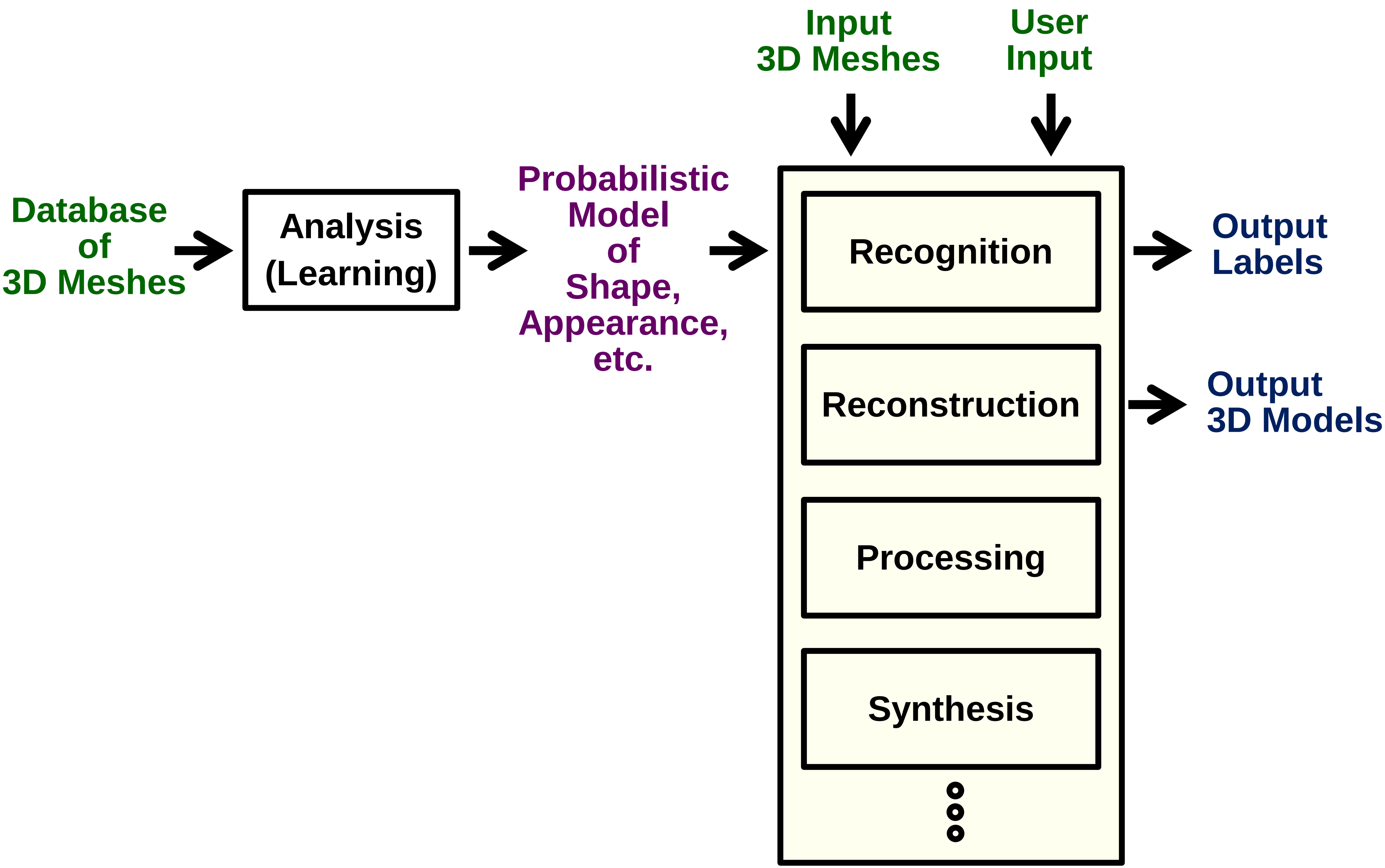
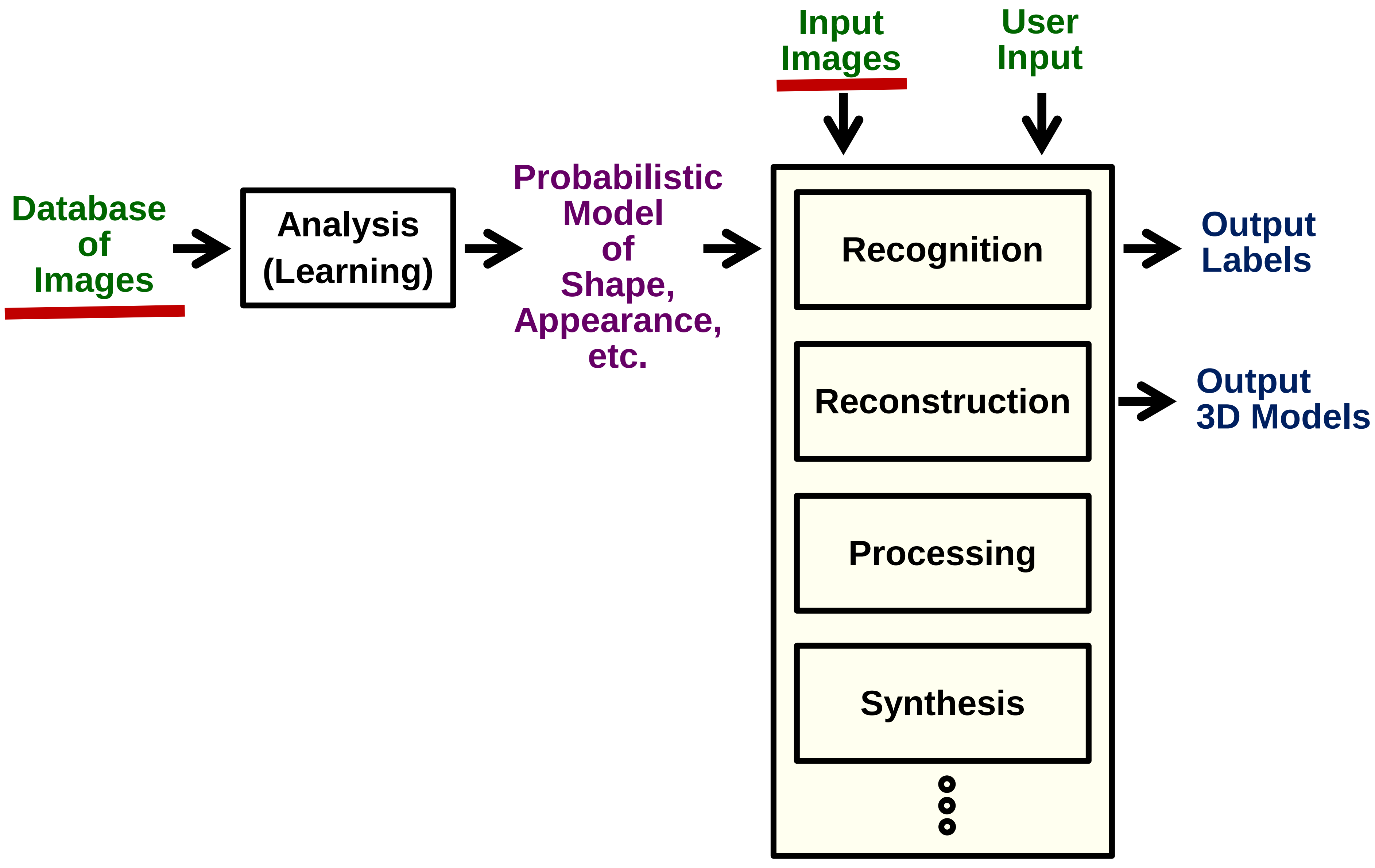
Learning Probabilistic Models from Collections of 3D Meshes

Sid Chaudhuri, Steve Diverdi, Matthew Fisher,
Pat Hanrahan, Vladimir Kim, Wilmot Li, Niloy Mitra,
Daniel Ritchie, Manolis Savva, and Thomas Funkhouser

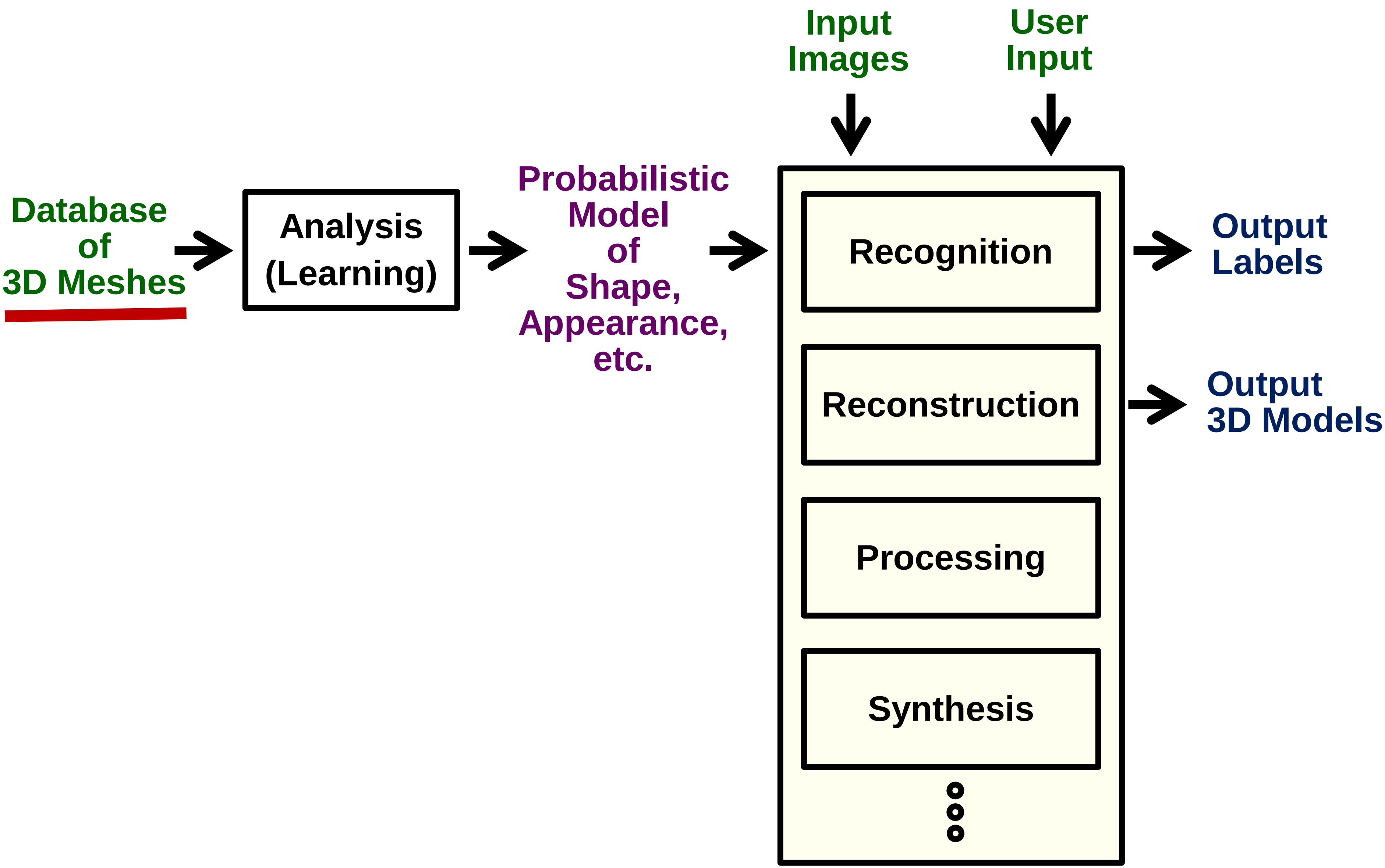
3D Shape Analysis



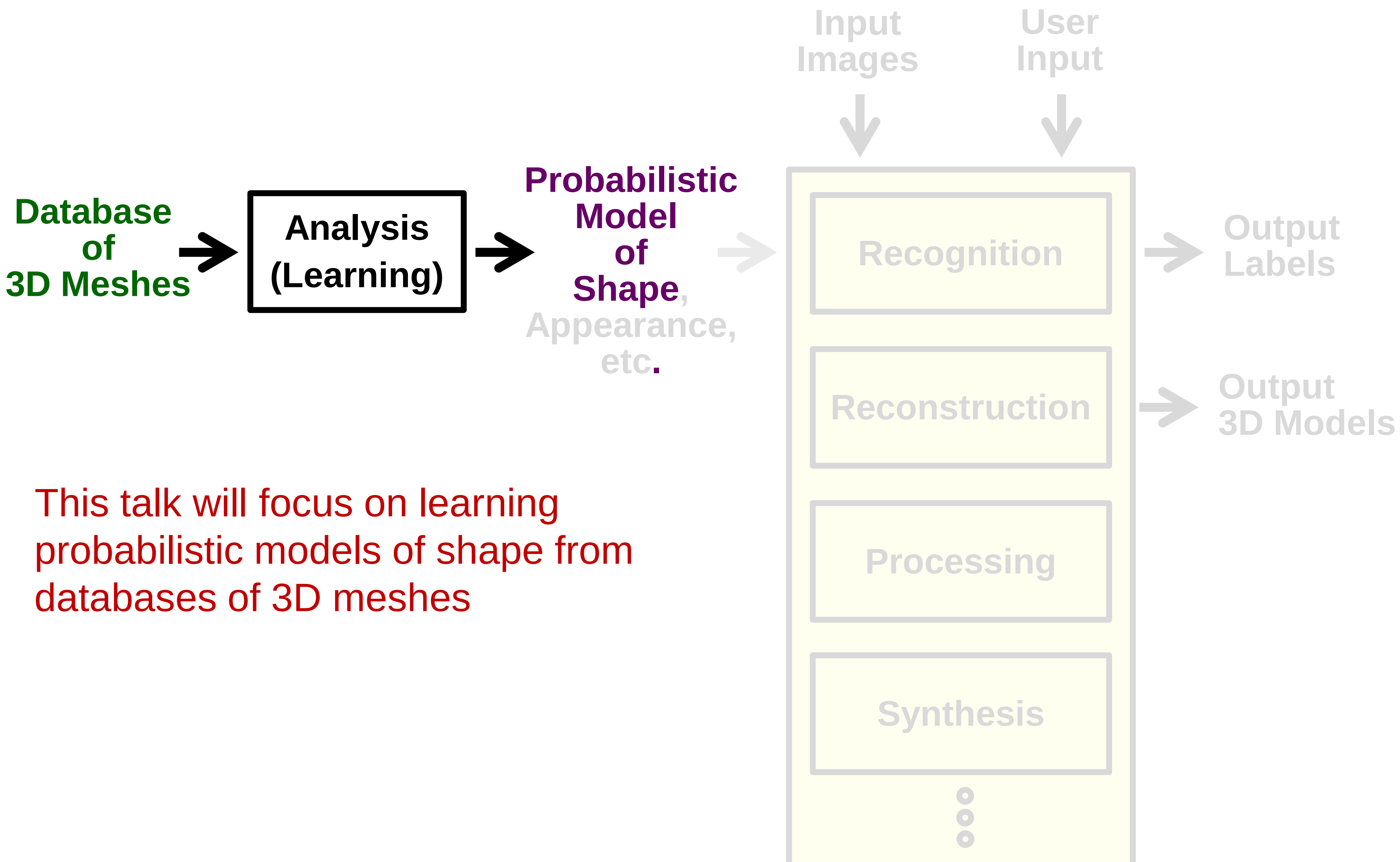
Computer Vision



3D Shape Analysis for Computer Vision?



