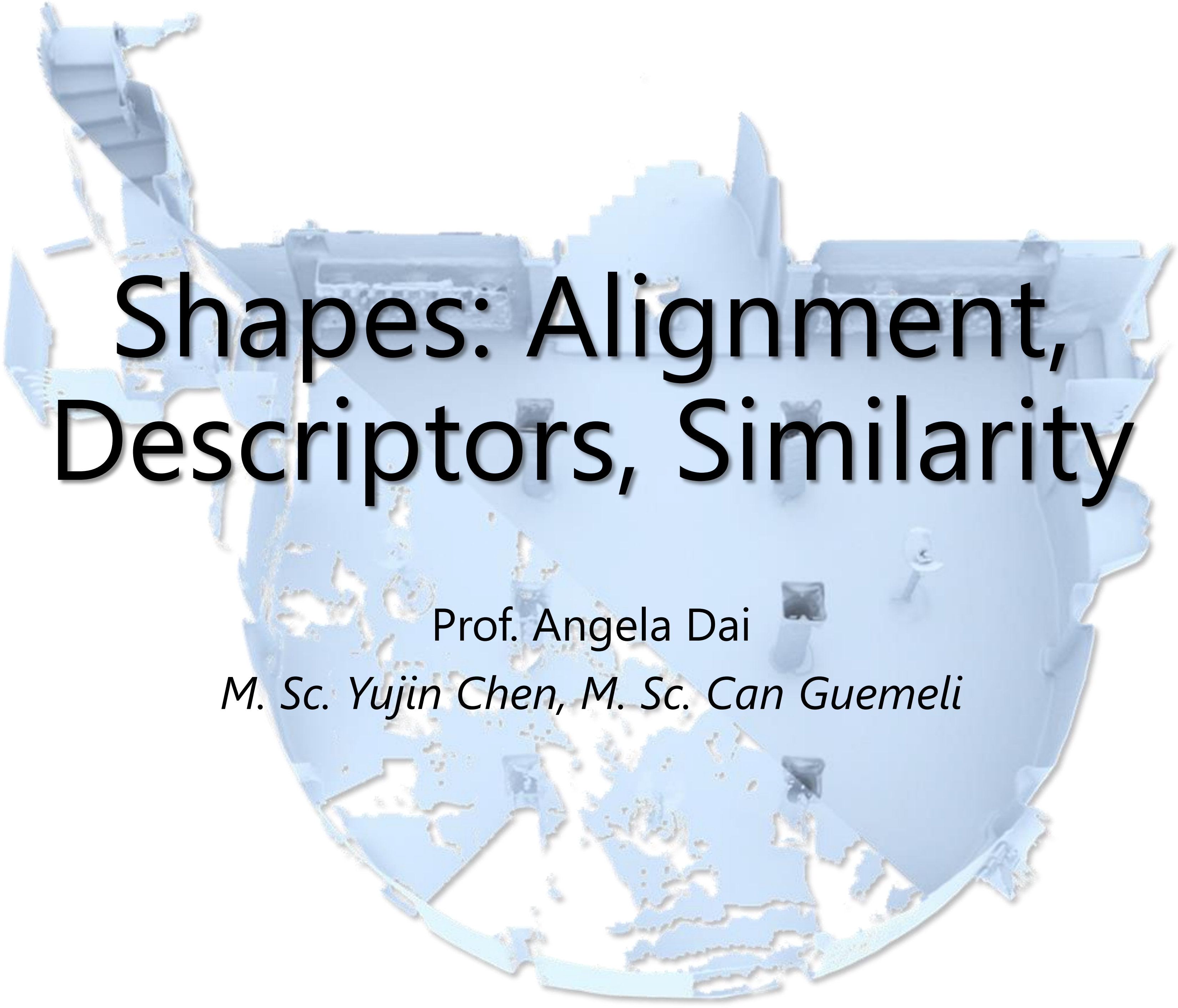


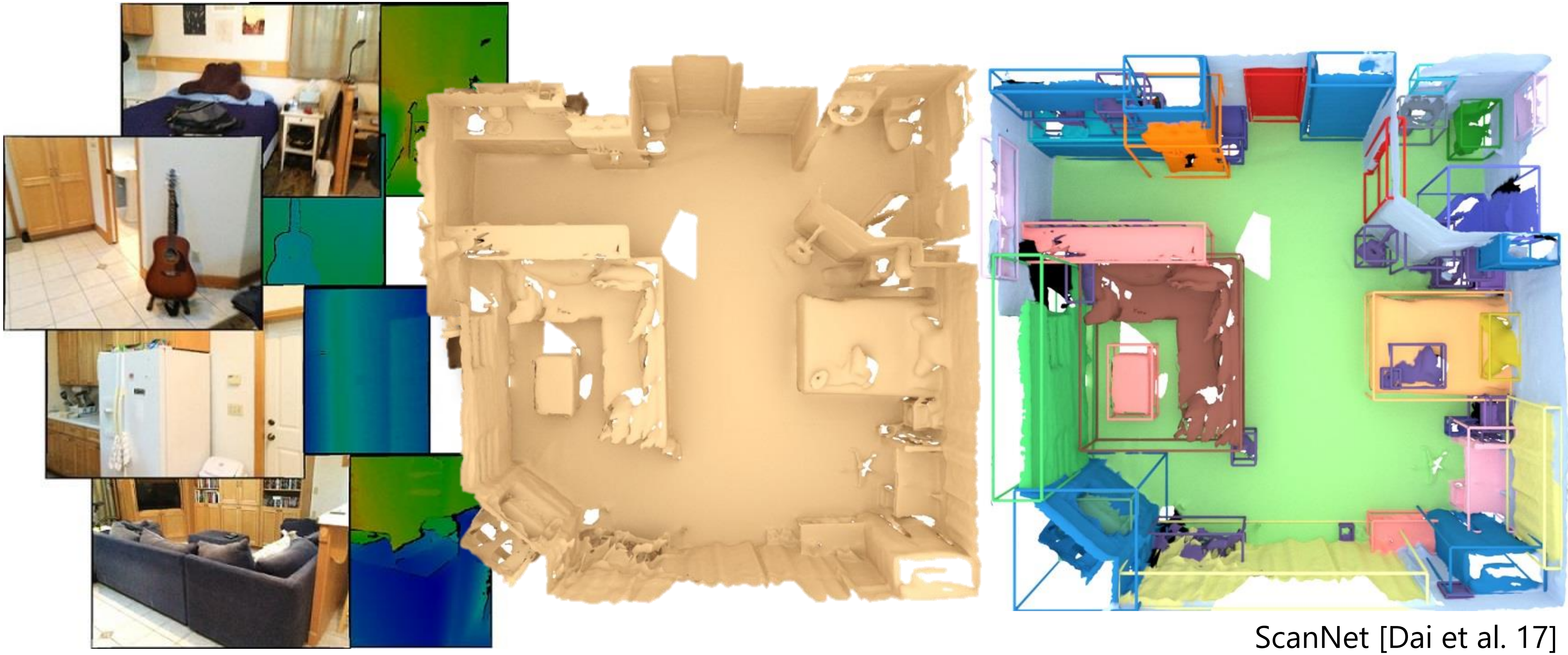
Shapes: Alignment, Descriptors, Similarity

Prof. Angela Dai

M. Sc. Yujin Chen, M. Sc. Can Guemeli

Brief Recap

Machine Perception of Real-World Environments



ScanNet [Dai et al. 17]

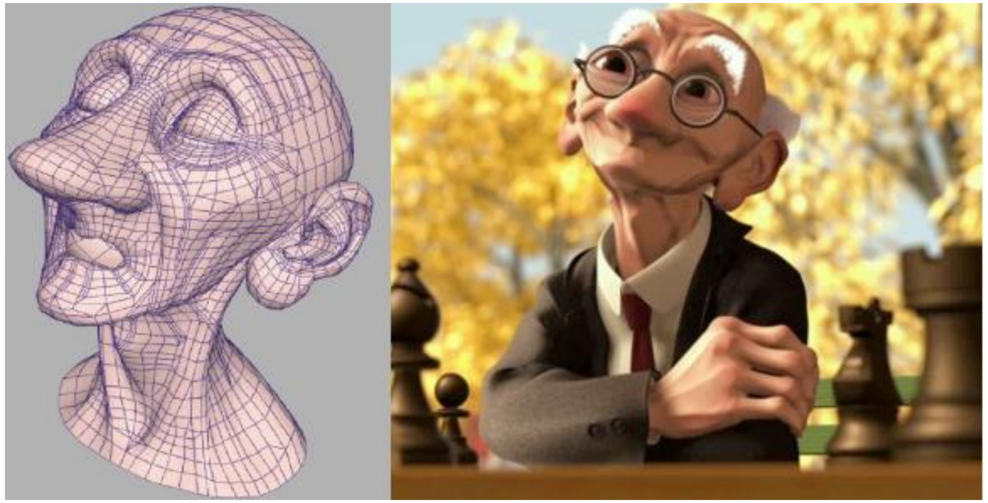
We *perceive* and *interact* with a 3D world

ASIMO, Honda

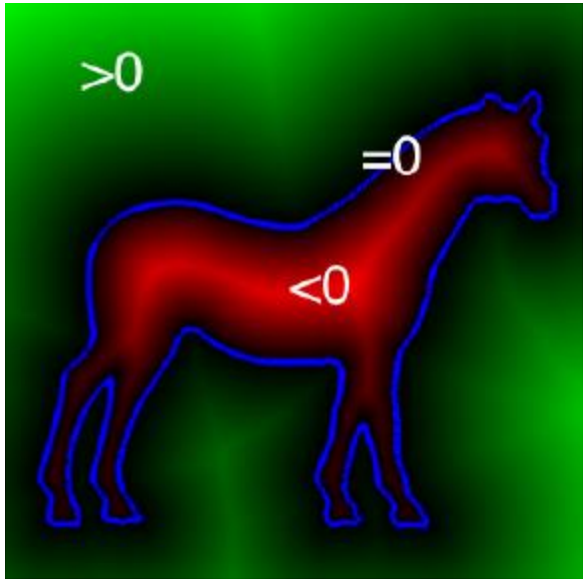
Star Trek TNG (Phantasms)

How to represent 3D?

