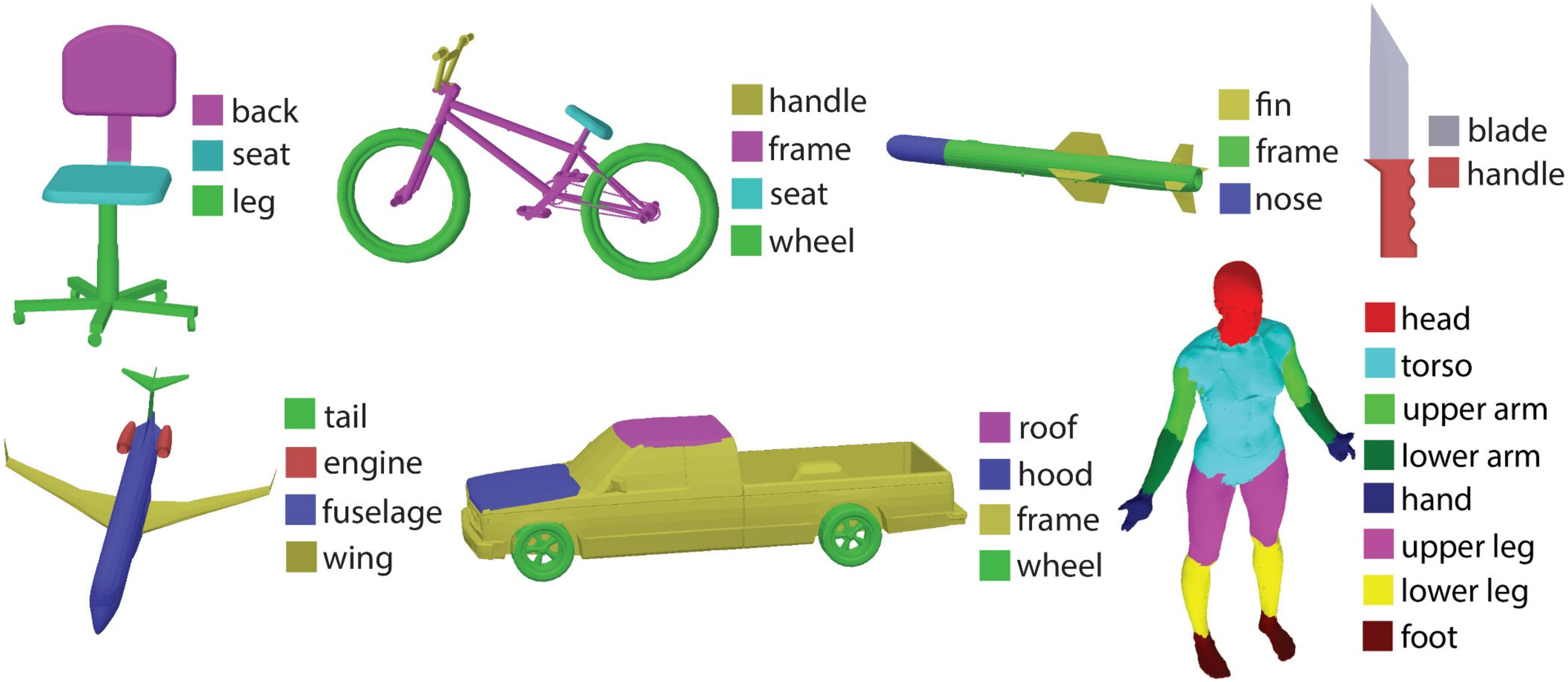
Deep learning architectures for 3D shape analysis and synthesis



