Shape Correspondence

Richard (Hao) Zhang

CMPT 464/764: Geometric Modeling in Computer Graphics

Lecture 11

Reading

A Survey on Shape Correspondence

Oliver van Kaick[1], Hao Zhang[1], Ghassan Hamarneh[1], Daniel Cohen-Or[2]

[1] School of Computing Science, Simon Fraser University, Canada
[2] School of Computer Science, Tel Aviv University, Israel

Computer Graphics Forum, 2011.

Abstract
We review methods designed to compute correspondences between geometric shapes represented by triangle meshes, contours, or point sets. This survey is motivated in part by recent developments in space-time registration, where one seeks a correspondence between non-rigid and time-varying surfaces, and semantic shape analysis, which underlines a recent trend to incorporate shape understanding into the analysis pipeline. Establishing a meaningful correspondence between shapes is often difficult since it generally requires an understanding of the structure of the shapes at both the local and global levels, and sometimes the functionality of the shape parts as well. Despite its inherent complexity, shape correspondence is a recurrent problem and an essential component of numerous geometry processing applications. In this survey, we discuss the different forms of the correspondence problem and review the main solution methods, aided by several classification criteria arising from the problem definition. The main categories of classification are defined in terms of the input and output representation, objective function, and solution approach. We conclude the survey by discussing open problems and future perspectives.

1. Introduction

Finding a meaningful correspondence between two or more shapes is a fundamental shape analysis task. The problem can be generally stated as: *given input shapes* $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_N$*, find a meaningful relation (or mapping) between their elements*, e.g., see Figure 1. Under different contexts, the problem has also been referred to as registration, alignment, or simply matching. Shape correspondence is a key algorithmic component in tasks such as 3D scan alignment and space-time reconstruction, as well as an indispensable prerequisite in diverse applications including attribute transfer, shape interpolation and statistical modeling.

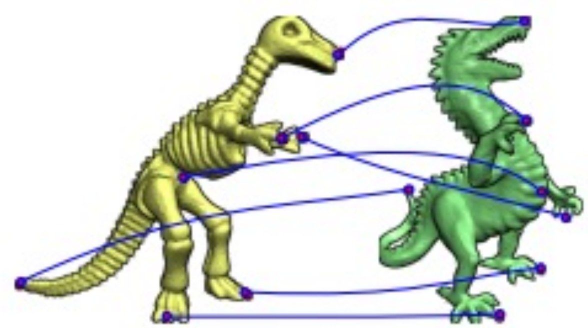


Figure 1: *A meaningful correspondence (blue lines) between a sparse set of feature points on two shapes. Note the large amount of geometric variations between the shapes which*

Correspondence

Webster dictionary definition

- the agreement of things with one another
- a particular similarity
- a relation between sets in which each member of one set is associated with one or more members of the other

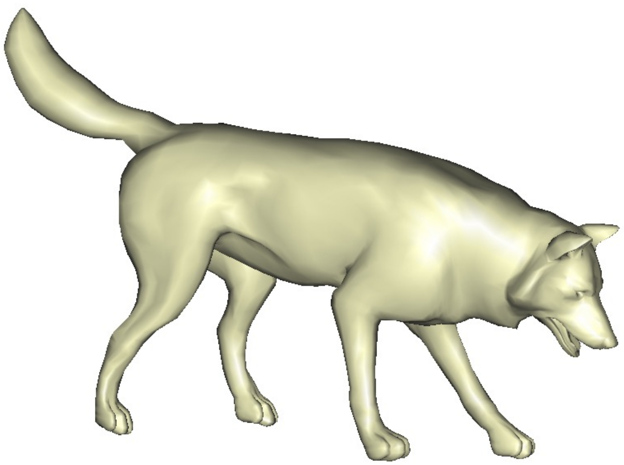
Correspondence

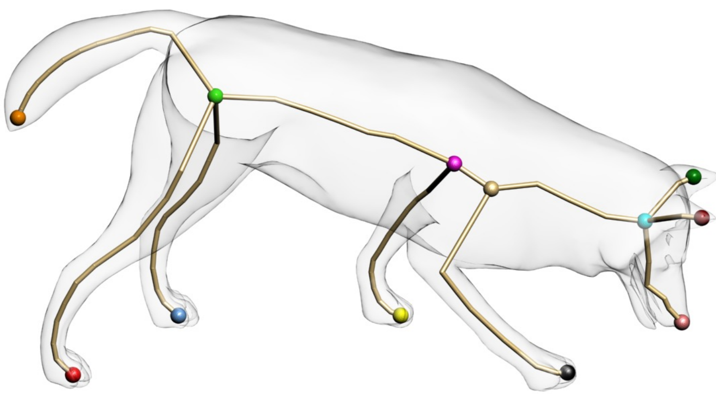
Webster dictionary definition

- the agreement of things with one another
- a particular similarity
- a relation between sets in which each member of one set is associated with one or more members of the other

Focus on shapes

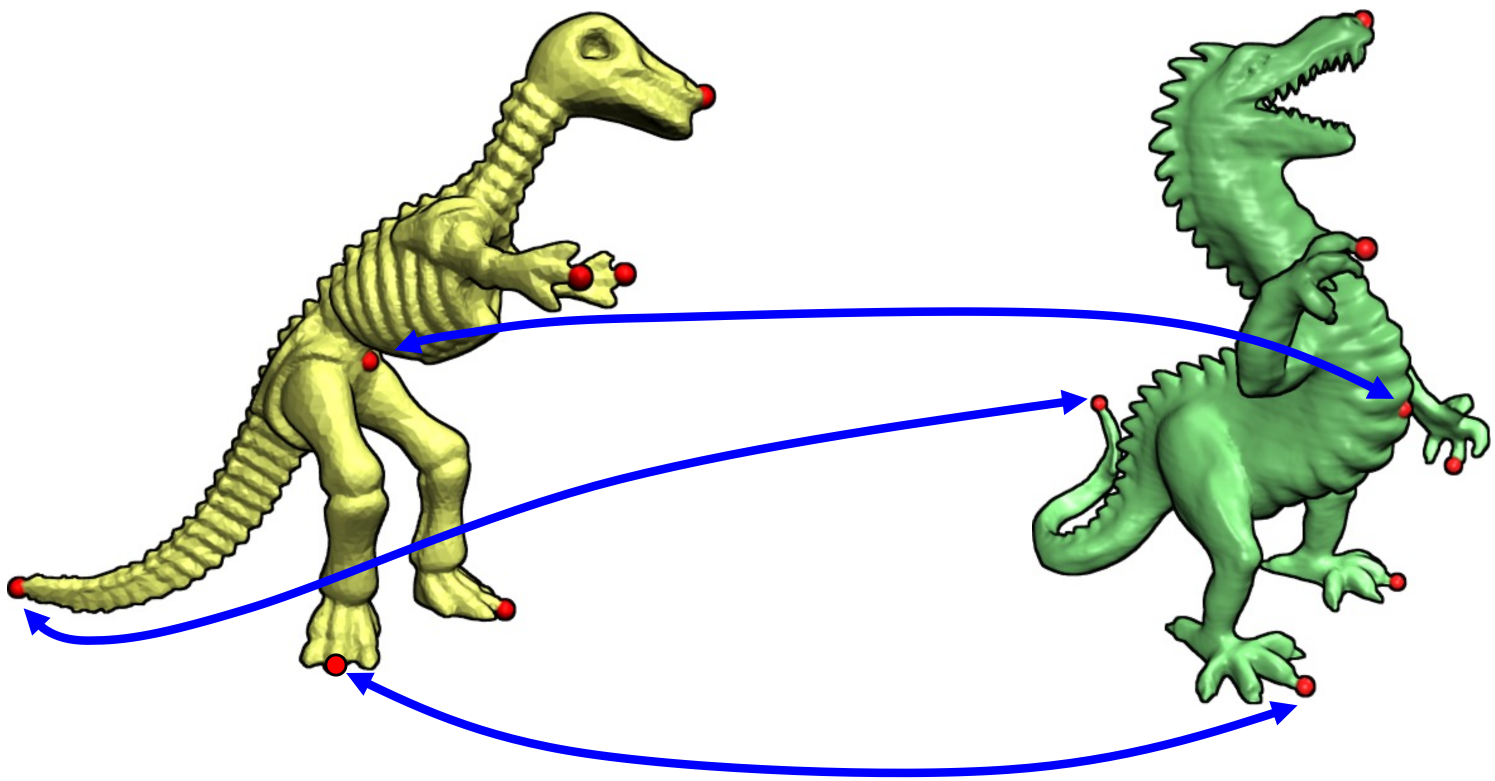
- Correspondence between **shape** representations





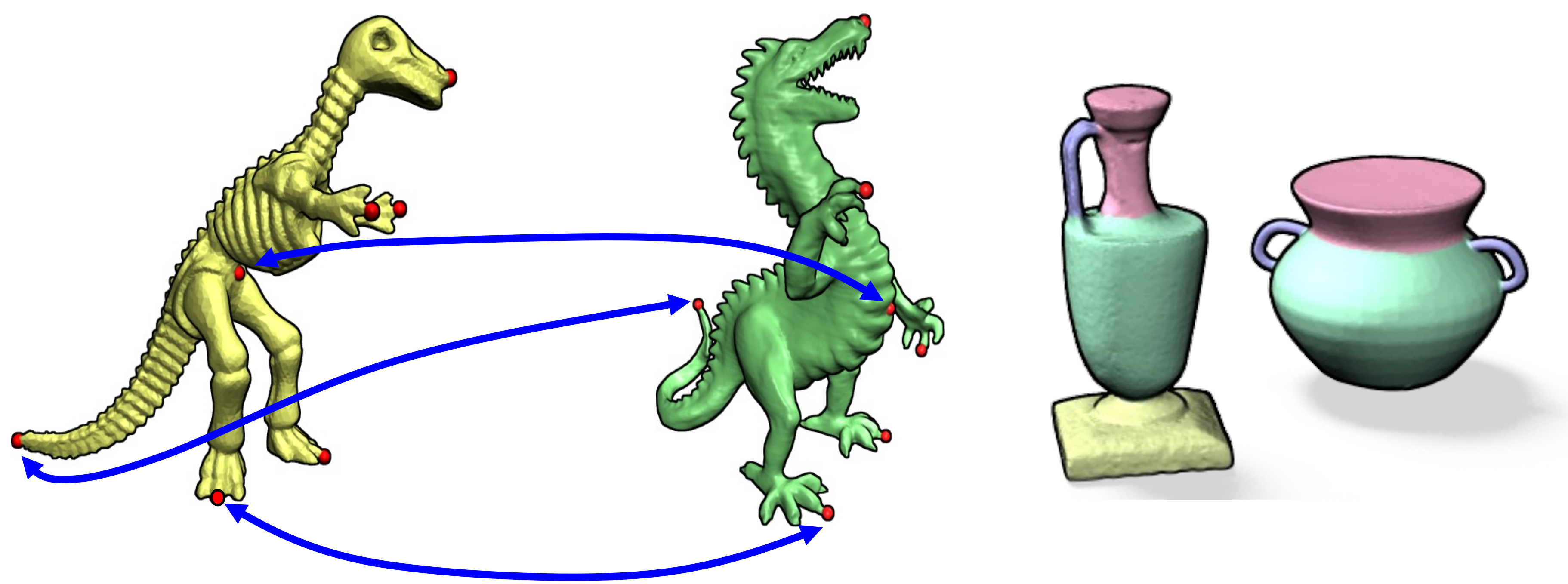
Problem statement

- Given *n* input shapes, search for a meaningful relation *R* between their elements



Why is correspondence hard?

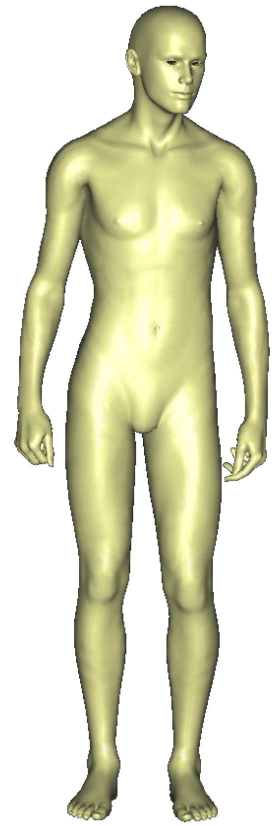
- Given *n* input shapes, search for a $\boxed{\text{meaningful}}$ relation *R* between their elements



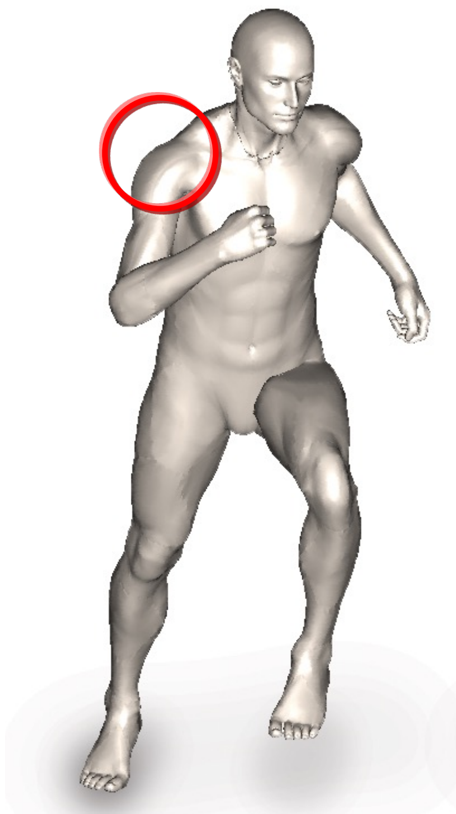
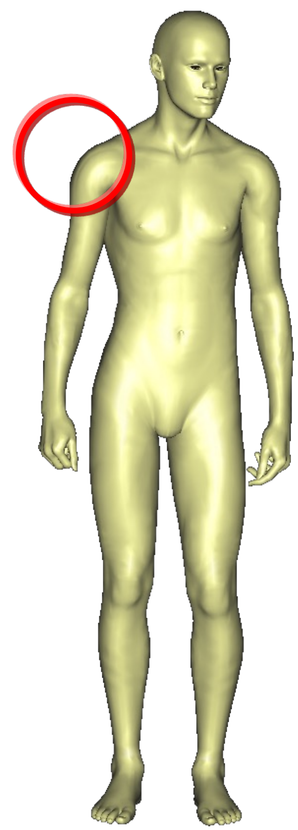
One interpretation

- Appropriate notion of “similarity” or “agreement”
- Similarity: corresponding points/regions look similar — local geometric similarity
- Agreement: close-by points should match close-by points — proximity or distortion

