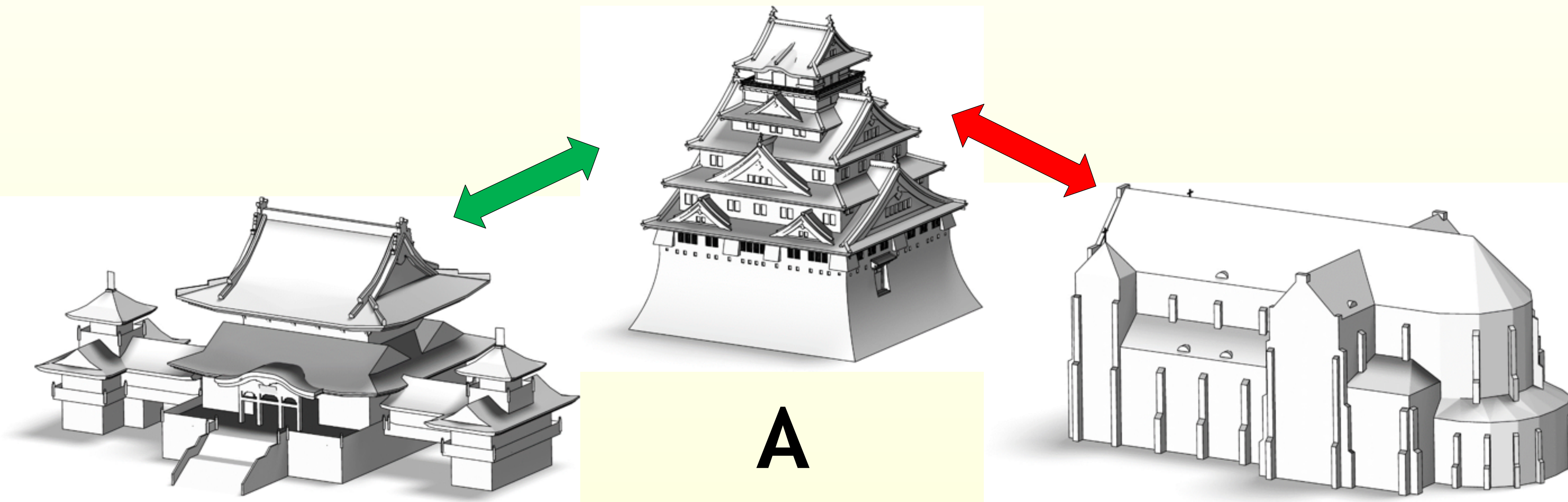
Learning perceptual style similarity



Lun et al. 2015

Learning perceptual style similarity

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



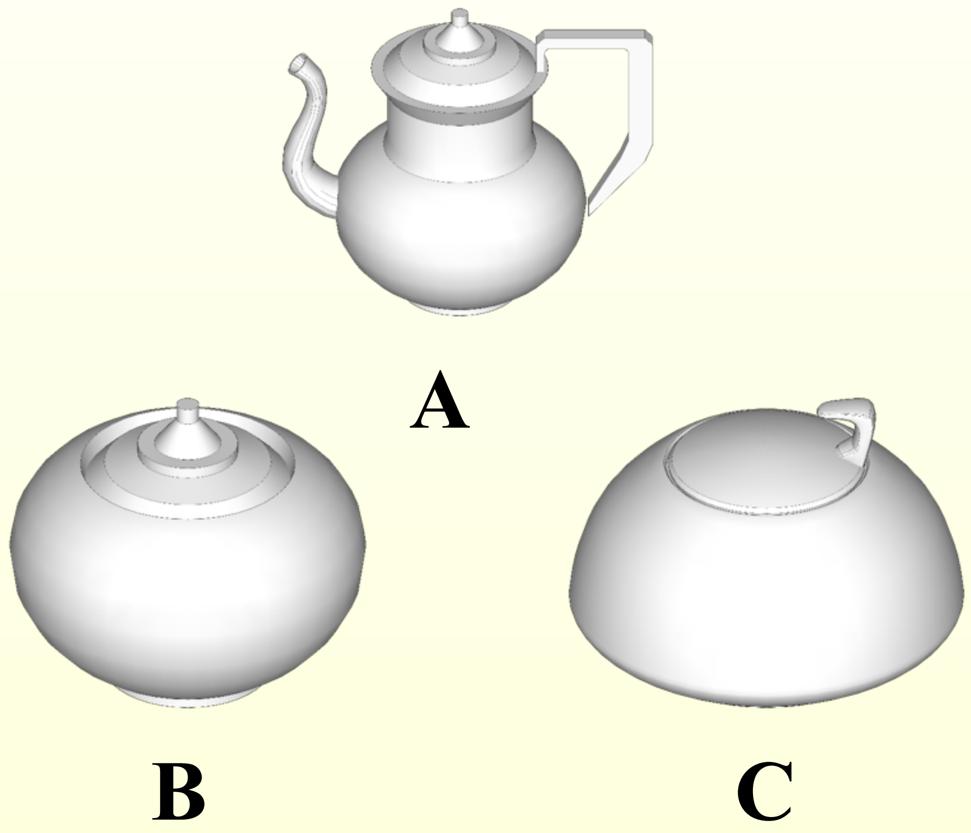
A

B

C

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**

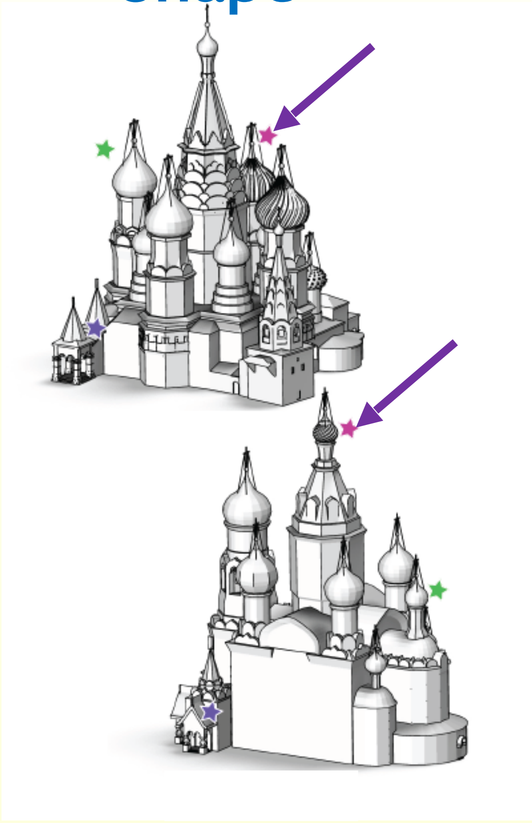
(iii) **Both**

(iv) **Neither**

Slide credit: Z. Lun

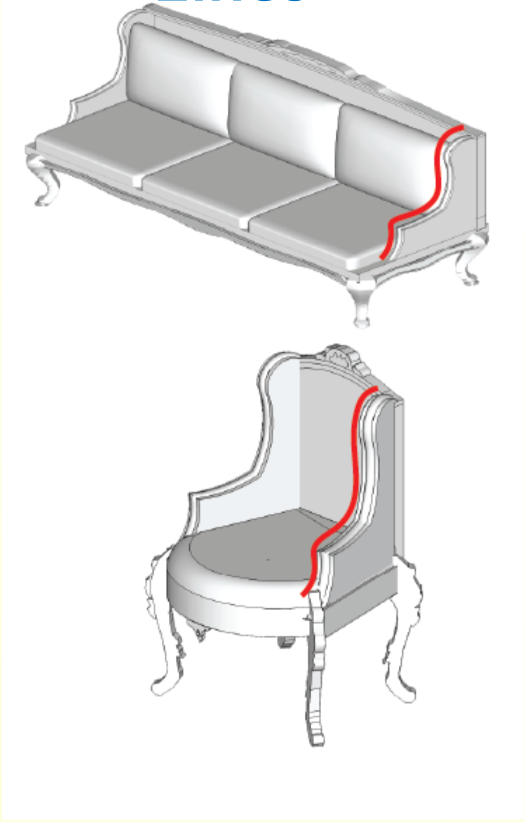
Geometric criteria for element similarity