Discrete:
Meshes,
Point Samples



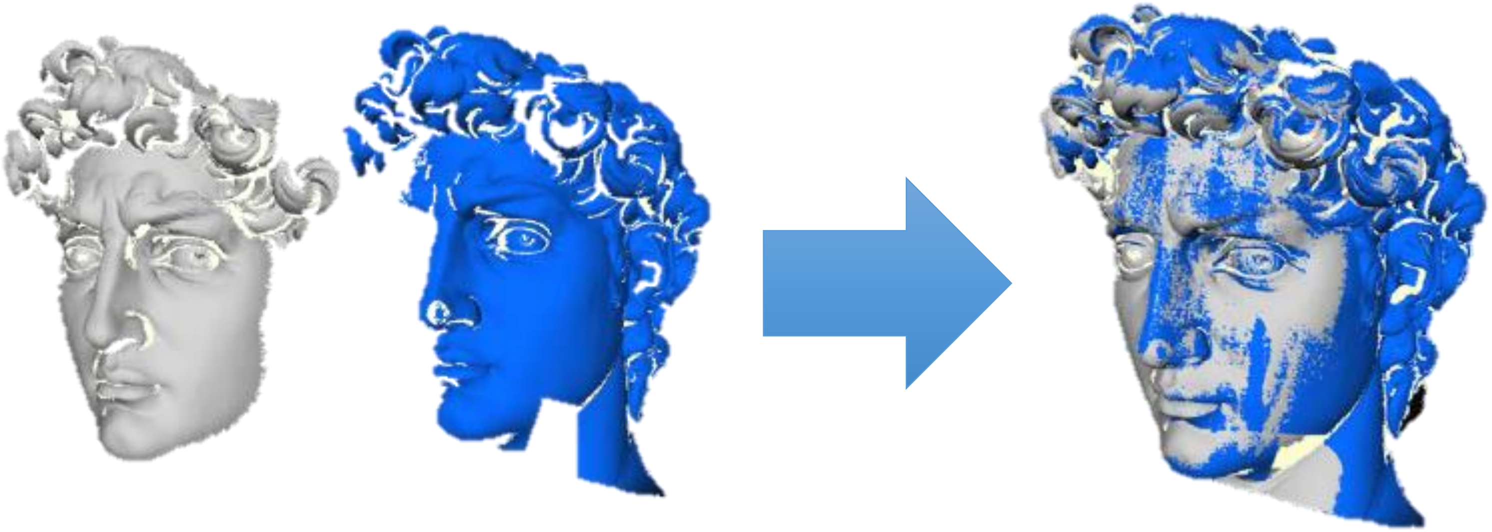
Parametric



Implicit:
Distance Fields

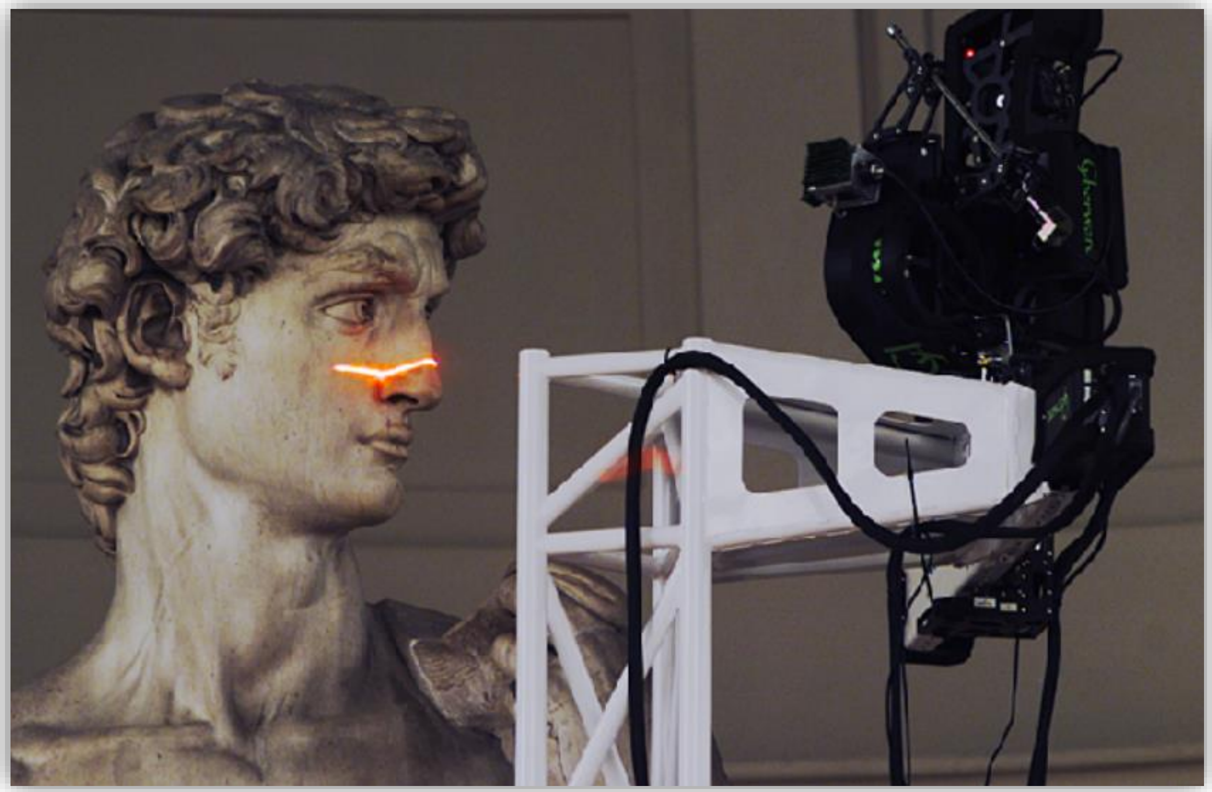
All about shapes

- Alignment
- Correspondences
- Descriptors



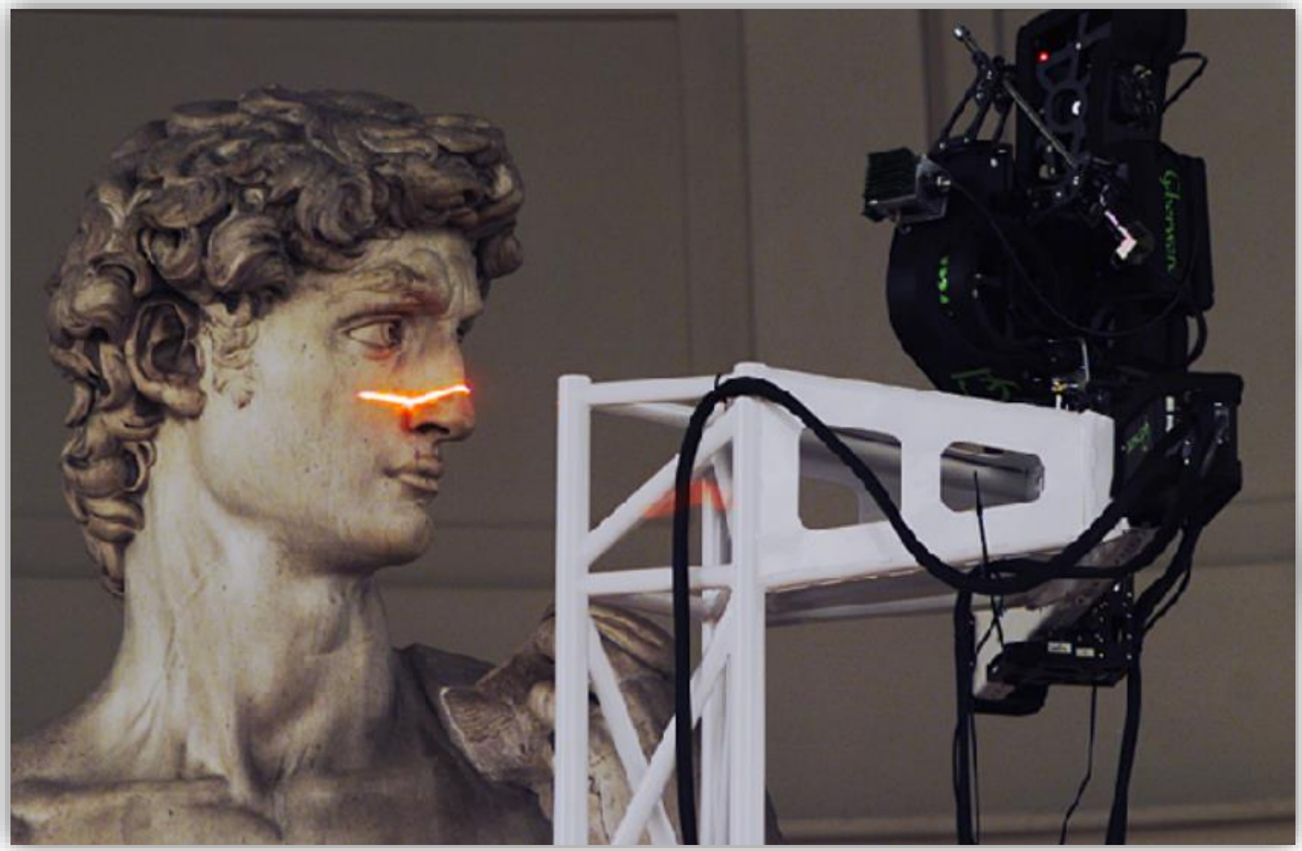
Shape Acquisition

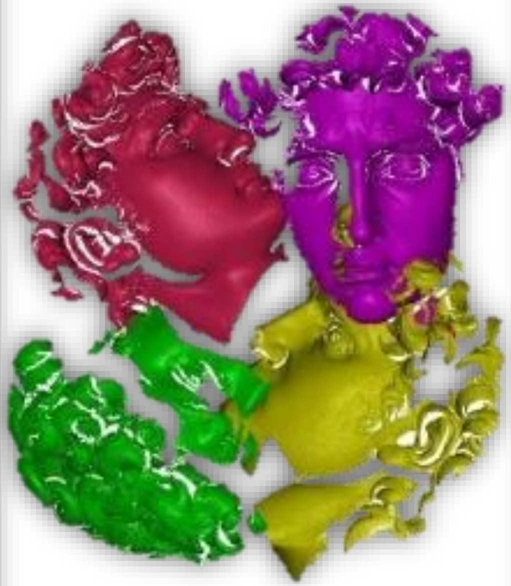
- 3D Scanning and Motion Capture (IN2354)

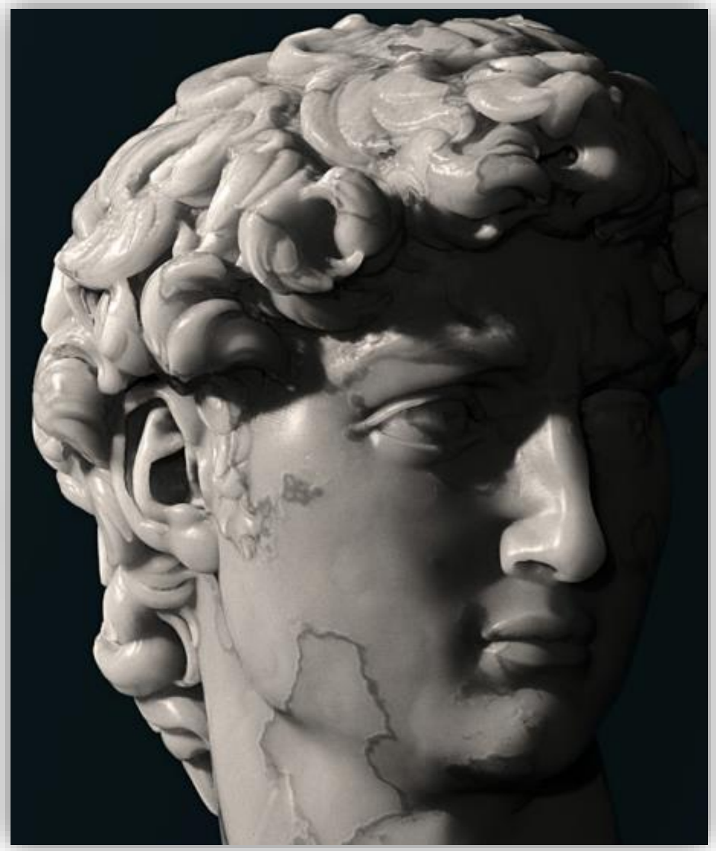


3D Alignment

- Many applications, e.g., 3D scanning, SLAM





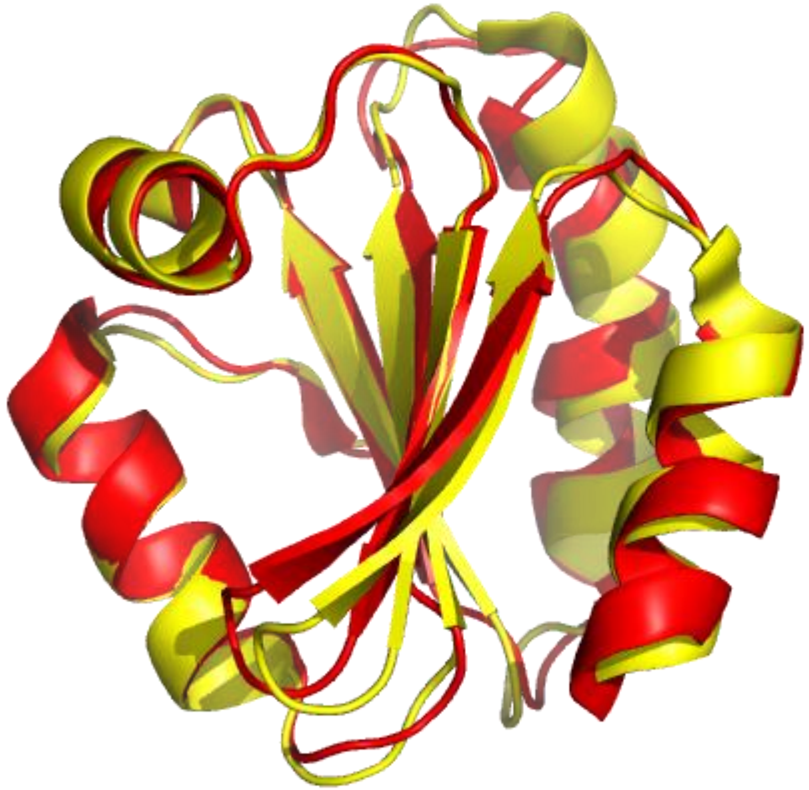


3D Alignment

- Many applications, e.g., 3D scanning, SLAM

3D Alignment

- Many applications

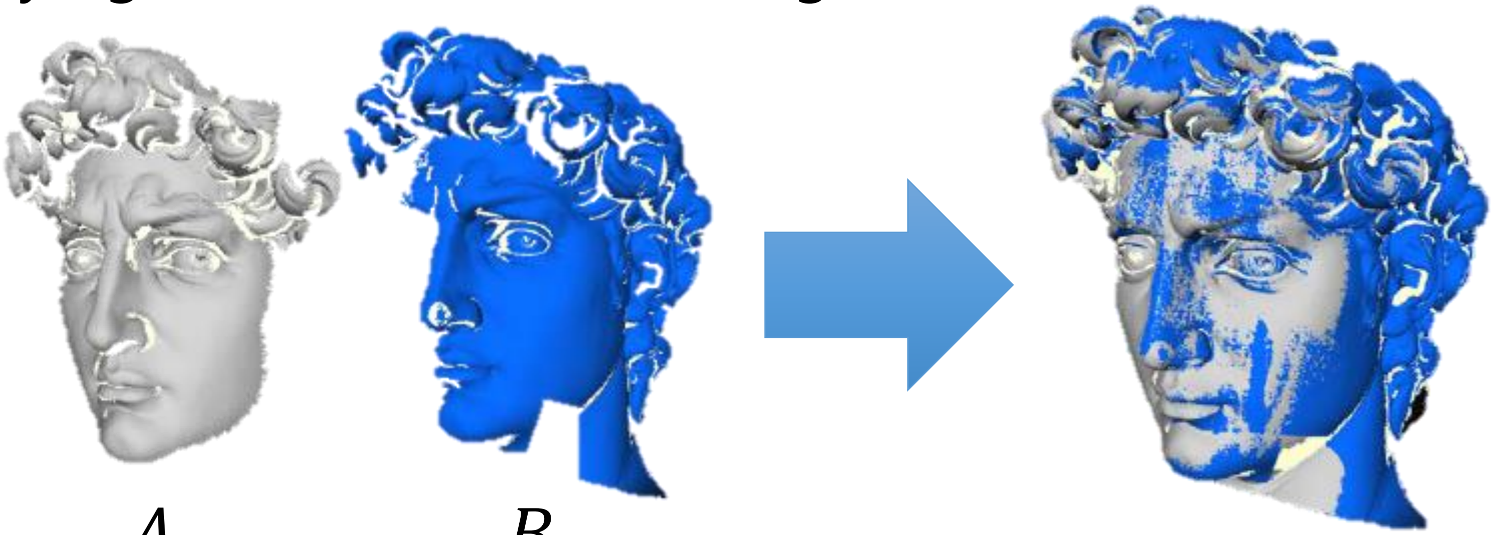


Protein Structure Alignment:

- (red) from humans
- (yellow) from fly Drosophila melanogaster.

3D Alignment (Registration)

- Input:
  - 2 shapes A and B with partial overlap
- Problem:
  - Register B to A by rigid transform, minimizing distance between A and B



A B

Shape Distance

- Measure of success for registration problem

$$\min_{T} \delta(A, T(B))$$

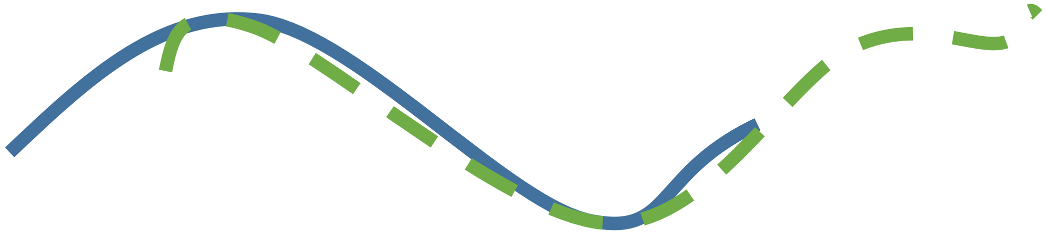
T: rigid transform to bring B to A

- Fundamental for shape similarity, classification, general machine learning losses

How to evaluate 3D distance?

- What about common function norms, e.g., ℓ_2?
  - We don't have correspondences across 3D structures, shapes

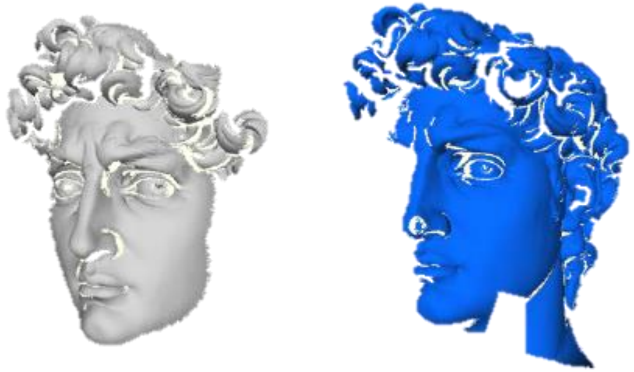
- Should support partial matches



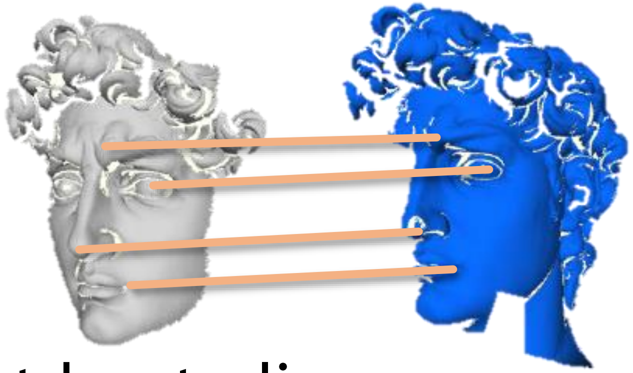
  - Trade-off between support size and aggregated distance
  - Distance for partial matches not a metric

Alignment Estimation

- Given shapes A and B



- Establish correspondences between A and B



- Find optimal transform that best aligns correspondences together, based on a distance measure