Focus of This Talk



This talk will focus on learning probabilistic models of shape from databases of 3D meshes

Why 3D Shape Analysis?

Why analyze 3D meshes rather than images/scans?

- No noise
- No lighting
- No perspective
- No occlusions
- No pose estimation
- Easier segmentation
- Enough availability
- Large variety

Trimble 3D Warehouse

Why 3D Shape Analysis?

Why analyze 3D meshes rather than images/scans?

- No noise
- No lighting
- No perspective
- No occlusions
- No pose estimation
- Easier segmentation
- Enough availability
- Large variety
- ➢ Quality?

Trimble 3D Warehouse

Why 3D Shape Analysis?

Why analyze 3D meshes rather than images/scans?

- No noise
- No lighting
- No perspective
- No occlusions
- No pose estimation
- Easier segmentation
- Enough availability
- Large variety
- ➢ Quality?

Ikea

Related Work

Using databases of meshes for scene understanding

- Fitting 3D meshes to images
  - Lai 2009, Xu 2011, Satkin 2013, Aubry 2014, etc.
- Fitting 3D meshes to range scans
  - Nan 2012, Shen 2012, Kim 2012, Song 2014, etc.
- Using 3D meshes to learn parameters
  - Zhao 2013, etc.

Analyzing databases of meshes

- Consistent segmentation, labeling, correspondence, …
  - Golovinskiy 2009, Sidi 2011, Kim 2013, Mitra 2013, etc.
- Learning probabilistic models
  - Chaudhuri 2010, Kalogerakis 2012, Fisher 2012, Kim 2013, etc.

Outline of Talk

Introduction

Learning probabilistic models from 3D collections

- Part-based templates
- Generative model

Conclusions

Outline of Talk

Introduction

Learning probabilistic models from 3D collections

- Part-based templates
- Generative model

Conclusions

Vladimir G. Kim, Wilmot Li, Niloy J. Mitra, Siddhartha Chaudhuri, Stephen DiVerdi, and Thomas Funkhouser. "Learning Part-based Templates from Large Collections of 3D Shapes," SIGGRAPH 2013.

Goal for This Project

Database of 3D meshes representing an object class

Probabilistic Model of Shape

Consistent part segmentations, labels, and correspondences

Goal for This Project

Database of 3D meshes representing an object class

Probabilistic Model of Shape

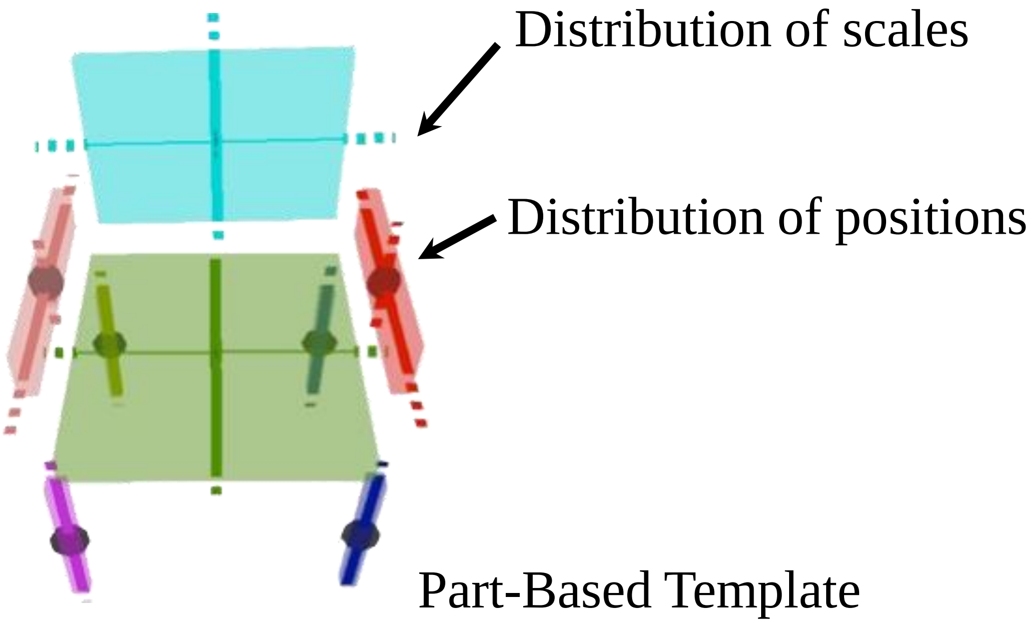
Consistent part segmentations, labels, and correspondences

Challenge

Need to discover segmentations, labels, correspondences, and deformation modes all together

Part-Based Templates

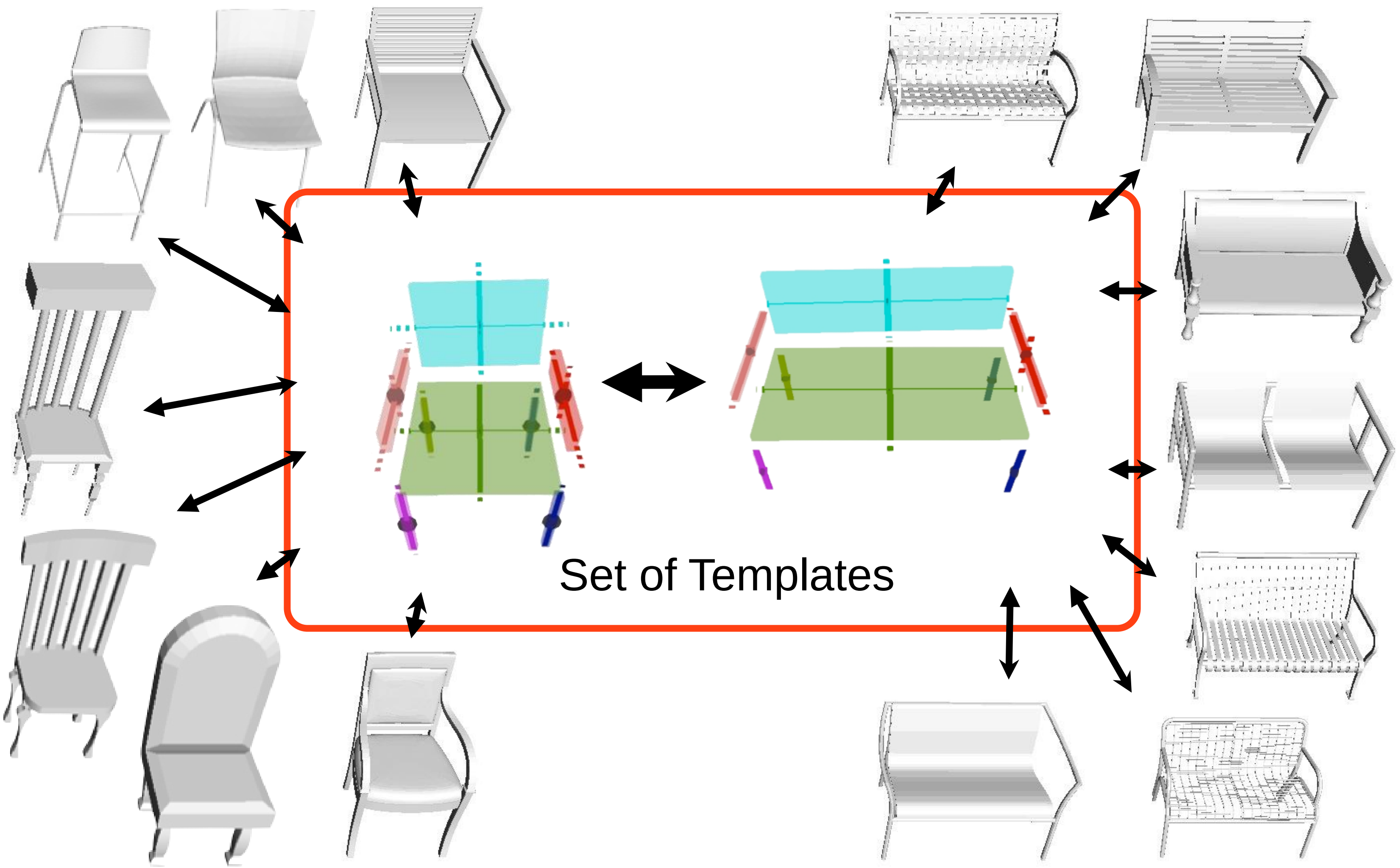
Represent object class by part-based templates where each template has a set of parts, and each part has probability distributions for its shape, position, and anisotropic scales



Part-Based Template

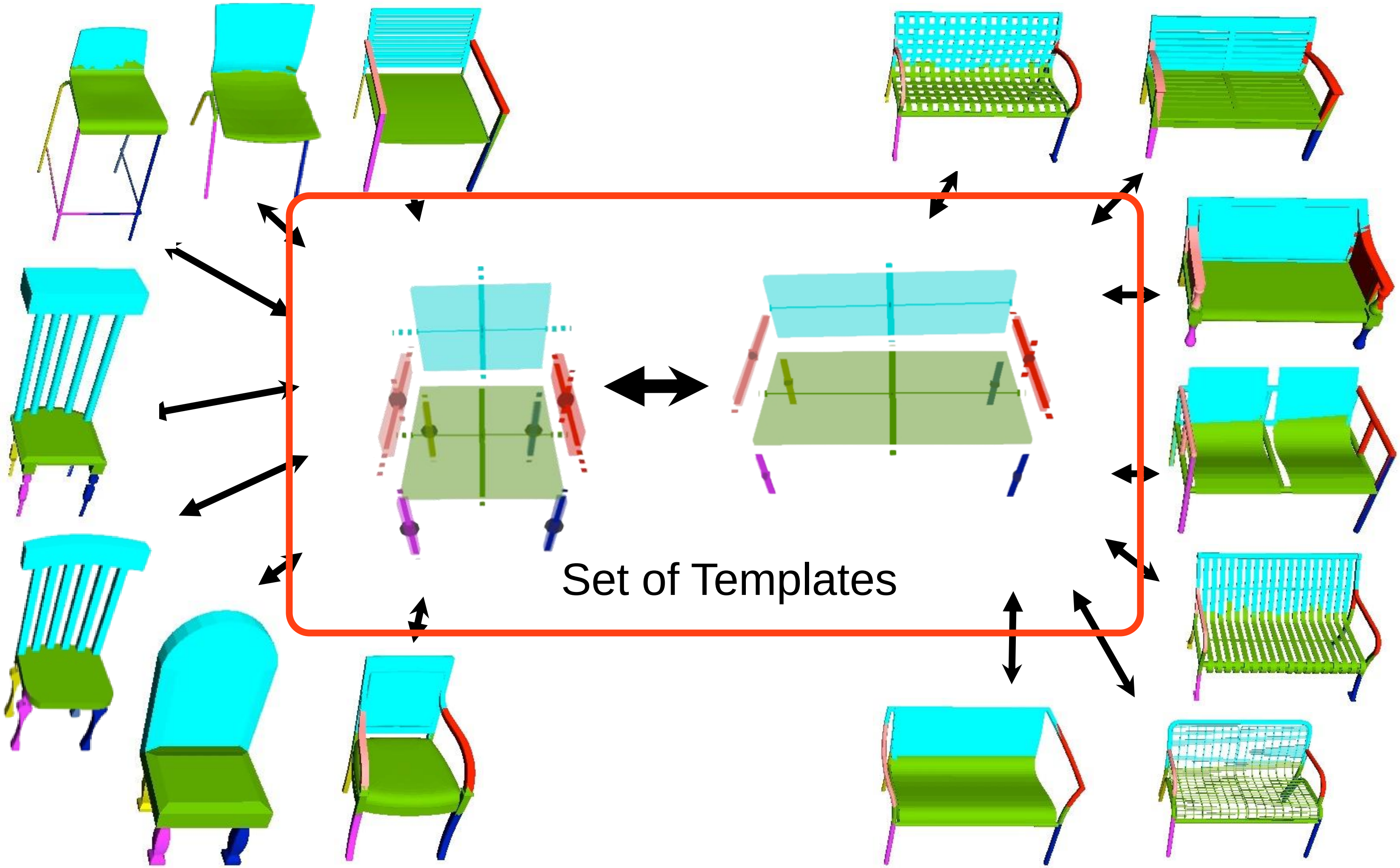
Template Learning and Fitting

Aim to learn a set of corresponding templates that provides a good fit to every mesh in the database