Shape



Proportions

Lines



- Style-related elements are frequently designed to be distinct

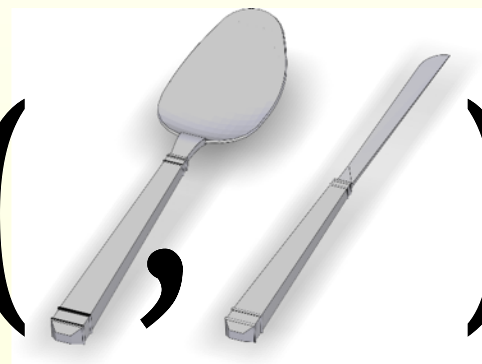
…

Slide credit: Z. Lun

Algorithm for measuring style similarity

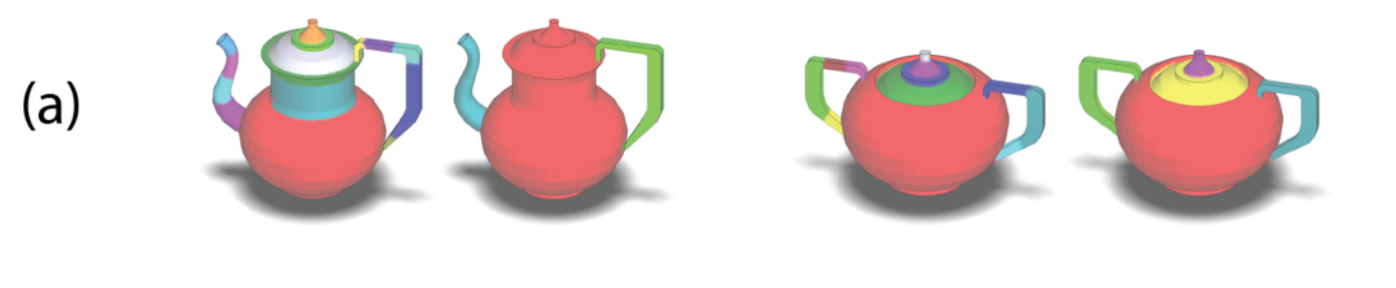
Input: a pair of shapes

Output: a measure of style dissimilarity (distance)

D()=?

Slide credit: Z. Lun

Extraction of matching elements

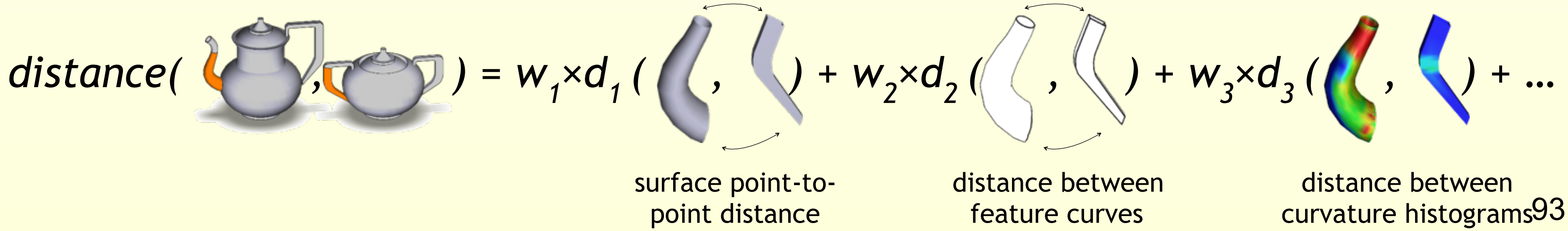


- **Multi-scale** segmentation
- Patches as **initial seeds** to detect elements

Slide credit: Z. Lun

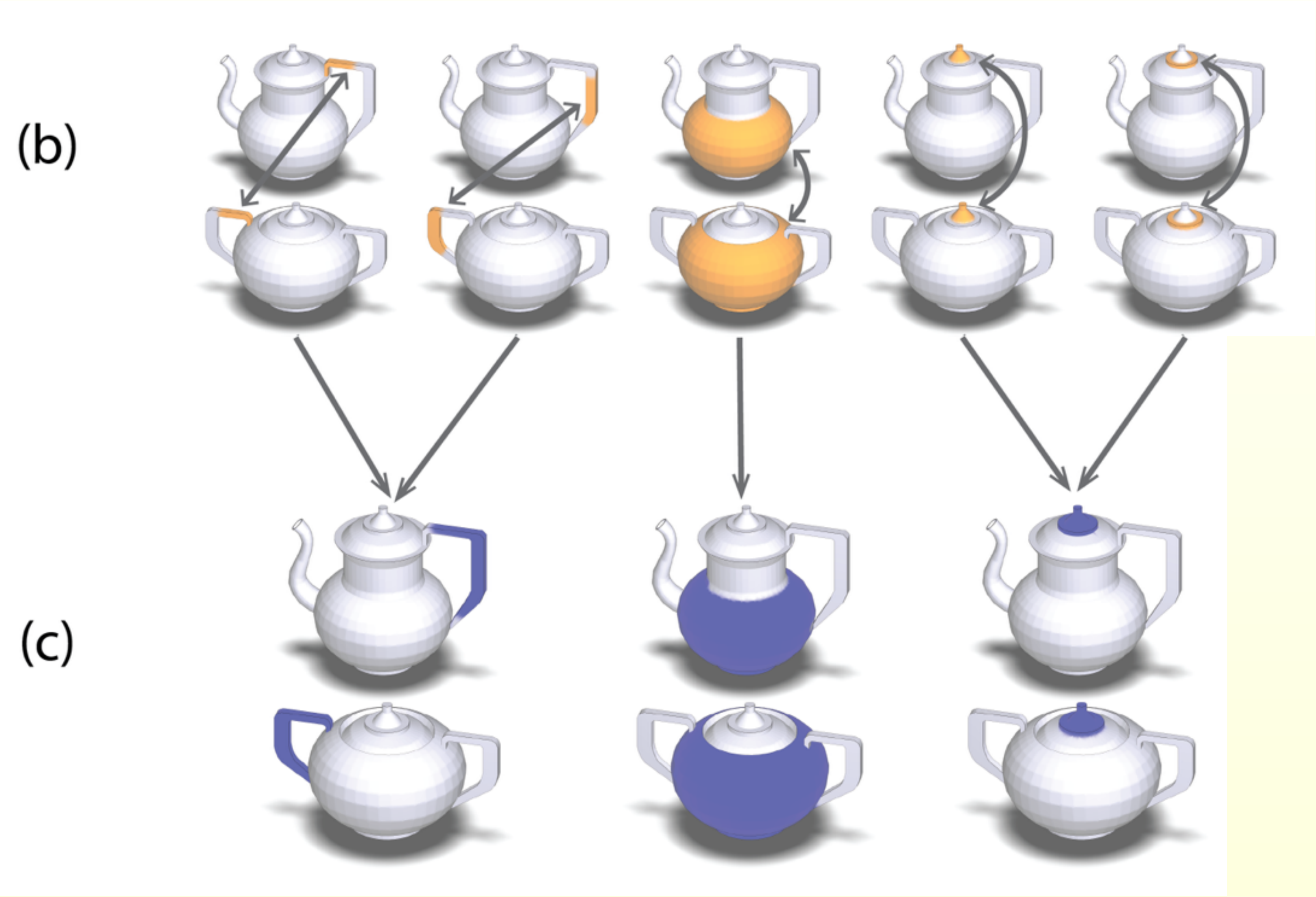
Extraction of matching elements

Align with **affine transformation**, measure patch **stylistic similarity**:



Slide credit: Z. Lun

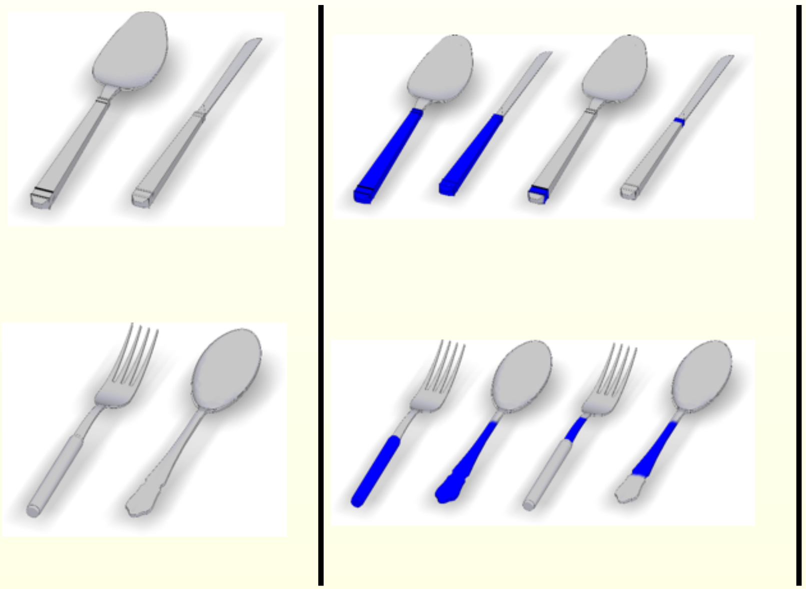
Extraction of matching elements



Group patches into **matching elements**

Slide credit: Z. Lun

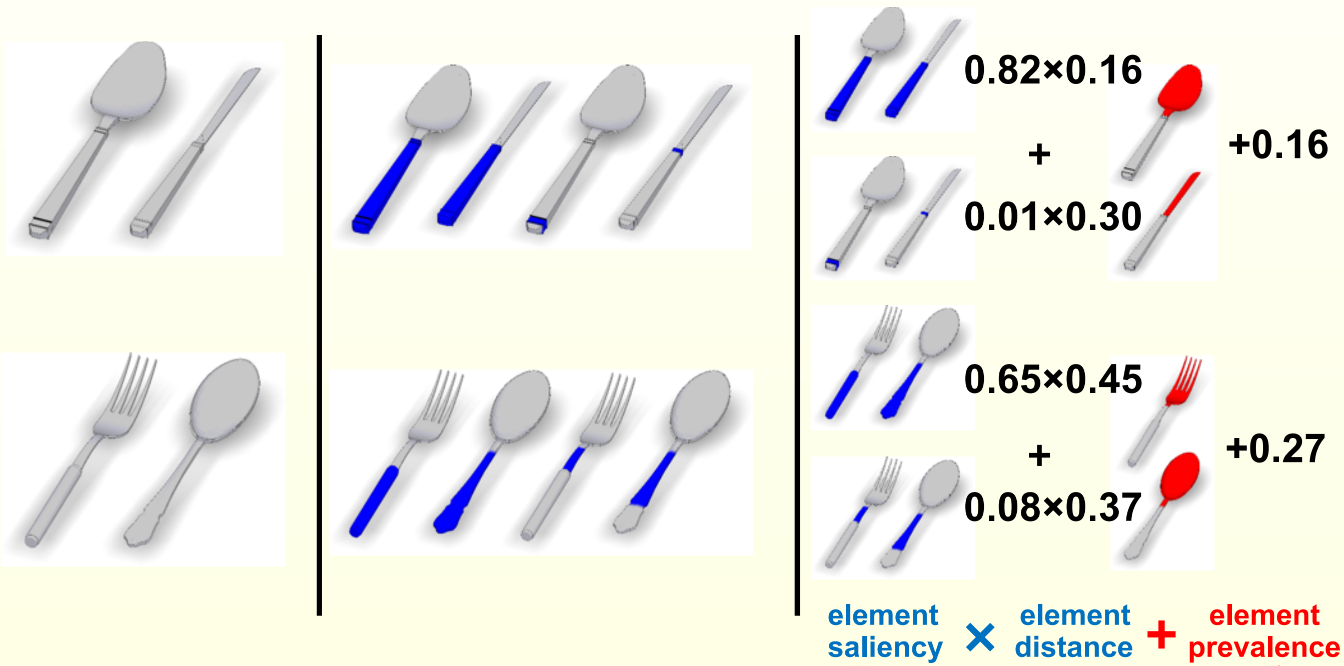
Algorithm for measuring style similarity



input shapes matching elements

Slide credit: Z. Lun

Algorithm for measuring style similarity



input shapes   matching elements   distance components

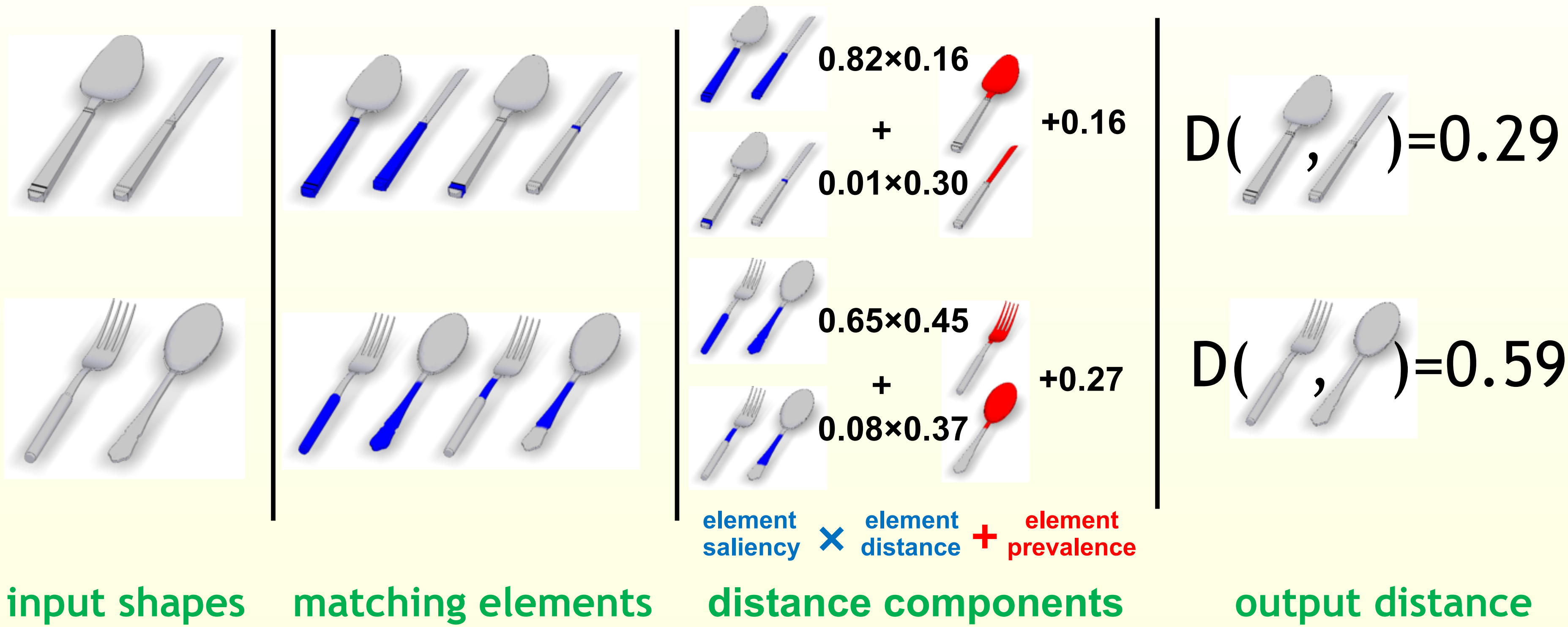
Computed for each element using geometric cues

Same distance we used to match elements

Percentage of the area on both models not covered by any matched elements, weighted by their saliency

Slide credit: Z. Lun

Algorithm for measuring style similarity



Slide credit: Z. Lun

Parameter learning

Learn parameters from training triplets:

- element-similarity weights ($\mathbf{w}$)
- saliency weights ($\mathbf{v}$)
- prevalence penalty (t)

that maximize likelihood function & regularizer to promote sparsity:

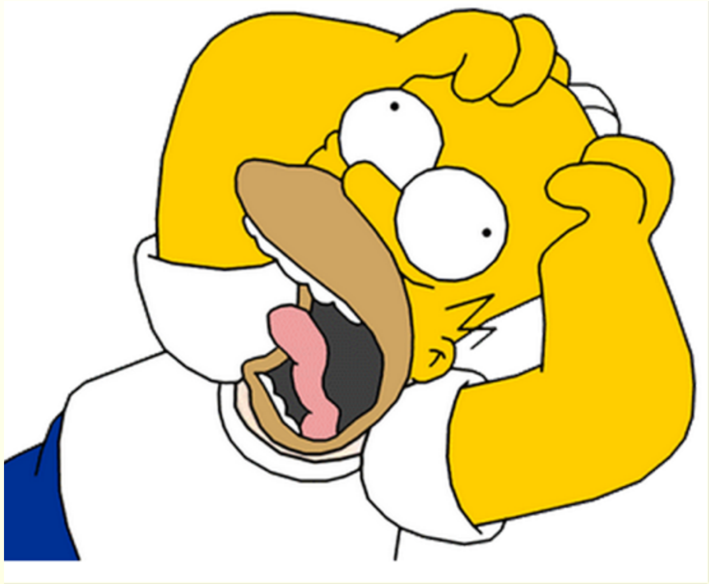
$$L(\mathbf{w}, \mathbf{v}, t) = \sum_{\text{triplet } \{A,B,C\}} \mathit{confidence}(\mathrm{B}) \cdot \log P(\mathrm{B \text{ is more similar to } A \text{ than } C})$$

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

$$+ \boxed{\mathit{regularizer}(\mathbf{w}, \mathbf{v}, t)}$$

Slide credit: Z. Lun

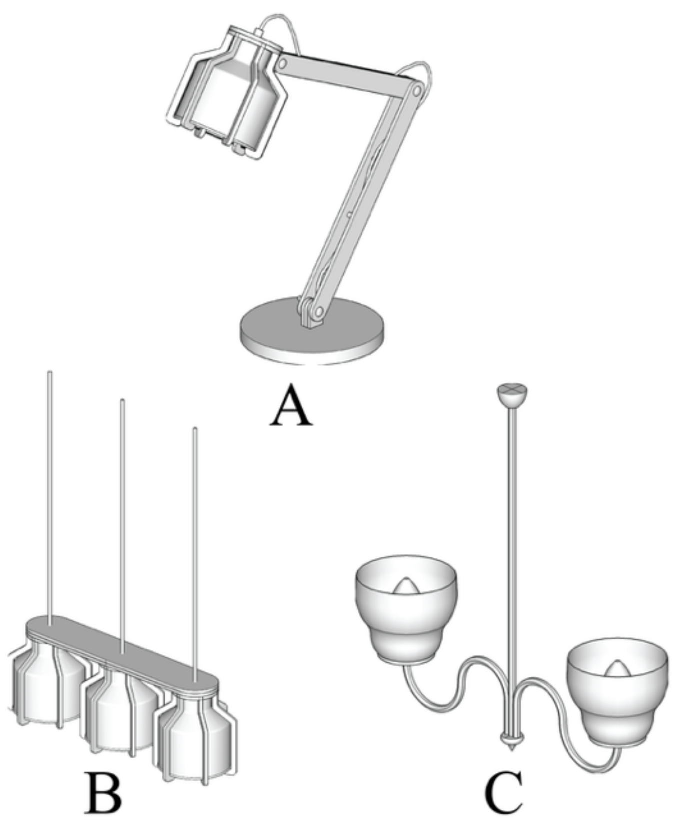
Validation



Does it work?