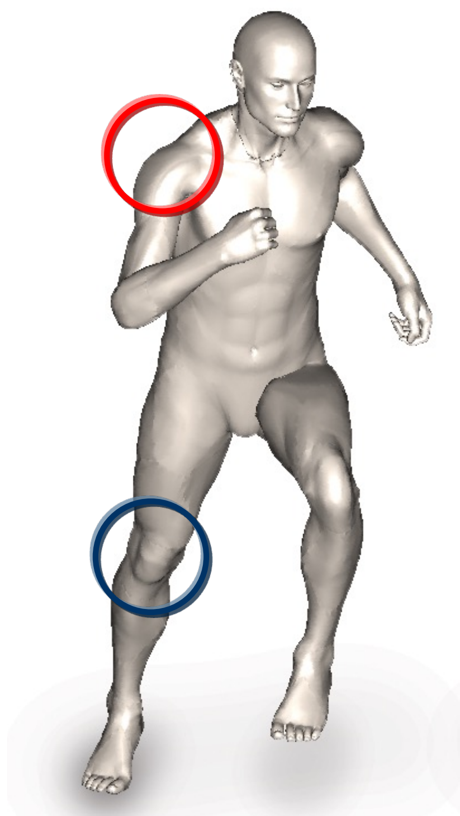
Local similarity + distortion

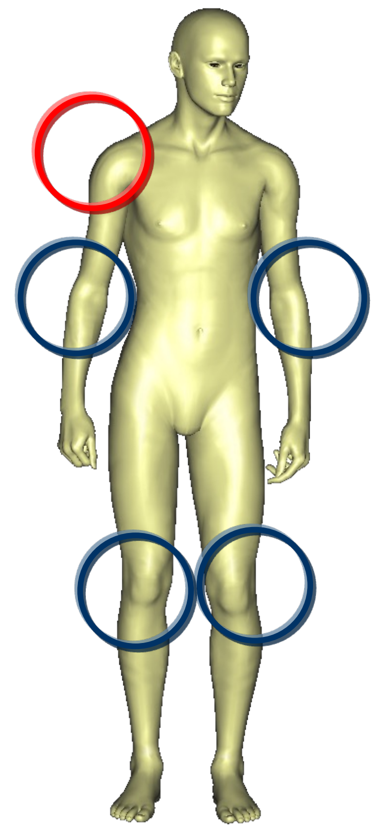


Local similarity

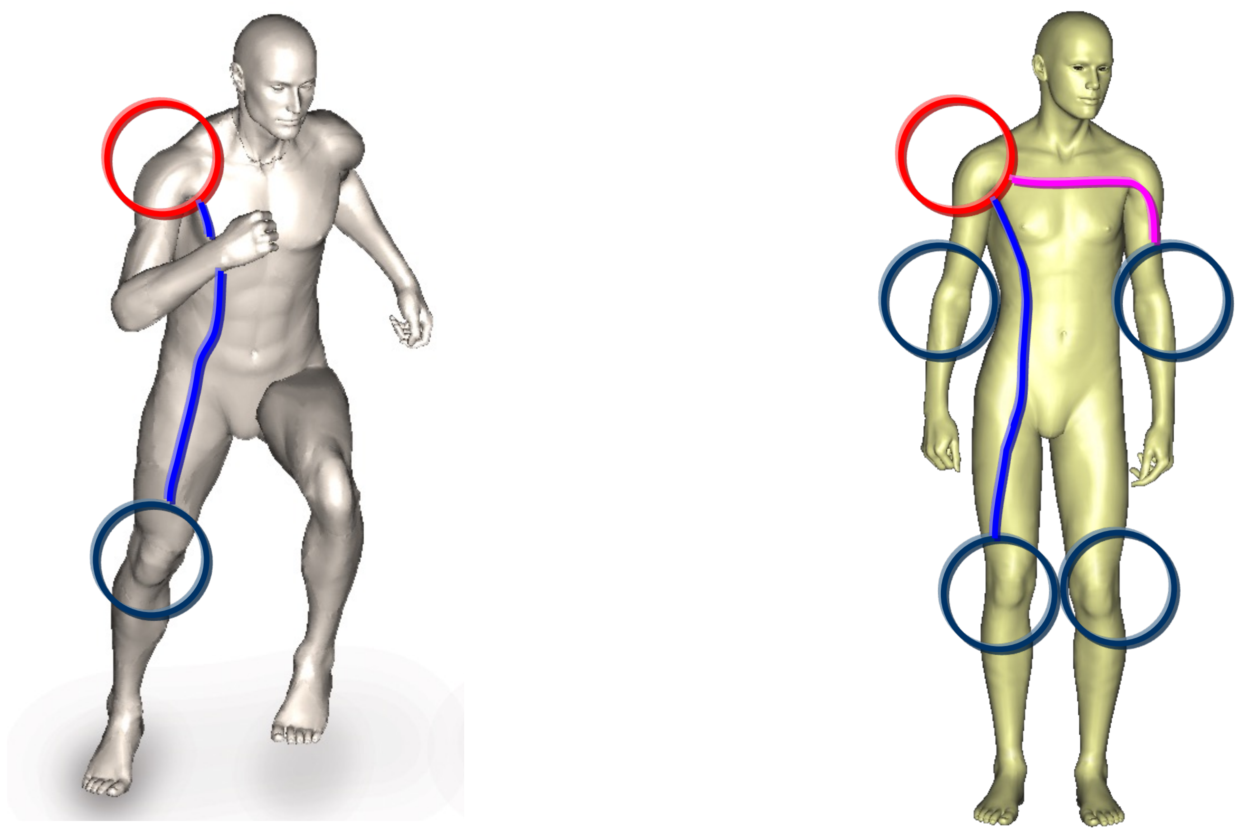



Local similarity

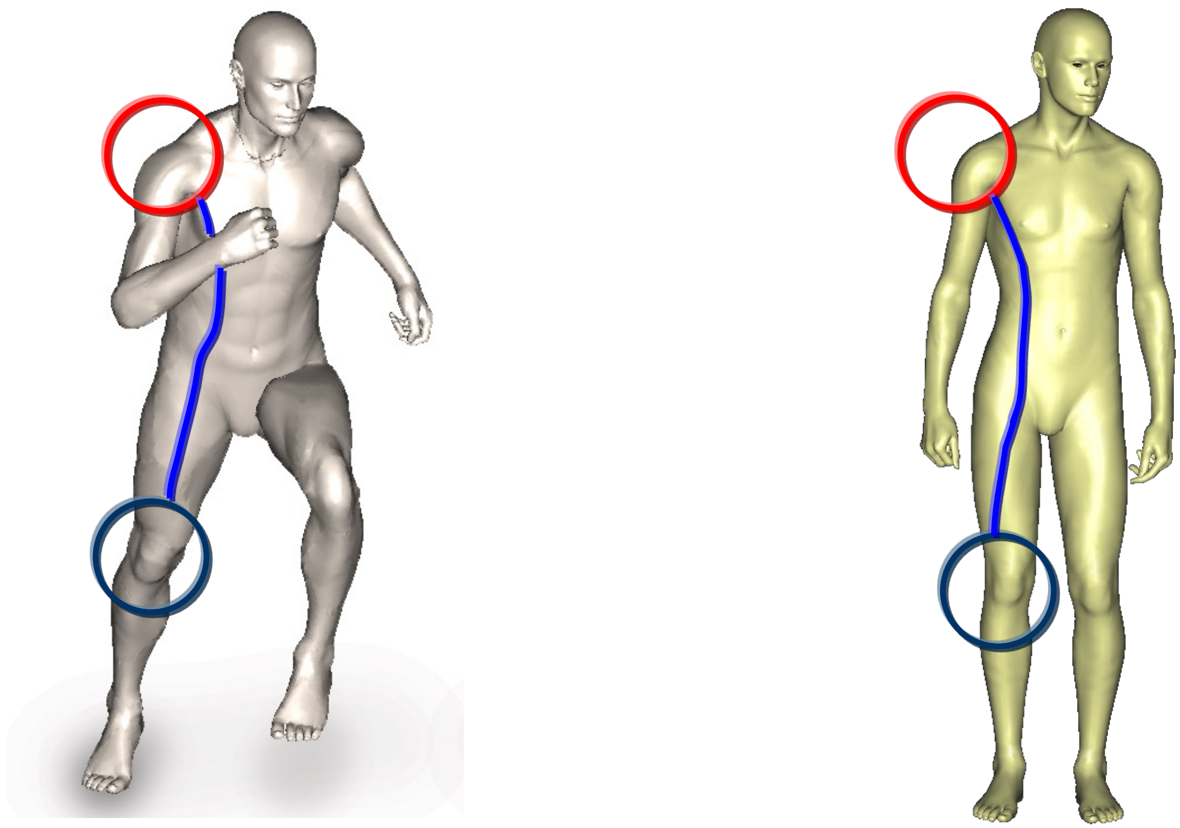




Proximity or distortion

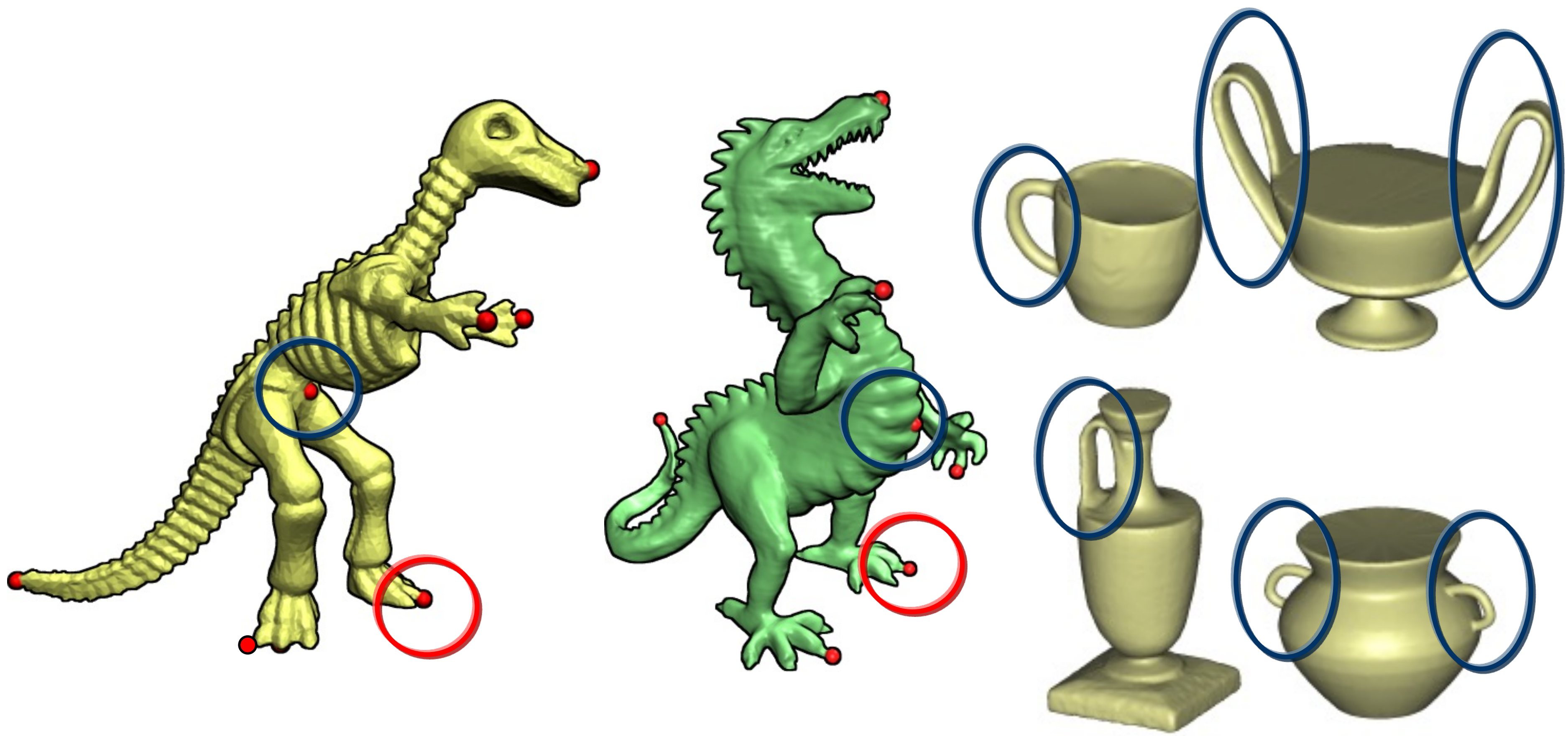


Local similarity + distortion



Not always straightforward

- Local similarity + distortion often does not work



Another interpretation

- Given *n* input shapes, search for a $\boxed{\text{meaningful}}$ relation *R* between their elements

Dictionary

Definitions from Oxford Languages · Learn more

se·man·tics

/səˈman(t)iks/

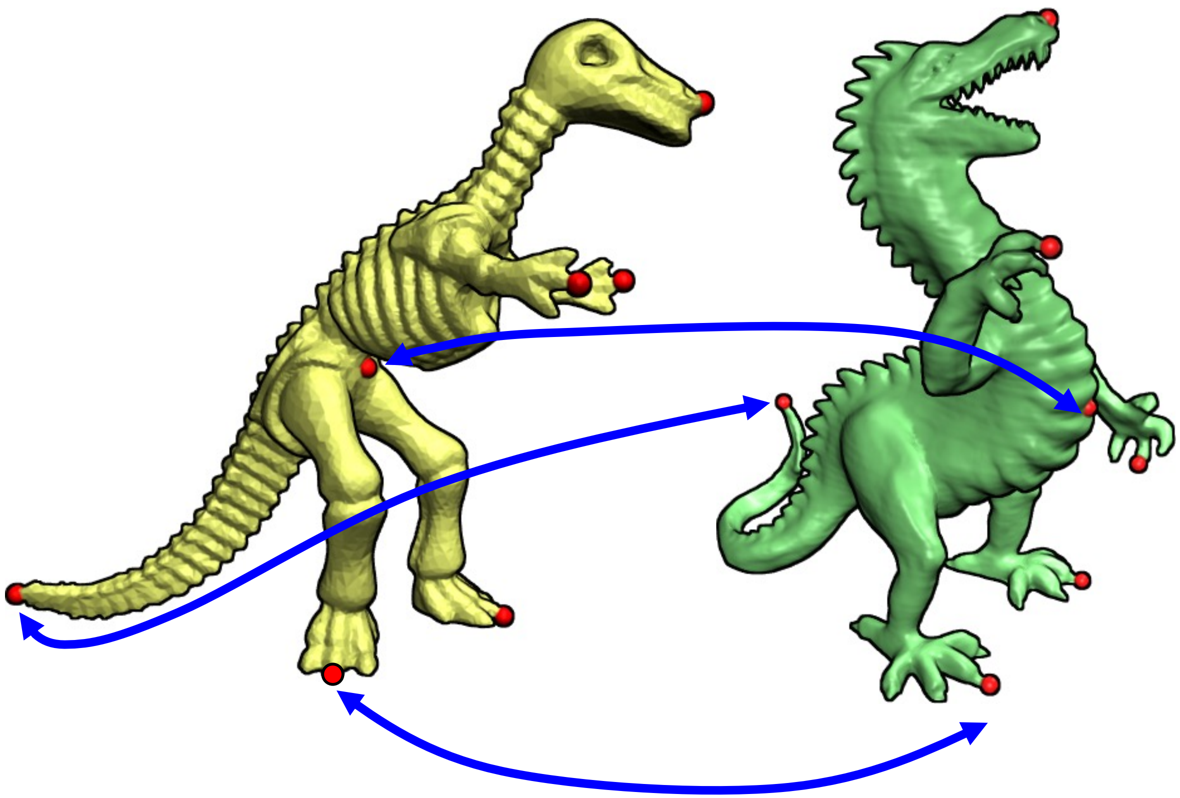
noun

the branch of linguistics and logic concerned with meaning. There are a number of branches and

Meaningful correspondence/segmentation = semantic correspondence/segmentation

Correspondence by humans

- Recognition is involved: more semantical

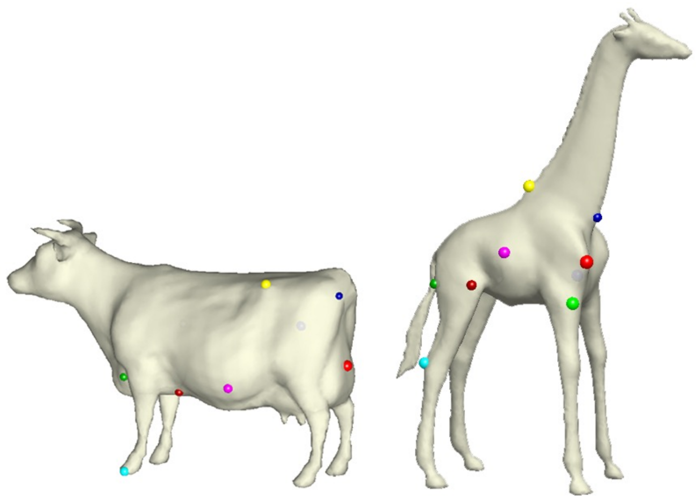


Human recognition

- Can tolerate many shape variations
  - Rigid transforms: e.g., rotation
  - Isometric (distance-preserving) transform and pose
  - Local geometric details