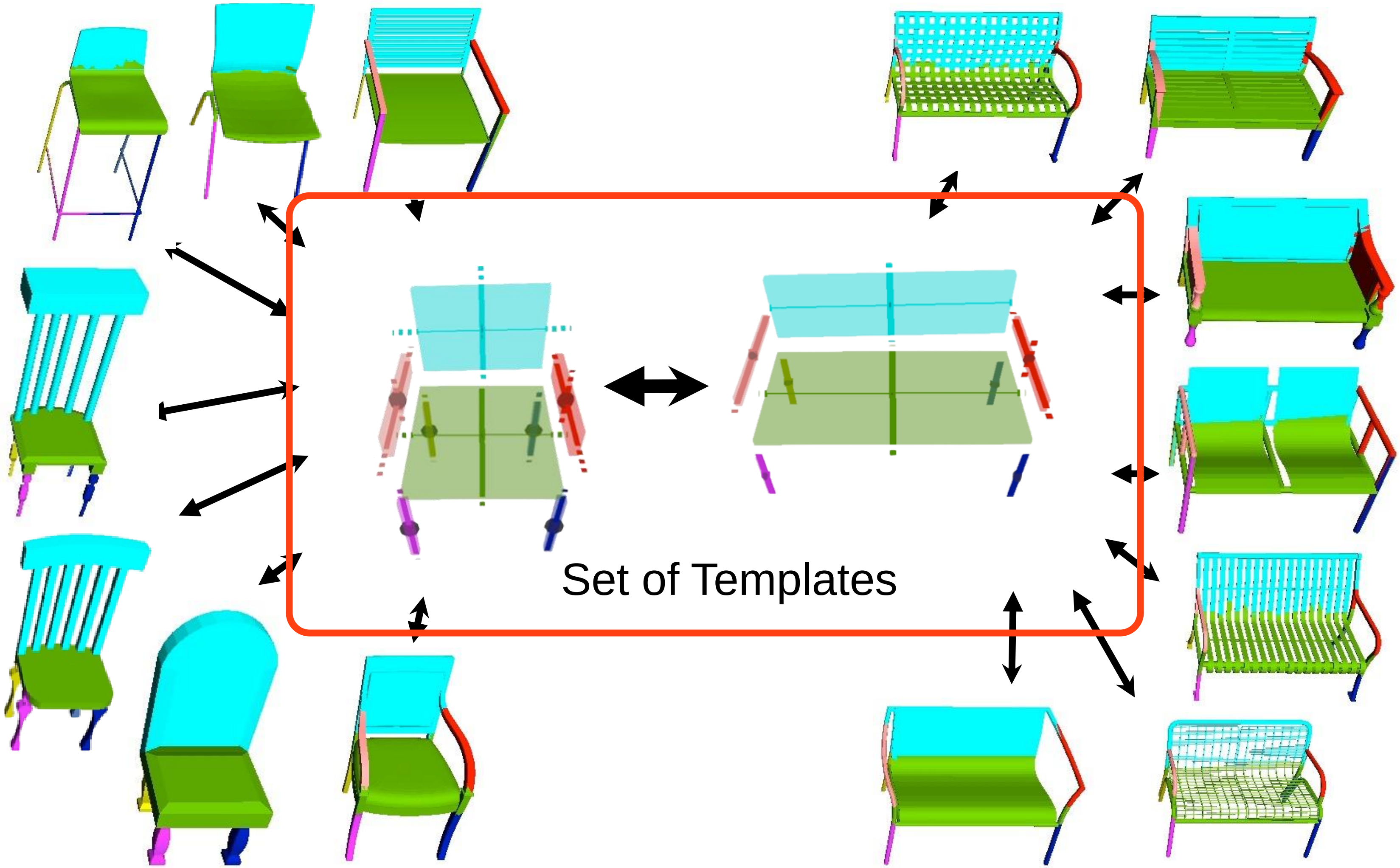
Template Learning and Fitting

Aim to learn a set of corresponding templates that provides a good fit to every mesh in the database



Template Learning and Fitting

Aim to learn a set of corresponding templates that provides a good fit to every mesh in the database

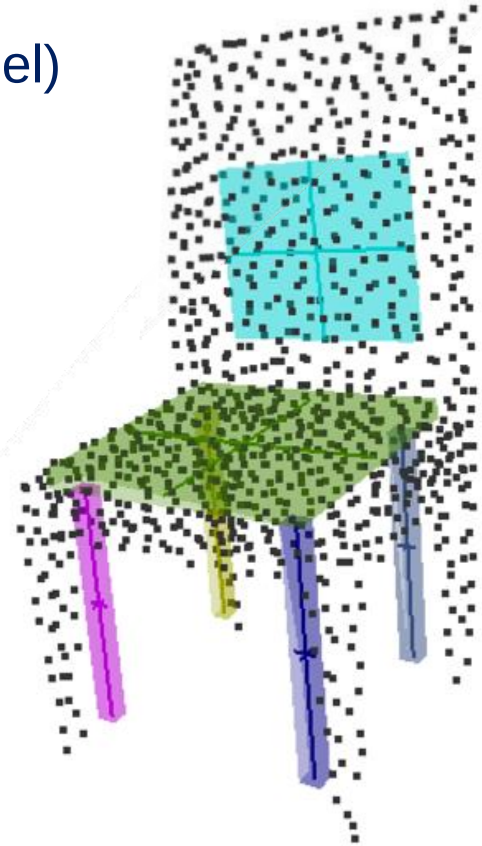


Template Fitting Problem

For a given template and mesh, aim to minimize:

$$E = E_{\mathrm{data}} + \gamma E_{\mathrm{deform}} + \beta E_{\mathrm{smooth}}$$

- E_{data} (template ↔ shape distance + local shape features)
- E_{deform} (plausibility of template deformation)
- E_{smooth} (close & similar regions get same label)



Unknowns are:

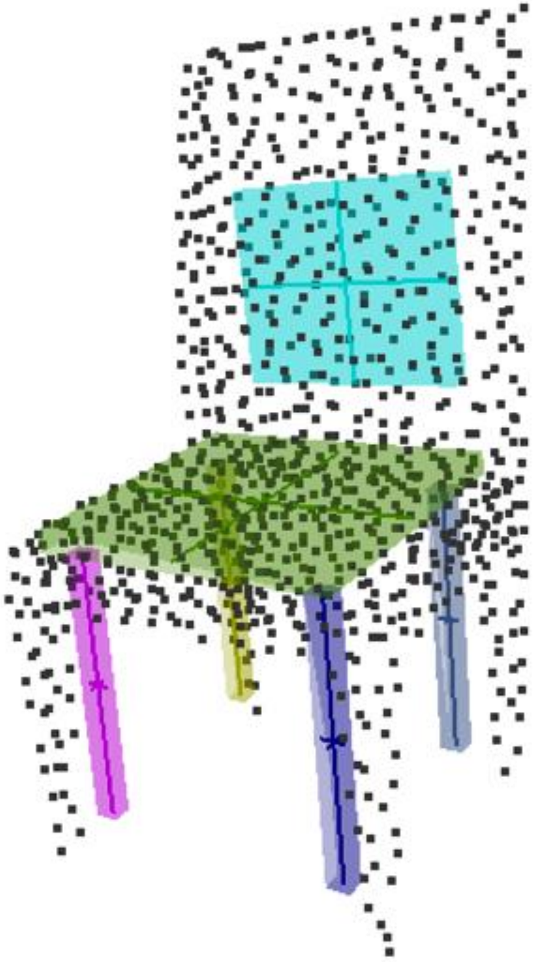
- Point segmentations and labels
- Point correspondences
- Part center positions
- Part anisotropic scales

Template Fitting Algorithm

Solve by iteratively minimizing different energy terms:

- Segmentation and labeling
- Point correspondence
- Part-aware deformation

$$E = E_{\text{data}} + \gamma E_{\text{deform}} + \beta E_{\text{smooth}}$$

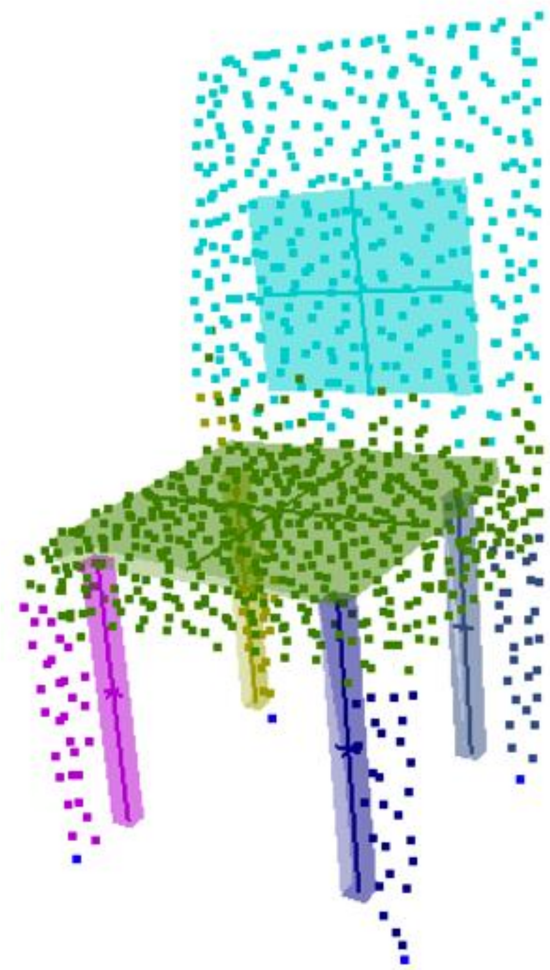


Template Fitting Algorithm

Solve by iteratively minimizing different energy terms:

- ➢ Segmentation and labeling
- ◦ Point correspondence
- ◦ Part-aware deformation

$$E = E_{\text{data}} + \gamma E_{\text{deform}} + \beta E_{\text{smooth}}$$



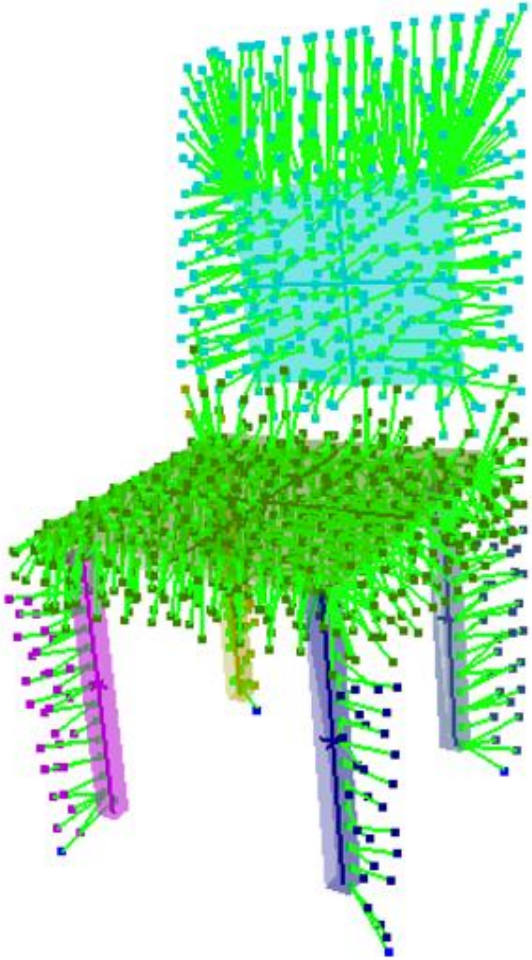
Solve with graph cut [Boykov 2001]

Template Fitting Algorithm

Solve by iteratively minimizing different energy terms:

- Segmentation and labeling
- ➢Point correspondence
- Part-aware deformation

$$E = E_{\text{data}} + \gamma E_{\text{deform}} + \beta E_{\text{smooth}}$$



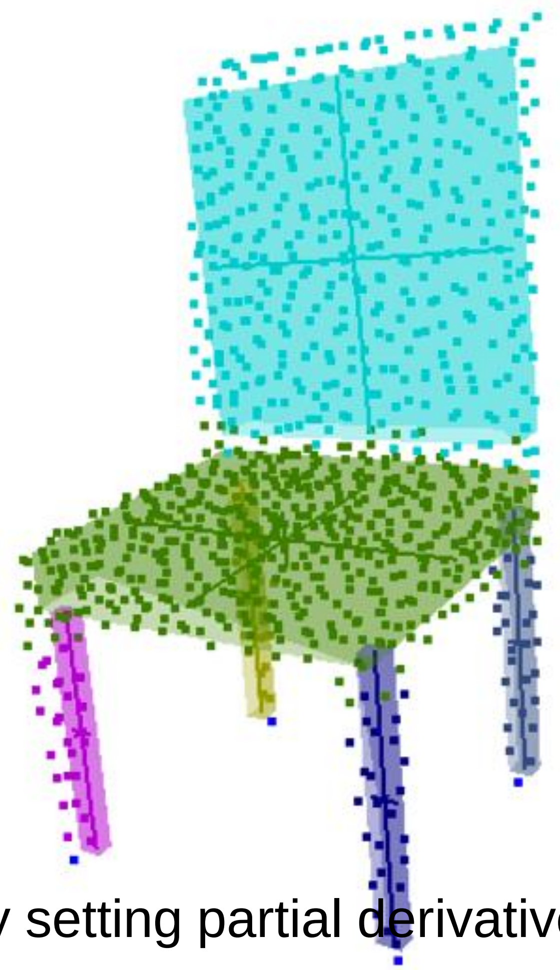
Solve with part-aware closest points

Template Fitting Algorithm

Solve by iteratively minimizing different energy terms:

- ◦ Segmentation and labeling
- ◦ Point correspondence
- ➢ Part-aware deformation