Transform Estimation

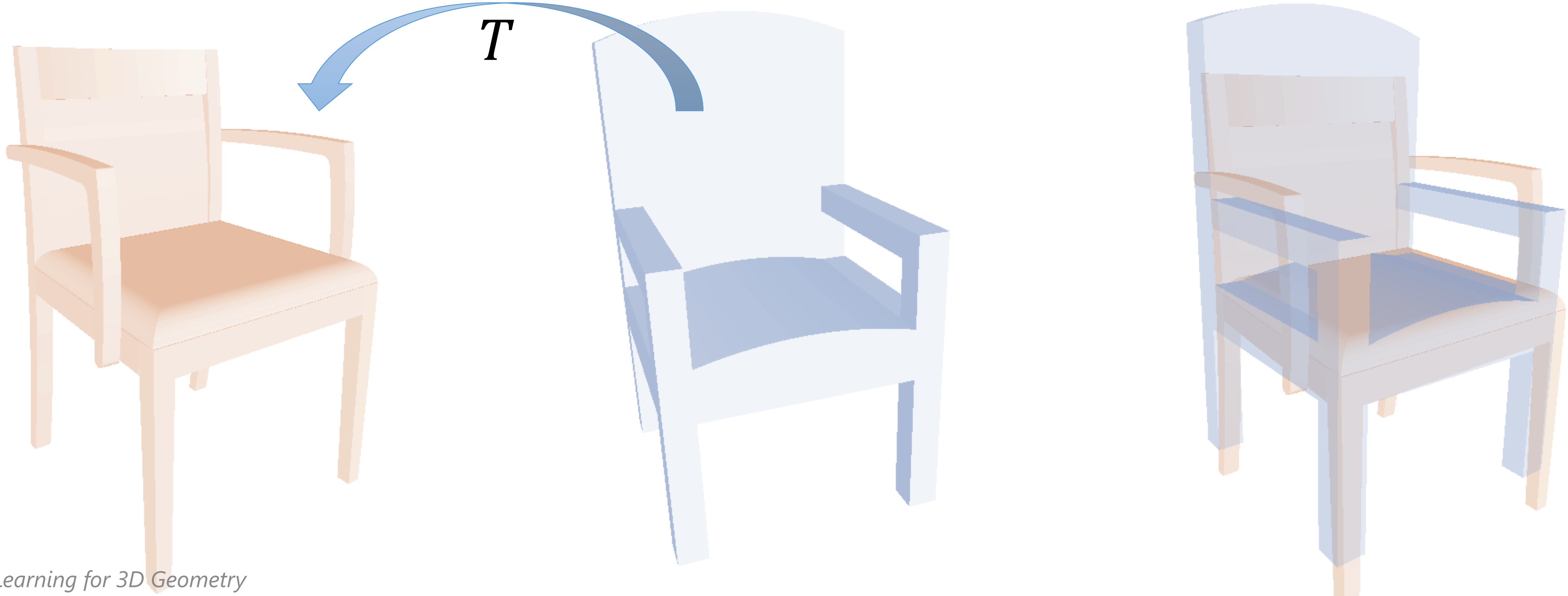
- Degrees of freedom
- Rigid motion has 6 degrees of freedom (3 rotation, 3 translation)
- Typically estimate with more correspondences -> overdetermined problem
- More general transforms -> more degrees of freedom, e.g., nonrigid deformations

Alignment Challenges

- Correspondence Estimation
  - Combinatorial search
- Transform Estimation
  - Transforms can be non-linear
- Difficult optimization -> look for good features, low-dimensional transforms

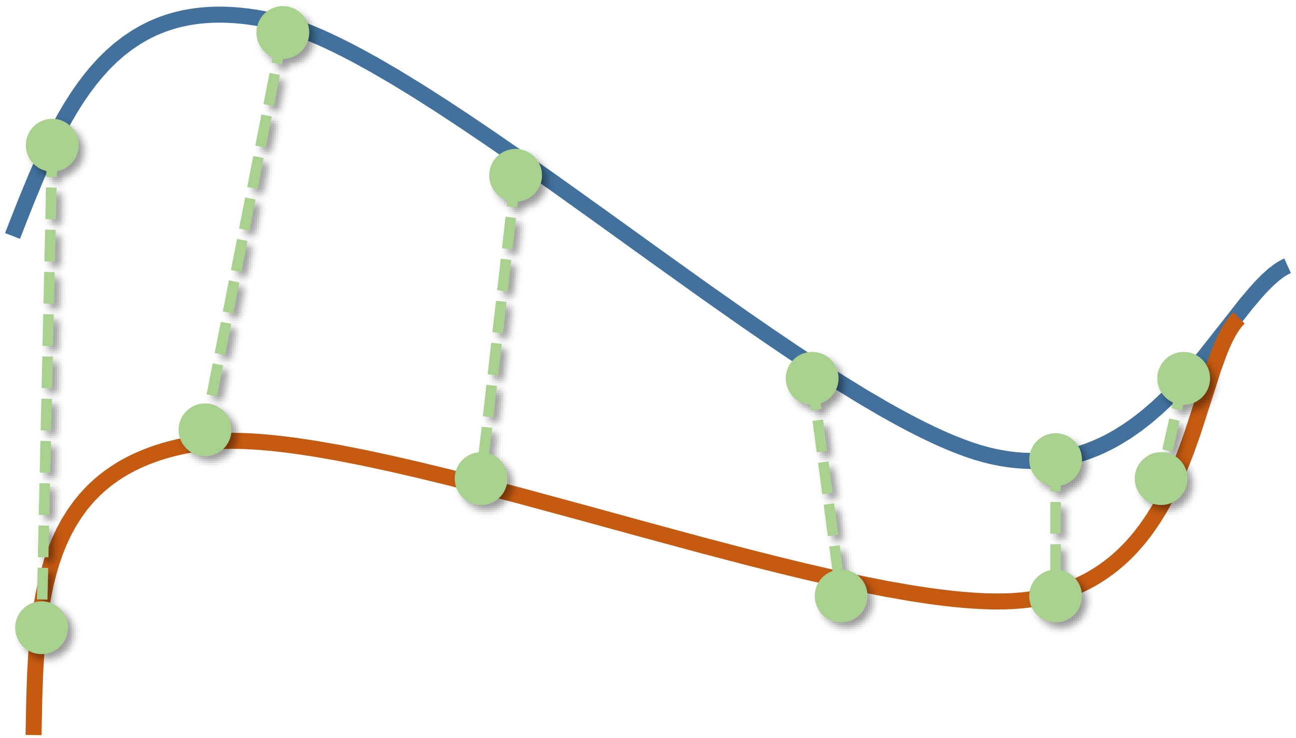
Rigid 3D Alignment

- Find 6DoF rigid transform that best aligns shapes, even if the shapes are different



Rigid 3D Alignment (Given Correspondences)

- Given correspondences $\{x_i\}, \{y_i\} \in \mathbb{R}^3$
- Find rigid transform $\boldsymbol{R}, t$ that minimizes $\sum_{i=1}^{N} \|\boldsymbol{R}x_i + t - y_i\|_2^2$



Solved as *orthogonal Procrustes problem* in 1966

Rigid 3D Alignment (Given Correspondences)

$$\min_{\boldsymbol{R},t} \sum_{i=1}^{N} \|\boldsymbol{R}x_i + t - y_i\|_2^2$$

- How to solve for $\boldsymbol{R}, t$?

- Consider coordinate system centered at the mean of the x_i

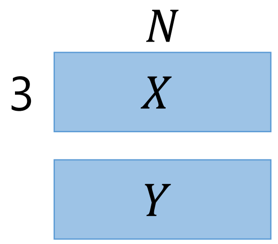
$$\min_{\boldsymbol{R},t} \sum_{i=1}^{N} \underbrace{\|t - y_i\|_2^2}_{\text{translation part}} - 2 \sum_{i=1}^{N} \underbrace{\langle \boldsymbol{R}x_i, y_i \rangle}_{\text{rotation part}}$$

Rigid 3D Alignment (Given Correspondences)

$$\min_{\boldsymbol{R},t} \sum_{i=1}^{N} \|t - y_i\|_2^2 - 2\sum_{i=1}^{N} \langle \boldsymbol{R} x_i, y_i \rangle$$

- Translation: $t = \frac{1}{N}\sum_{i=1}^{N} y_i$ (align centroids)
- Remove translation by mean-centering:

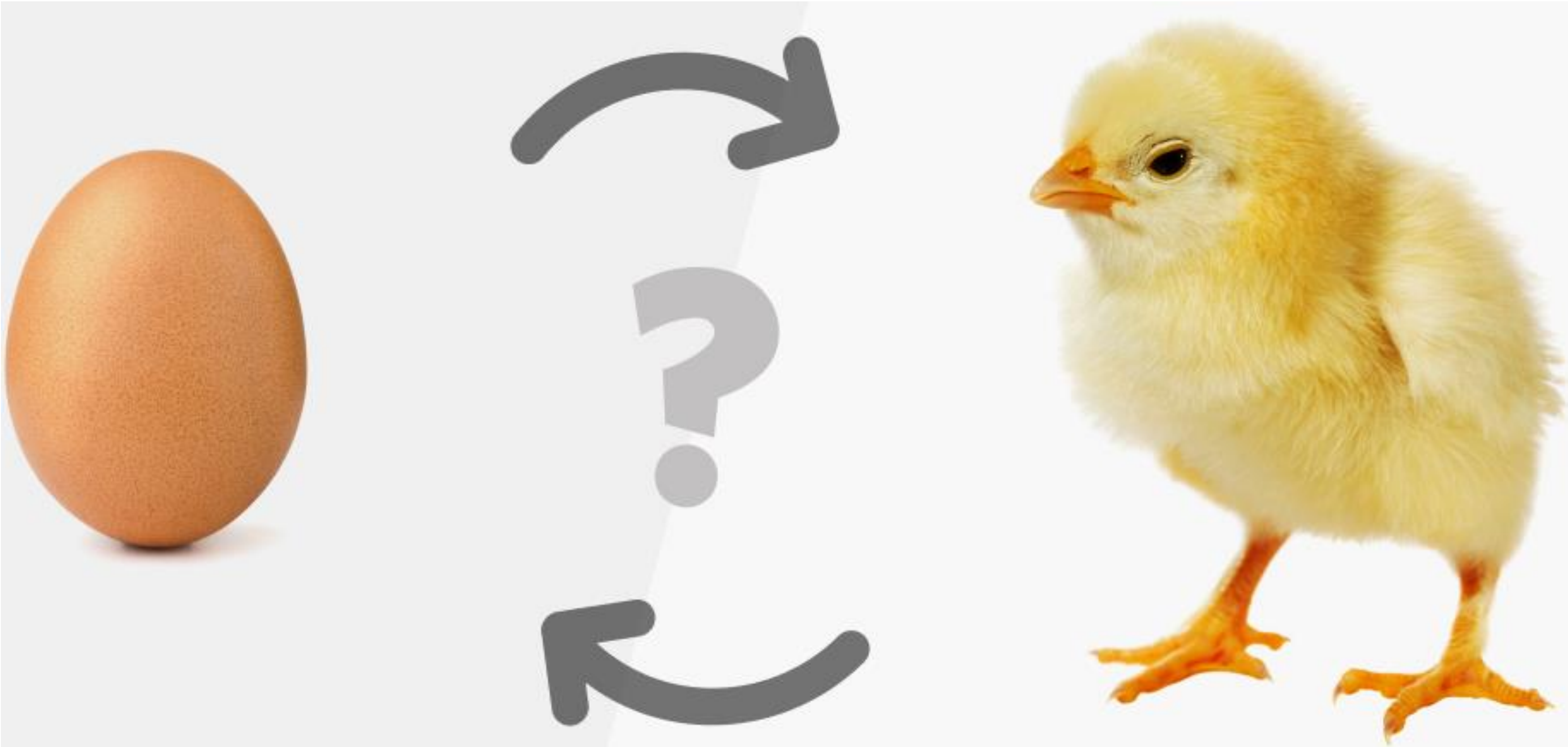
$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad \bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i \qquad X = [x_0 - \bar{x}, \dots, x_n - \bar{x}]^T \qquad Y = [y_0 - \bar{y}, \dots, y_n - \bar{y}]^T$$



- Compute SVD: $XY^T = UDV^T$ ← 3×3 matrix
- Define $S = \begin{cases} I, & \text{if } \det(U)\det(V) = 1 \\ diag(1, \dots, 1, -1) & \text{otherwise} \end{cases}$

$$\boxed{R = USV^T}$$

How to get correspondences?



How to get correspondences?

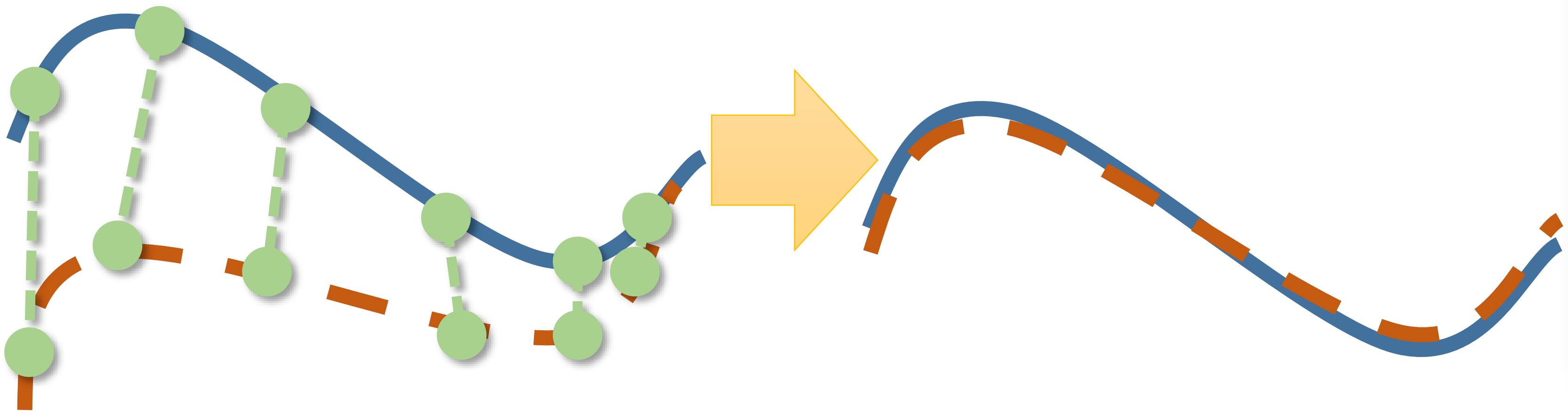
- Iterate between finding correspondences and solving for the best transform for those correspondences

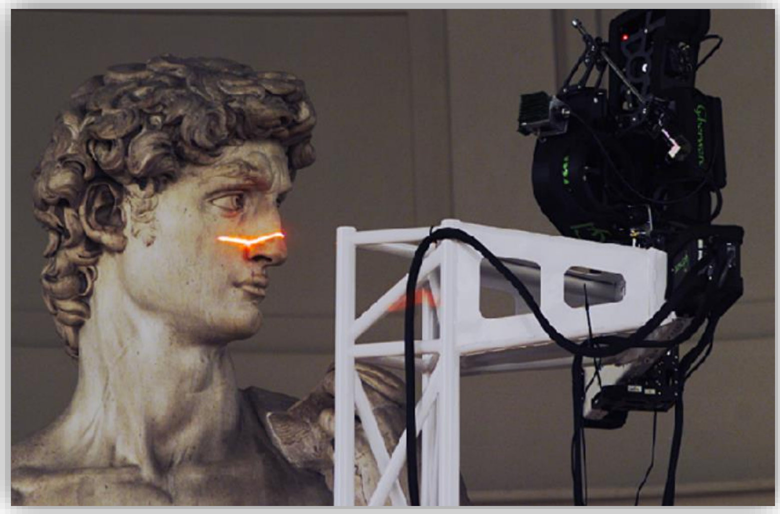
➢ Iterative Closest Points (ICP)

- Various methods to explicitly match features (handcrafted or learned), which can also be refined with ICP

Iterative Closest Points (Besl and McKay '92)