Human recognition

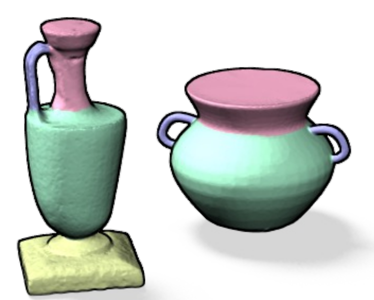
- More drastic shape variations
  - Non-homogenous part scaling
  - Even topological differences



[Lipman et al. 2009]

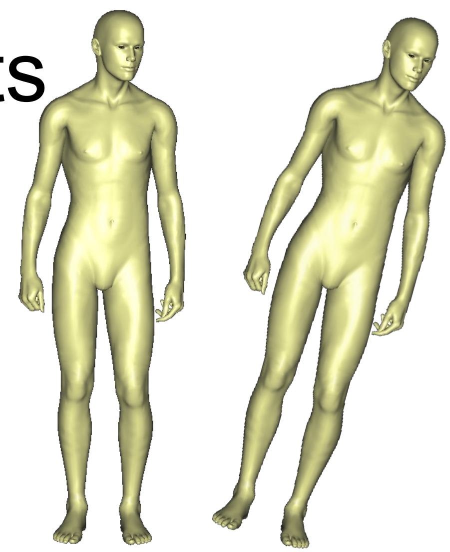
- Human recognition is often
  - beyond individual components ⇒ in a **context**
  - beyond what something looks like

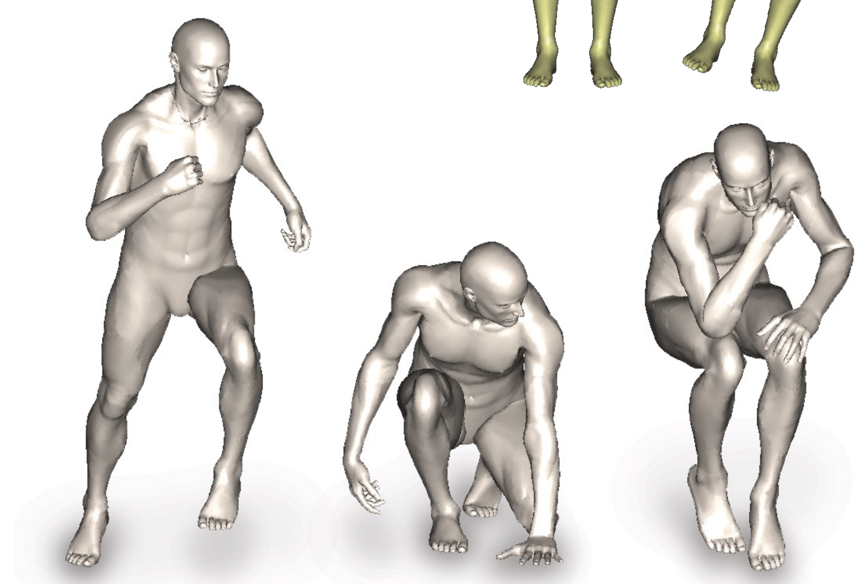
    ⇒ what it what it does — **functionality**



Computer algorithms

- Relatively strict **geometry** constraints
  - Rigid transforms: e.g., rotation
  - Isometric transform and pose
  - Local geometric details





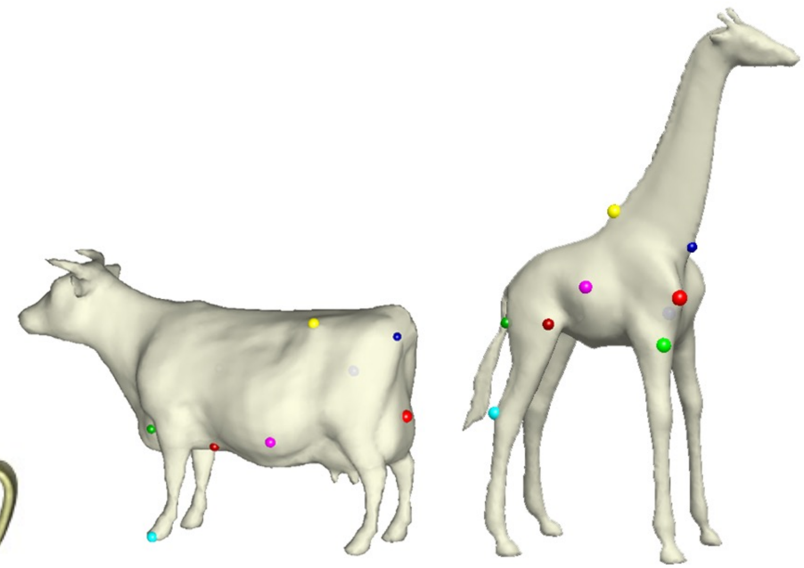
[Ovsjanikov et al. 2008]

Computer algorithms

- Challenge: more drastic variations; **semantics**

  - Non-homogenous part scaling

  - Topological differences



[Lipman et al. 2009]

Challenge: objective

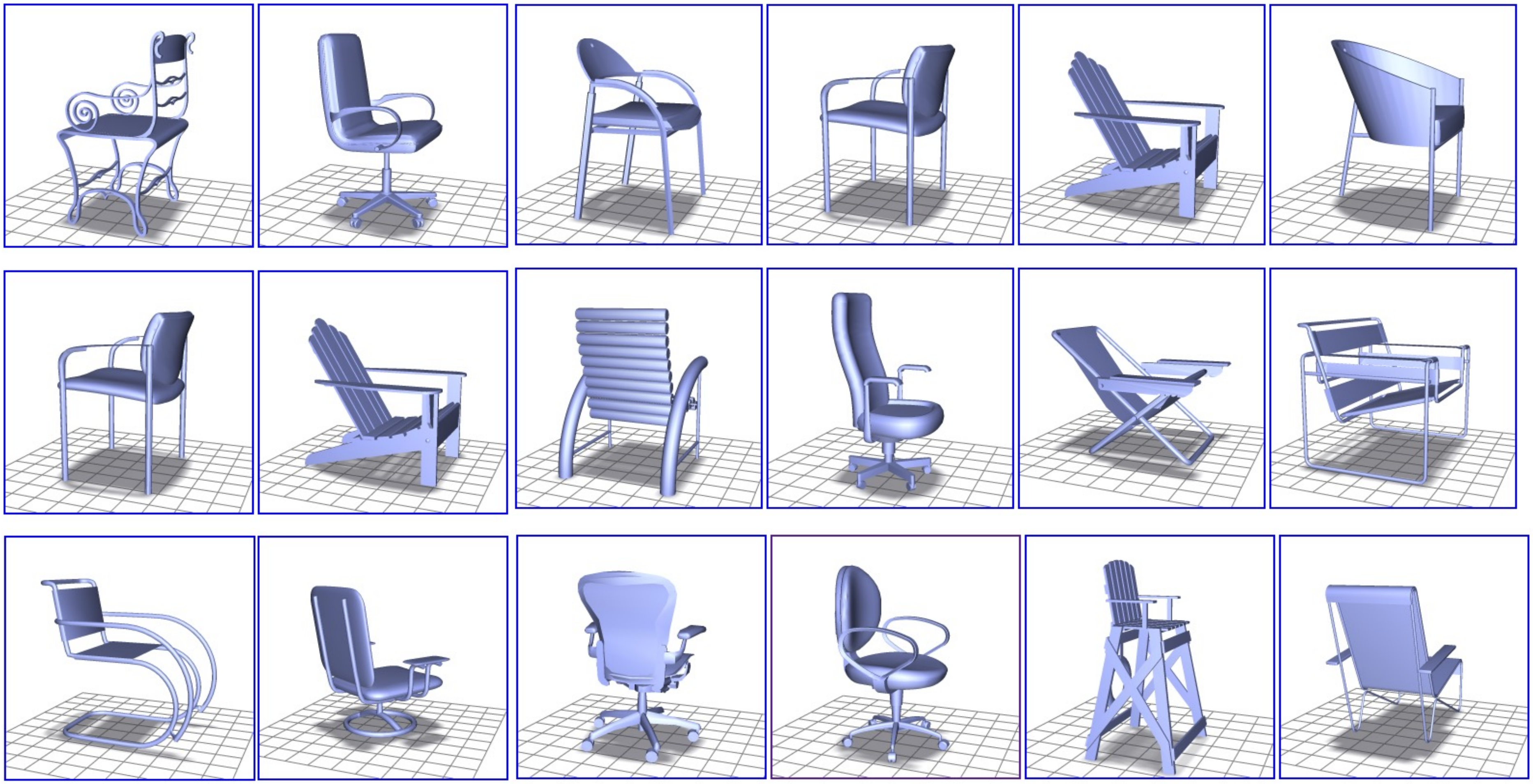
- Correspondence is a **search problem**
- **Search objective** can be hard to define

  Easy: relatively strict geometry constraint/criteria

  Hard: shape **semantics,** modeling of **functionality**

What is a handle?

What is an armrest?



Challenge: search space

- Search space can be large to intractable

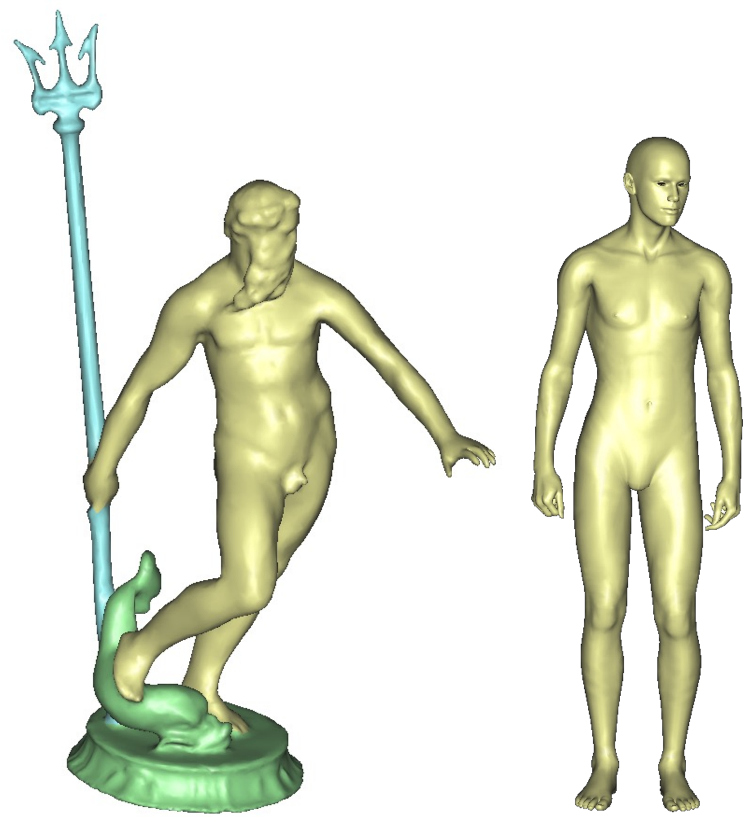
  Perhaps only rigid transforms are low-dim (6D)

  Difficult to parameterize search space for non-rigid or topology-modifying transforms

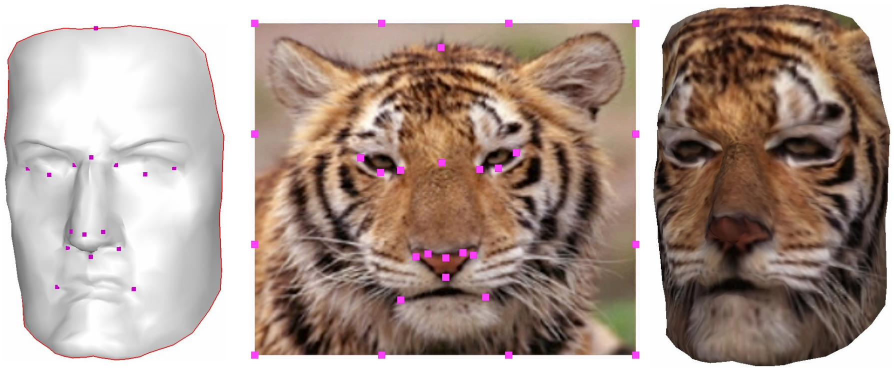
Challenge: partial matching

- Matching partial shapes — more challenge
  - Larger search space: also need to **find the subsets** — many (2^n) of them!

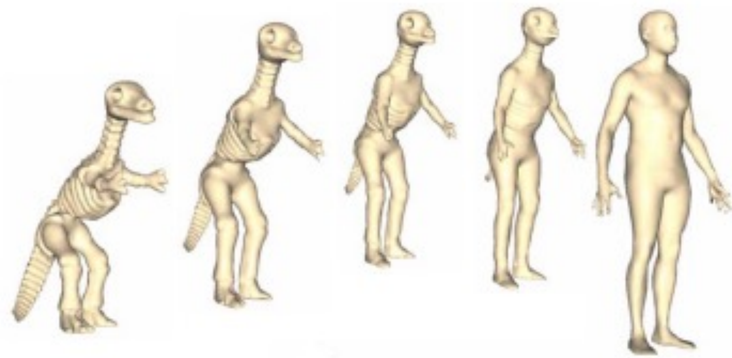
These are probably not as relevant with ML methods



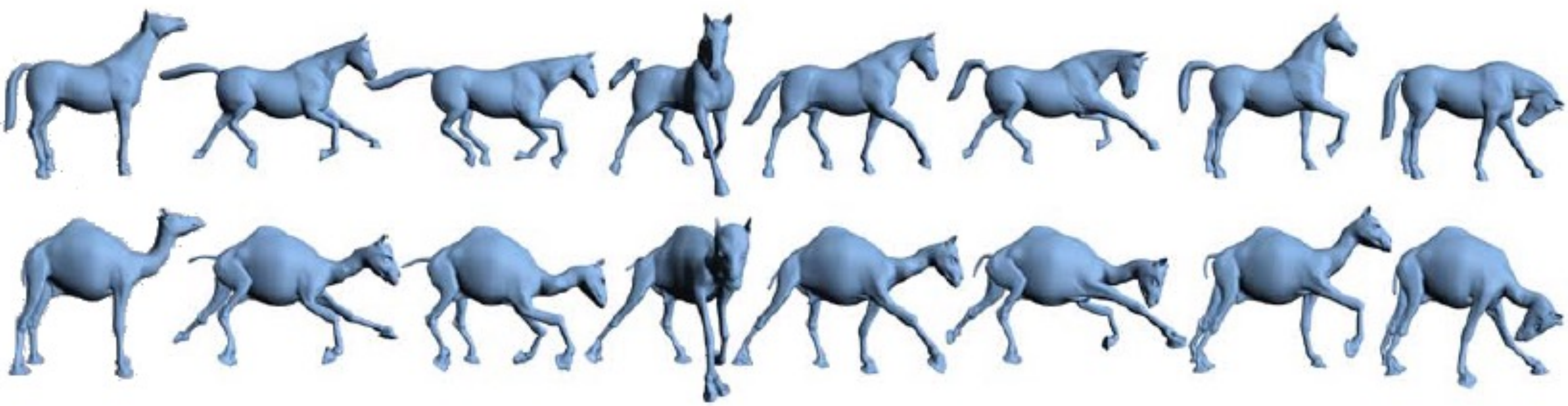
Some applications



Texture mapping [KSG03]



Morphing [KS04]



Deformation transfer [SP04]

Shape assembly

Modeling via part re-assembly [Funkhouser et al., SIG 2004]