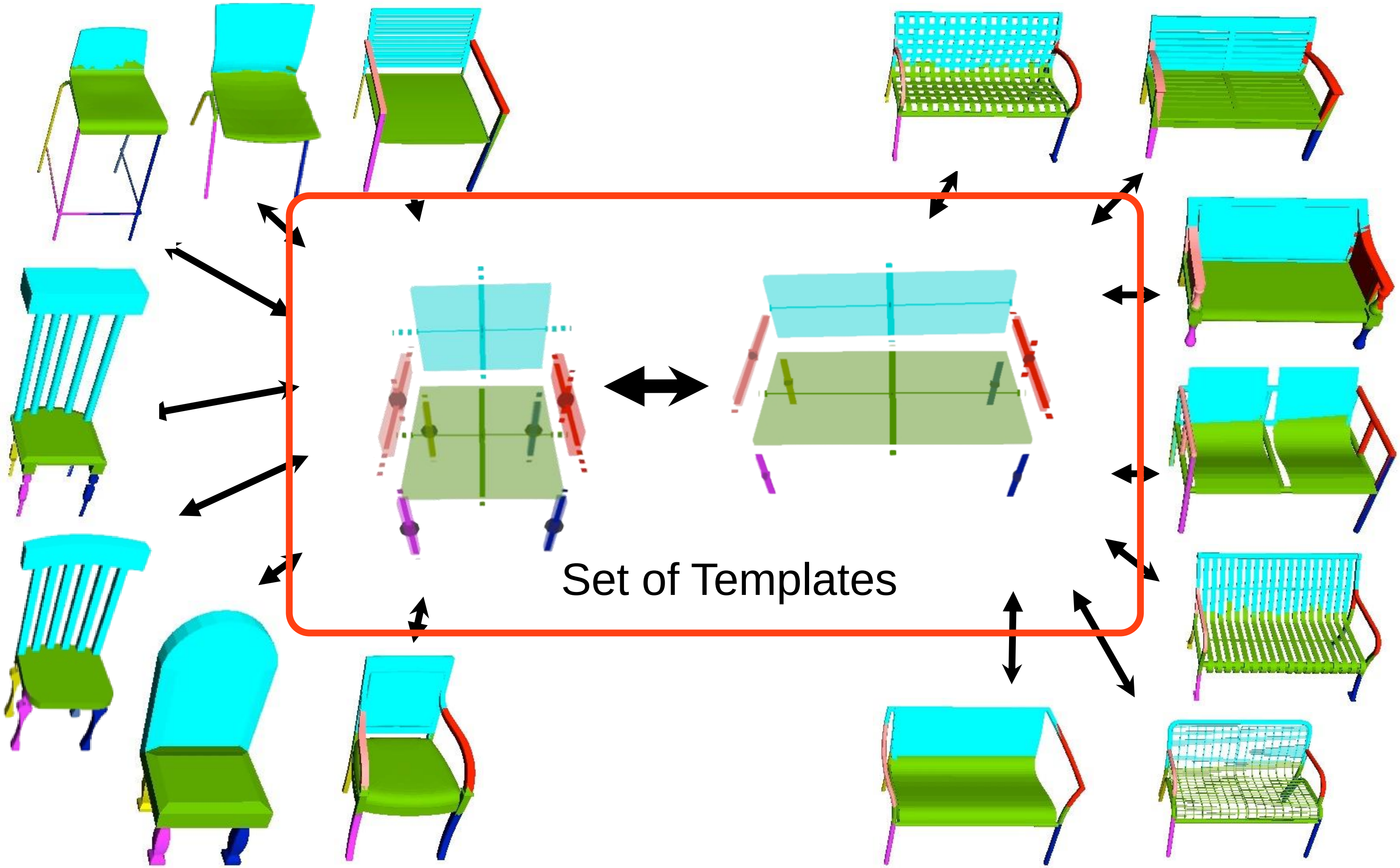
$$E = E_{\text{data}} + \gamma E_{\text{deform}} + \beta E_{\text{smooth}}$$



Solve for positions and scales of each part by setting partial derivatives to zero.

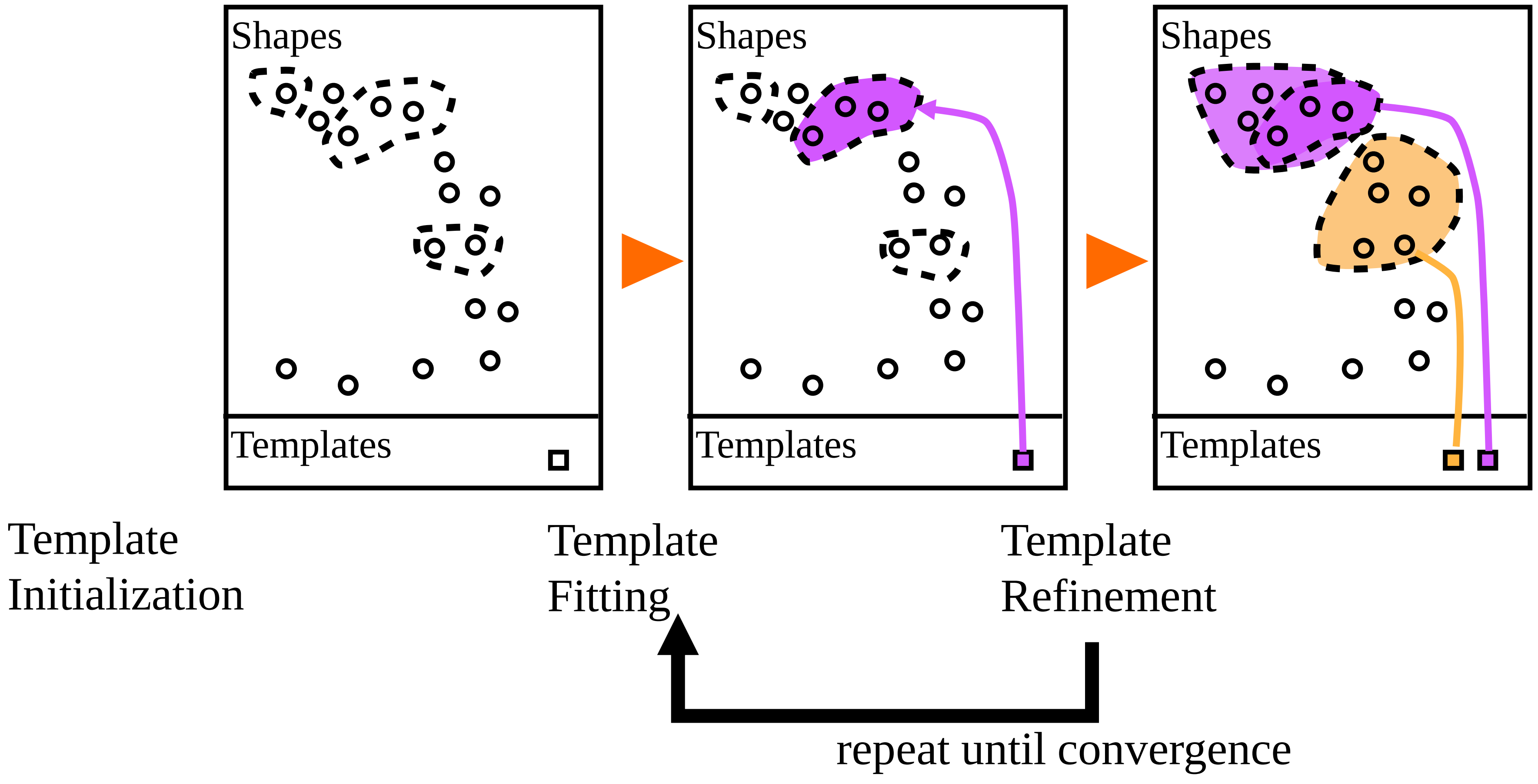
Template Learning Problem

Aim to learn a set of corresponding templates that provides a good fit to every mesh in the database

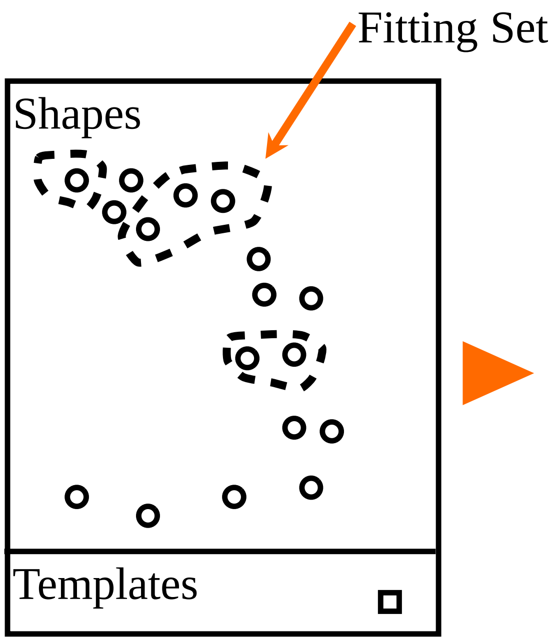


Template Learning Algorithm

Iteratively grow a set of templates with each optimized to fit a different cluster of meshes