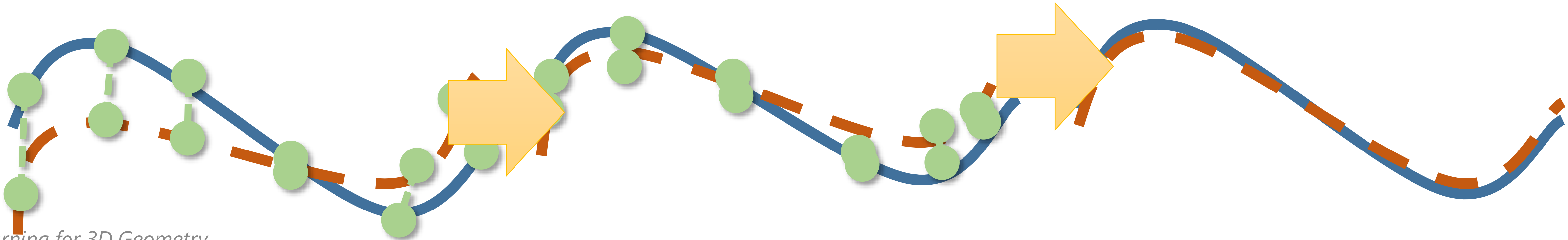
- Developed for aligning 3D shapes
- Nice analysis: *Efficient variants of the ICP algorithm* (Rusinkiewicz and Levoy 2001)



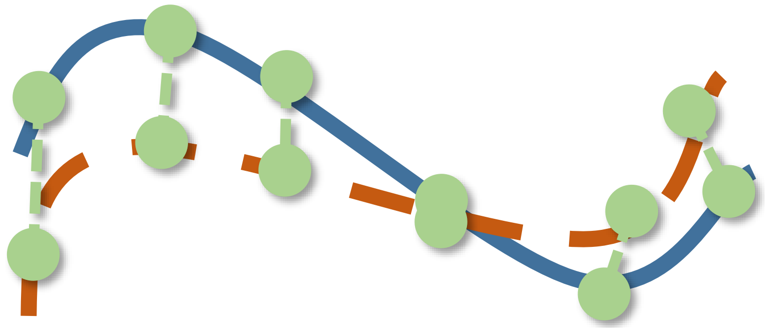


Iterative Closest Points

- How to find correspondences?
- Assume that closest points correspond
- Align the P_a points to their closest P_B neighbors; repeat
- Converges if starting positions are "close enough"



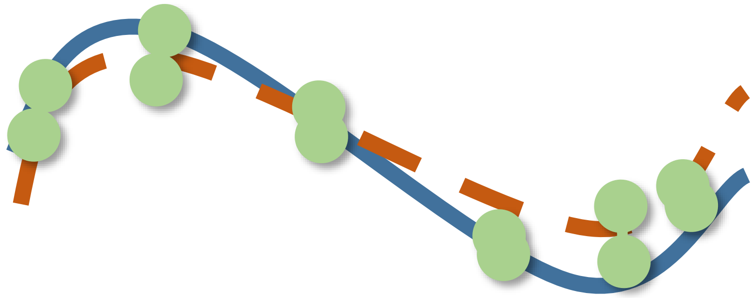
Iterative Closest Points



- Given a pair of shapes, A and B

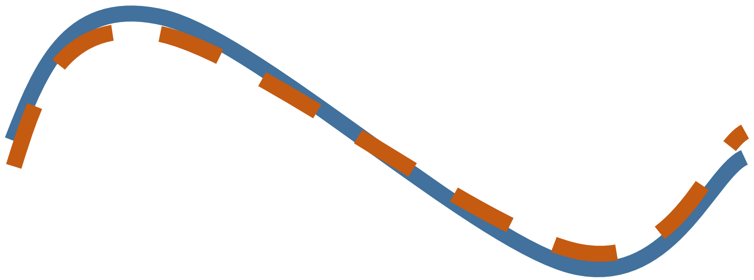
- Iterate:
  - Find corresponding points P_A and P_B based on proximity
  - Find optimal transform $\boldsymbol{R}, t$ minimizing $\underset{\boldsymbol{R},t}{\operatorname{argmin}} \sum_i \|\boldsymbol{R}x_i + t - y_i\|_2^2$
  - Apply optimized $\boldsymbol{R}, t$

Iterative Closest Points



- Given a pair of shapes, A and B
- Iterate:
  - Find corresponding points P_A and P_B based on proximity
  - Find optimal transform $\boldsymbol{R}, t$ minimizing $\underset{\boldsymbol{R},t}{\operatorname{argmin}} \sum_i ||\boldsymbol{R}x_i + t - y_i||_2^2$
  - Apply optimized $\boldsymbol{R}, t$

Iterative Closest Points



- Given a pair of shapes, A and B
- Iterate:
  - Find corresponding points P_A and P_B based on proximity
  - Find optimal transform $\boldsymbol{R}, t$ minimizing $\underset{\boldsymbol{R},t}{\operatorname{argmin}} \sum_i \|\boldsymbol{R}x_i + t - y_i\|_2^2$
  - Apply optimized $\boldsymbol{R}, t$

ICP: Runtime Analysis

- Each iteration:
  - Find closest points:
    - $O(N_B)$ per point
    - $O(N_B * N_A)$ total
  - Compute optimal alignment: $O(N_A)$
  - Update scene $O(N_A)$

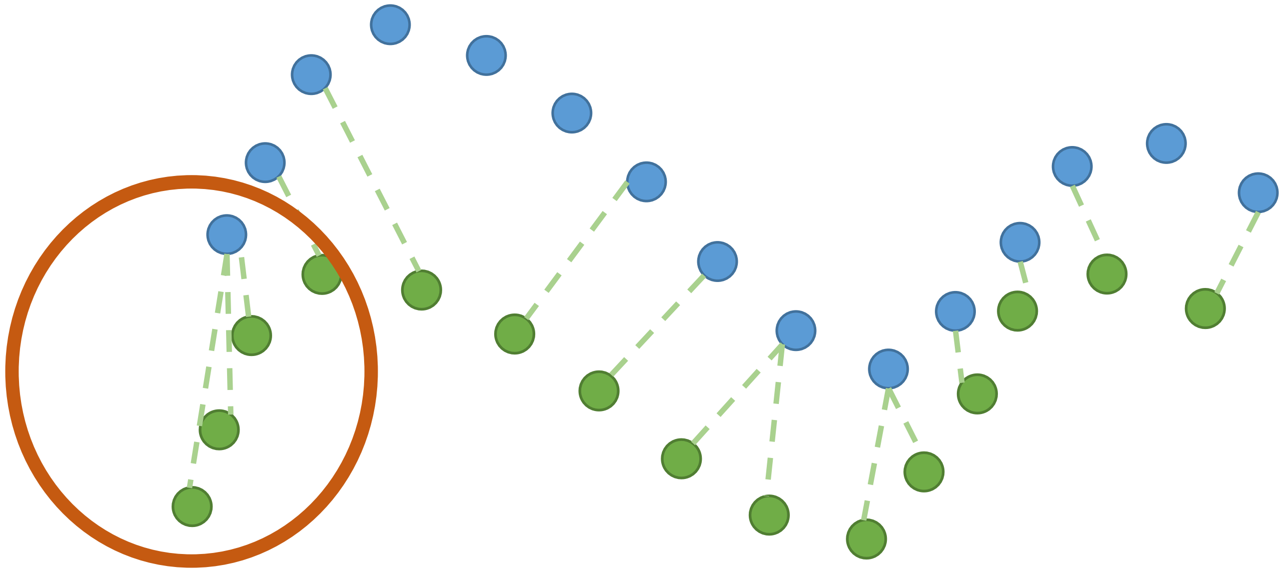
- Speed up with fast or approximate nearest-neighbor data structures, e.g., *kd*-tree

ICP Analysis

- Selection of points
- Matching correspondences
- Weighting correspondences
- Rejecting outlier correspondences
- Assigning error metric to the current transform
- Minimizing error metric w.r.t transform

ICP Analysis

- How to select correspondences?

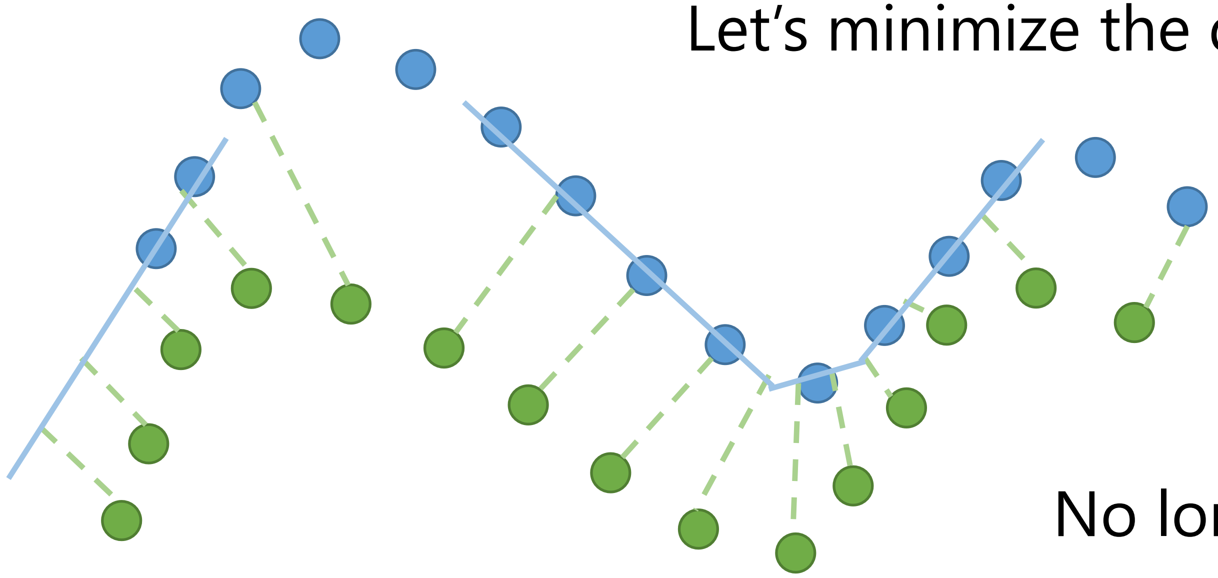


But: Uneven Sampling

Ideally, 1:1 correspondences

ICP Analysis

- How to select correspondences?



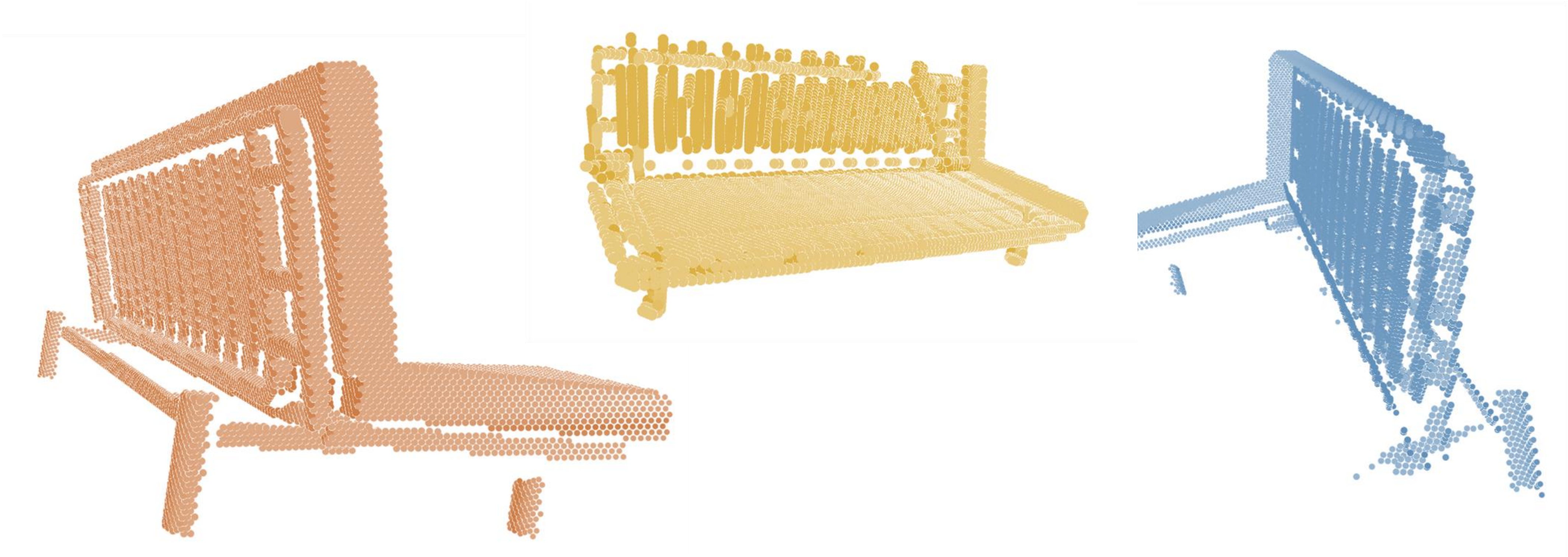
Let’s minimize the distance to the tangent plane!

No longer a closed form solution

In practice: faster convergence than point-point ICP

Global Registration

- Given shapes in arbitrary positions, find the alignments



- Often approximate – to be refined (e.g., by ICP)

Global Registration

- Various Approaches
  - Exhaustive Search
  - Normalization
  - Random Sampling
  - Invariance

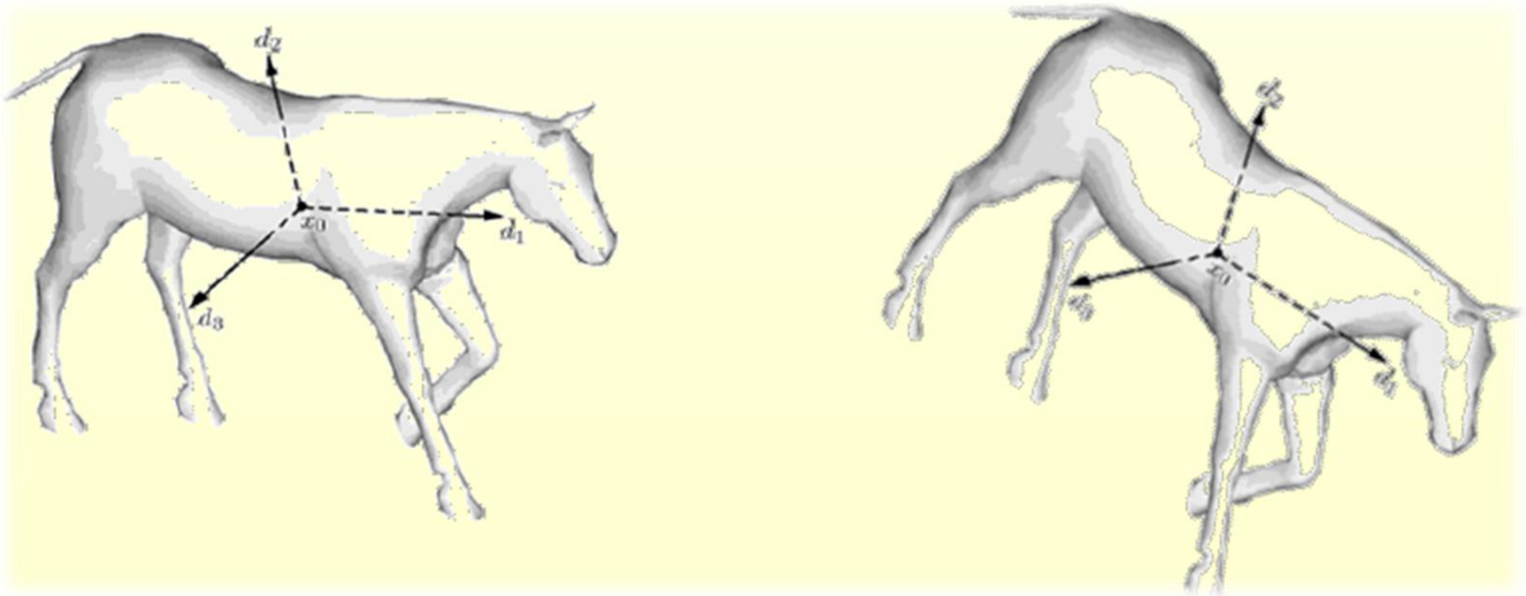
Global Registration: Exhaustive Search

- Compare all alignments
  - Sample space of possible initial alignments
  - Find alignment at which models are closest
  - (Refine with ICP)