Representative solutions

- Rigid transform
  - ICP: iterative closest points
  - Transformation search by voting
- Piece-wise rigidity and isometry
  - Rigidity decomposition or spectral embedding
- Large non-rigid deformation
  - Deformation-driven search
- Data-driven and learning-based solutions

Representative solutions

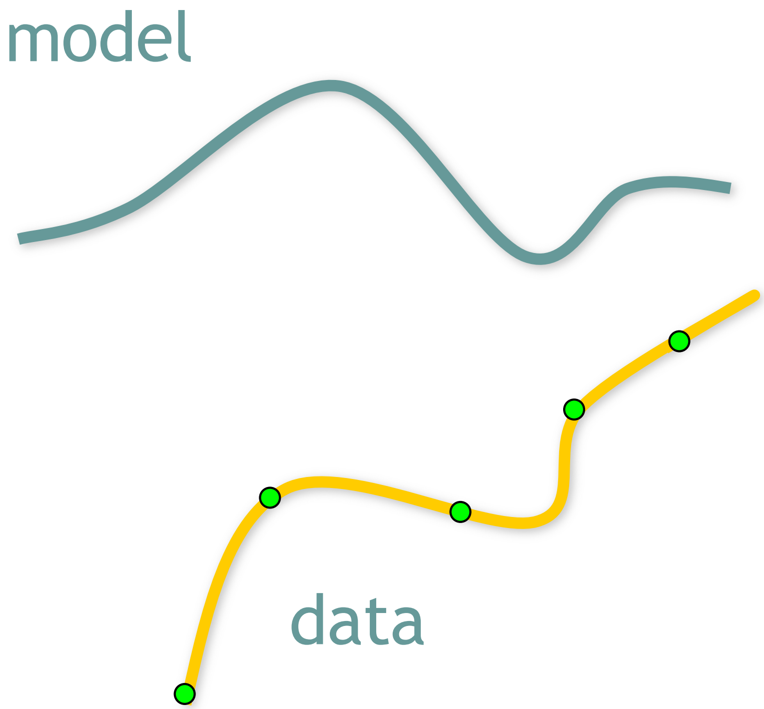
- Rigid transform
  - ICP: iterative closest points
  - Transformation search by voting
- ~~Piece-wise rigidity and isometry~~
  - ~~Rigidity decomposition or spectral embedding~~
- Large non-rigid deformation
  - Deformation-driven search
- Data-driven and learning-based solutions

Iterative closest points (ICP)

- One of the most classic correspondence schemes
- Input: **data** and **model shapes**
- Objective:
  - Rigid transform = rotation + translation
  - **Minimize mean squared error** from data points to closest points in model
- Correspondence obtained by Euclidean proximity

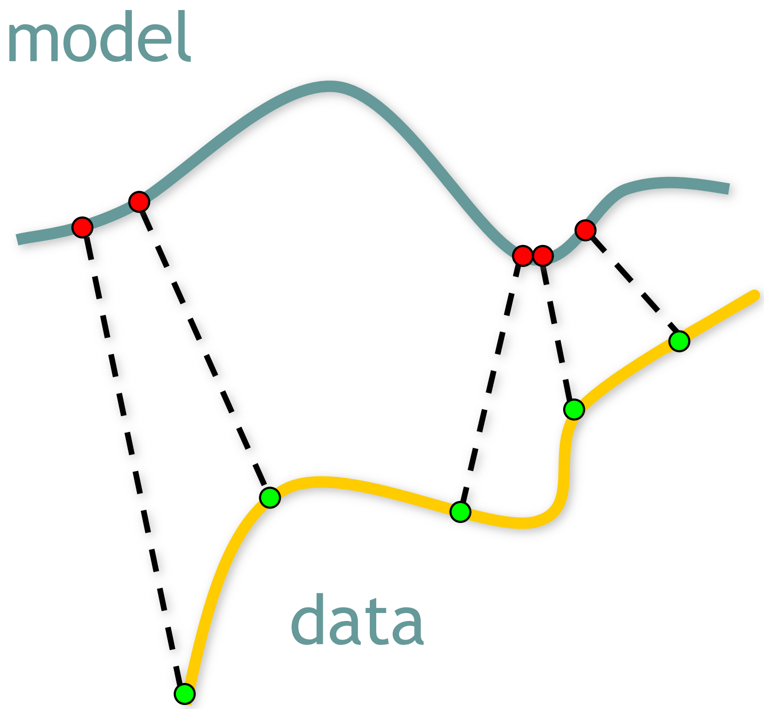
[Besl and Mckay 92]

ICP



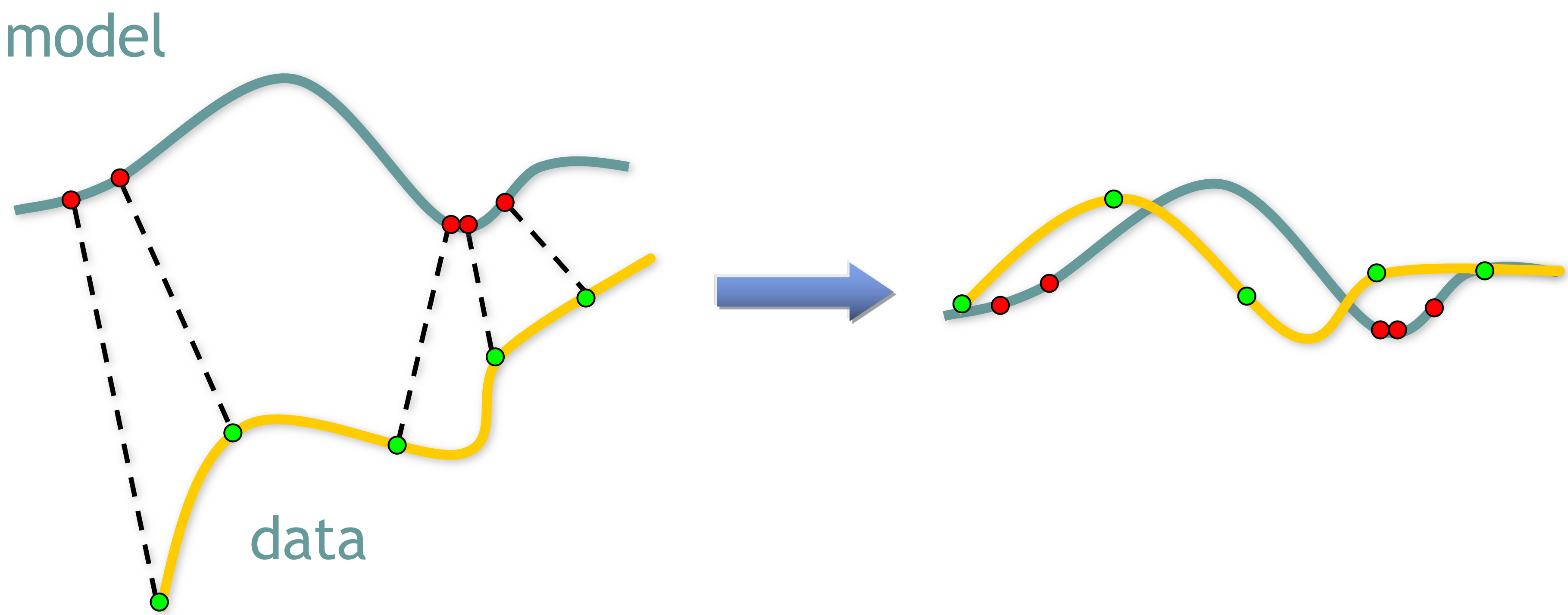
Model and data shapes (point samples)

ICP



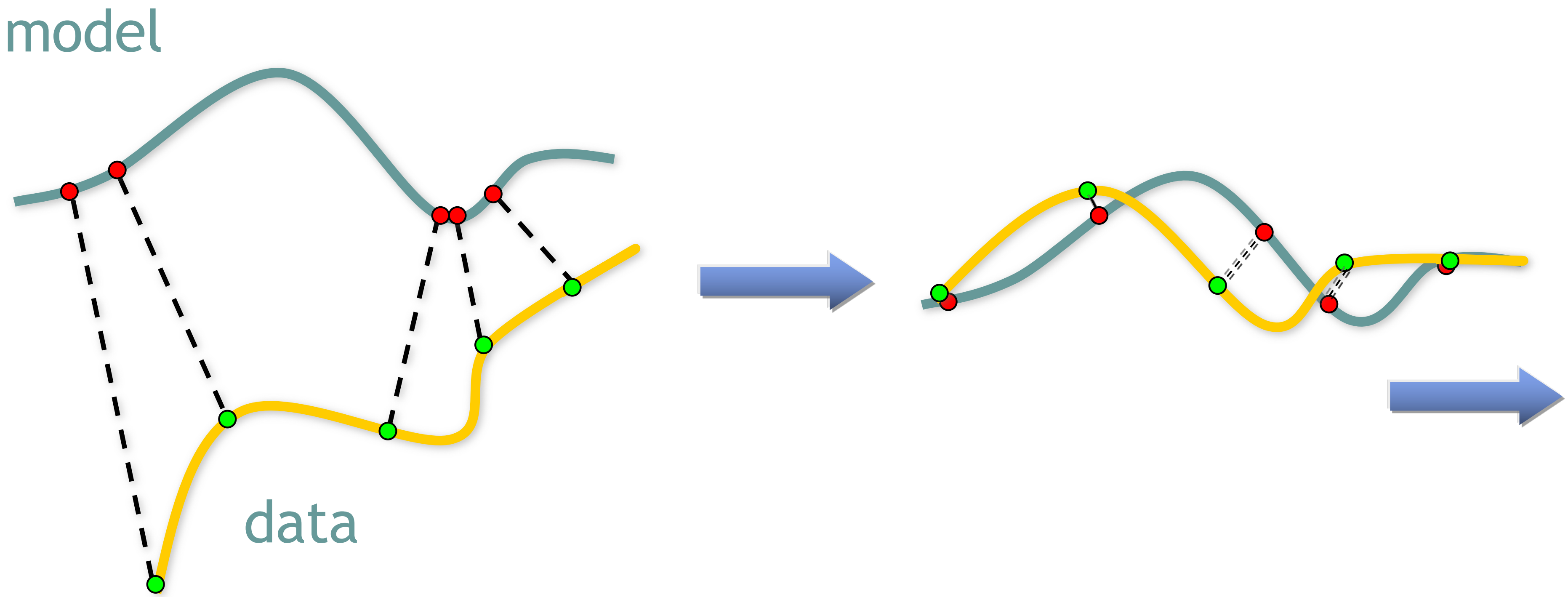
Find closest points from data to model

ICP



Find best rigid transform to align the corresponding points

ICP



Iterate …

Many follow-ups and extensions

- Extend the allowable transformation from rigid to affine and then to general non-rigid deformations, e.g., see: https://en.wikipedia.org/wiki/Point_set_registration
- Expand the notion of "closest points" to account for feature matching and similarity
- Use of deep learning approaches to ICP

Deep Closest Point: Learning Representations for Point Cloud Registration

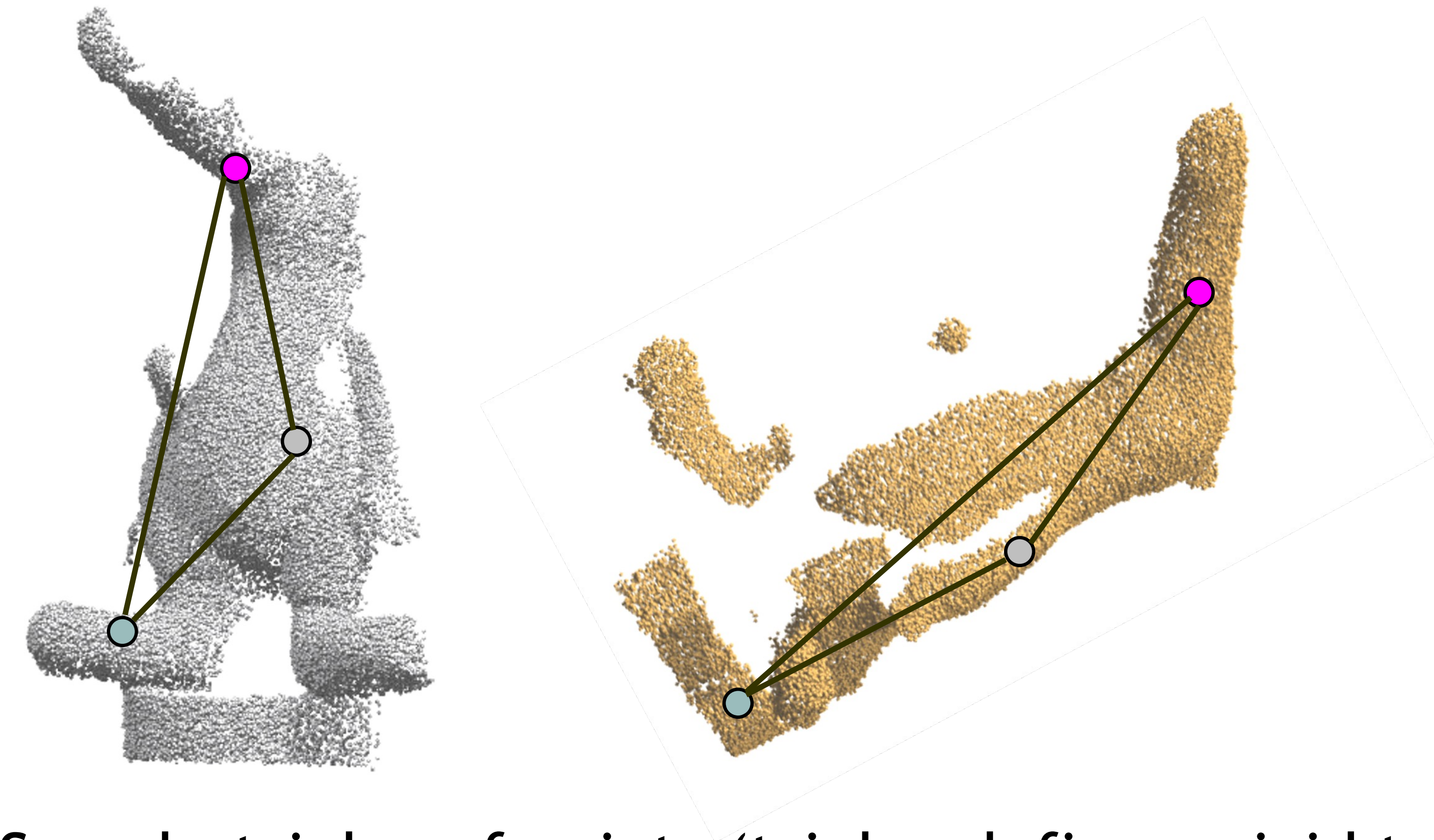
Yue Wang, Justin M. Solomon

(Submitted on 8 May 2019)

Point cloud registration is a key problem for computer vision applied to robotics, medical imaging, and other applications. This problem involves finding a rigid transformation from one point cloud into another so that they align. Iterative Closest Point (ICP) and its variants provide simple and easily-implemented iterative methods for this task, but these algorithms can converge to spurious local optima. To address local optima and other difficulties in the ICP pipeline, we propose a learning-based method, titled Deep Closest Point (DCP), inspired by recent techniques in computer vision and natural language processing. Our model consists of three parts: a point cloud embedding network, an attention-based module combined with a pointer generation layer, to approximate combinatorial matching, and a differentiable singular value decomposition (SVD) layer to extract the final rigid transformation. We train our model end-to-end on the ModelNet40 dataset and show in several settings that it