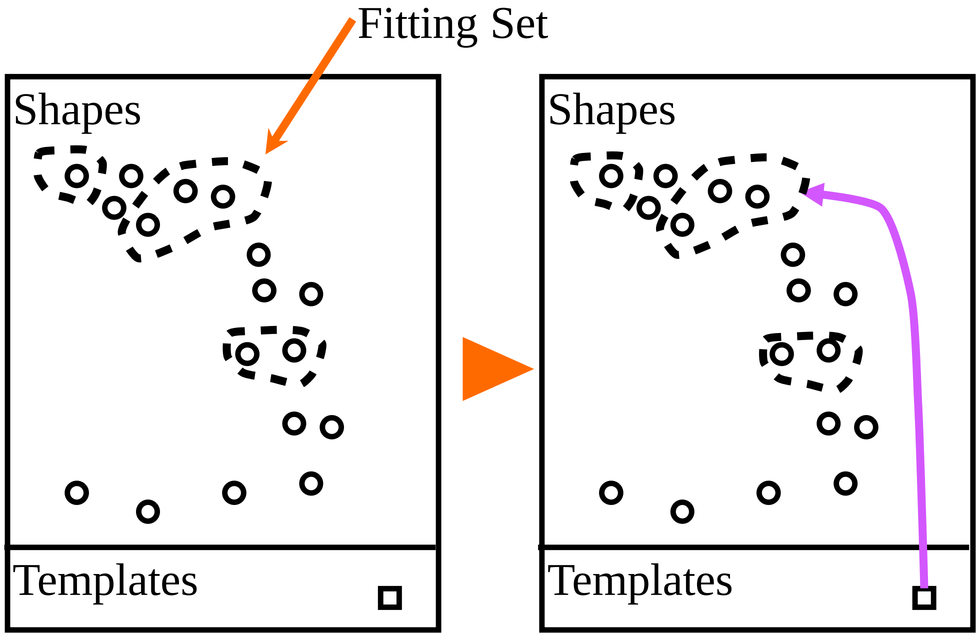
Template Learning Algorithm



Template Initialization

Template Fitting

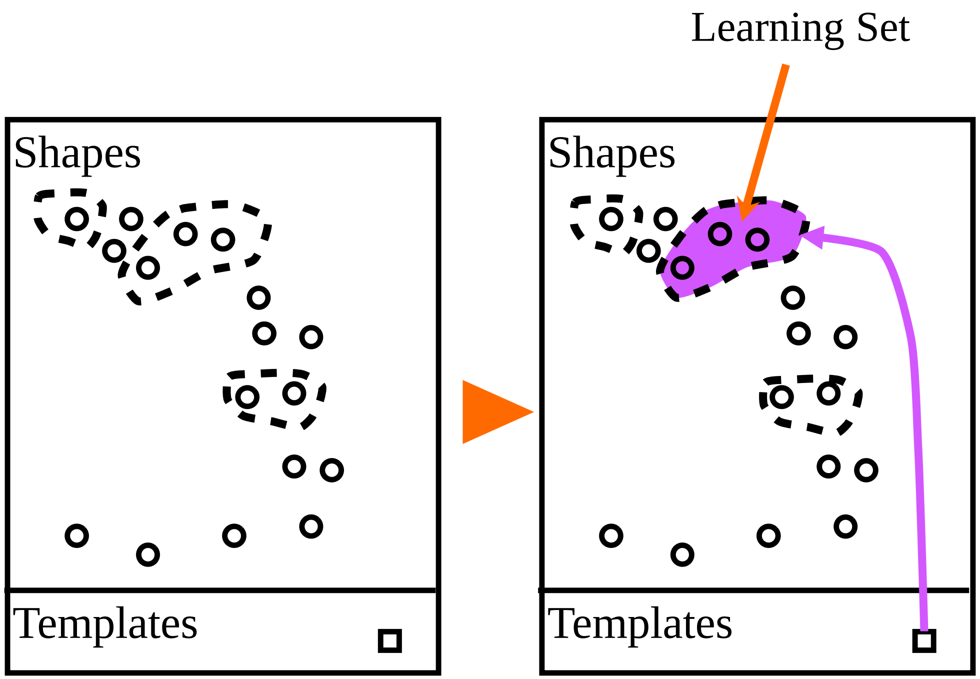
Template Learning Algorithm



Template
Initialization

Template
Fitting

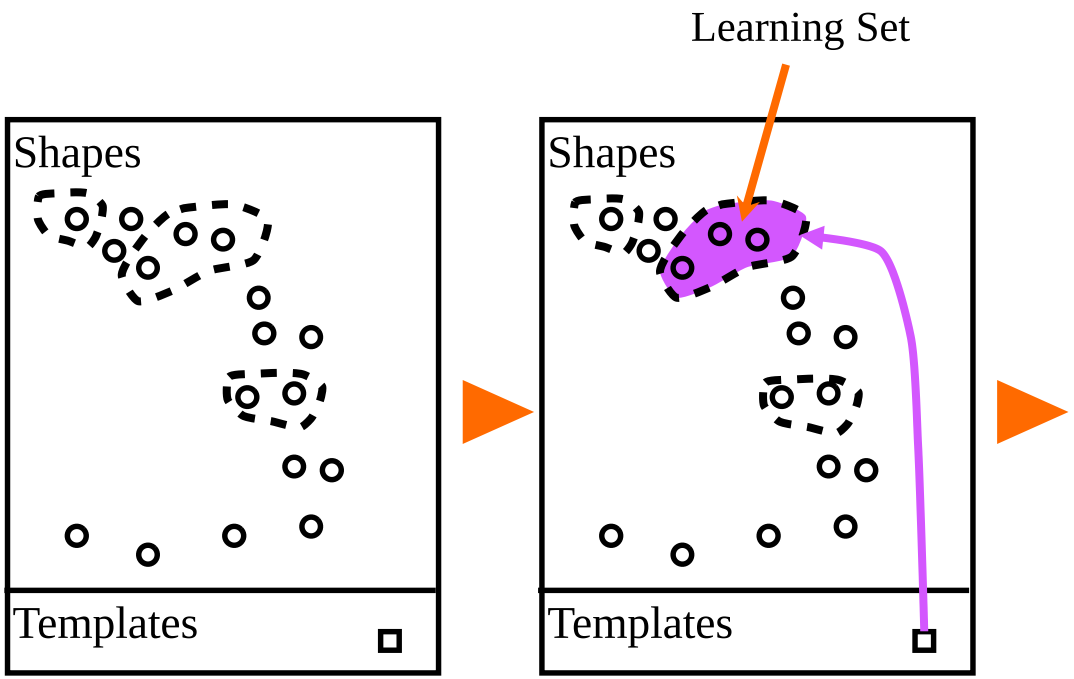
Template Learning Algorithm



Template
Initialization

Template
Fitting

Template Learning Algorithm

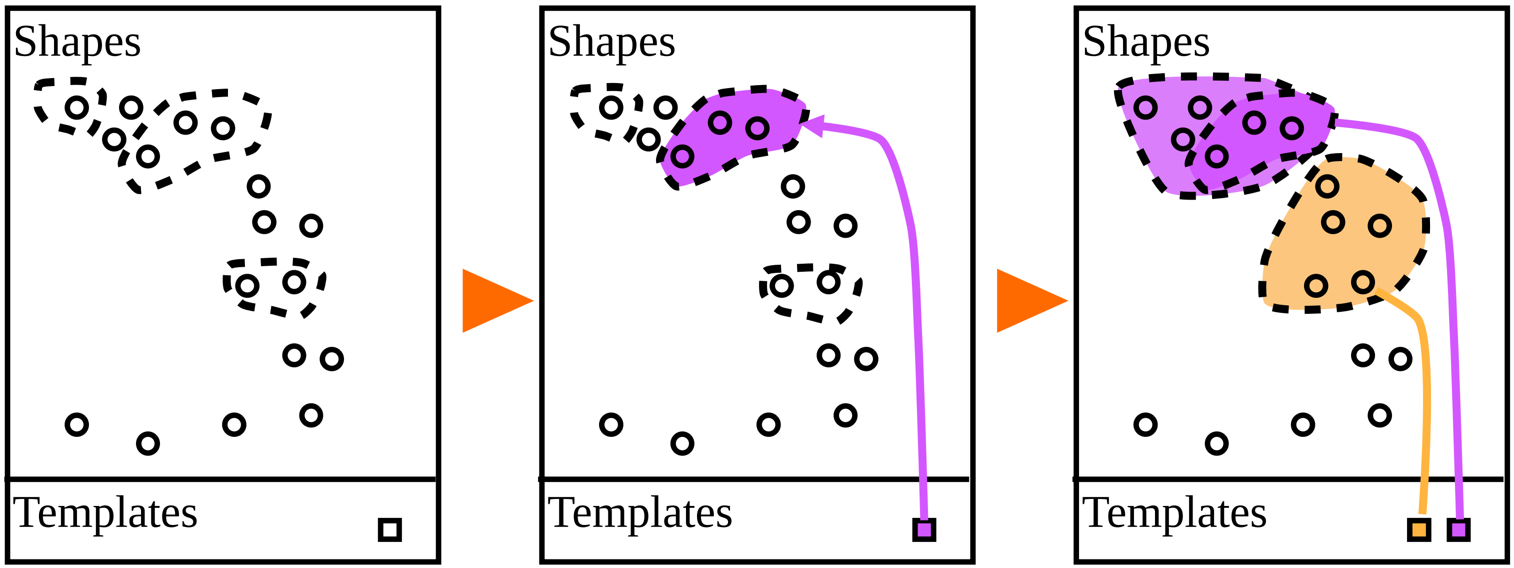


Template Initialization

Template Fitting

Template Refinement

Template Learning Algorithm

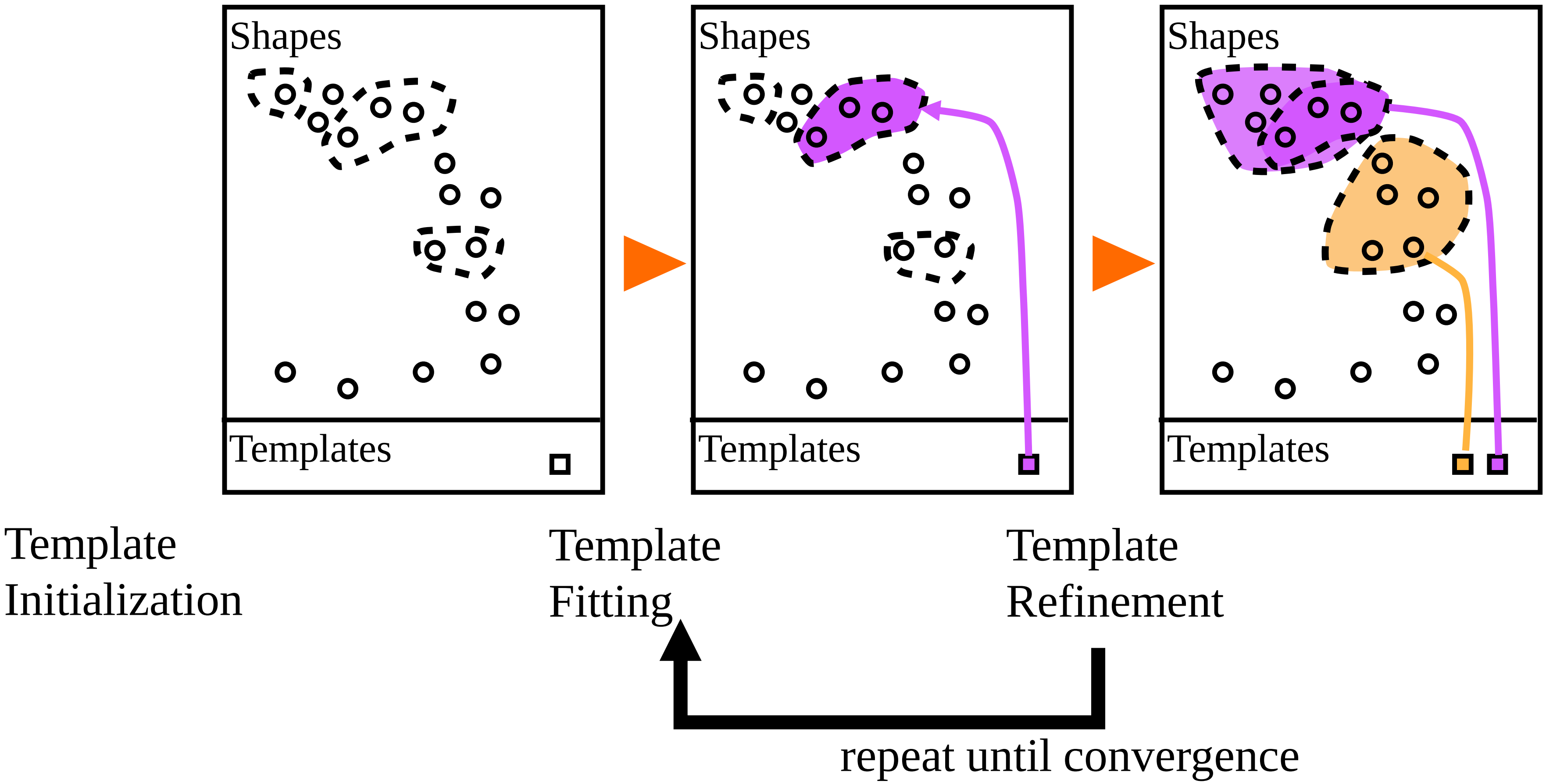


Template
Initialization

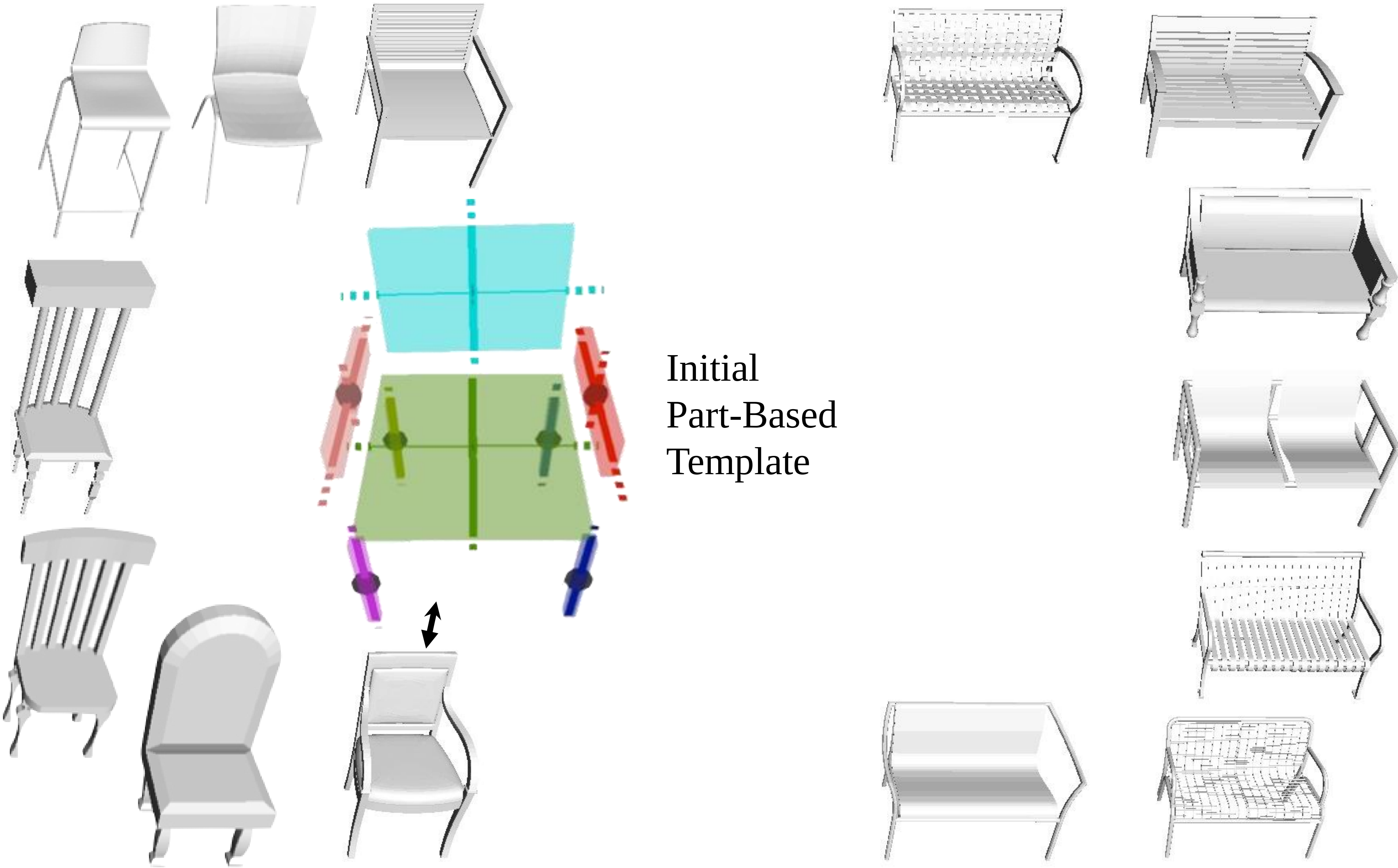
Template
Fitting

Template
Refinement