  - Can find optimal result
  - Can be unnecessarily slow
  - Often intractable for larger degrees of freedom (e.g., non-rigid deformations)

Global Registration: Normalization

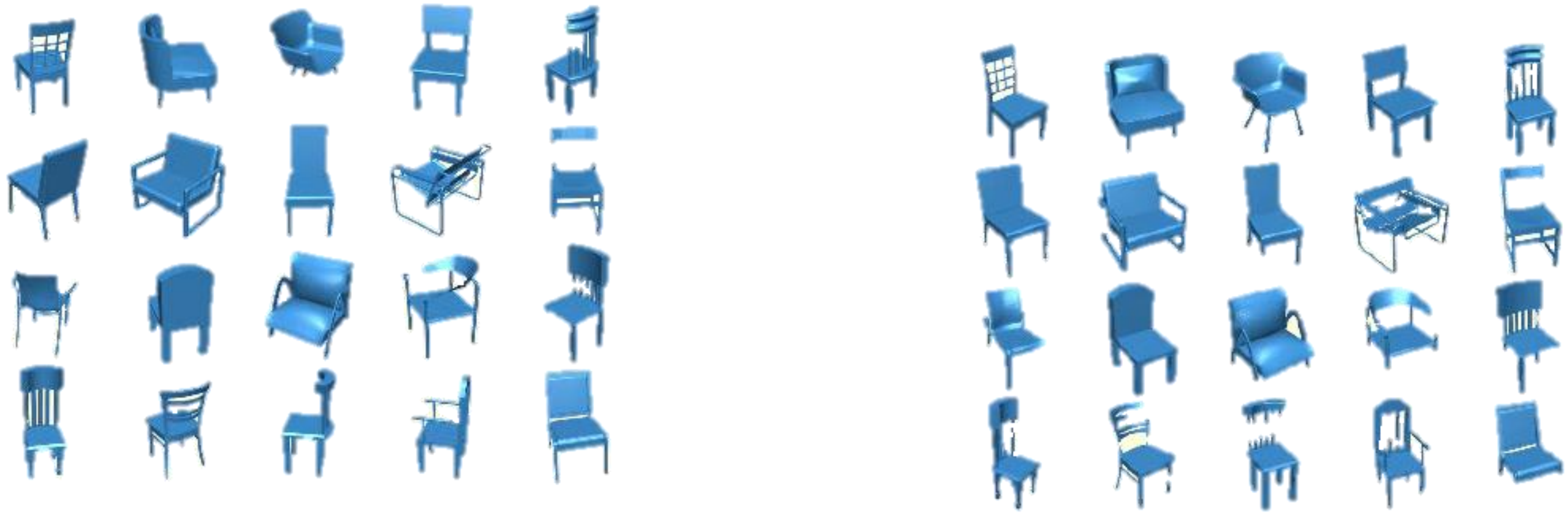
- If: only a handful of initial configurations



- Center all shapes at the origin and use PCA to find the principal directions

Global Registration: Normalization

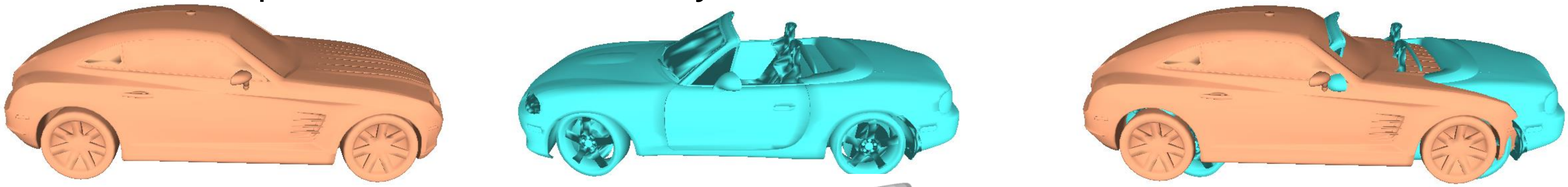
- Align a collection of shapes



- Works well for complete shapes with no noise

Global Registration: Normalization

- Problems with PCA:
  - Principal axes not consistently oriented



  - Unstable axes:

  - Partial Similarity

Global Registration: Random Sampling

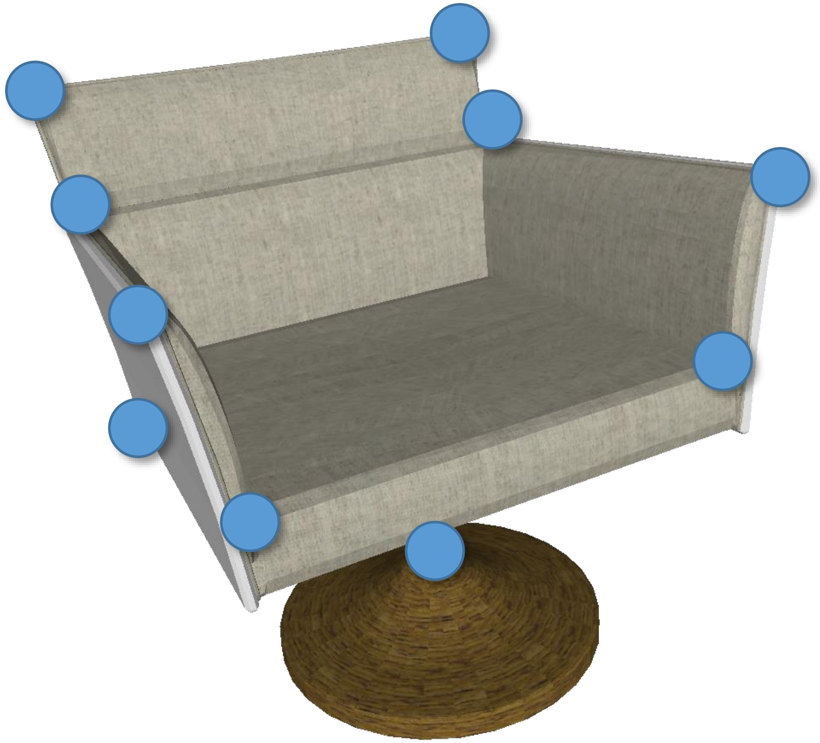
- RANSAC

- Iterate:
  - Pick random pair of n (3 for rigid) points on both shapes
  - Estimate alignment, and check for error

- Guess and verify

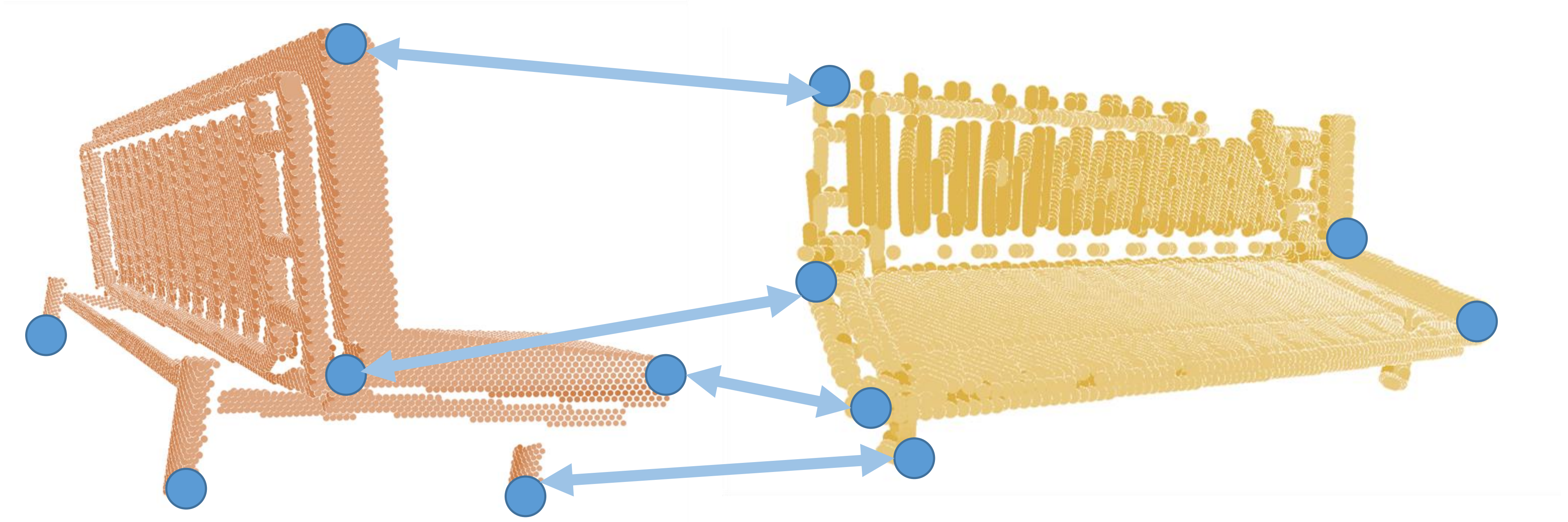
Global Matching: Invariant Features

- Characterize shape with properties that are invariant under the desired transform
- Often trade-off: invariance vs informative
- Identify salient feature points
- Compute informative descriptors



Matching Feature Points

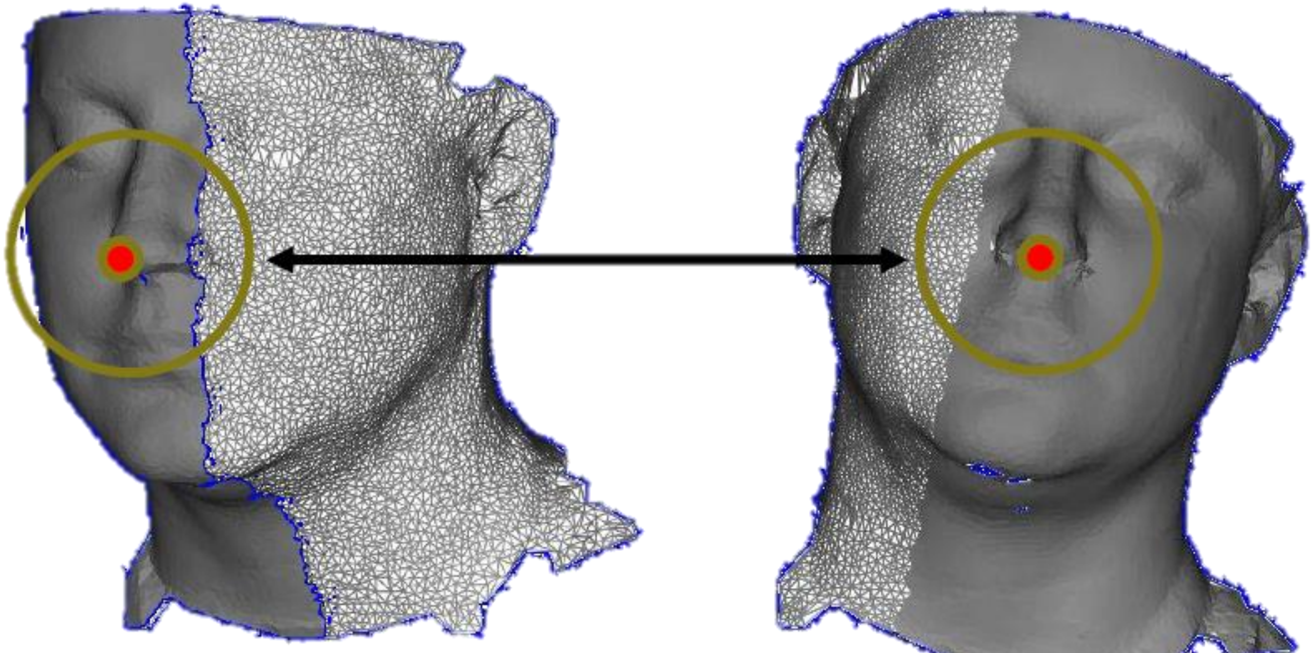
- Find feature points on each shape
- Establish correspondences
- Compute transform that aligns correspondences



How to establish correspondence?

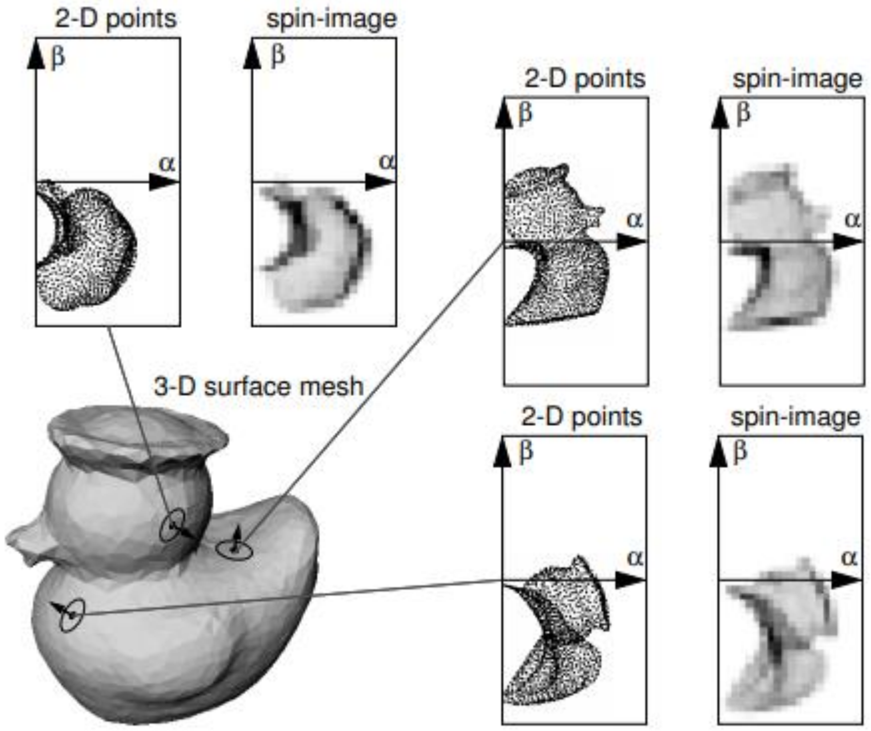
- When do two points on different shapes/scans represent the same feature?

- Are the surrounding regions similar?