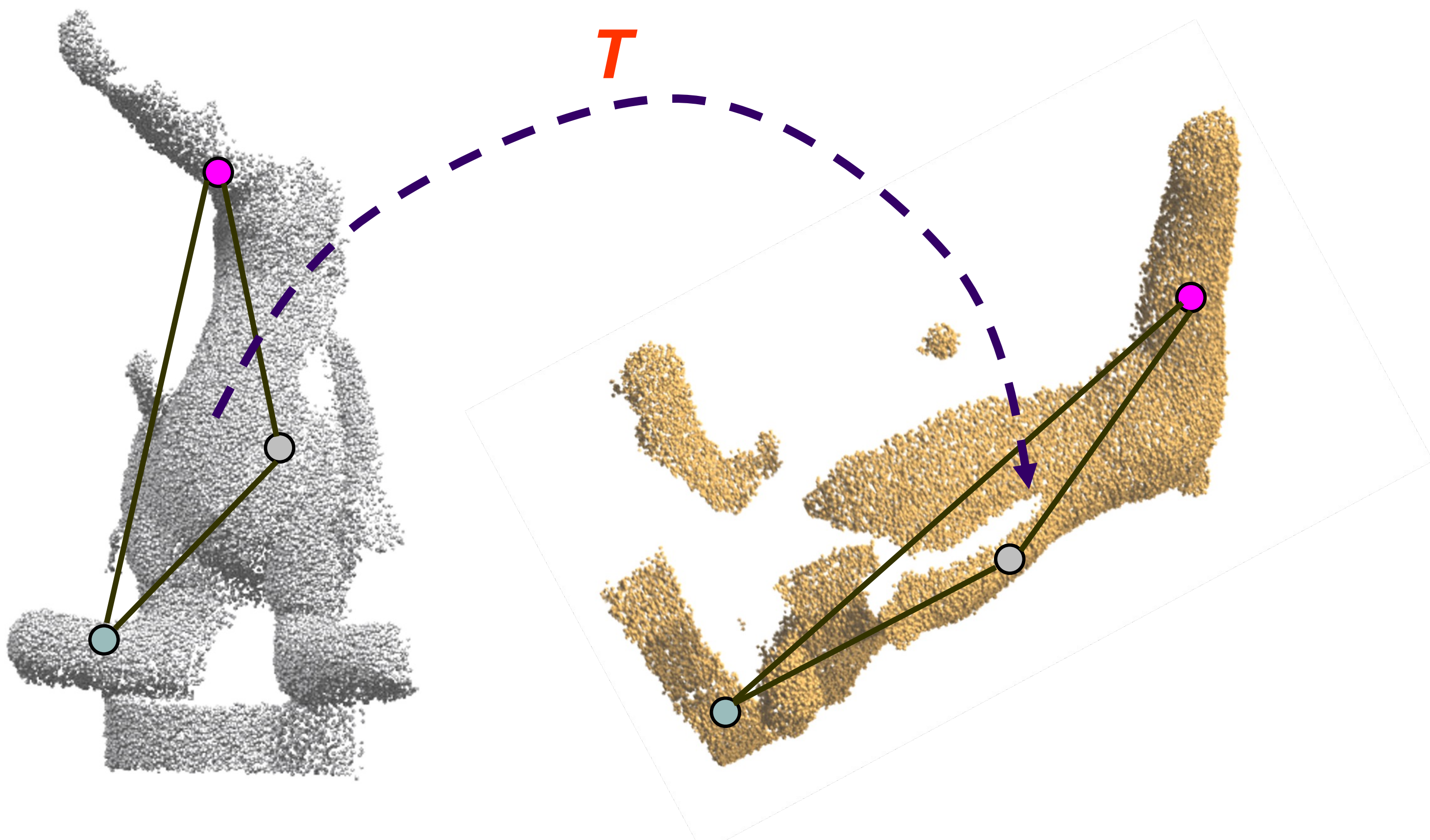
Transformation-space voting

Transformation-space voting



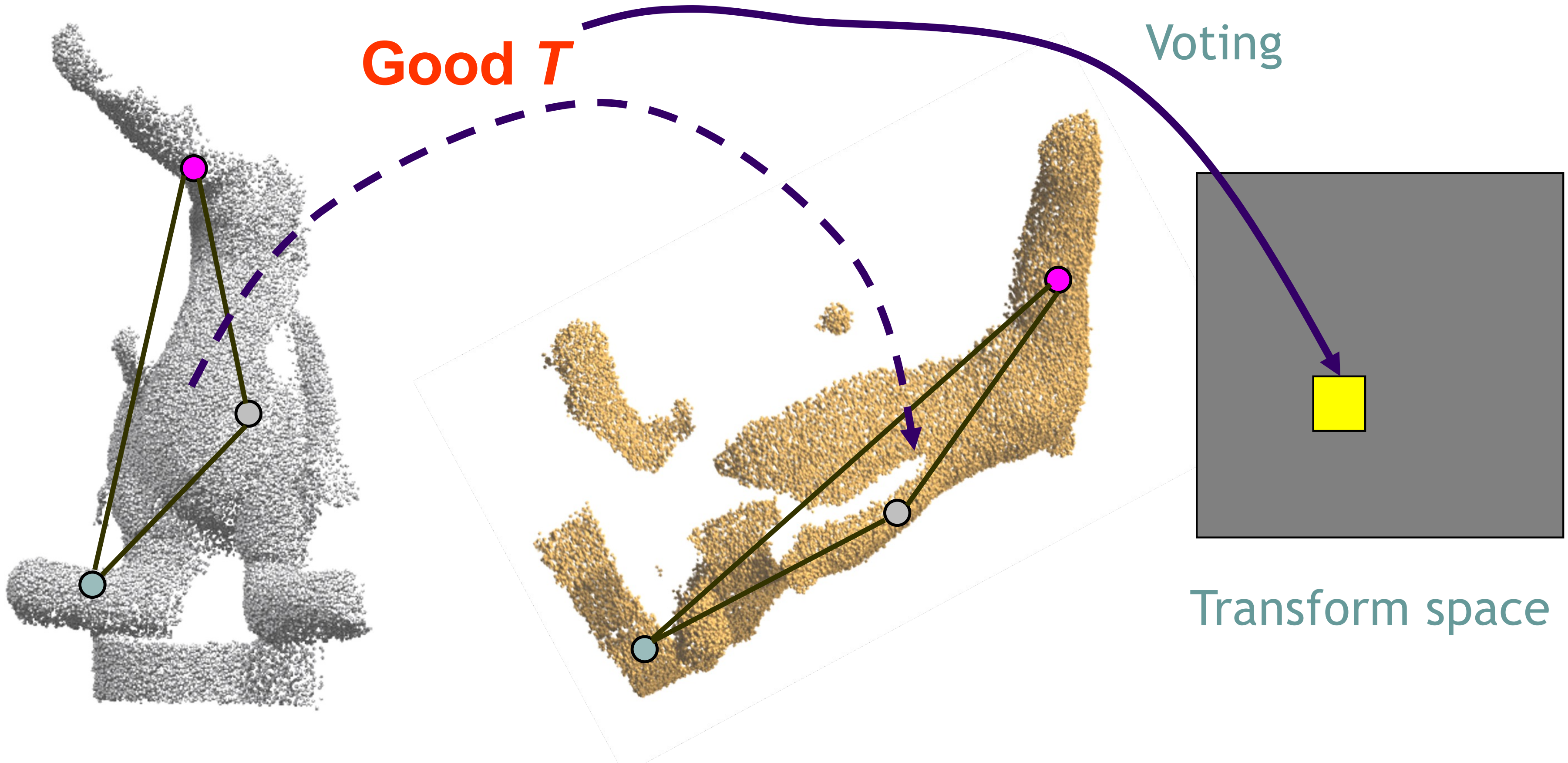
Sample triples of points (triples define a rigid transform)

Transformation-space voting



Estimate transform and test quality of fit

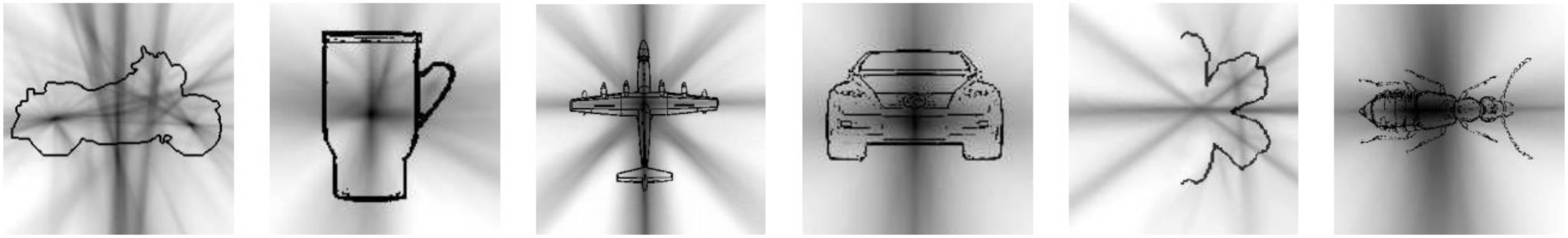
Transformation-space voting



Good transform casts vote in **low-D transform space**

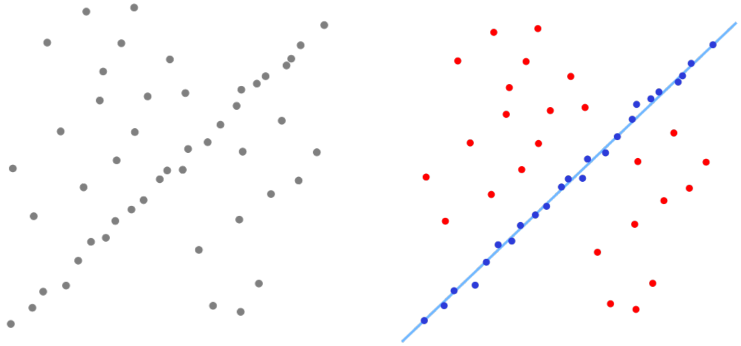
Voting is a meta idea

- Many applications, e.g., symmetry detection



- RANSAC (Random Sampling Consensus)

  https://en.wikipedia.org/wiki/Random_sample_consensus



- Randomly sample points to "vote" for model parameters, e.g., a line
- Randomness pays off: effectively reducing the impact of outliners

Large non-rigid deformation

Typical objective function for correspondence search

$$\omega \sum_{i:\text{feature}} \text{SimlarityDist}(p_i, q_i) + (1-\omega) \sum_{i,j:\text{feature}} \text{Distortion}(p_i, p_j, q_i, q_j)$$

Large non-rigid deformation

Typical objective function for correspondence search

$$\omega \sum_{i:\text{feature}} \boxed{\text{SimlarityDist}(p_i, q_i)} + (1-\omega) \sum_{i,j:\text{feature}} \boxed{\text{Distortion}(p_i, p_j, q_i, q_j)}$$

Descriptor must be insensitive to pose and local geometry change

No longer appropriate in the case of large deformations

Large non-rigid deformation

Typical objective function for correspondence search

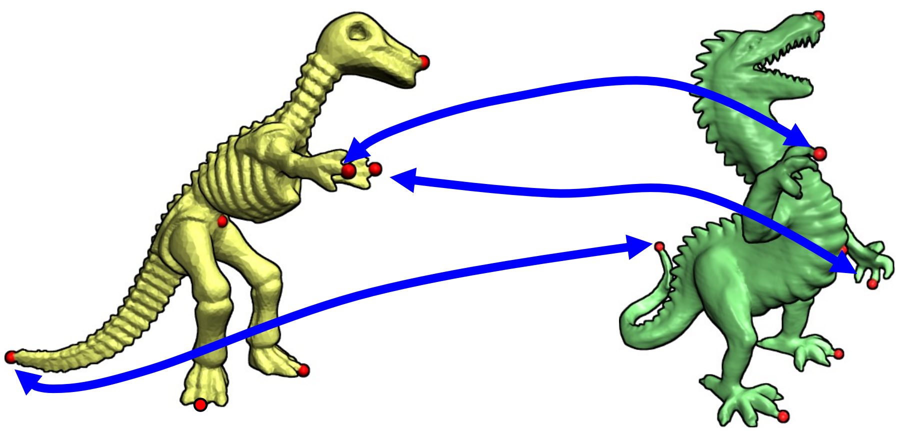
$$\omega \sum_{i:\text{feature}} \boxed{\text{SimlarityDist}(p_i, q_i)} + (1-\omega) \sum_{i,j:\text{feature}} \boxed{\text{Distortion}(p_i, p_j, q_i, q_j)}$$

Descriptor must be insensitive to pose and local geometry change

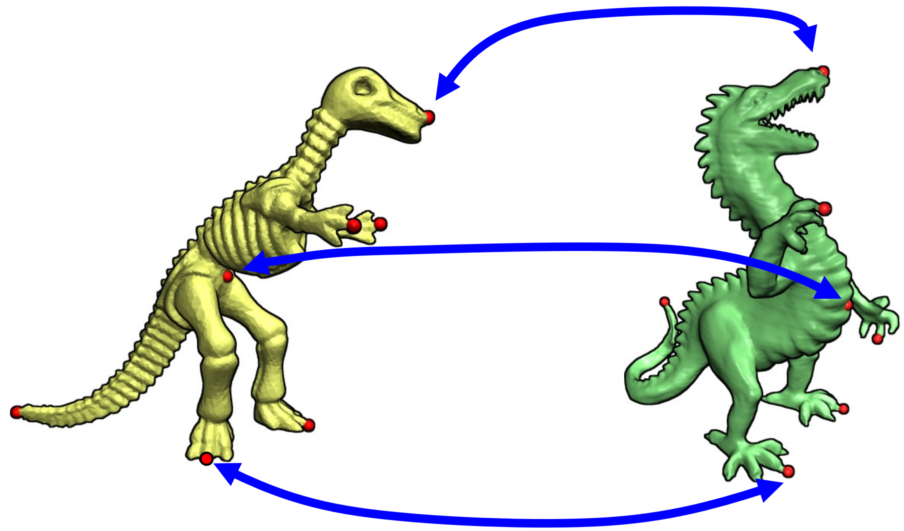
No longer appropriate in the case of large deformations

- More drastic shape variations
- Rigidity or isometry constraints no longer applies
- Fewer works to date [Zhang et al. 2008]

Large non-rigid deformation



Pose + non-homogenous part scaling



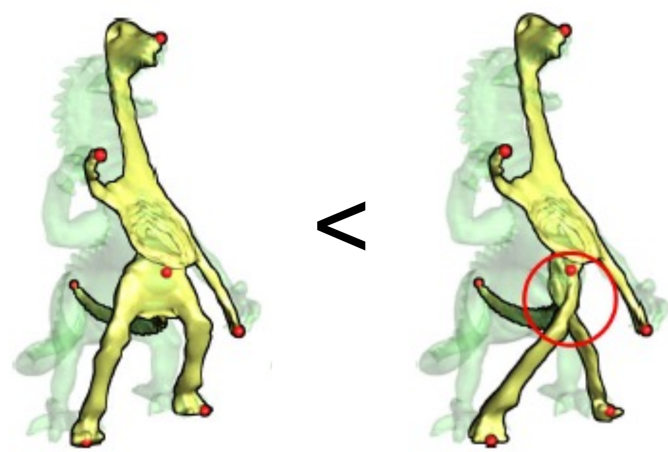
Local shape variability

[Zhang et al. 08]

A more global approach

- Local vs. global criteria
  - Local: feature similarity
  - **Local criterion less reliable with large shape variations**
  - Focus more on global consistency of correspondence
- Use of non-rigid mesh deformation [Zhang et al. 08]

Correspondence cost = effort to deform one mesh into other



A result: "symmetry switching"

[Zhang et al. 08]