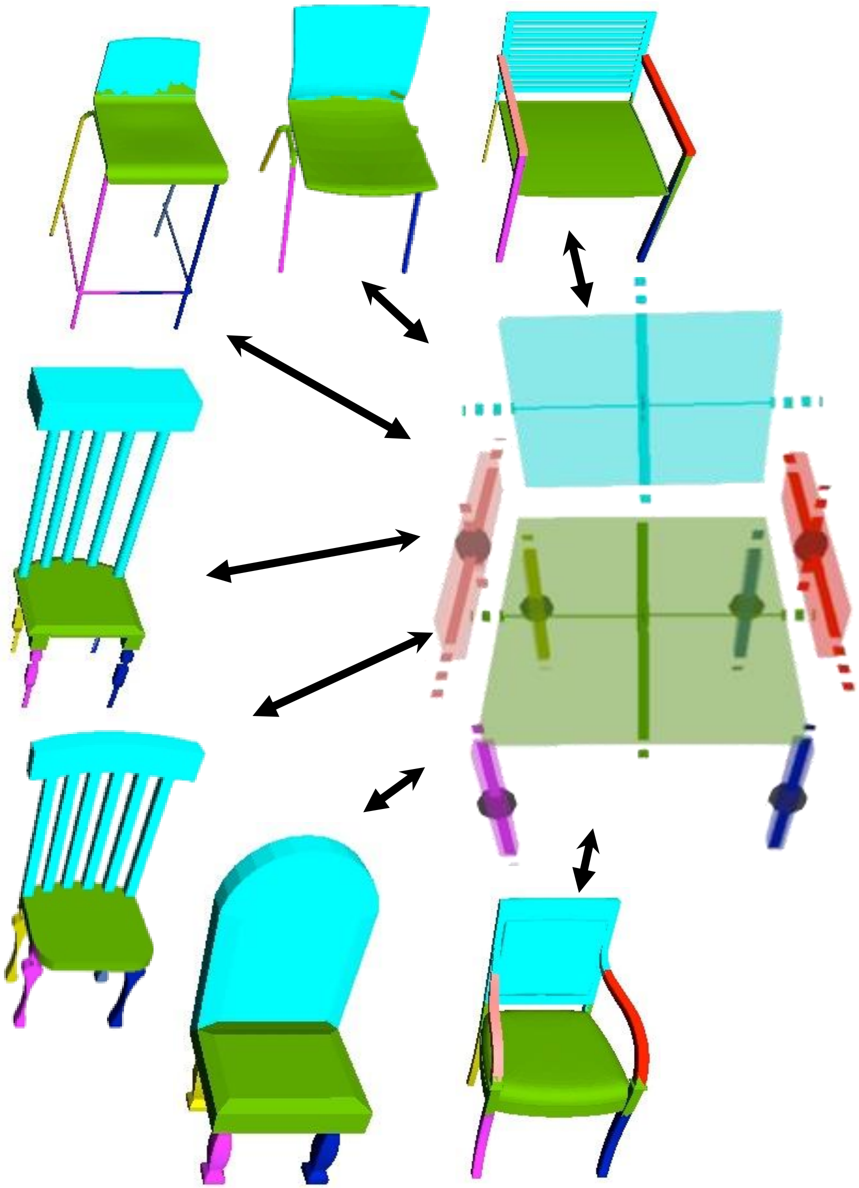
Template Learning Algorithm



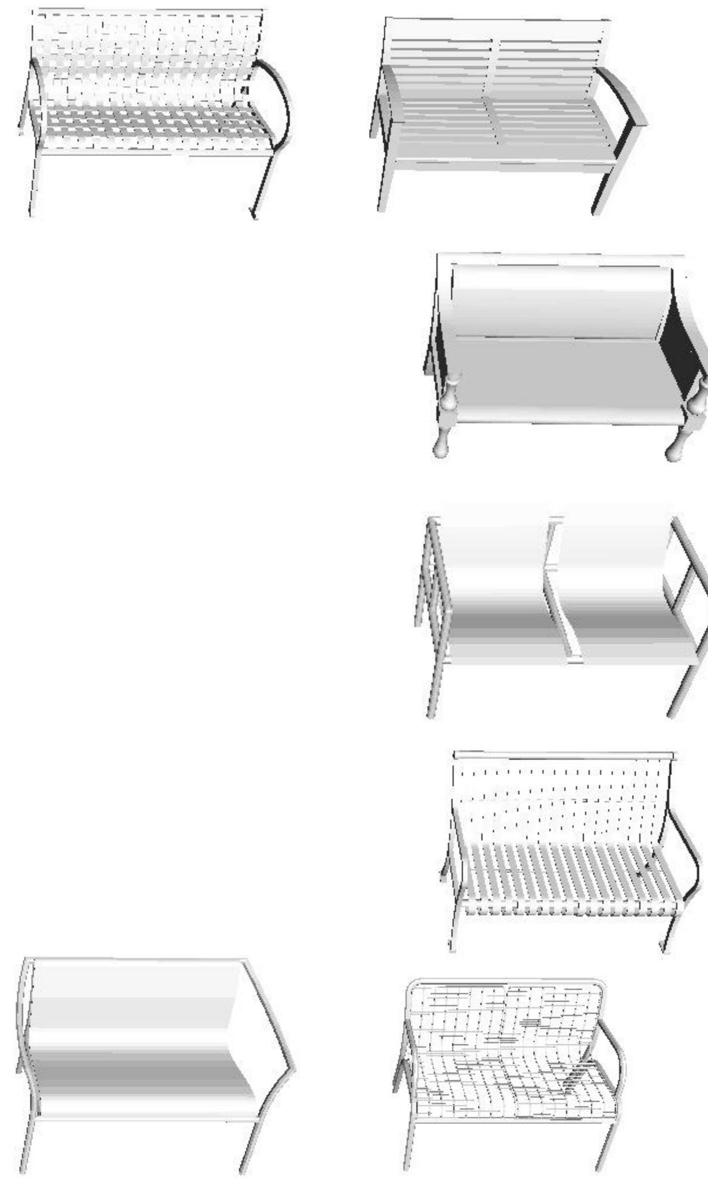
Template Learning Example



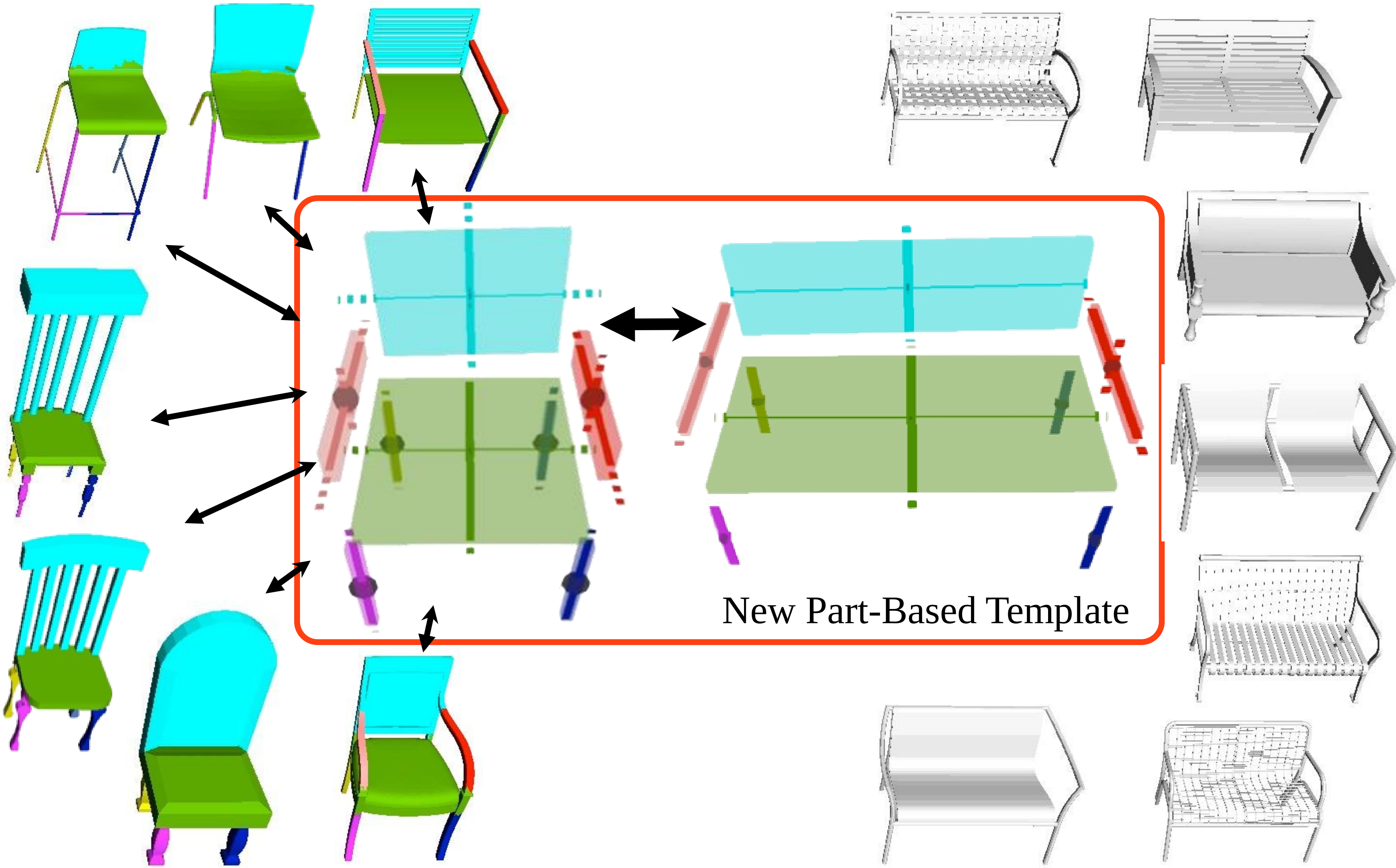
Template Learning Example



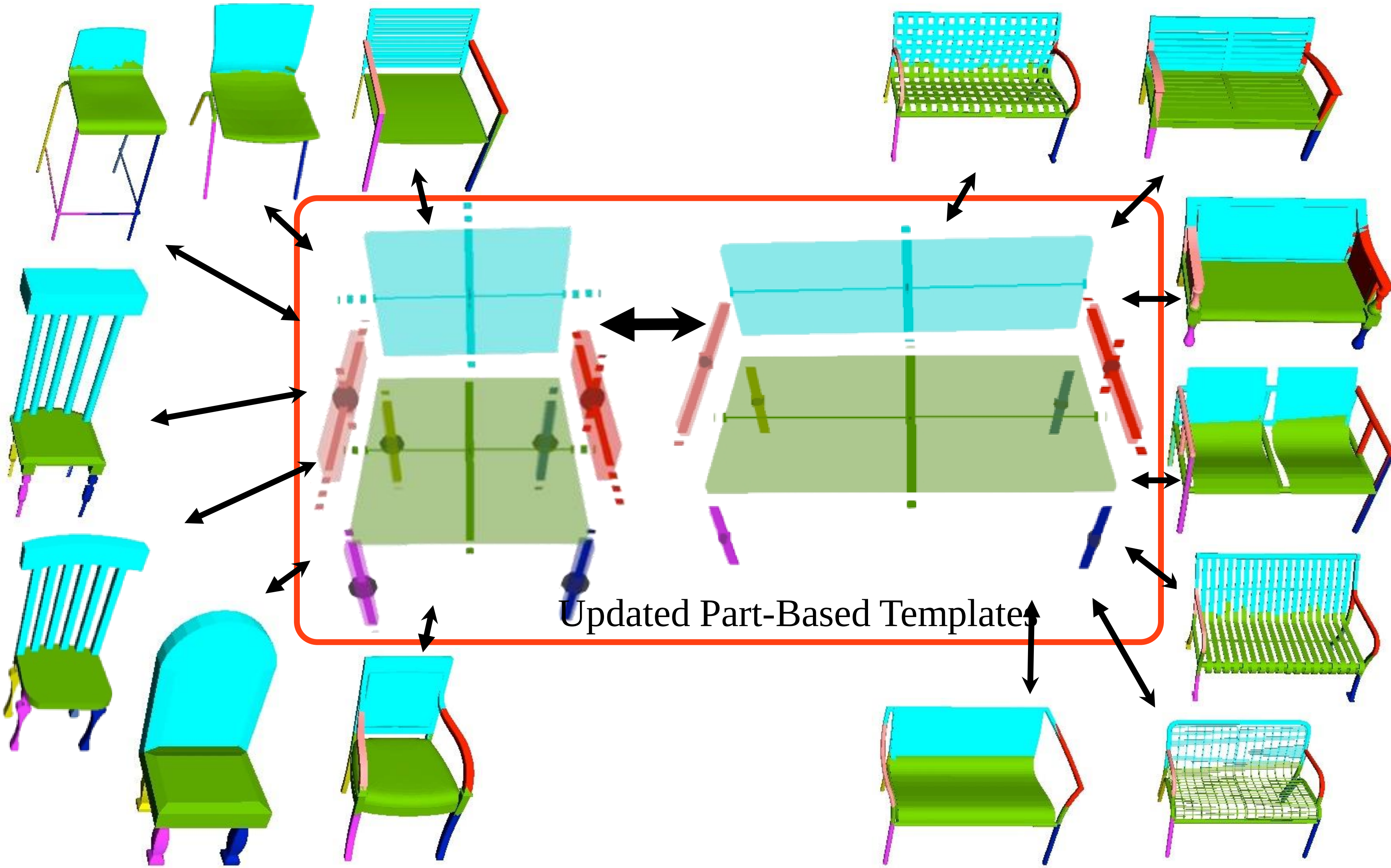
Updated
Part-Based
Template



Template Learning Example



Template Learning Example



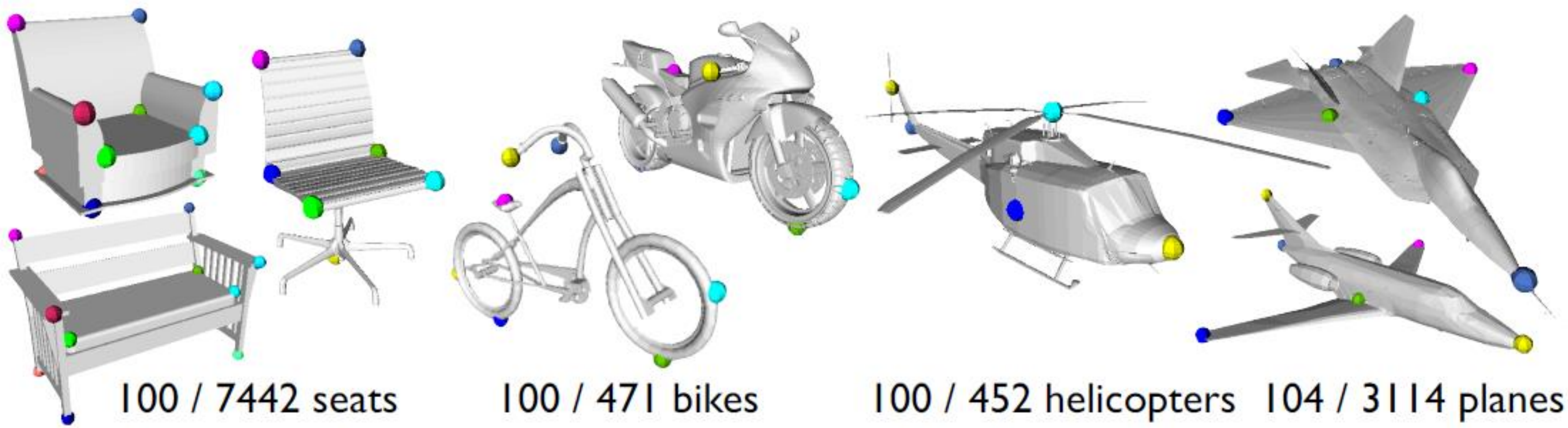
Template Learning and Fitting Results

Data sets:

- Crawl SketchUp Warehouse for collections by keyword
- Eliminate outliers with Mechanical Turk
- Specify manual correspondences for subset of models

Experiments:

- Solve for part-based templates for collection
- Evaluate correspondences & segmentations