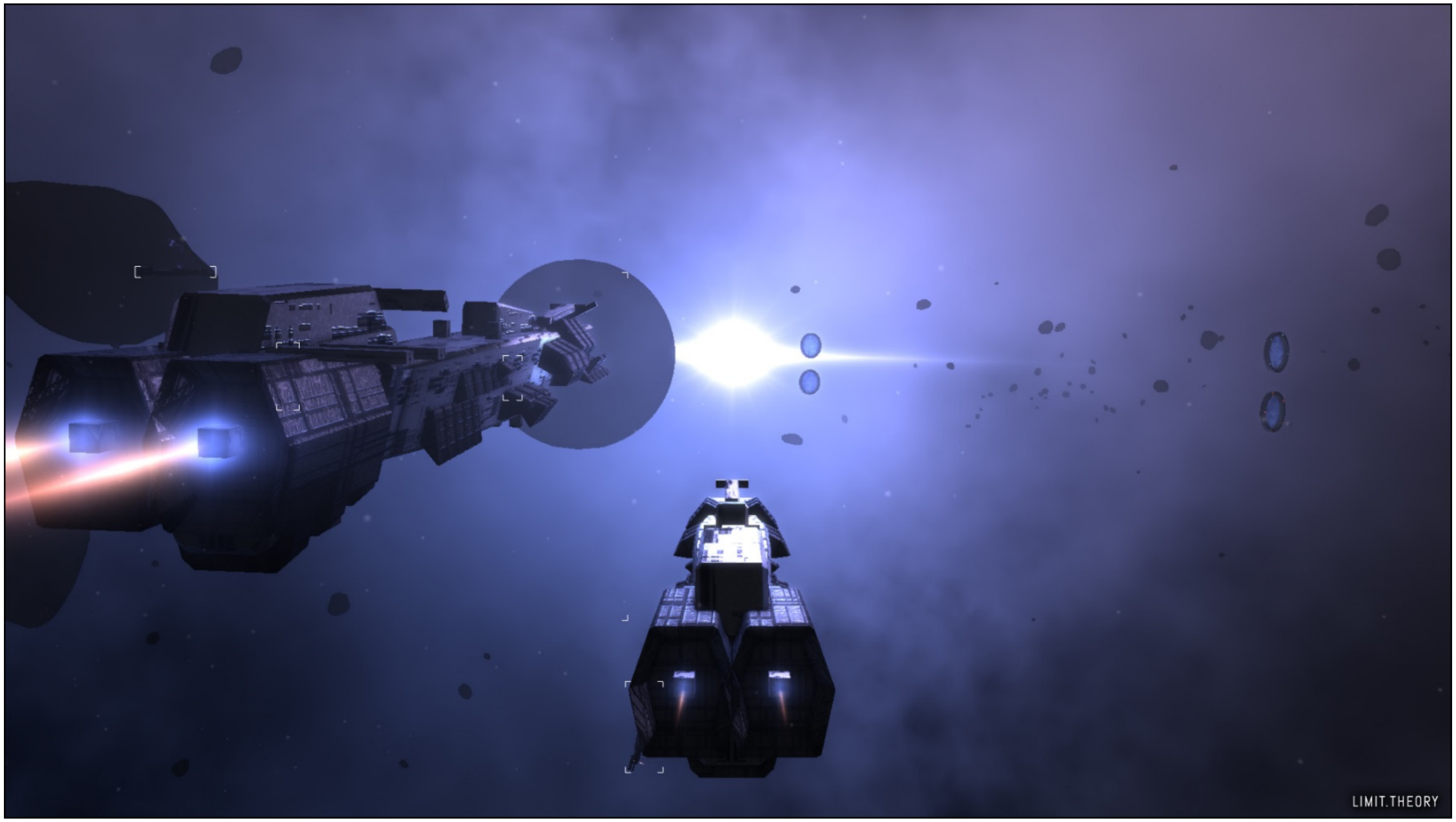
Evangelos Kalogerakis

UMASS AMHERST

3D models for architecture

Architect: Thomas Eriksson
Courtesy Industriromantik

3D models for digital entertainment