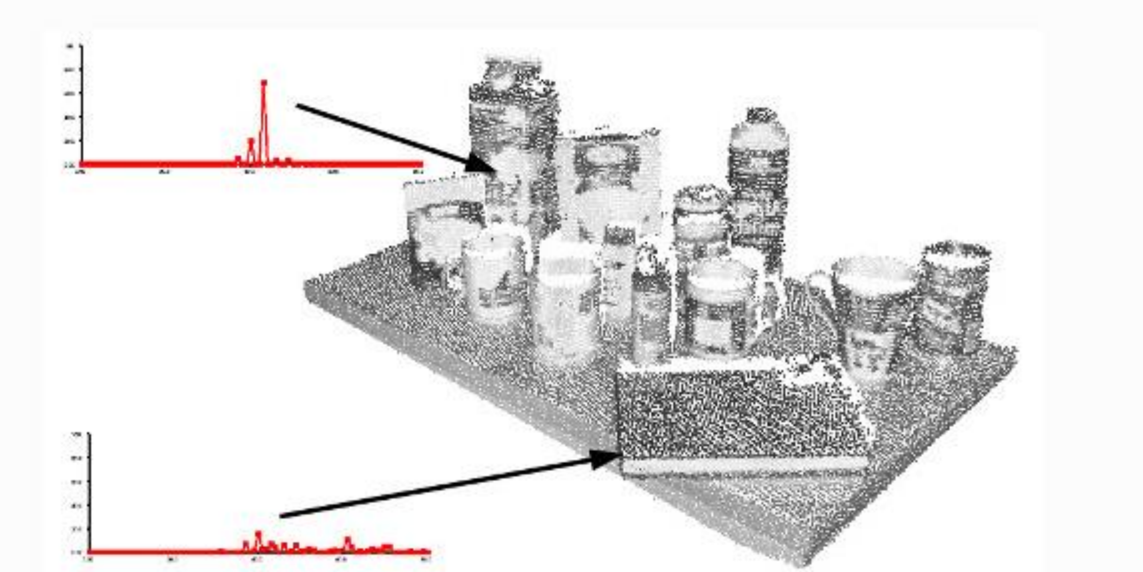


Feature descriptors summarize surrounding regions

Shape Descriptors



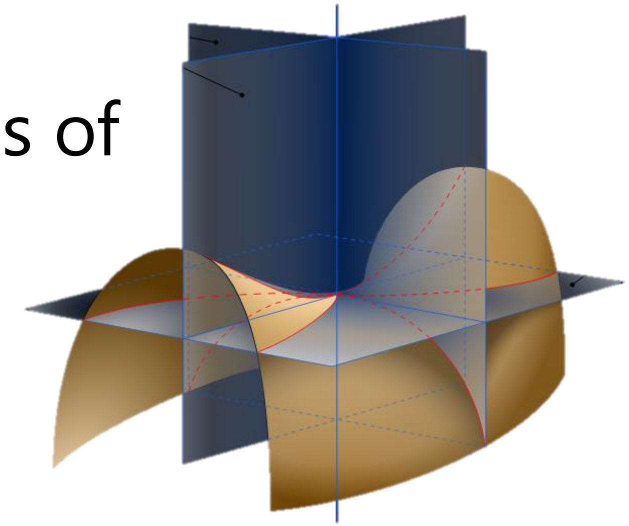
Spin Images [Johnson and Hebert '99]

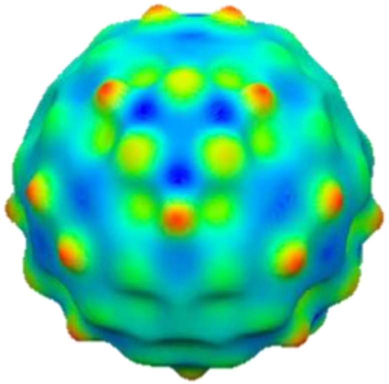


(Fast) Point Feature Histograms [Rusu et al. '09]

Classical Descriptors

- Curvature
- Differential features describe characteristics of surrounding surface

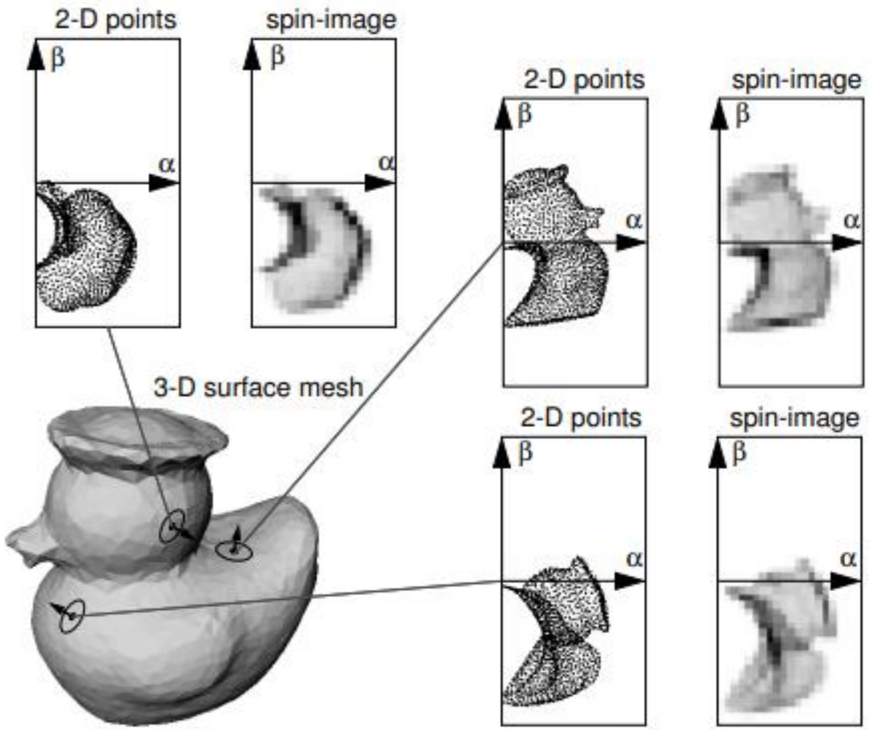




- Differential features can be noisy on meshes and real-world captured data

Classical Descriptors: Spin Images

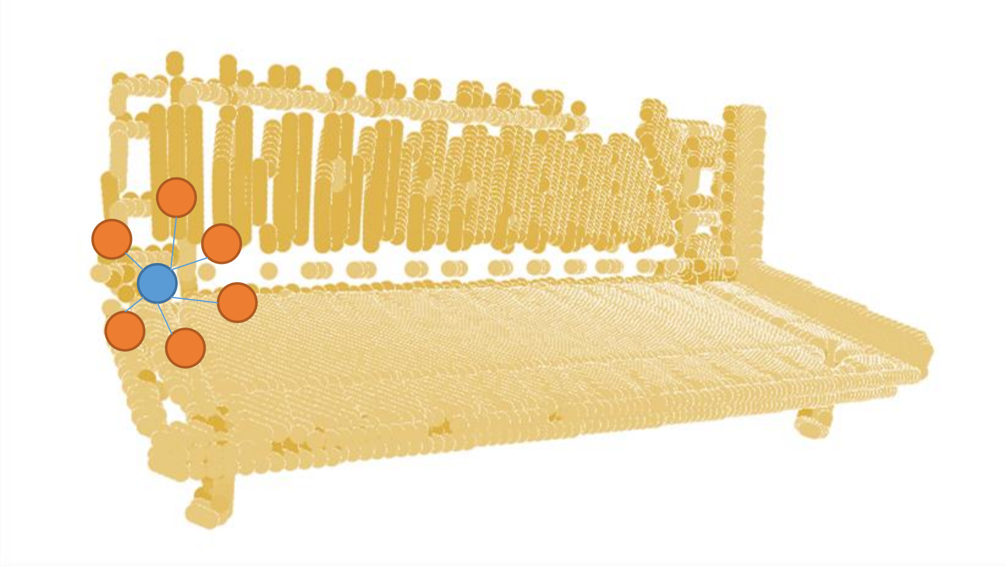
- Create image associated with neighborhood of a feature point
- "Spin" image along point normal
- Collect contributions of each other point by their distance to tangent and distance to normal
- 2D spin image comparison

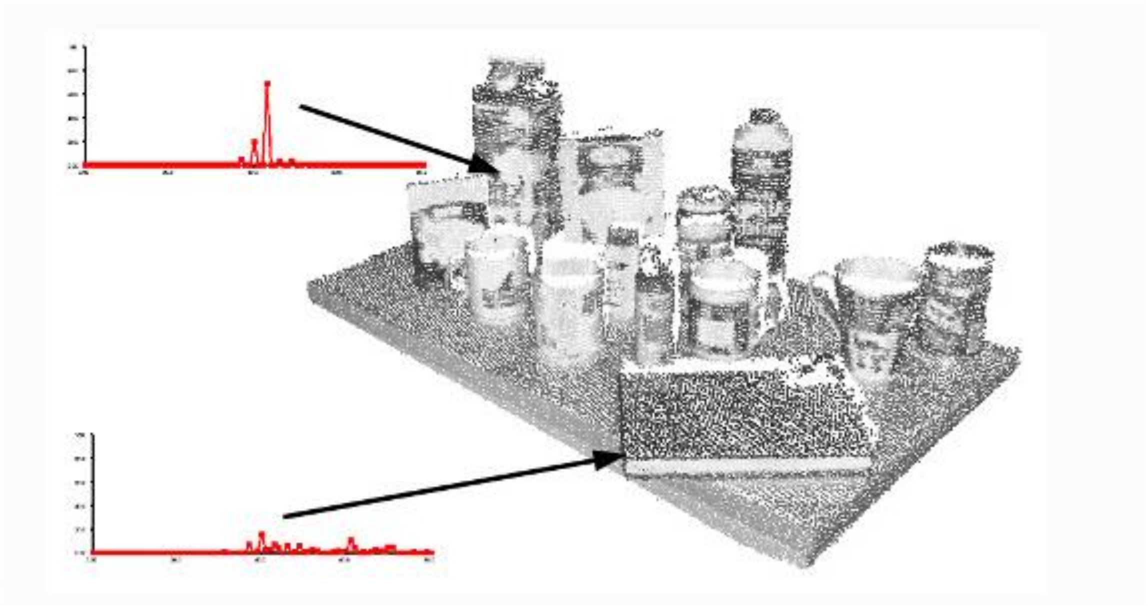


Spin Images [Johnson and Hebert '99]

Classical Descriptors: Point Feature Histograms

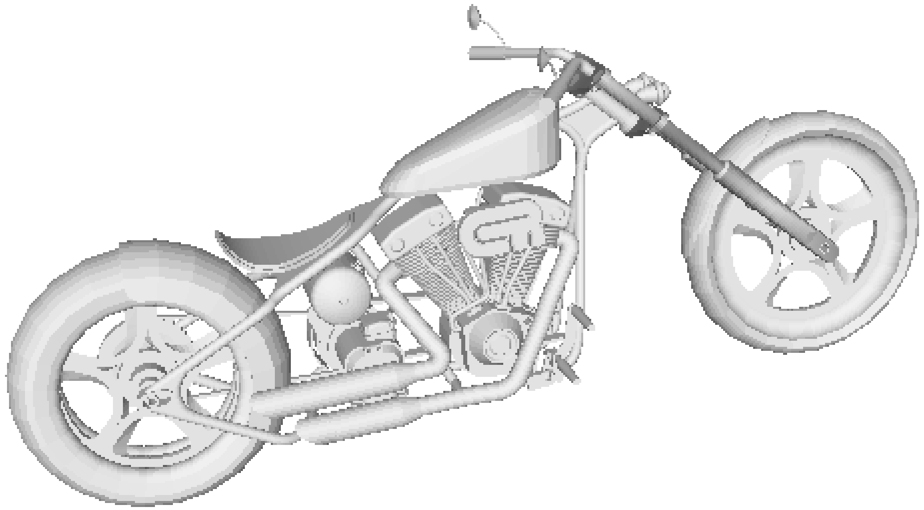
- For a point p find its k neighbors $\{q_i\}$
- Compute histogram from tuples of $\{(p, q_i)\}$ based on distances, normal, optionally curvature etc.

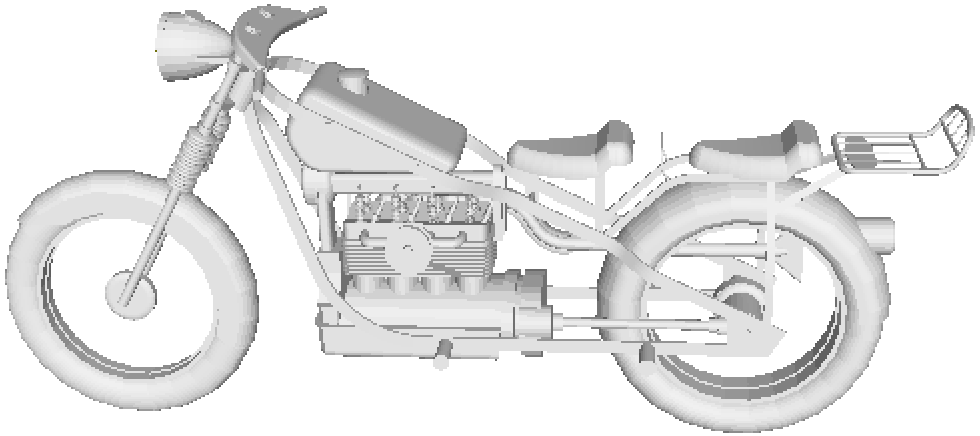




Global Shape Similarity

- Do two 3D models represent the same or similar shapes?



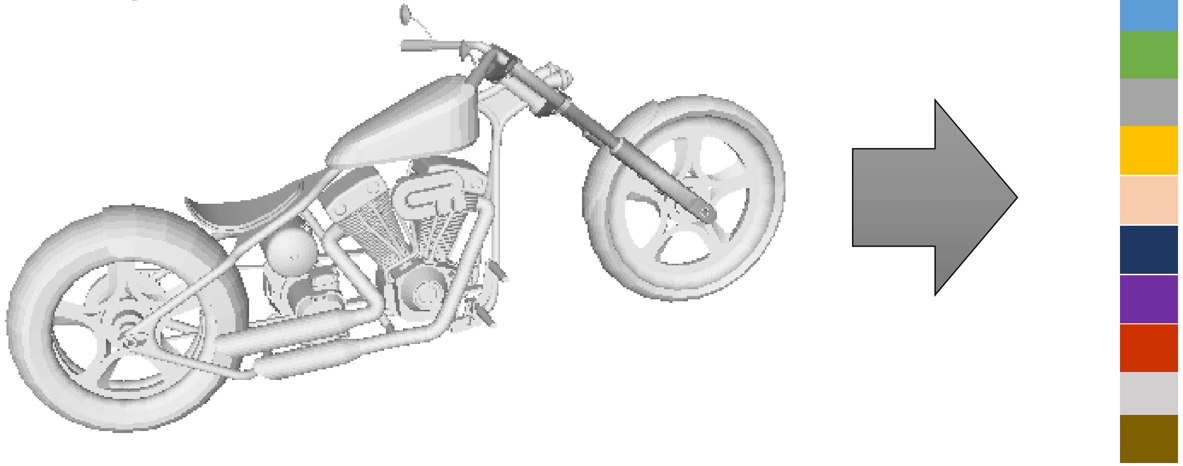


3D models can have different representations, tessellations, topologies, etc.

Global Shape Similarity

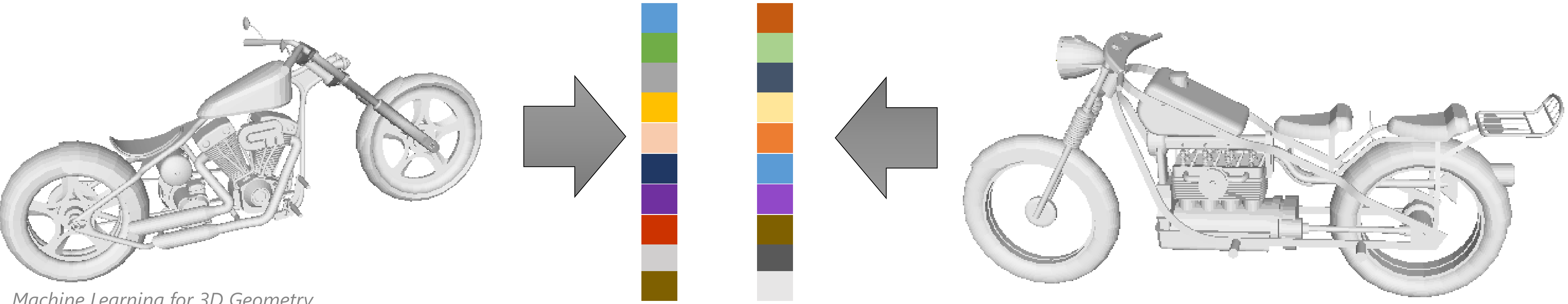
- Do two 3D models represent the same or similar shapes?