Template Learning and Fitting Results

2 Templates

3113 Airplanes

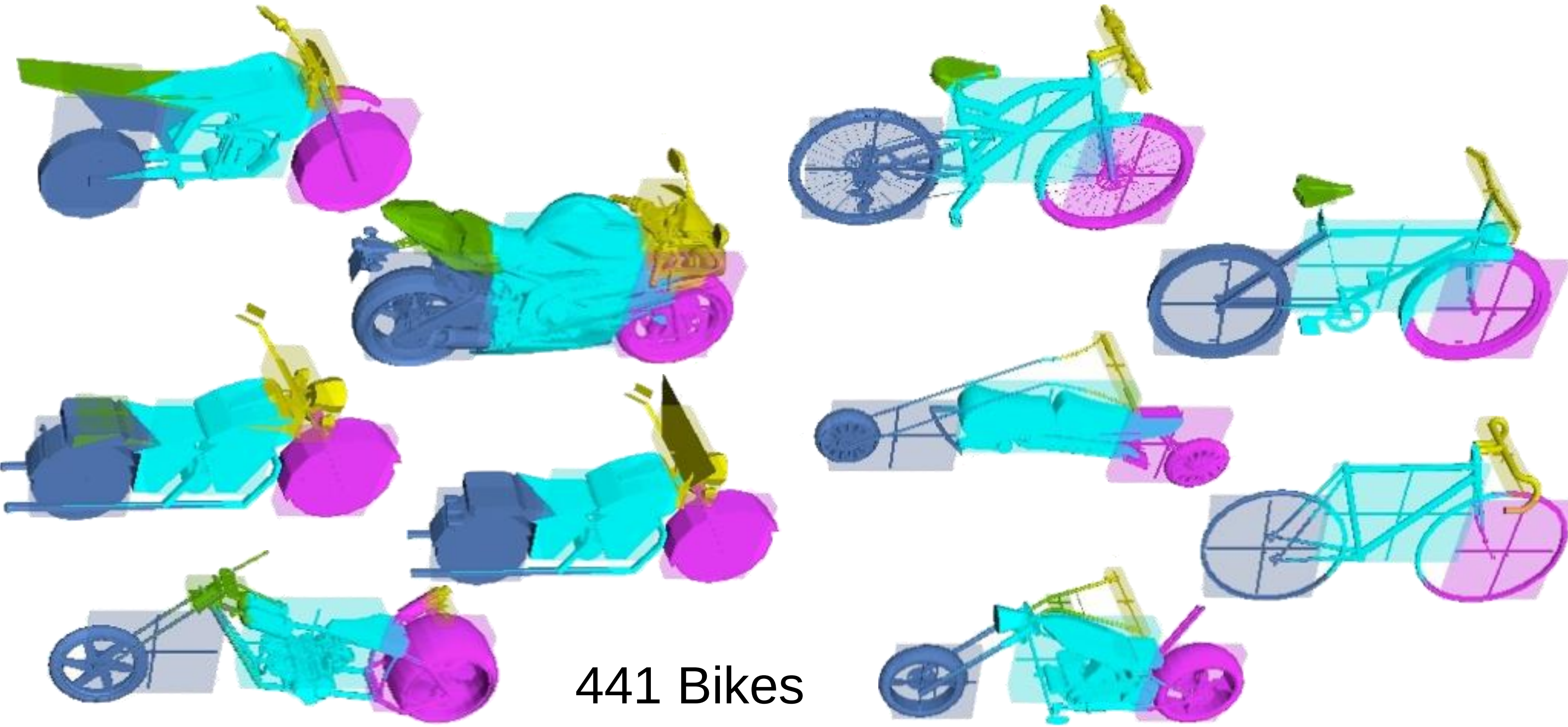
Template Learning and Fitting Results

2 Templates

3113 Airplanes

Template Learning and Fitting Results

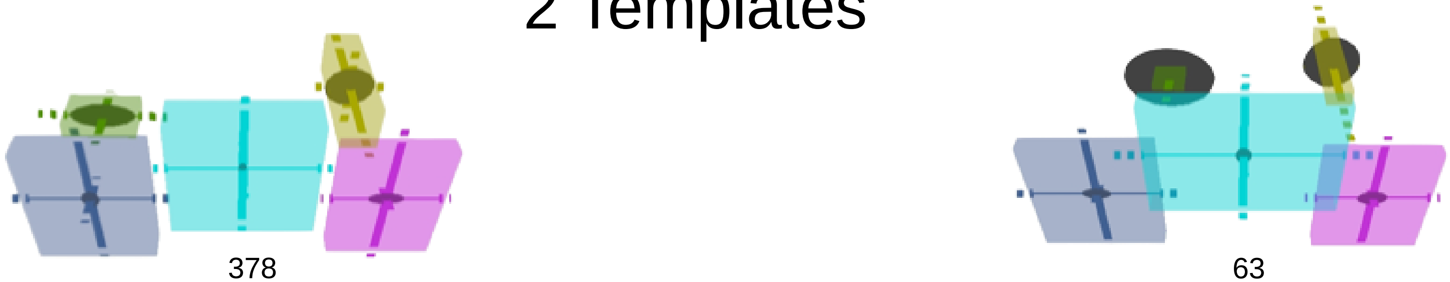
2 Templates

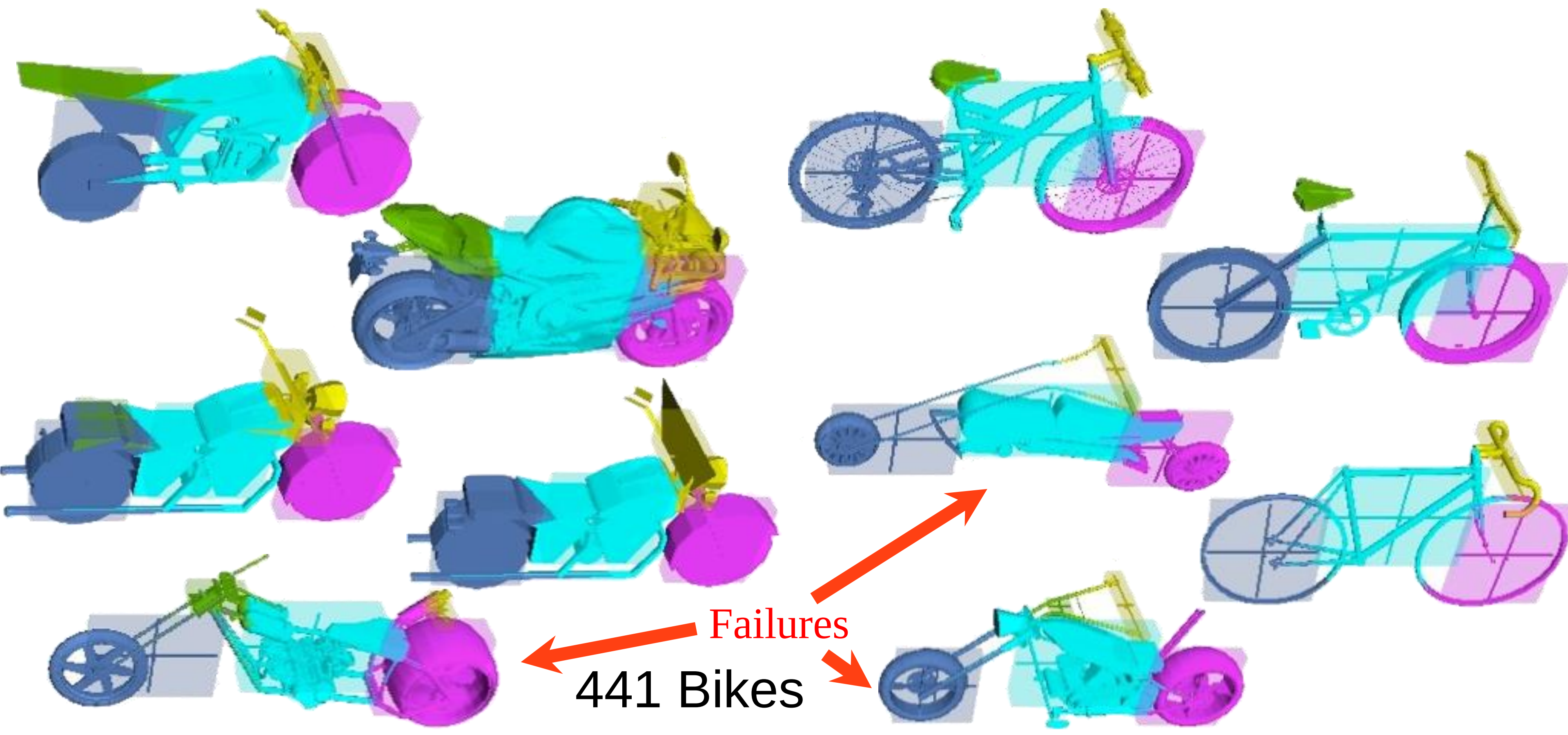


441 Bikes

Template Learning Results

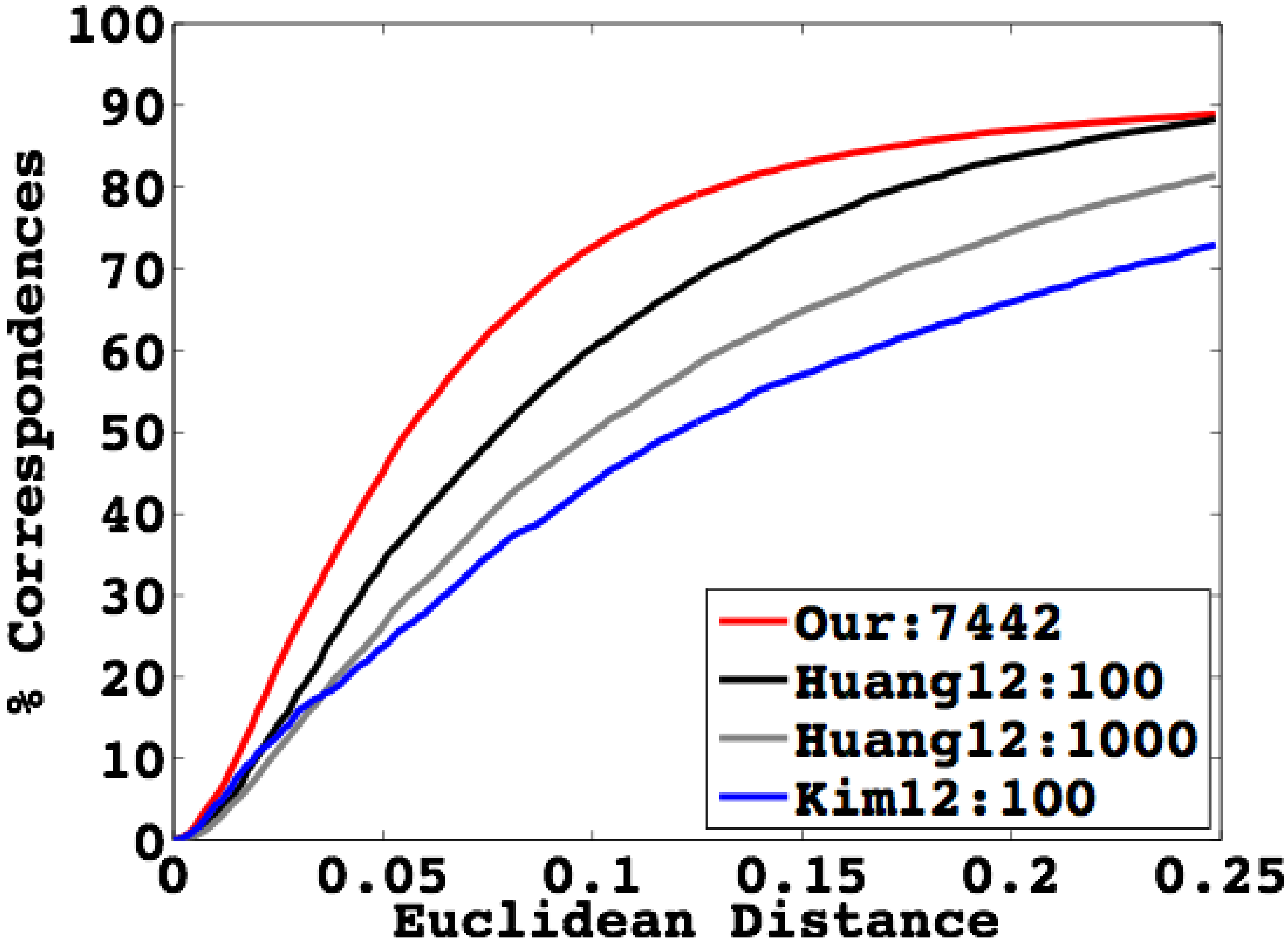
2 Templates





Surface Correspondence Results

Correspondence benchmark (7442 seats)



Surface Segmentation Results

Co-segmentation benchmark [Sidi et al, 2011]

Class	Hu	Our
Chairs	89.6	97.6
Lamps	90.7	95.2
FourLegged	88.7	86.9
Goblets	99.2	97.6
Vase	80.2	81.3
Guitars	98.0	88.5
Candelabra	93.9	82.4

within 2%
or ours
is better

Outline of Talk

Introduction

Learning probabilistic models from 3D collections

- Part-based templates
- Generative model

Conclusions

Matthew Fisher, Daniel Ritchie, Manolis Savva, Thomas Funkhouser, and Pat Hanrahan,
Example-based Synthesis of 3D Object Arrangements, SIGGRAPH Asia, 2012.

Goal for This Project

Exemplar scenes

+

Database of Scenes

Probabilistic Model of Shape

Synthesized novel scenes

Goal for This Project

Exemplar scenes

+

...