Slide credit: Z. Lun

Our result



(i) B - 90%

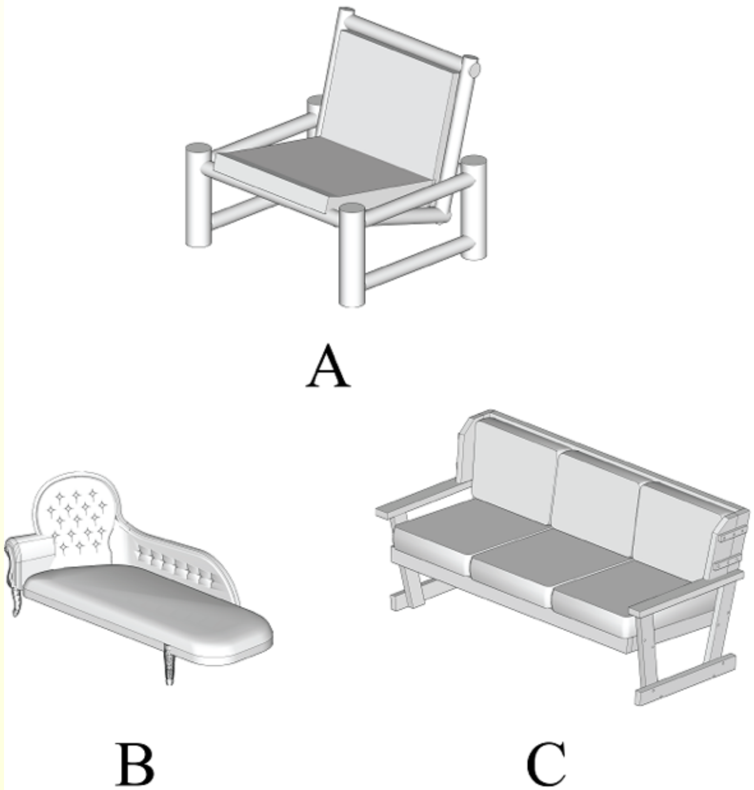
(ii) C - 0%

(iii) Both - 0%

(iv) Neither - 10%

Slide credit: Z. Lun

Failure case



(i) B - 0%

(ii) C - 70%

(iii) Both - 0%

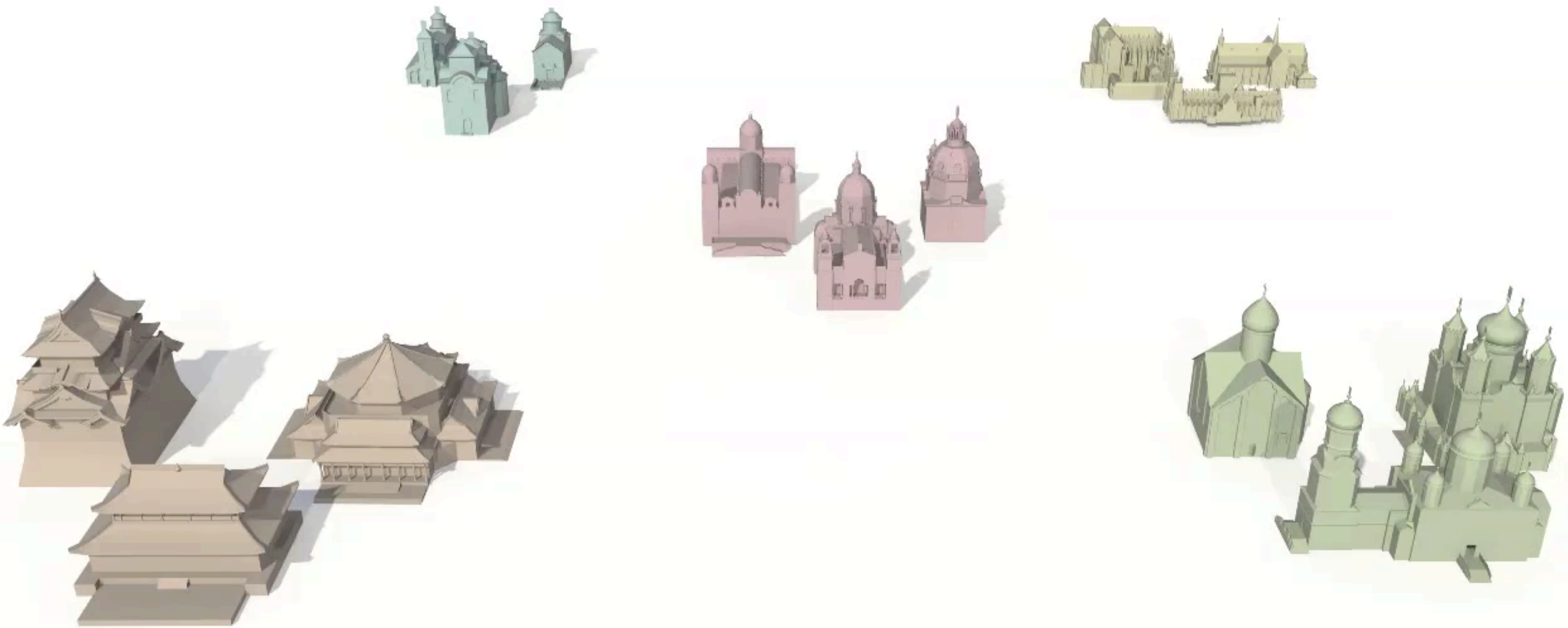
(iv) Neither - 30%

Slide credit: Z. Lun

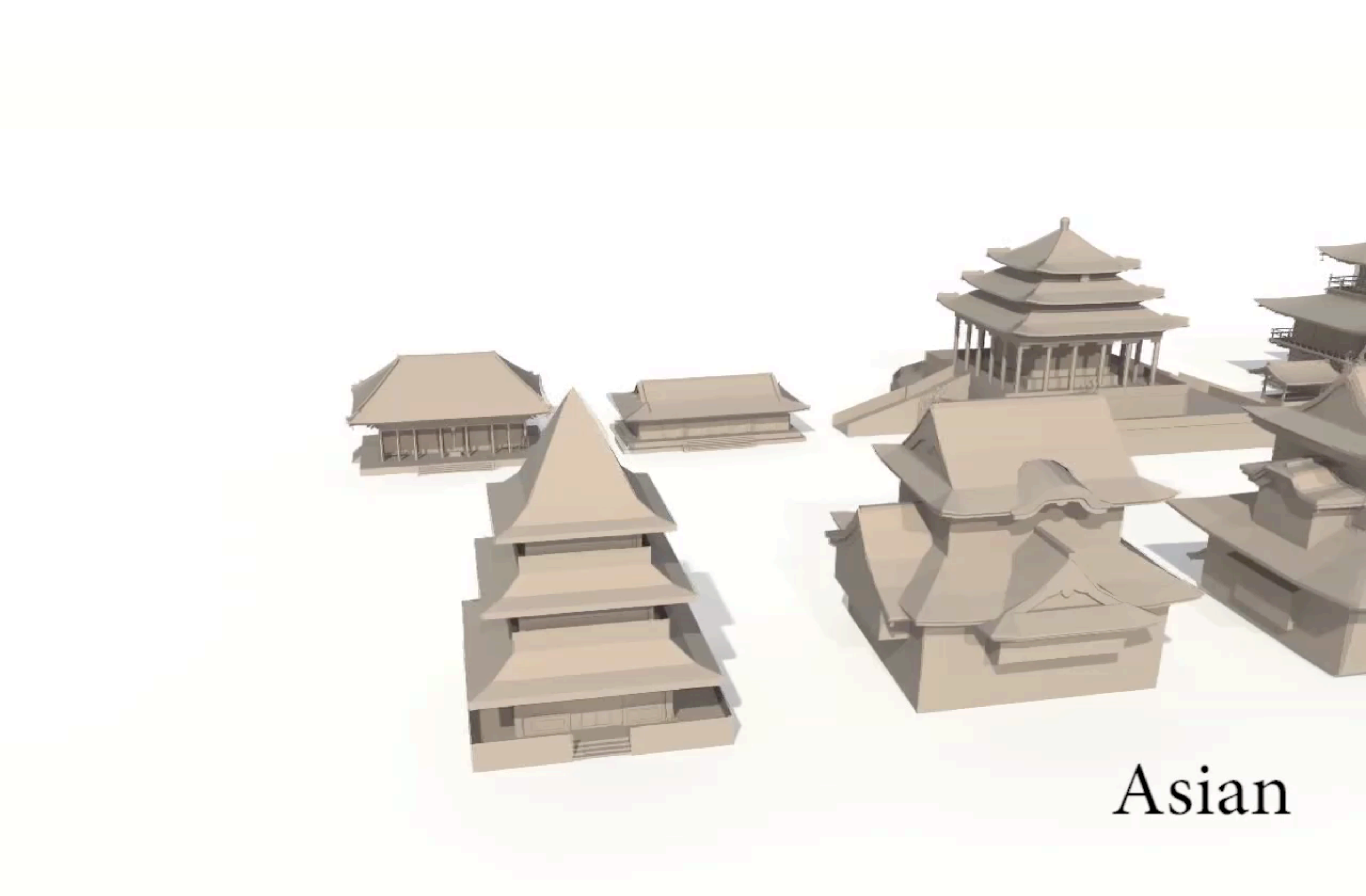
Application:

Style-based shape tagging

Slide credit: Z. Lun



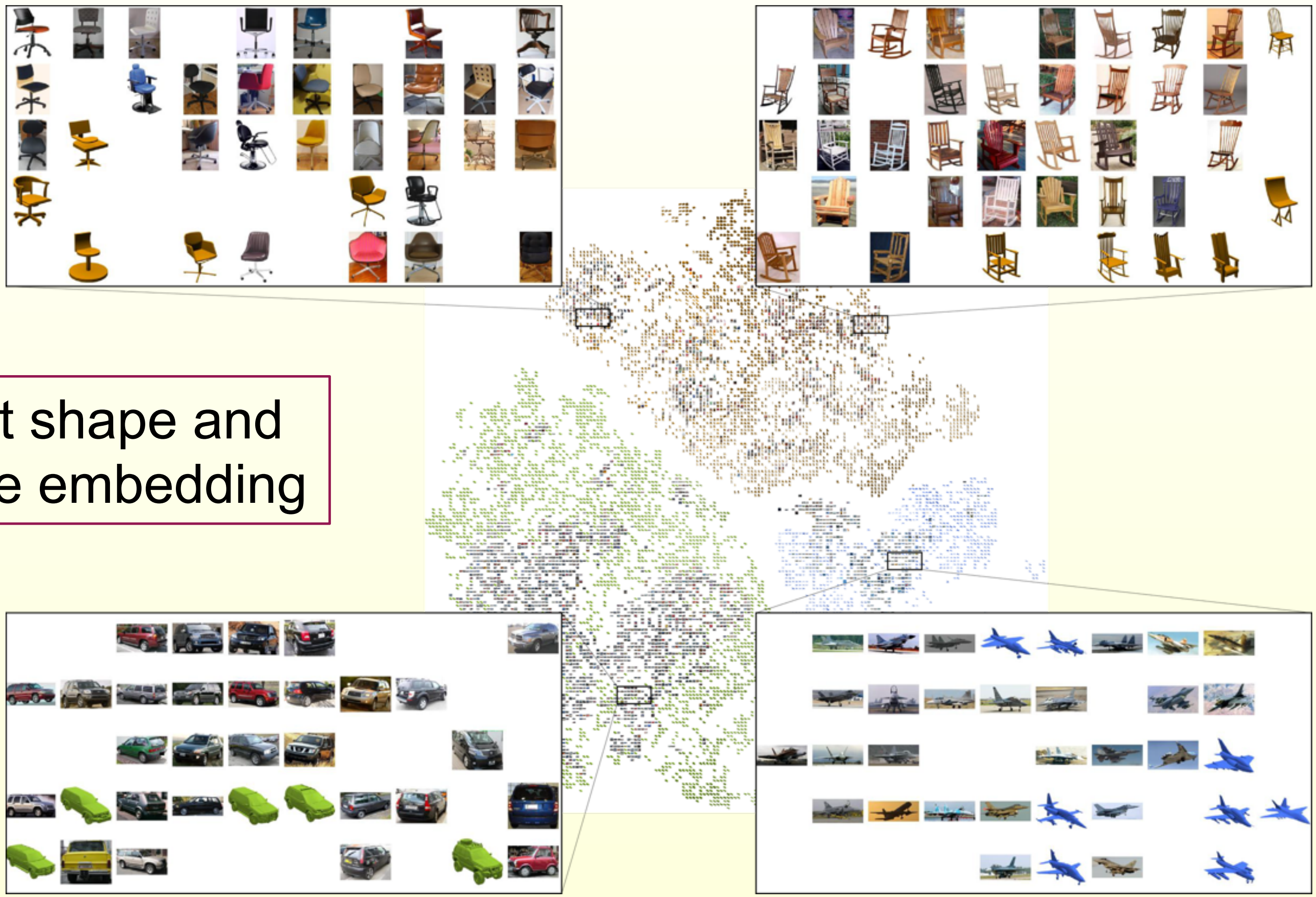
Slide credit: Z. Lun



Slide credit: Z. Lun

Questions?

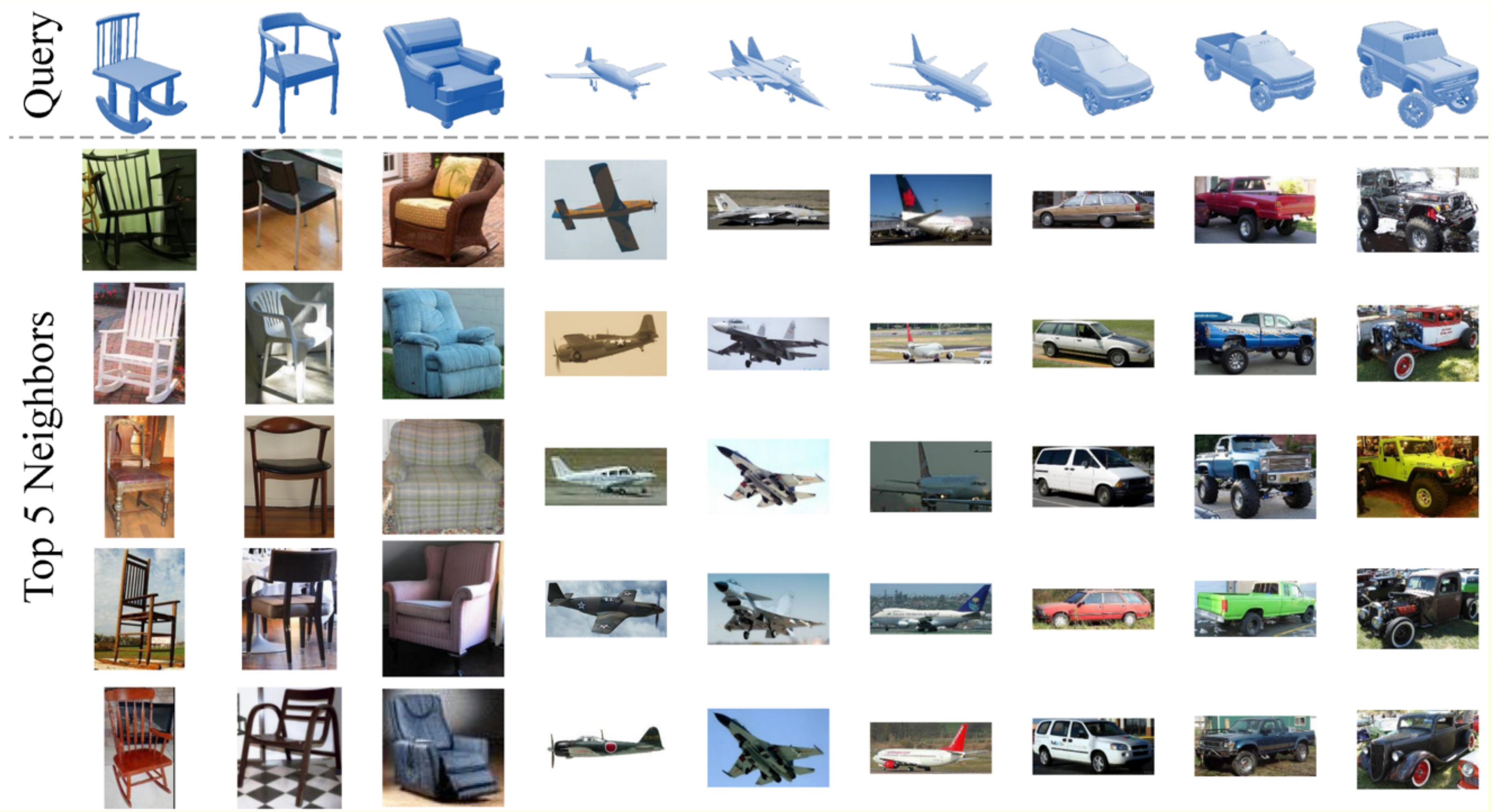
Shape similarity and retrieval - another flavor



Joint shape and image embedding

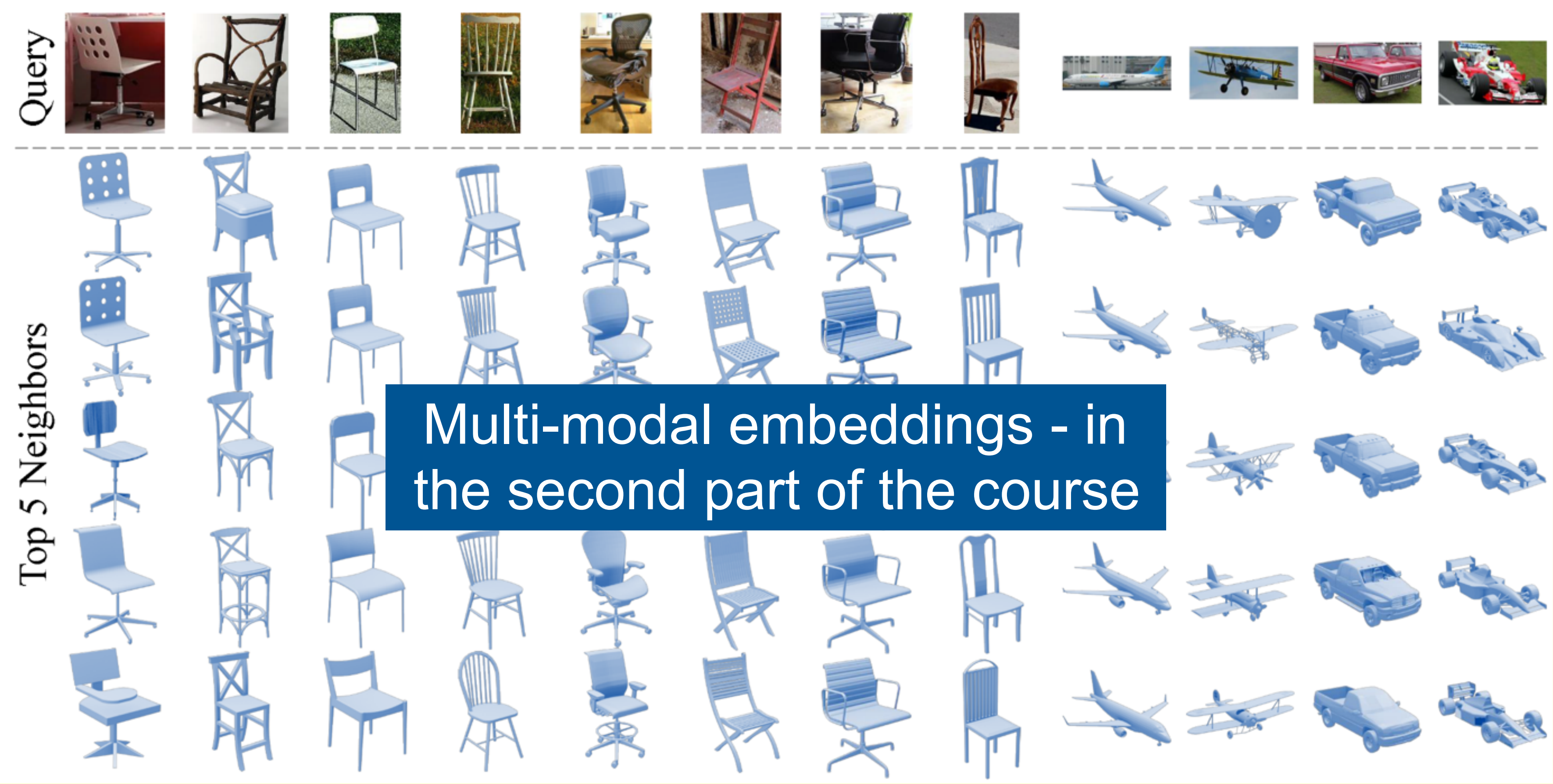
[Li et al. 2015]