Algorithm

Step 1: feature extraction

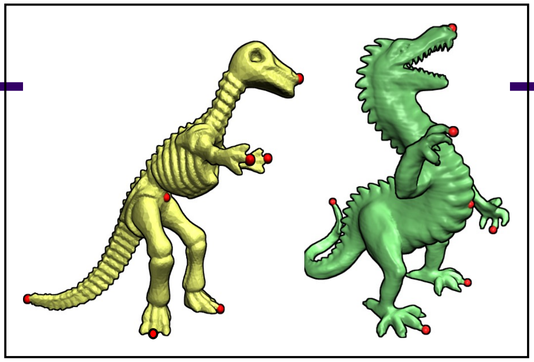
Algorithm

Step 1: feature extraction

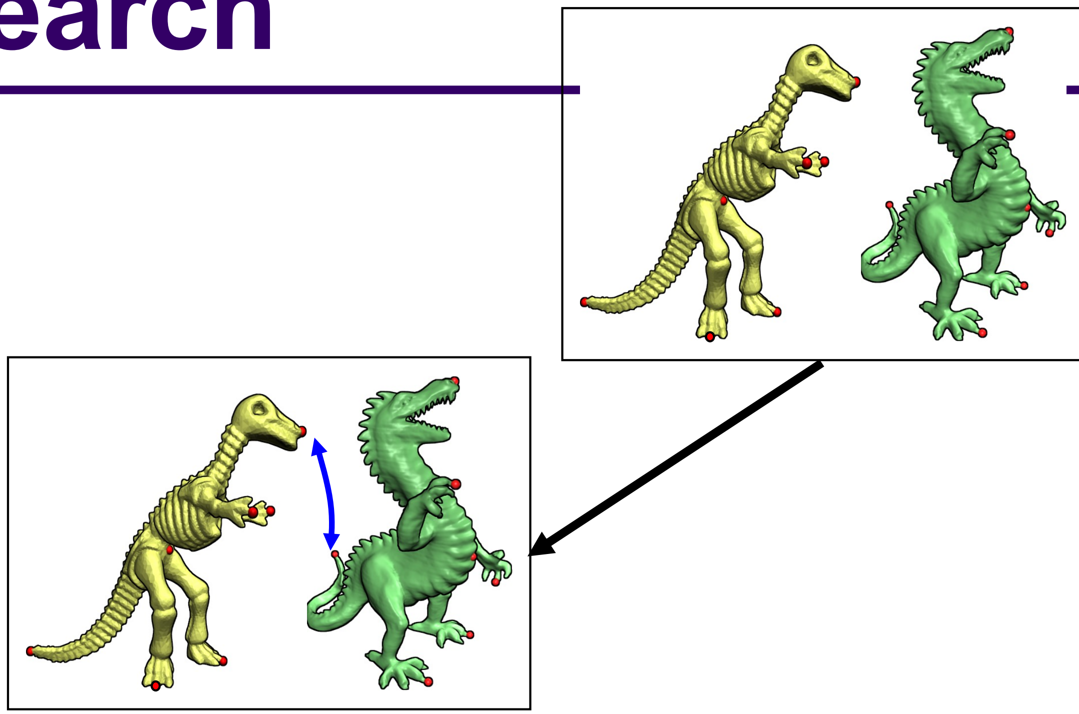
Step 2: combinatorial search

- Priority = deformation cost
- Pruning by feature similarity and geodesic distance