Database of Scenes

⇨ Probabilistic Model of Shape

⇩

Synthesized novel scenes

Challenge
Need to learn a model with great generality from few examples

Contextual Object Categories

Define categories of objects based on their contexts in a scene rather than basic functions

- Learned from examples by clustering of objects with similar spatial neighborhoods

Some Contextual Object Categories

Generative Model

Represent the probability of a scene S by a generative model based on category cardinalities (c), support hierarchy topology relationships (t), and spatial arrangement relationships (a)

$$P(S) = P(c,t,a) = P(a|t,c)\, P(t|c)\, P(c)$$

Exemplar scenes

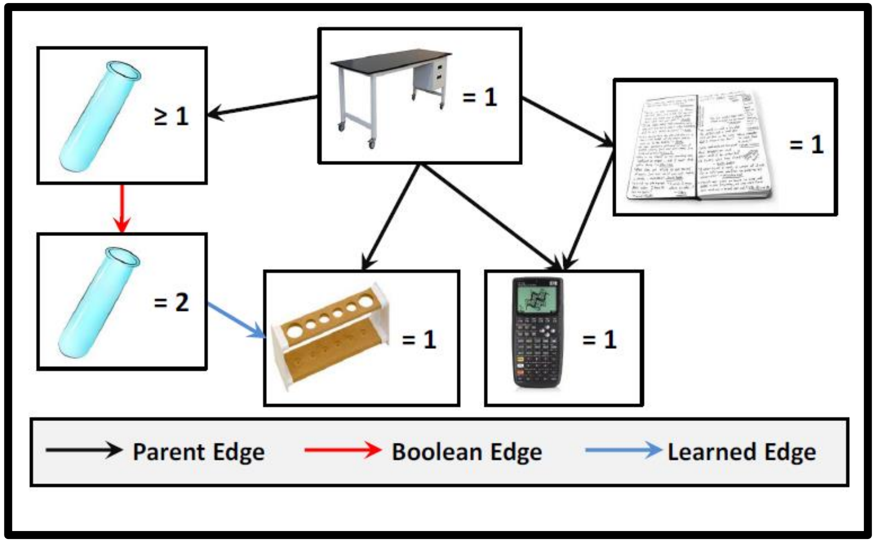
Generative Model Details

Category cardinalities: $P(c)$

- Represent with Bayesian network
- Boolean random variables (# desks > 1?)
- Add support surface constraints

Lab table	Test tube rack	Test tube	Notebook	Calculator
1	1	2	0	0
1	0	1	0	1
1	0	0	1	1

Object frequencies in target scenes
+ support constraints



Bayesian network

Generative Model Details

Support relationships: $P(t|c)$

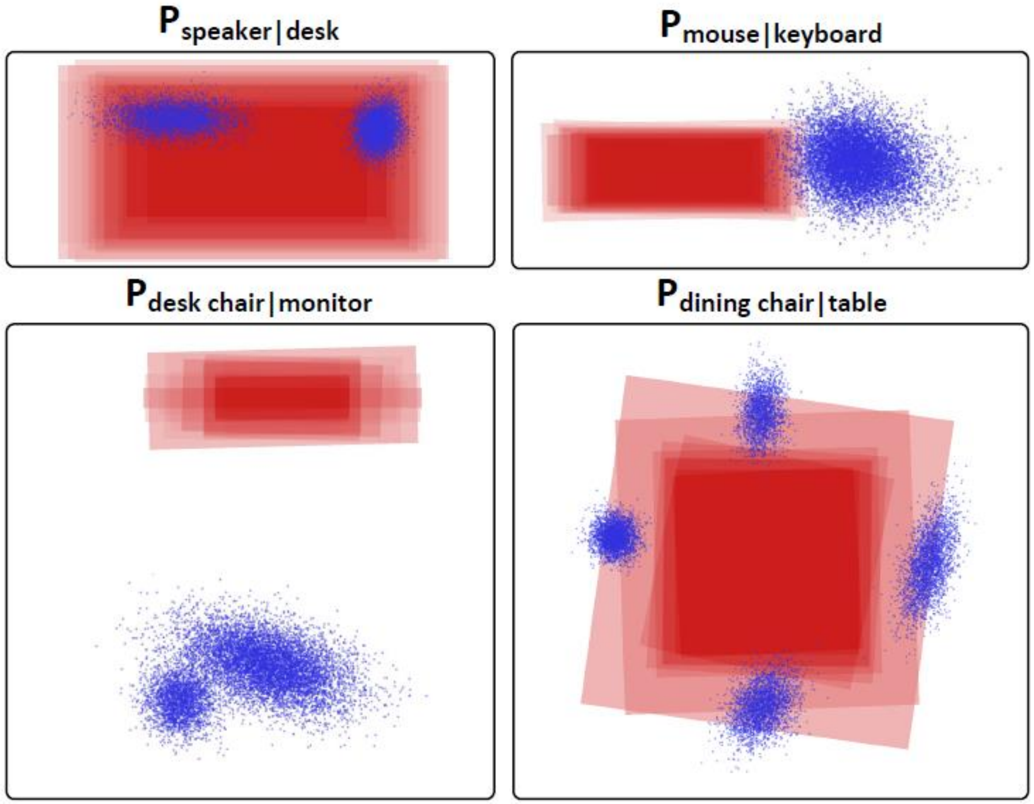
- Boolean random variables (desk supports keyboard?)
- Learn frequencies for pairs of categories
- Total probability is product over all objects in scene

$$P(t|c) = \prod_{o} P(C(o), C(support(o)))$$

Generative Model Details

Spatial arrangements: $P(a|t,c)=R(a,t,c)S(a,t,c)$

- Random variables for relative positions and orientations
- Pairwise distributions of spatial relationships

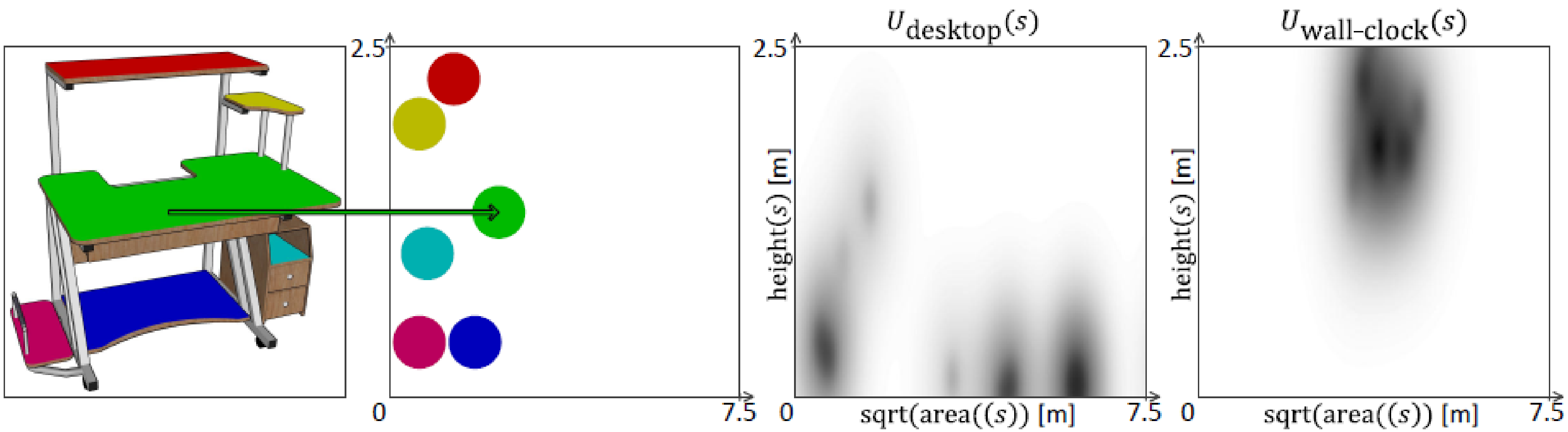


Distributions of spatial relationships for pairs of object categories

Generative Model Details

Spatial arrangements: $P(a|t,c)=R(a,t,c)S(a,t,c)$

- Random variables for relative positions and orientations
- Pairwise distributions of spatial relationships
- Feature distributions for positions on support surfaces



Distributions of geometric features of support surfaces

Scene Synthesis

Exemplar scenes

+

...

Database of Scenes

Probabilistic Model of Shape

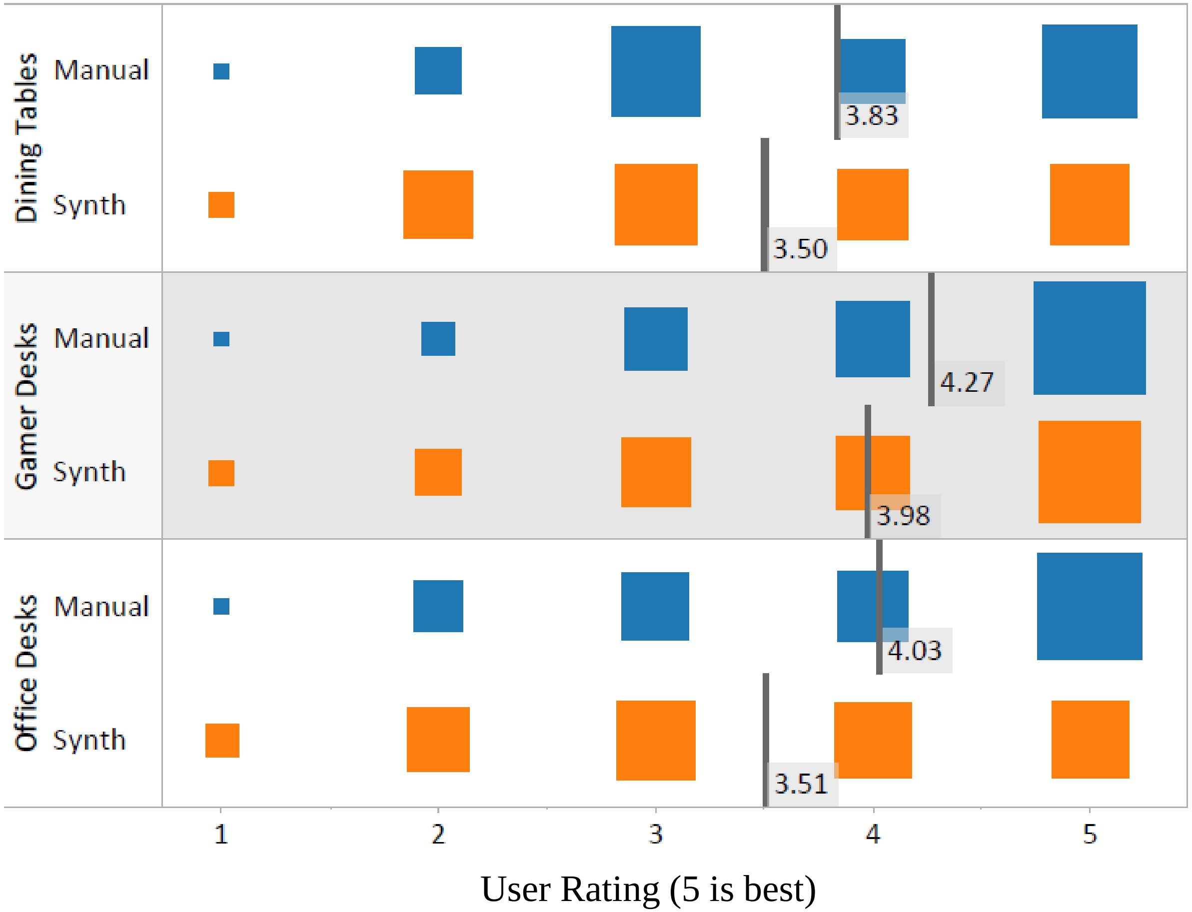
Synthesized novel scenes

Scene Synthesis Results

Synthesized novel scenes

Scene Synthesis Results

User study suggests that people find our synthesized scenes almost as good as manually created ones



Outline of Talk

Introduction

Learning probabilistic models from 3D collections

- Part-based templates
- Generative model

➢ Conclusions

Conclusions

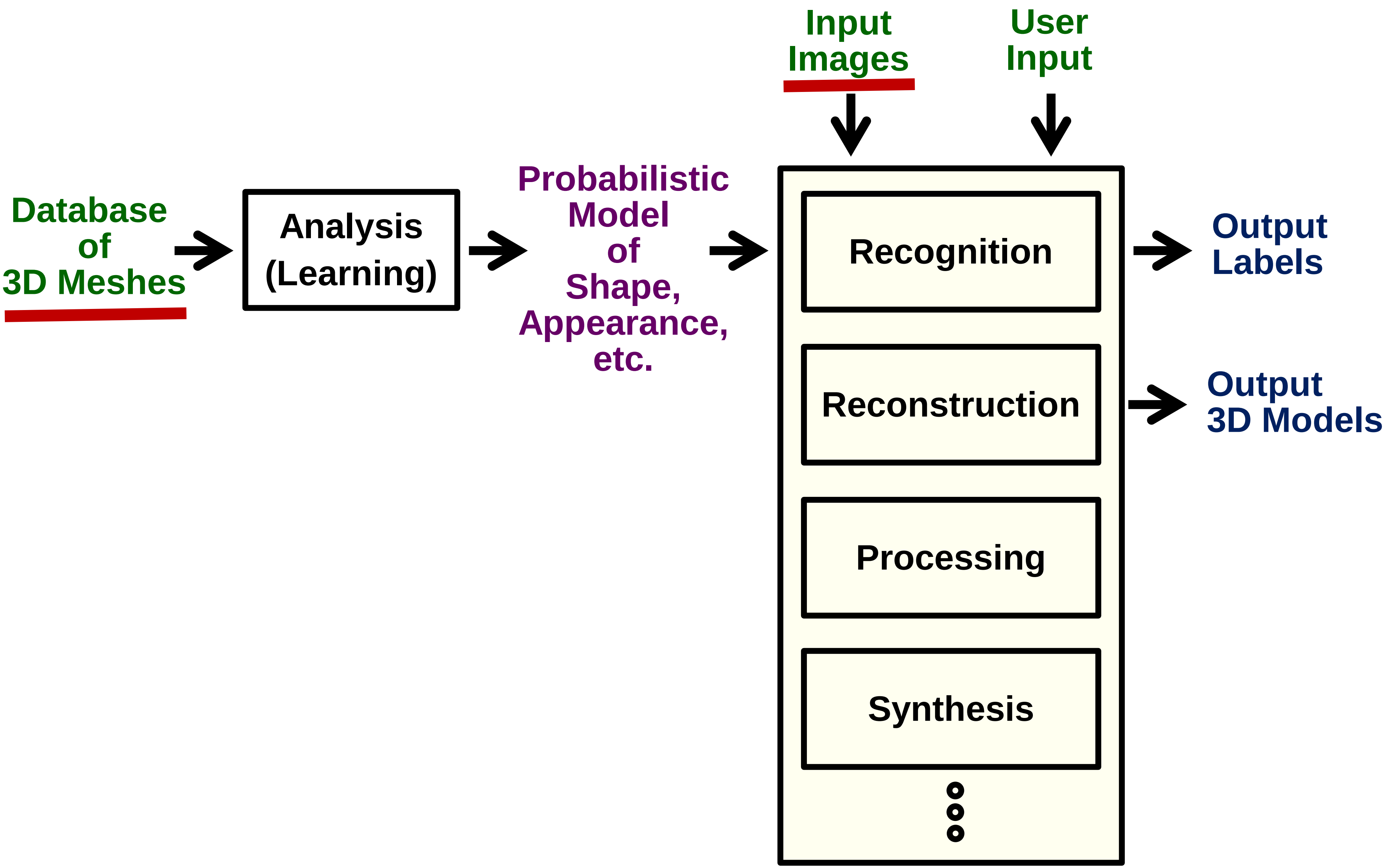
Main result:

- Probablistic models can be learned from collections of 3D meshes

Future work:

- It will be interesting to see if these models can be used effectively to understand scans and images

3D Shape Analysis for Computer Vision?



Acknowledgments

People:

- Sid Chaudhuri, Steve Diverdi, Matthew Fisher, Pat Hanrahan, Vladimir Kim, Wilmot Li, Niloy Mitra, Daniel Ritchie, Manolis Savva

Funding:

- NSF, NSERC, Intel, Adobe, Google

Data sets:

- Trimble 3D Warehouse

Thank You!