Shape based Image Retrieval



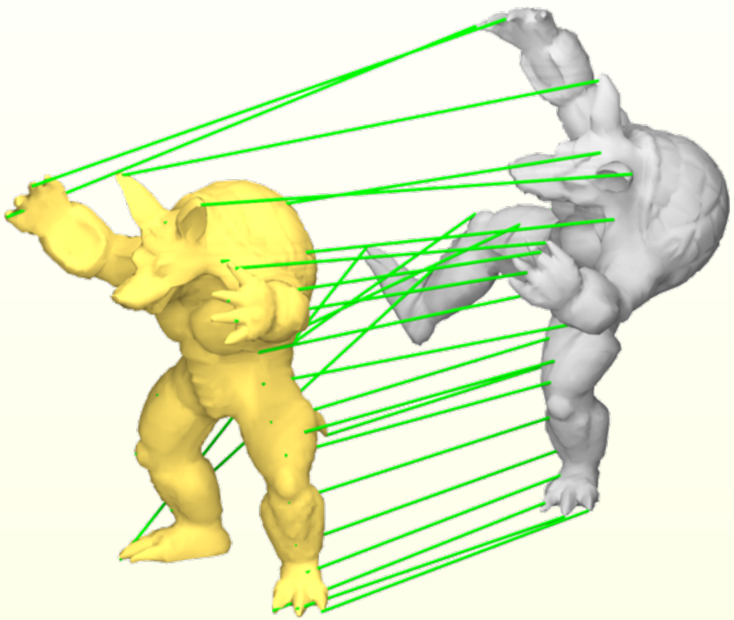
[Li et al. 2015]

Image based Shape Retrieval



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

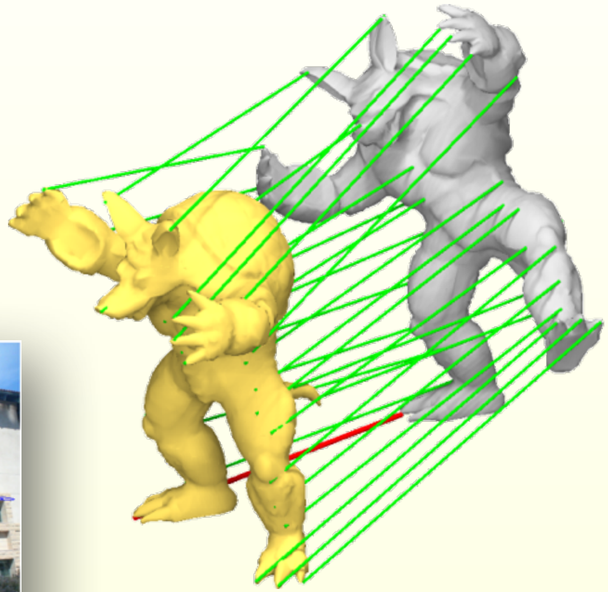
[Li et al. 2015]

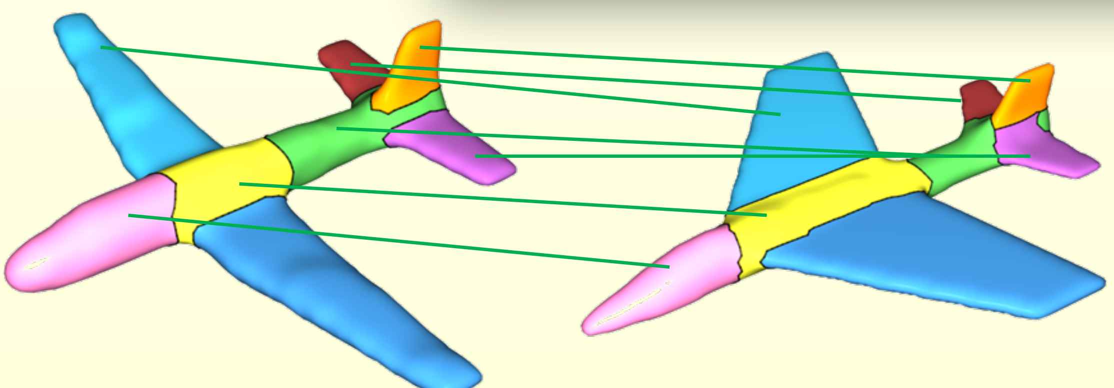


DEFORMABLE SHAPE MATCHING

Mapping Between Data Sets

- Multiscale mappings
  - Point/pixel level
  - Part level





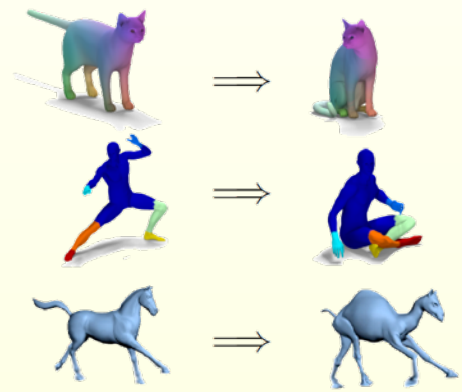
Maps capture what is the same or similar across two data sets

Slide credit: L. Guibas

Why Do We Care About Maps and Alignments?