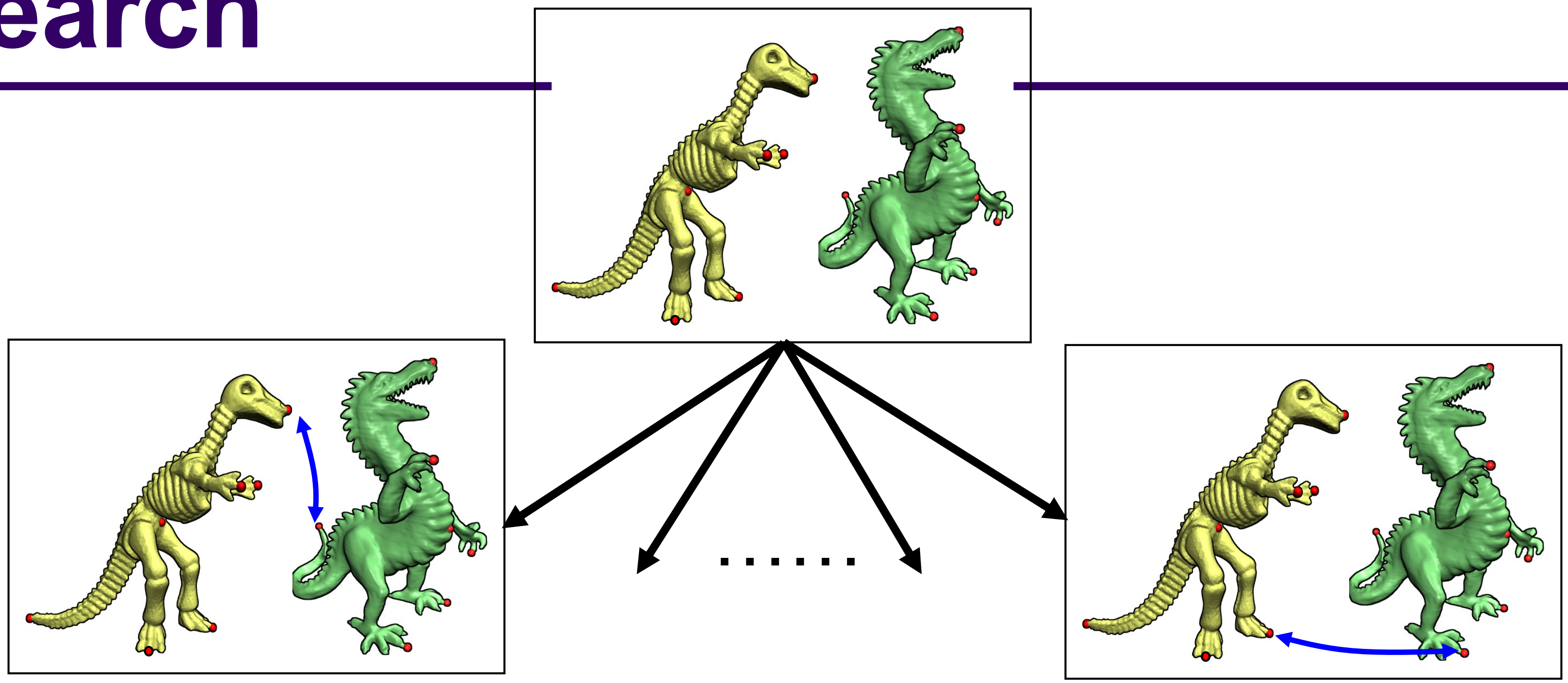
Search



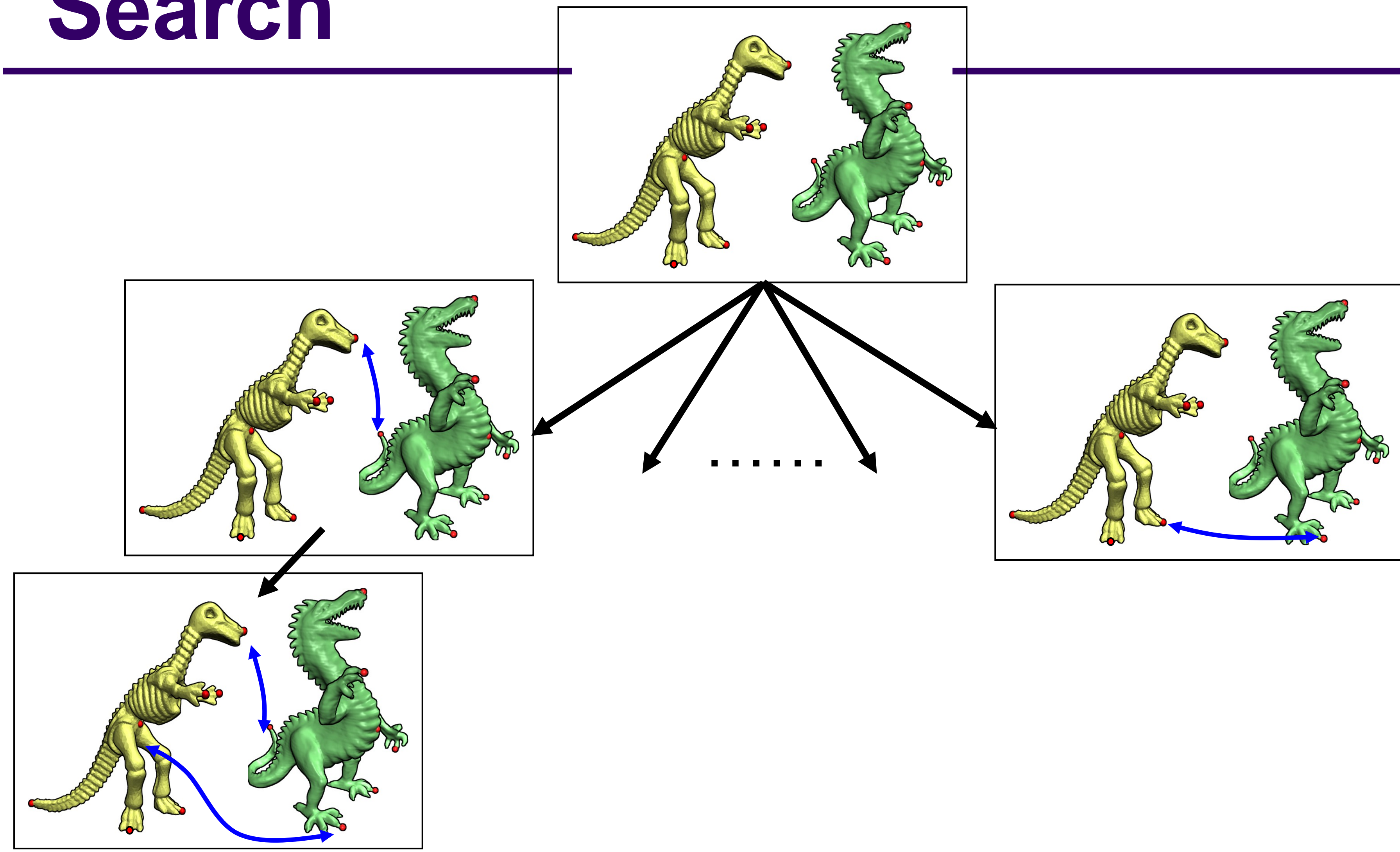
Search



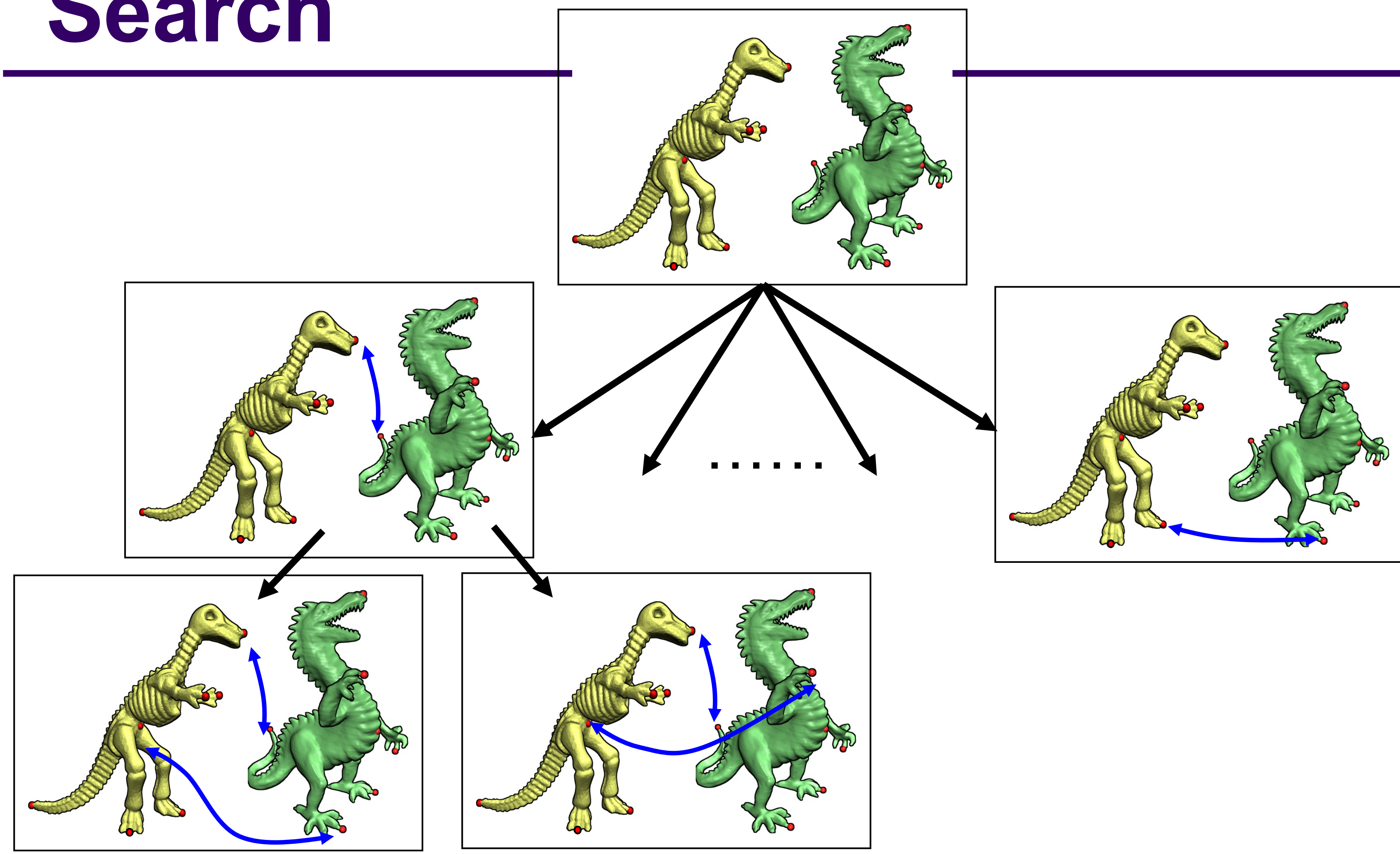
Search



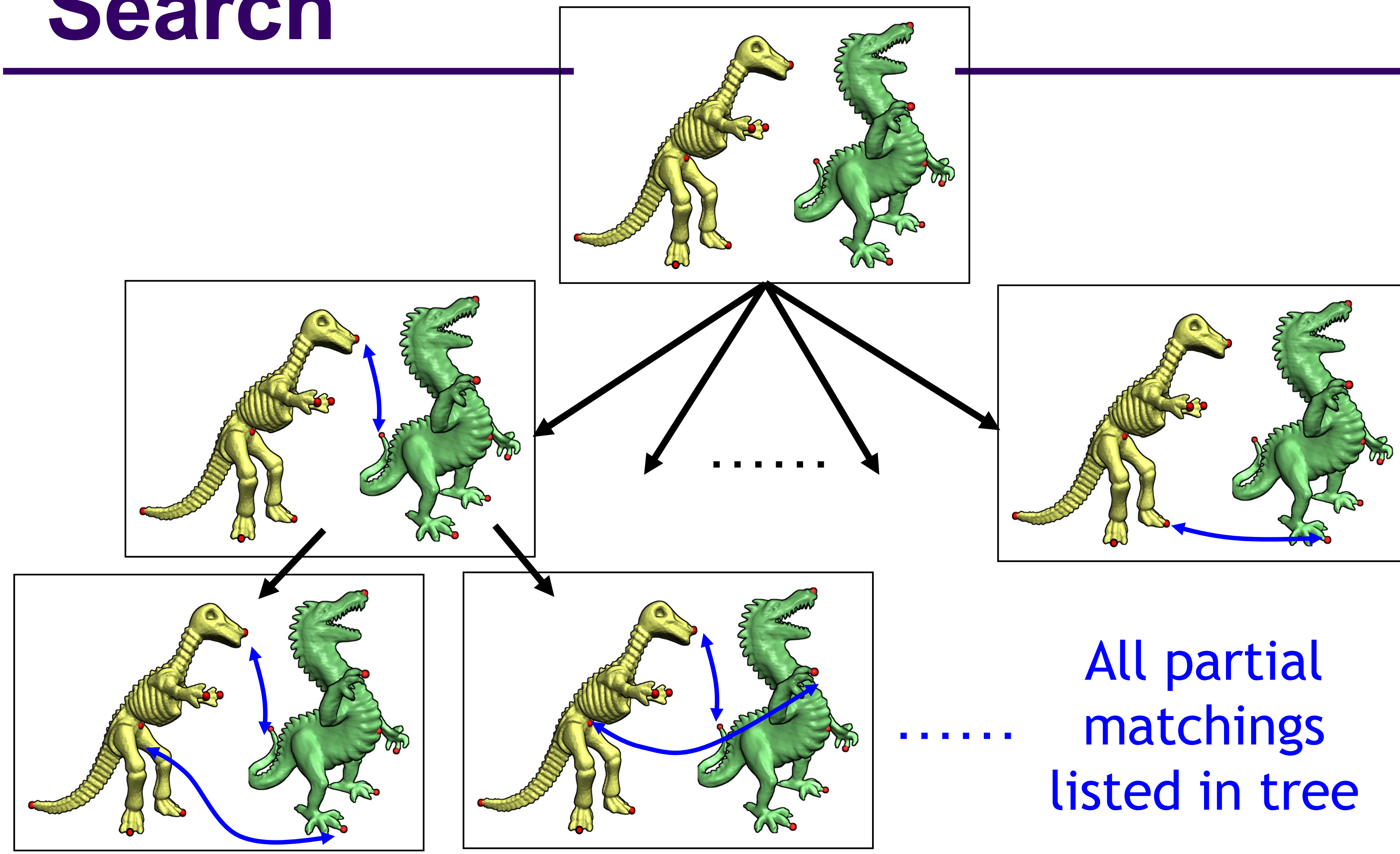
Search



Search



Search



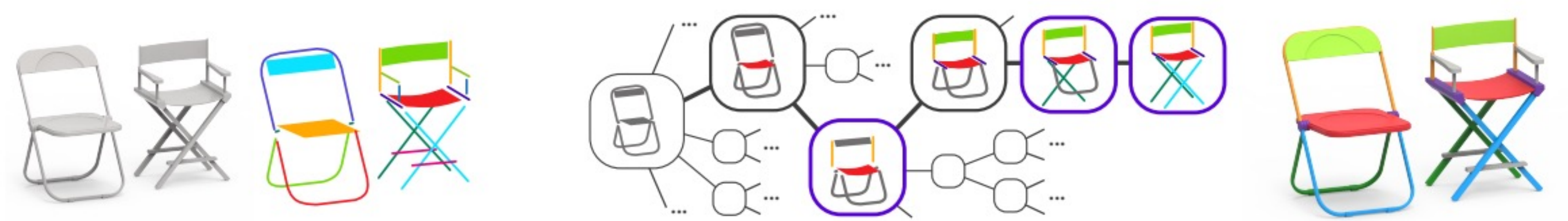
All partial matchings listed in tree

Extension to topology variation

Deformation-Driven Topology-Varying 3D Shape Correspondence

Ibraheem Alhashim[1] Kai Xu[2,3] Yixin Zhuang[3] Junjie Cao[4] Patricio Simari[5] Hao Zhang[1]
[1]Simon Fraser University [2]Shenzhen VisuCA Key Lab / SIAT
[3]National University of Defense Technology [4]Dalian University of Technology [5]The Catholic University of America

SIGGRAPH Asia 2015



(a) Source (left), target shapes, and curve-sheet abstractions (b) Search tree (c) Final correspondence result

- Deformation model allows topological changes

Extension to topology variation

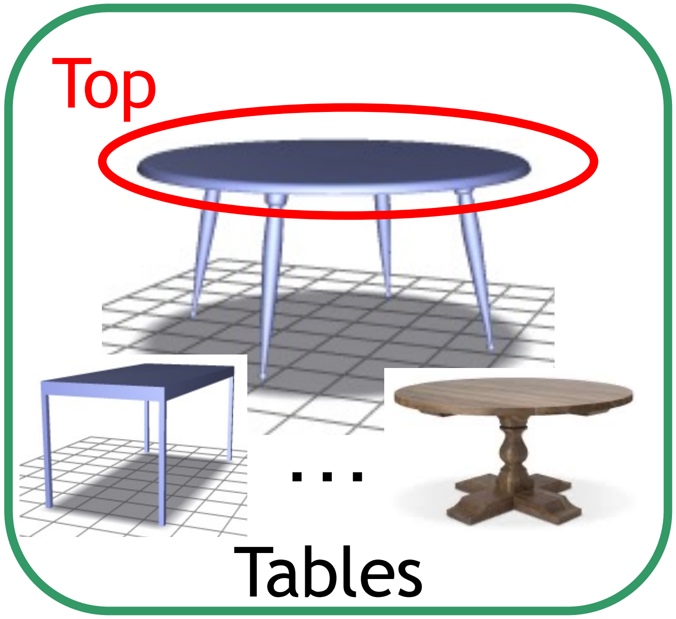
Deformation-Driven Topology-Varying 3D Shape Correspondence

Ibraheem Alhashim[1] Kai Xu[2,3] Yixin Zhuang[3] Junjie Cao[4] Patricio Simari[5] Hao Zhang[1]
[1]Simon Fraser University [2]Shenzhen VisuCA Key Lab / SIAT
[3]National University of Defense Technology [4]Dalian University of Technology [5]The Catholic University of America

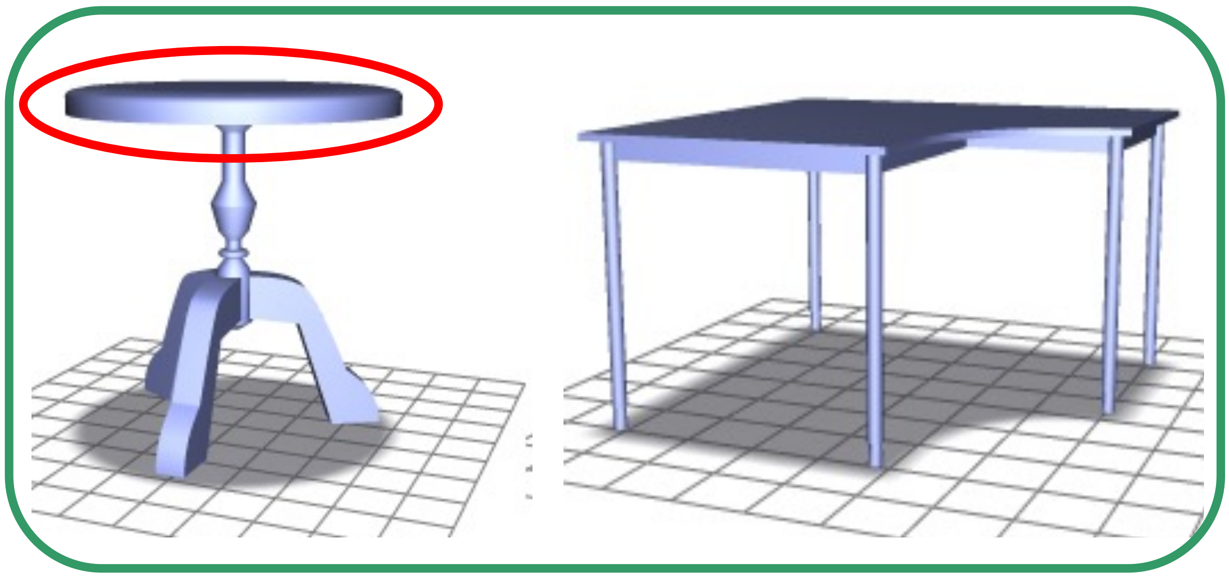
SIGGRAPH Asia 2015

Data-driven and learning

- Pure geometry-driven analysis inherently limited
- Semantics: correspondence through recognition



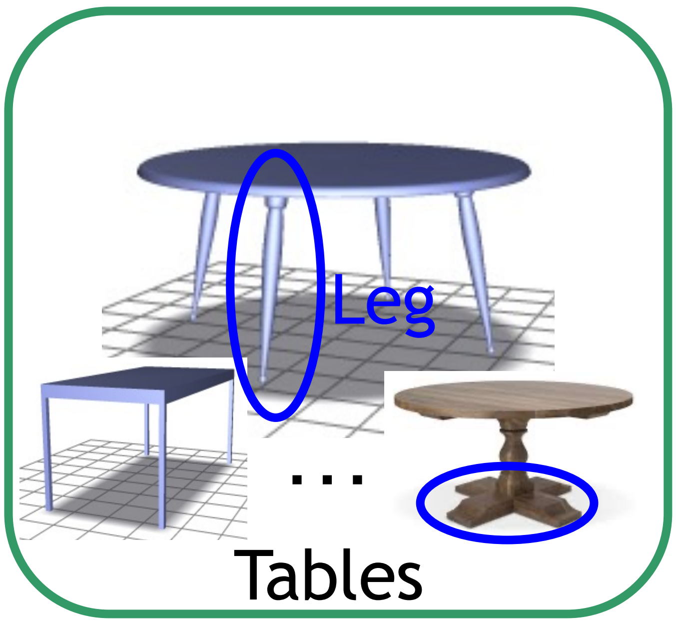
Memory/knowledge



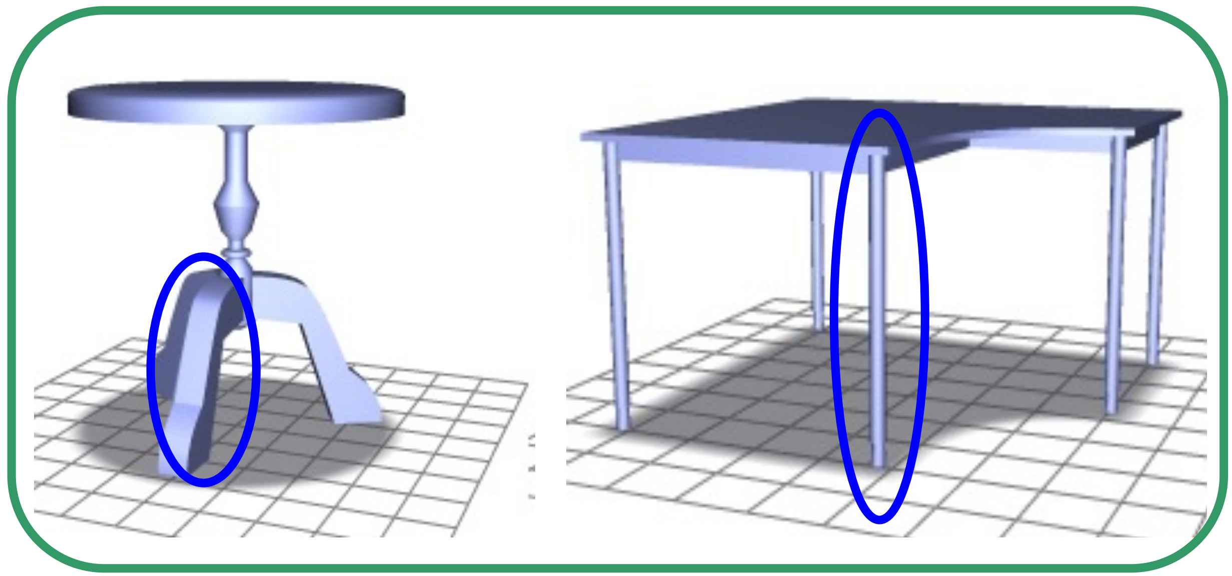
Queries

Data-driven and learning

- Pure geometry-driven analysis inherently limited
- Semantics: correspondence through recognition