- Represent each model by a shape descriptor
  - Structured, abstraction of a 3D model
  - Captures salient shape information
  - Typically a high-dimensional vector



Global Shape Similarity

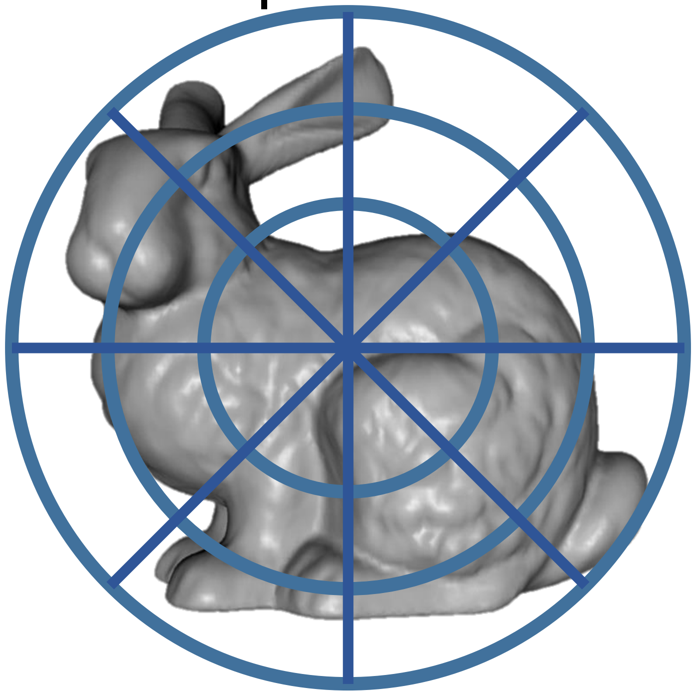
- Do two 3D models represent the same or similar shapes?
- Represent each model by a shape descriptor
- Compare shapes by comparing descriptors



Global Shape Descriptors

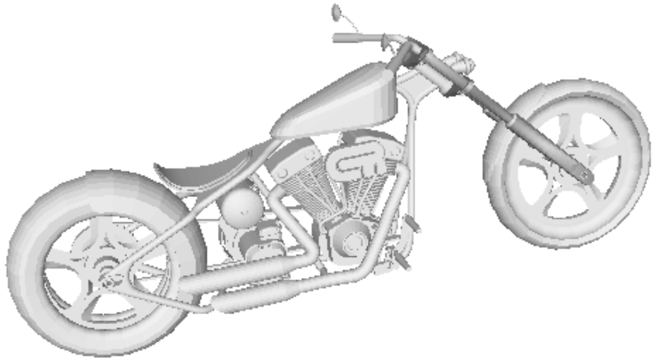
- Simple descriptor: Shape Histograms

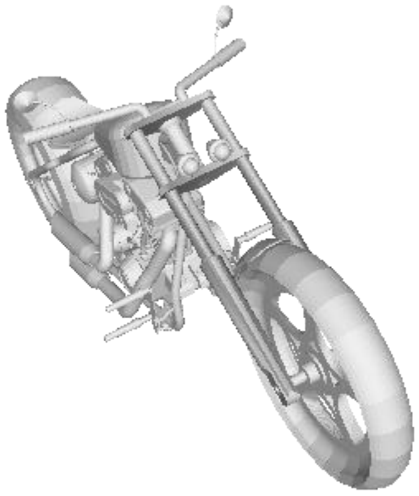
- Store histogram of how much surface area resides within different concentric shells in space



Global Shape Descriptors

- Should be invariant to rigid transforms of the shapes (rotation, translation)

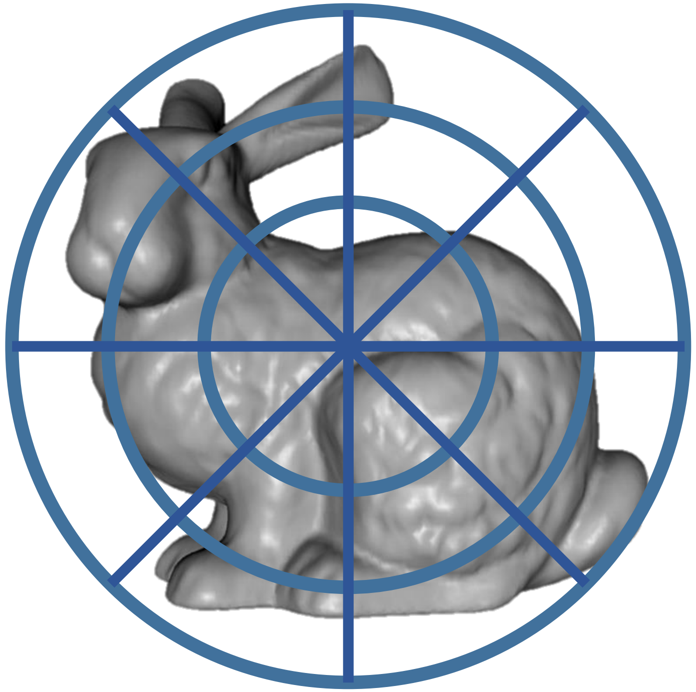


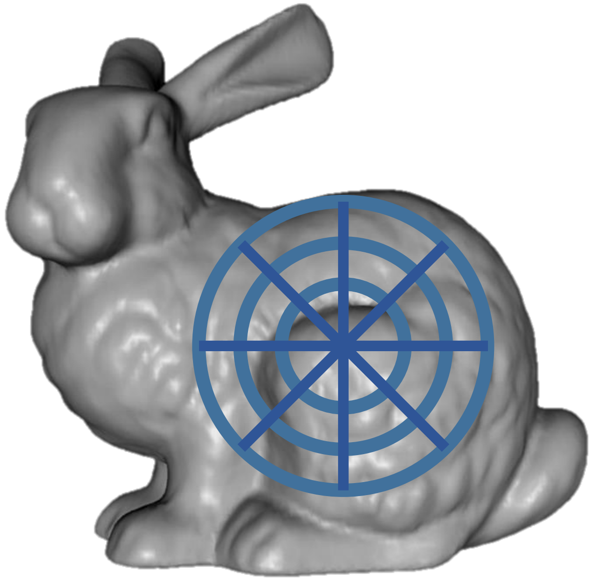


- Compute descriptors at the optimal alignment of shapes

Global -> Local Descriptors

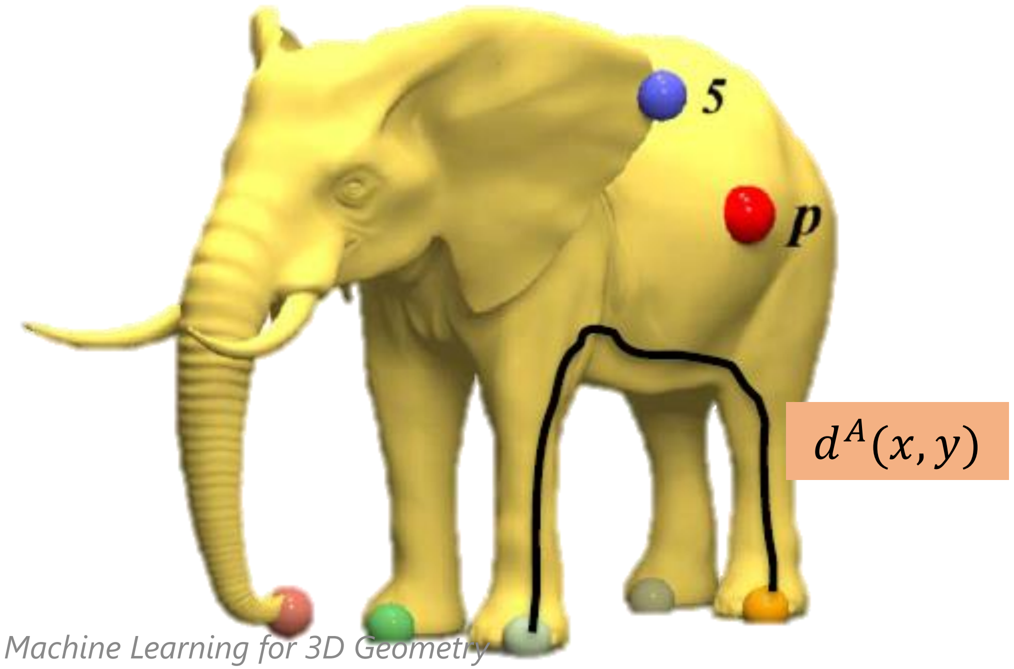
- Center and restrict around local region

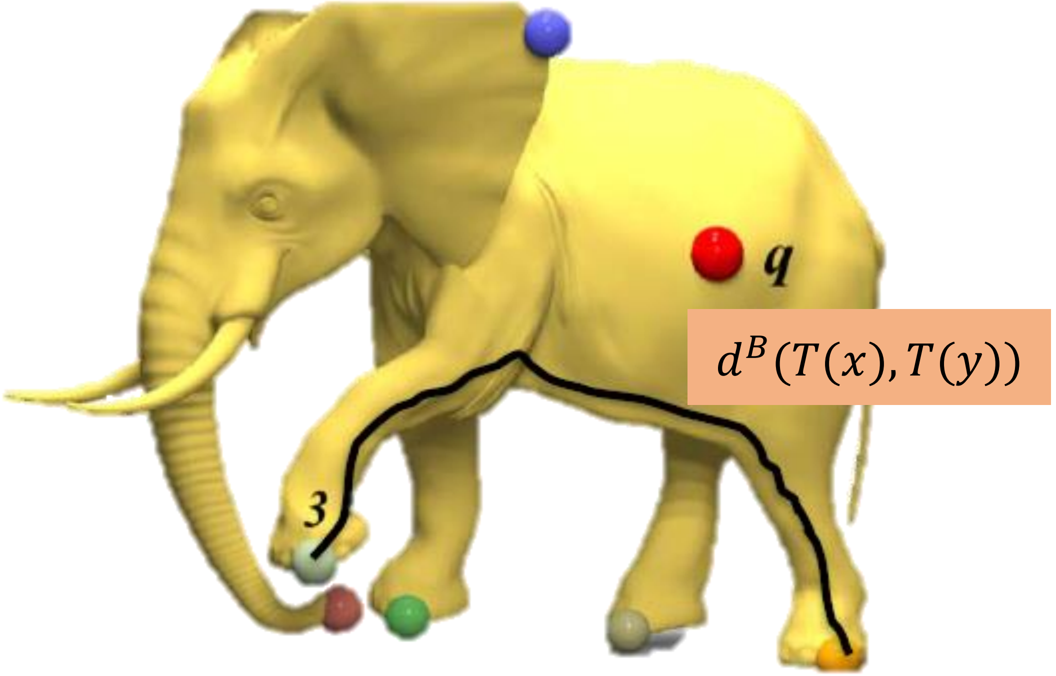




Non-Rigid Shape Matching

- Consider near-isometric cases
- Find correspondences that preserve intrinsic (geodesic) distances on the shapes

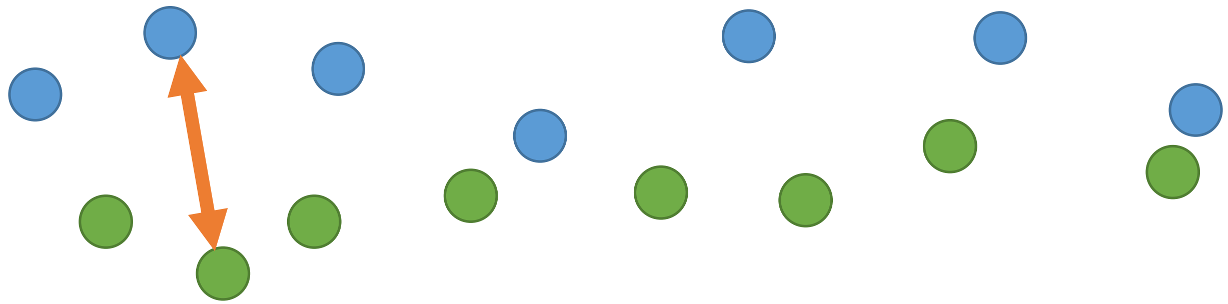




Measuring intrinsic shape similarity

- Gromov-Hausdorff distance

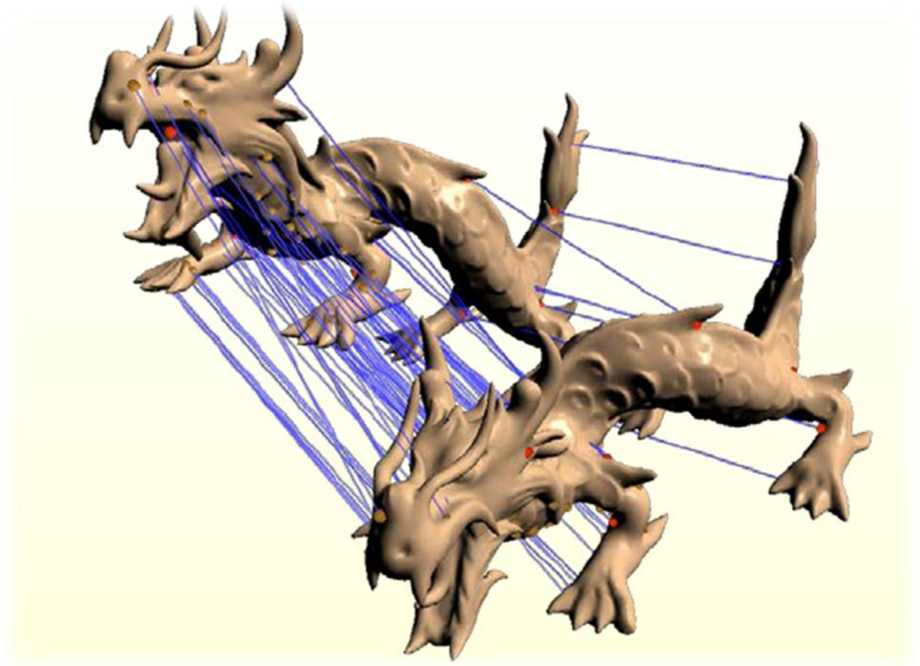
- Hausdorff distance: maximin
  - Maximum of all minimum distances between two sets of points

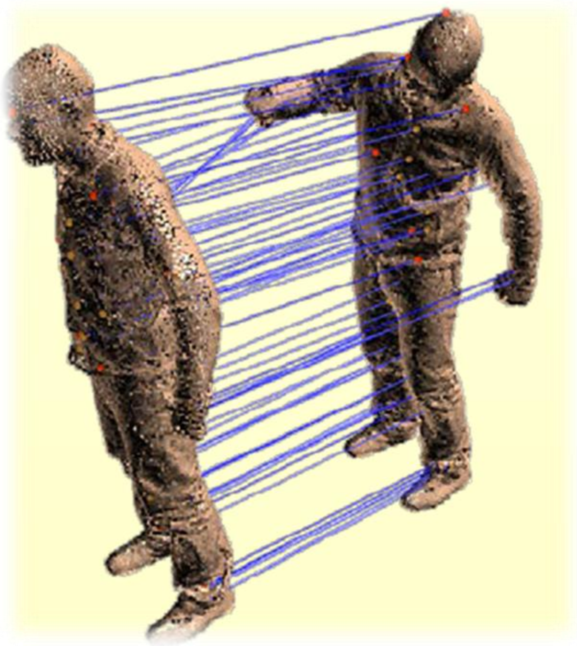


- Gromov-Hausdorff: infimum of all Hausdorff distances over mappings or correspondences
  - Over all correspondences -> difficult to compute!