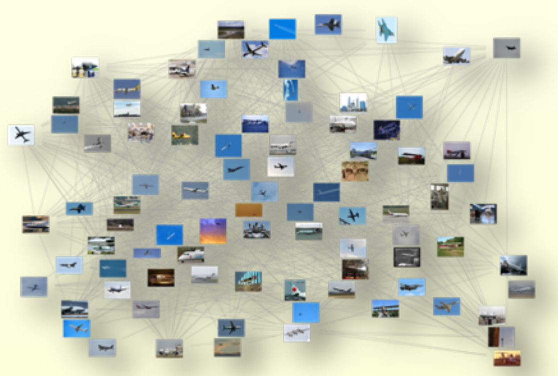
- To stitch data together

- To transfer information



- To compute distances and similarities

- To perform joint analysis



Slide credit: L. Guibas

Problem definition

- Given a pair of shapes, find corresponding points

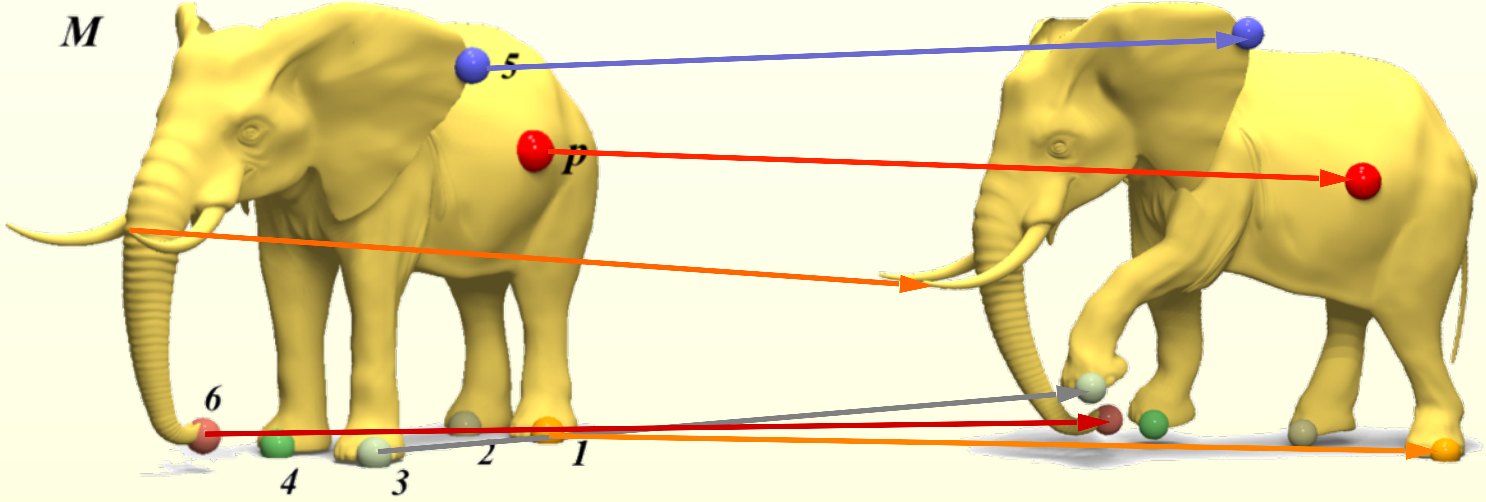
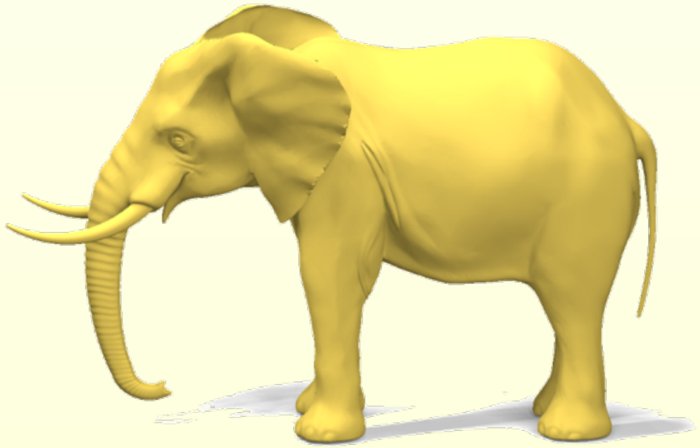


Image credit: L. Guibas, M. Ovsjanikov

Problem definition

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



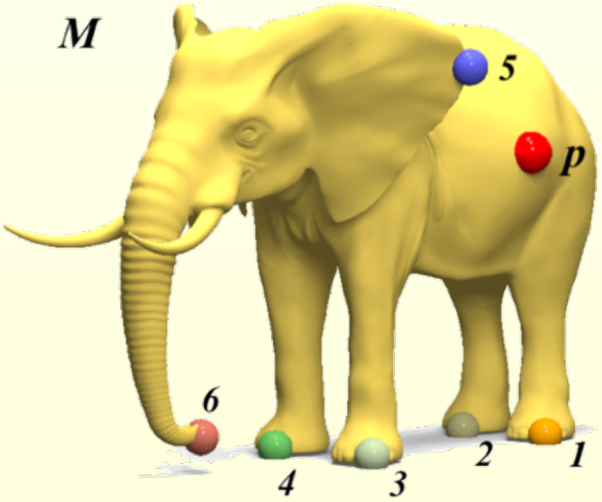
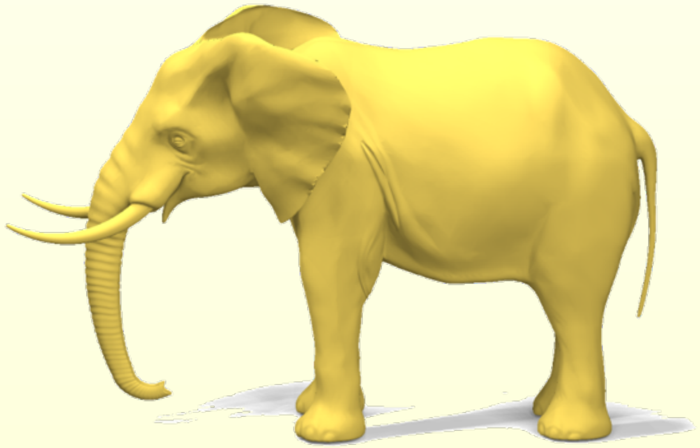


Image credit: L. Guibas, M. Ovsjanikov

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



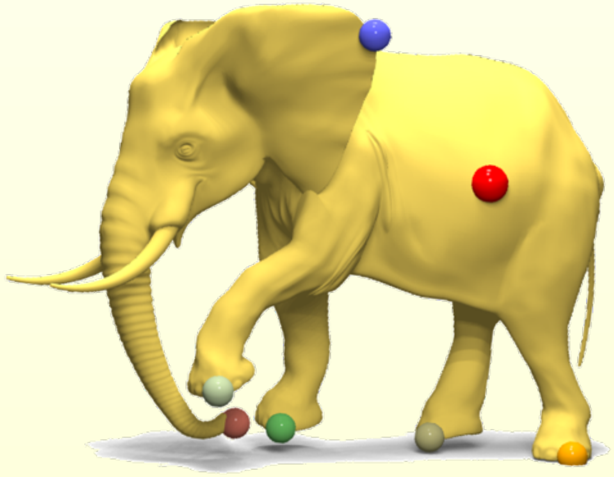
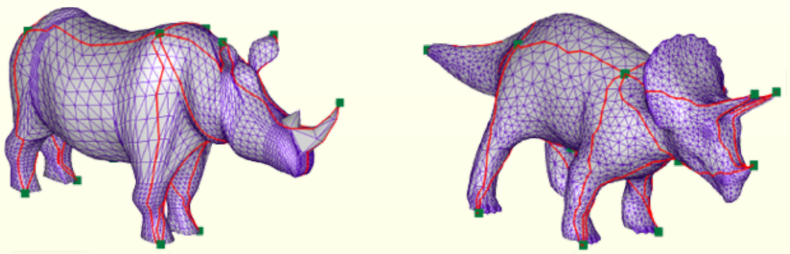


Image credit: L. Guibas, M. Ovsjanikov

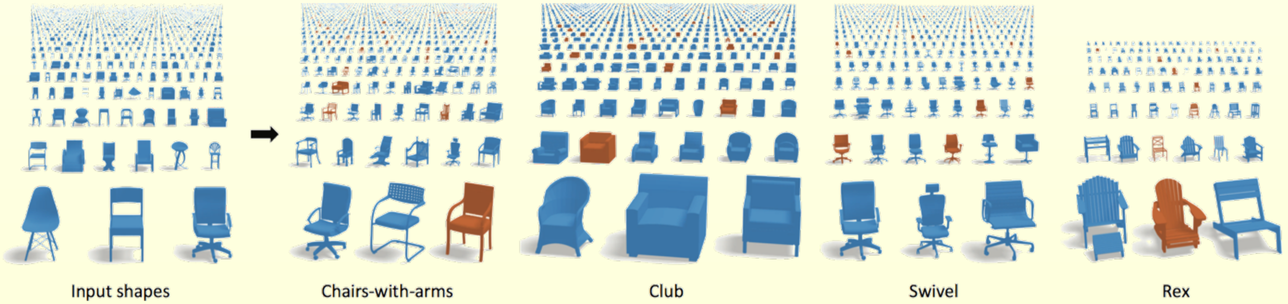
Problem definition

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

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



[Kreavoy and Sheffer 2004]



[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 that pairwise alignment
- More in the following lecture
- For overview of shape alignment methods take cs233