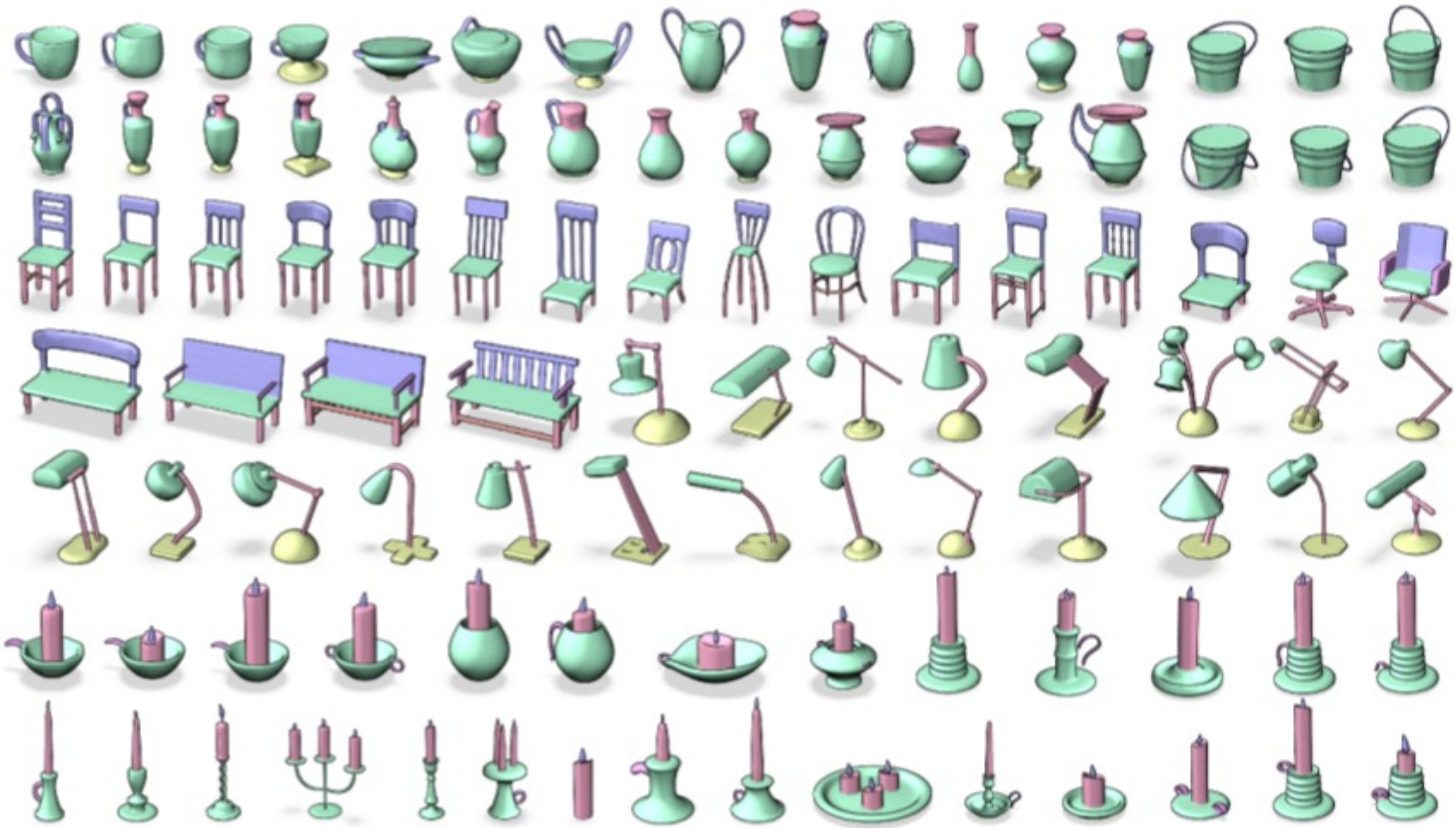


Memory/knowledge



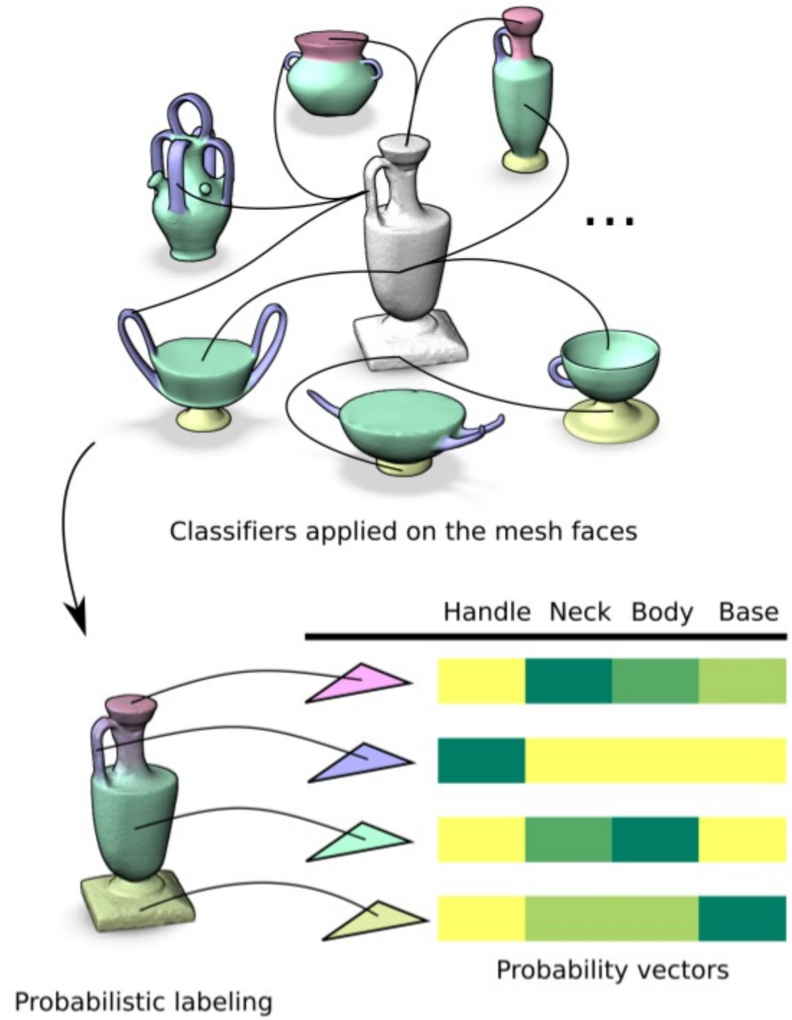
Queries

Knowledge = training set

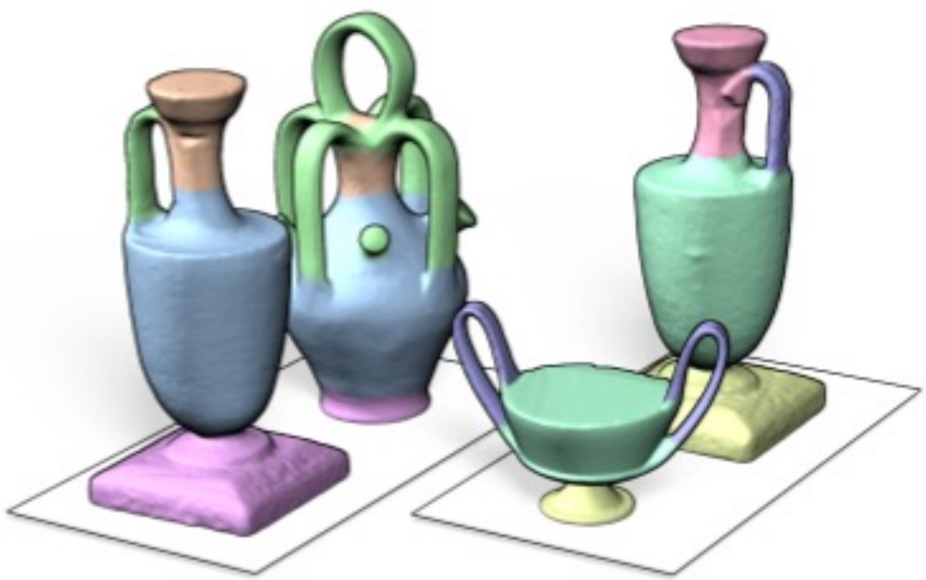


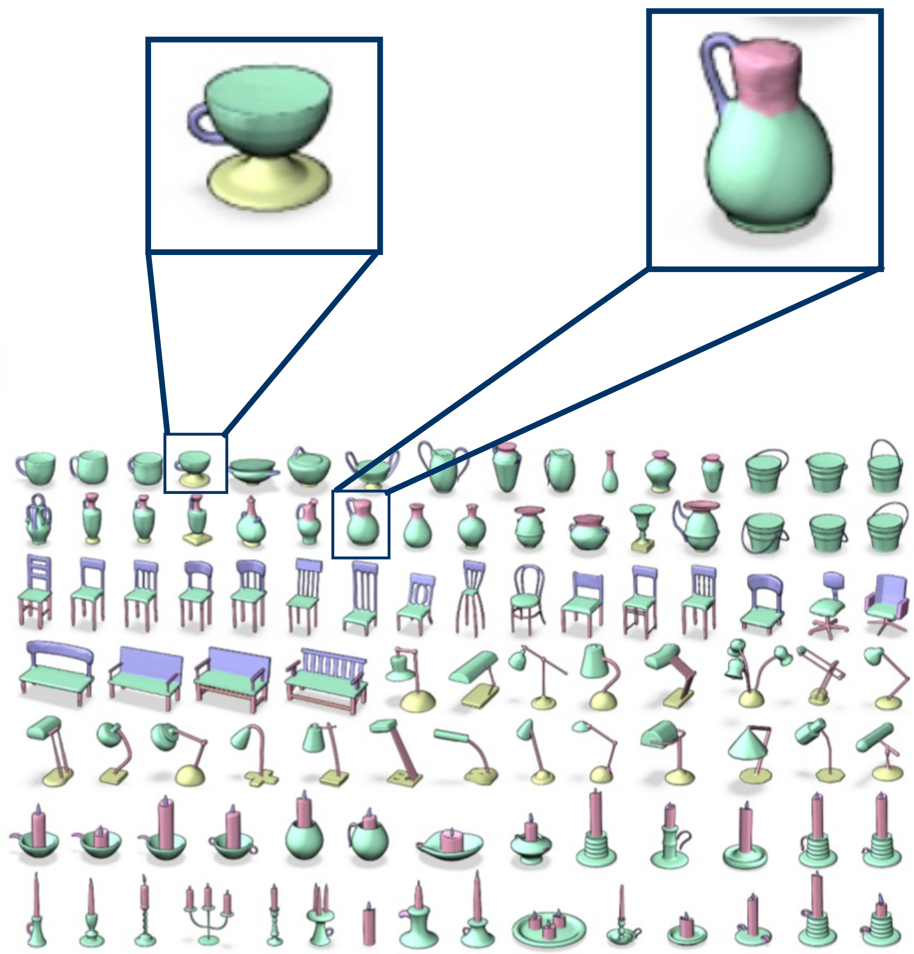
Learning to label

- Each face in training set has semantic label
- To correspond, assign semantic labels to query
- Labeling through training classifiers — standard machine learning



Impressive results?





Key idea

Difficult correspondence problem can be solved by recognition using knowledge from training set.

- A simple idea sorting to machine learning (ML)
- Classical ML: impressive results as long as there is knowledge (training data) to support it
  - Not generalizable: only do as well as training set allows

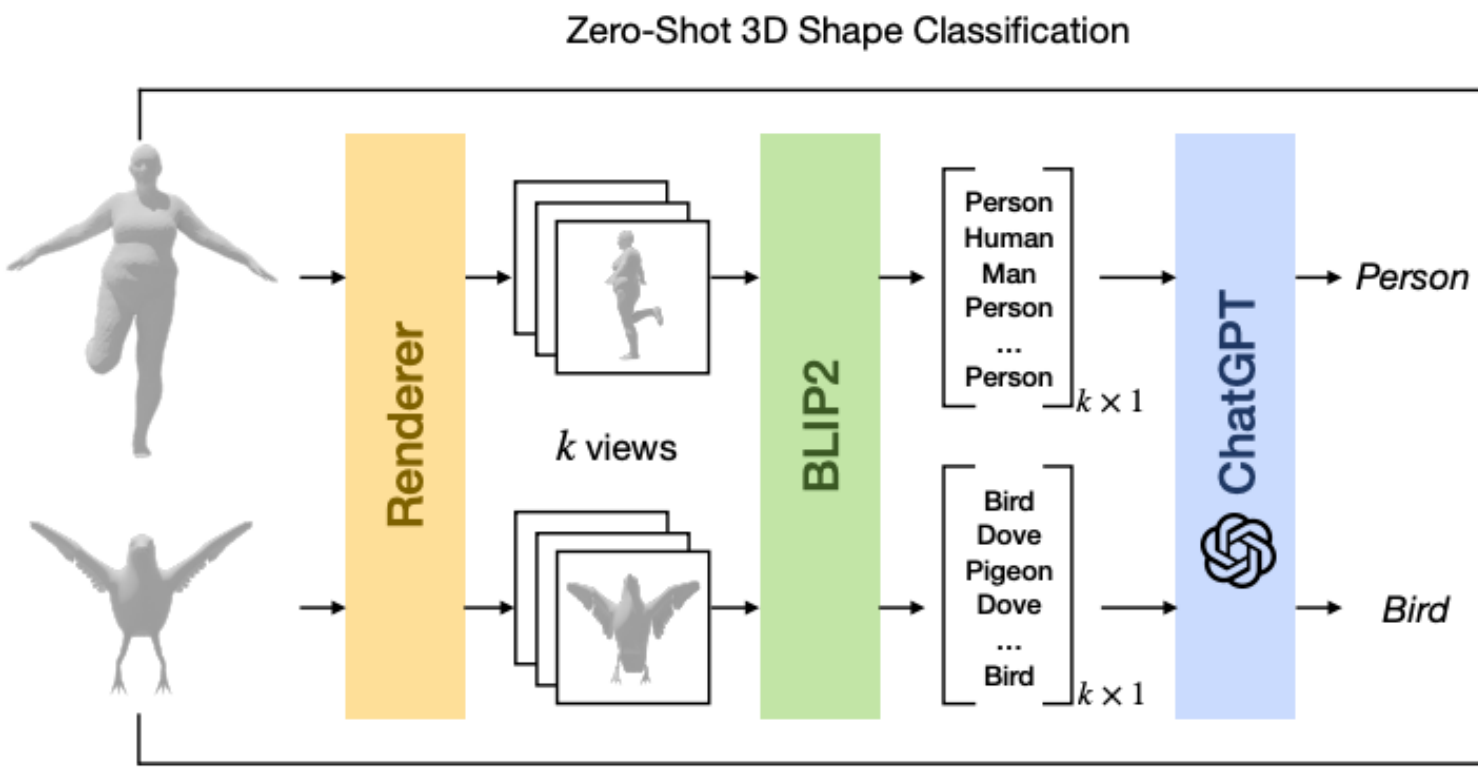
Fast track to 2024: no training data

Zero-Shot 3D Shape Correspondence

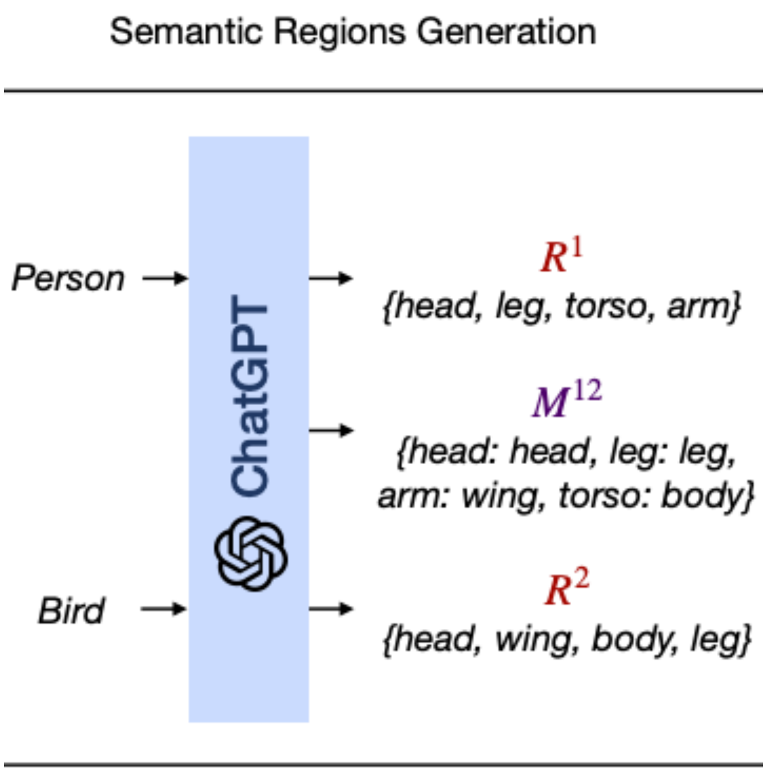
AHMED ABDELREHEEM, KAUST, Saudi Arabia
ABDELRAHMAN ELDESOKEY, KAUST, Saudi Arabia
MAKS OVSJANIKOV, LIX, École Polytechnique, France
PETER WONKA, KAUST, Saudi Arabia

Zero-shot

Zero-shot

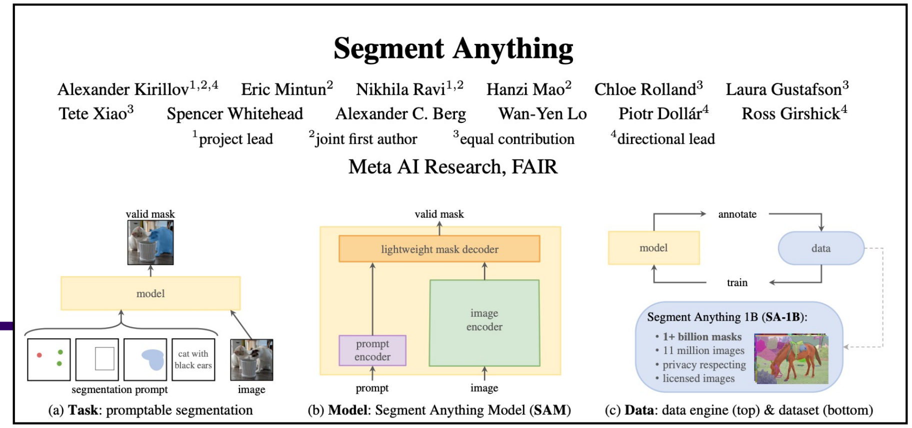

Zero-Shot 3D Shape Classification
Renderer
k views
BLIP2
Person
Human
Man
Person
...
Person
k × 1
Bird
Dove
Pigeon
Dove
...
Bird
k × 1
ChatGPT
Person
Bird

Zero-shot



Semantic Regions Generation
Person
Bird
ChatGPT
R^1
{head, leg, torso, arm}
M^{12}
{head: head, leg: leg,
arm: wing, torso: body}
R^2
{head, wing, body, leg}

Zero-shot


Segment Anything

Alexander Kirillov[1,2,4] Eric Mintun[2] Nikhila Ravi[1,2] Hanzi Mao[2] Chloe Rolland[3] Laura Gustafson[3]
Tete Xiao[3] Spencer Whitehead Alexander C. Berg Wan-Yen Lo Piotr Dollár[4] Ross Girshick[4]
[1]project lead [2]joint first author [3]equal contribution [4]directional lead

Meta AI Research, FAIR

(a) **Task**: promptable segmentation (b) **Model**: Segment Anything Model (SAM) (c) **Data**: data engine (top) & dataset (bottom)