Near isometries preserve local structure

- Optimal alignment can be defined, but difficult to compute
- Define descriptors of local regions -> establish good mappings

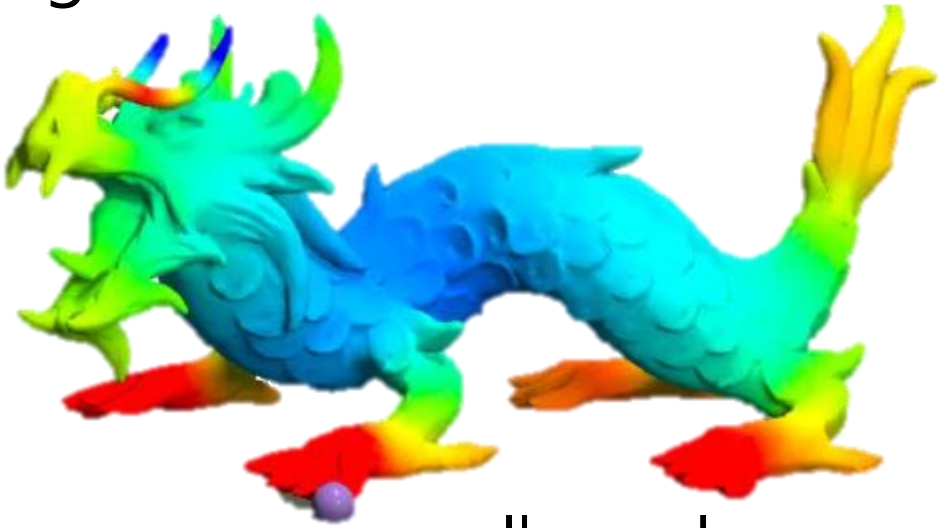




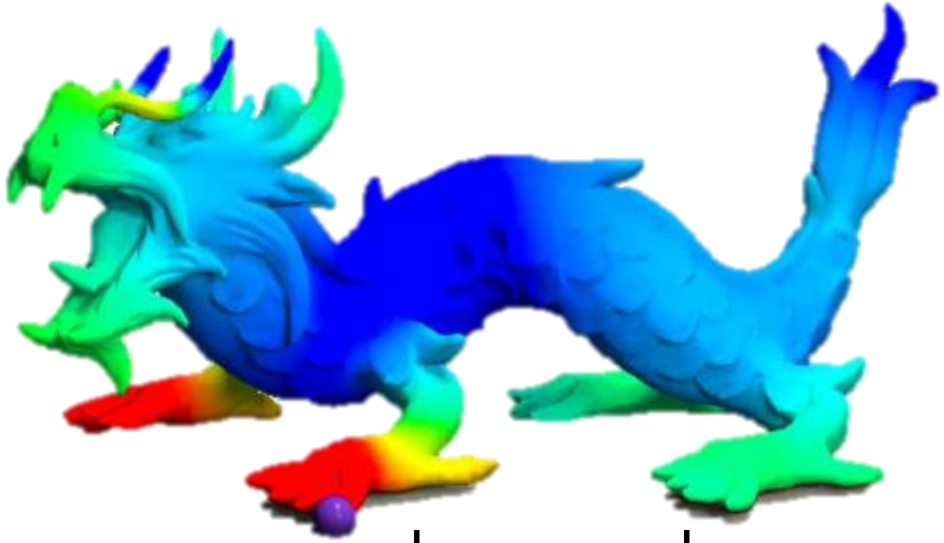
How large should a local region be?

Scale for local features?

- Given a point on a shape, find other points with similar neighborhoods



smaller scale



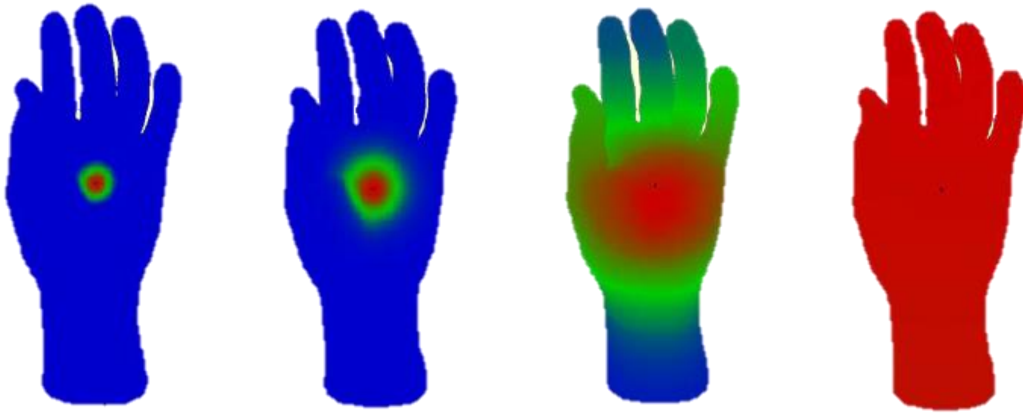
larger scale

- How to meaningfully compare neighborhoods at different scales?

Heat kernel signature

- Spectral shape analysis
- Heat kernel $k_t(x, y)$: amount of heat transferred from x to y in time t

$$f(x,t) = \int_M k_t(x,y) f(y) dy$$

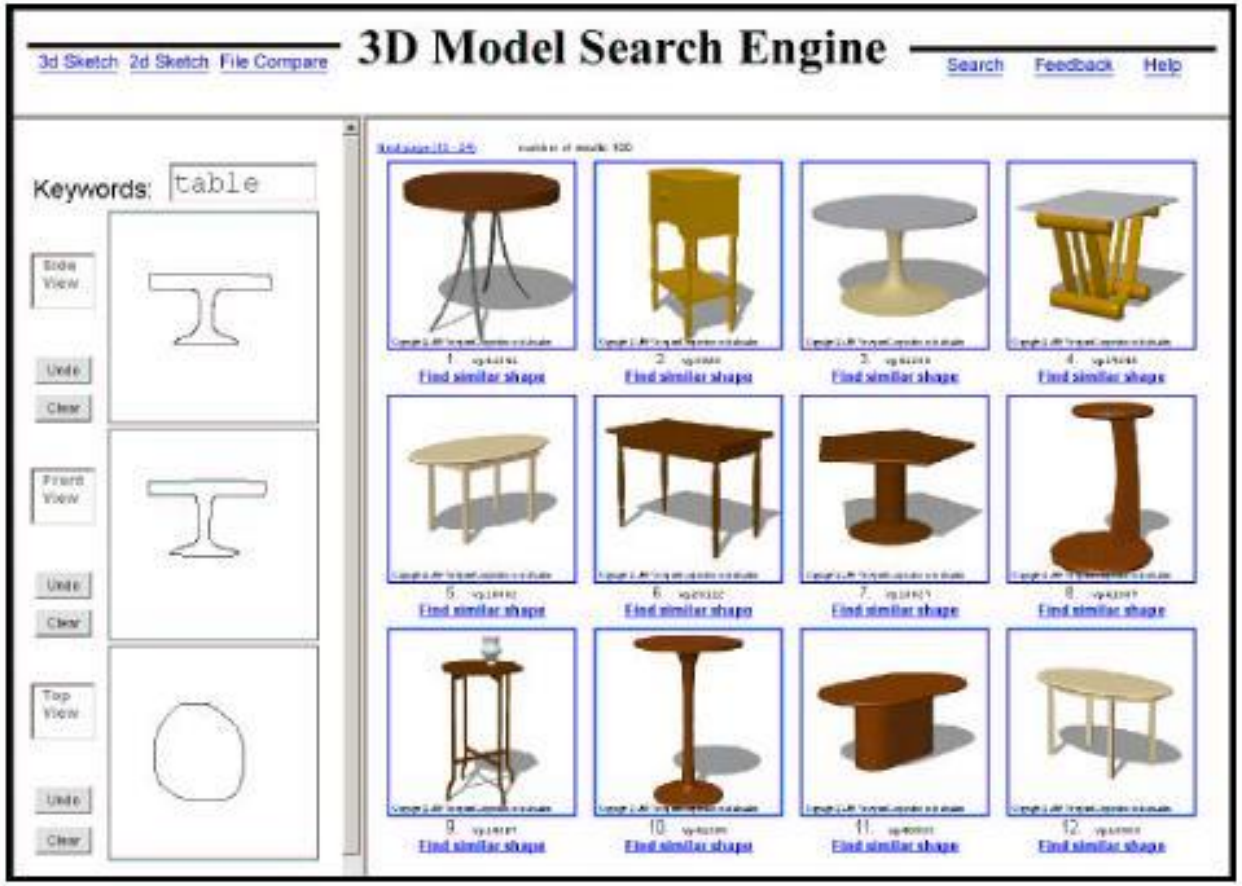


- Invariant under isometric deformations
- Multi-scale description

- In practice: can be difficult to apply to real scenarios (noisy, partial data)

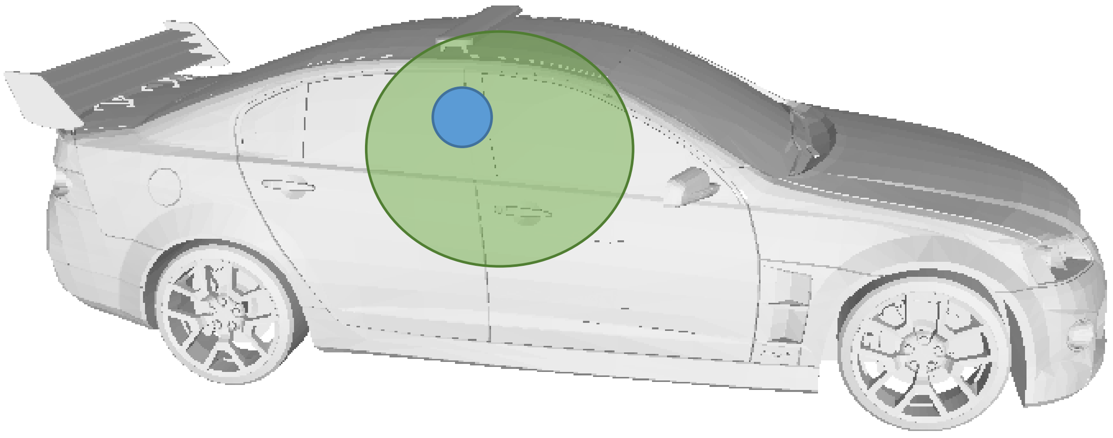
[Sun et al. '09]

Shape Search

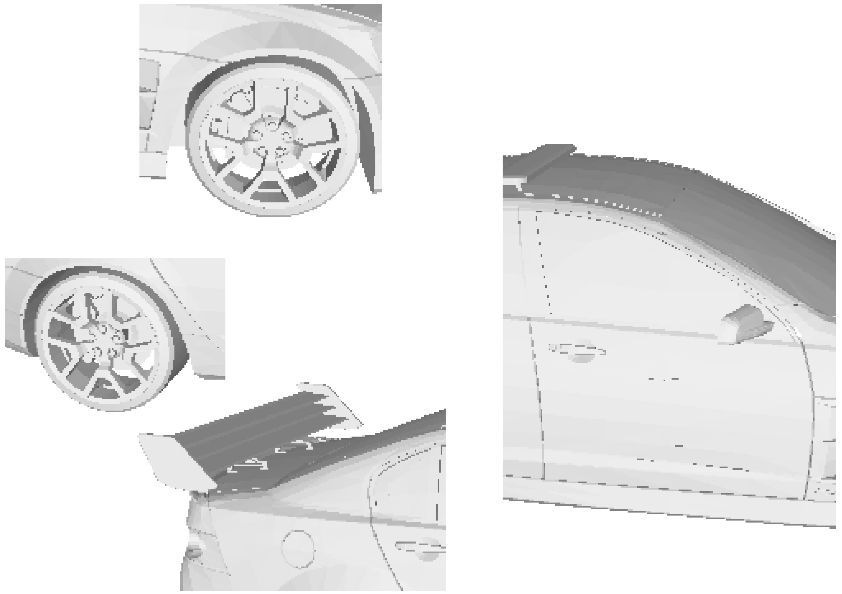


Shape Search: Bag of Words

Geometric Words



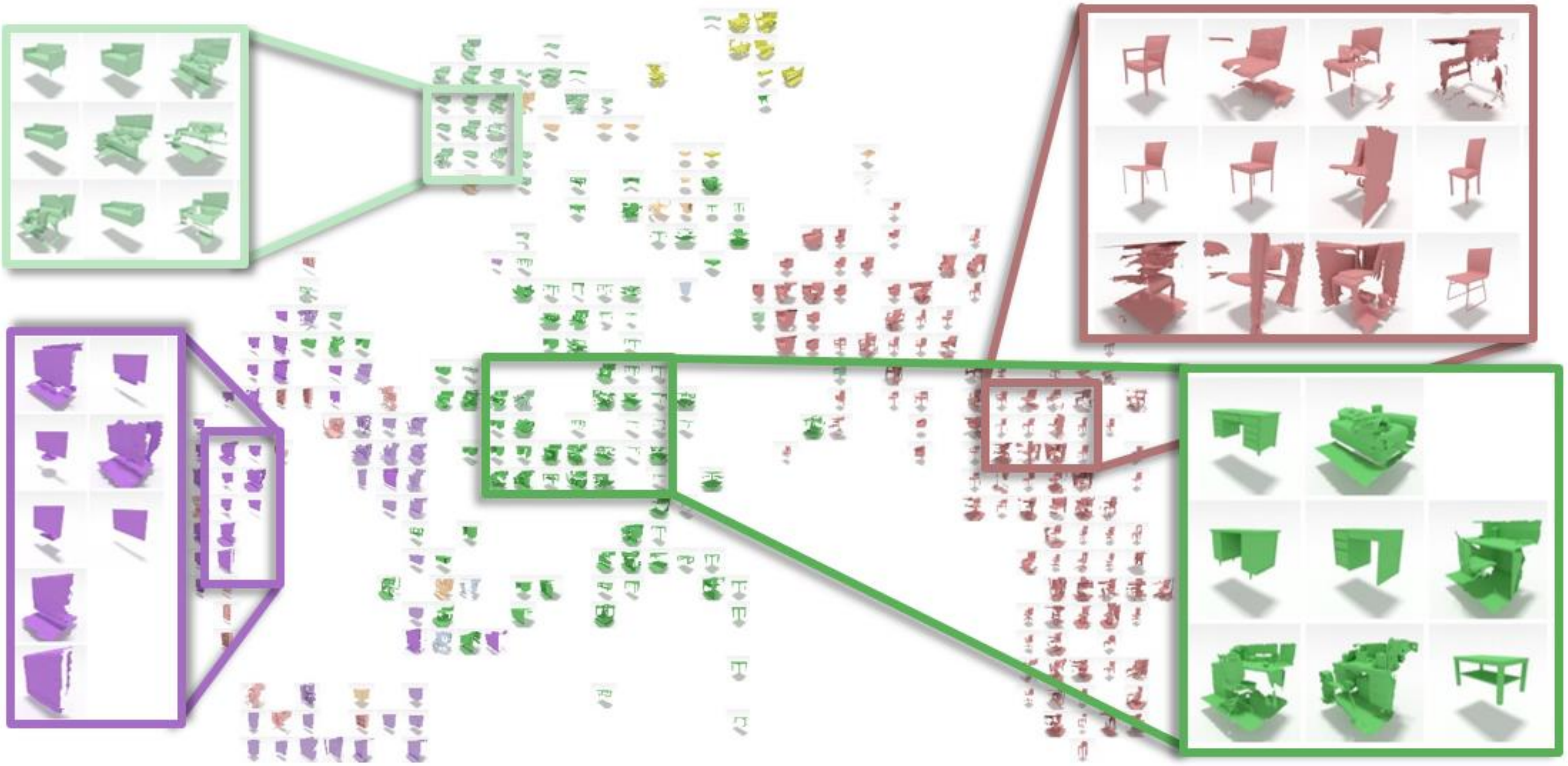
Features + Descriptors



Parts

Shape Search

- Retrieval through descriptor space



Learned Shape Search

Query

Nearest Neighbor Retrievals

[Dahnert et al. '19]

Additional references

- Efficient variants of the ICP algorithm [Rusinkiewicz et al. ‘01]
  - http://www.pcl-users.org/file/n4037867/Rusinkiewicz_Effcient_Variants_of_ICP.pdf
- Sparse Principal Component Analysis [Zou et al ‘16]
  - https://web.stanford.edu/~hastie/Papers/spc_jcgs.pdf