Limit Theory

3D models for cultural heritage

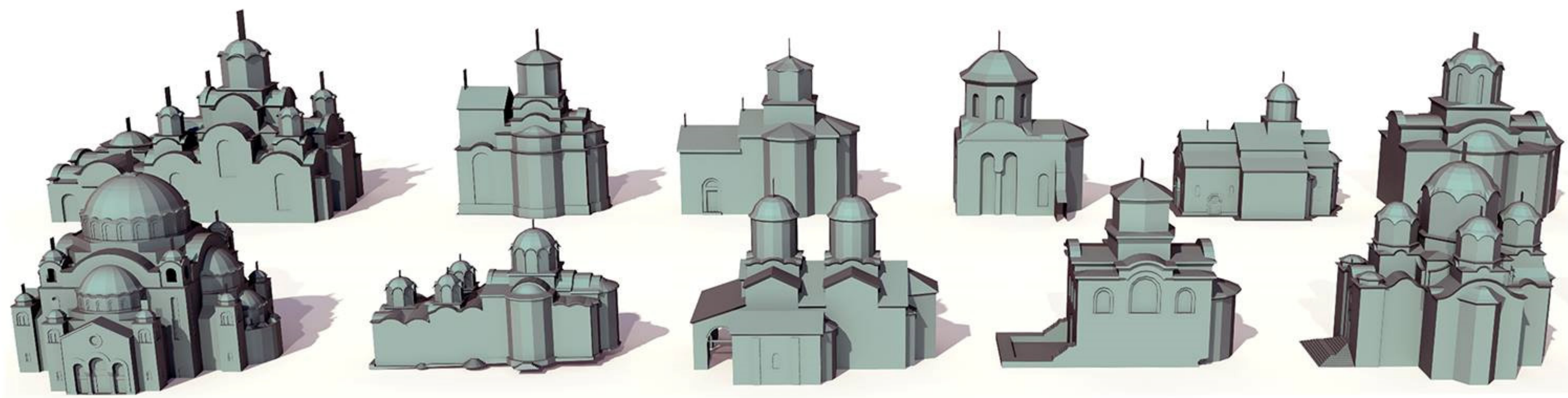
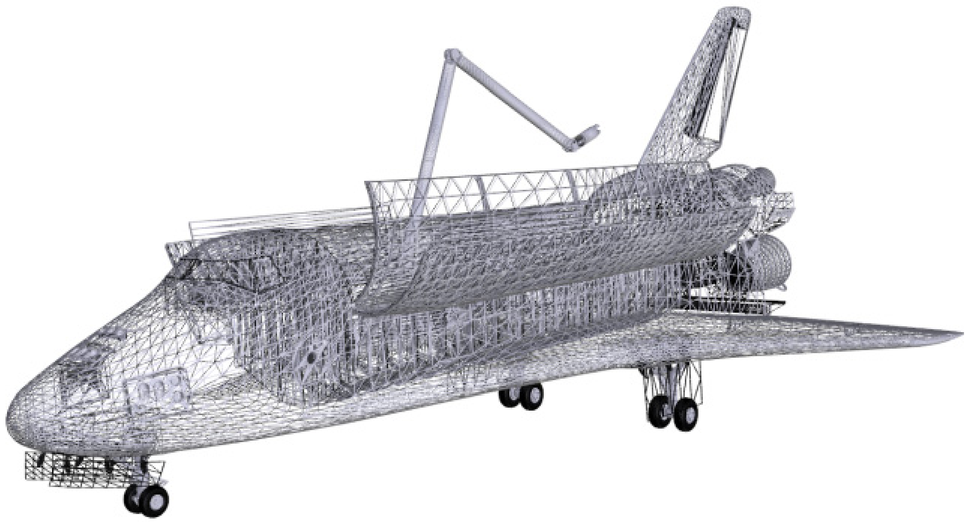
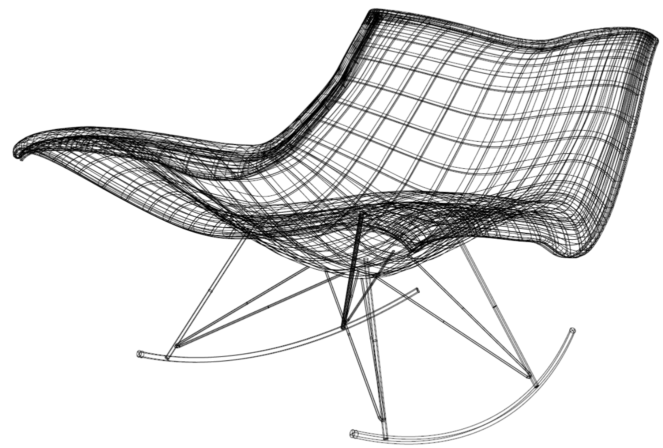


Image from [Lun, Kalogerakis, Sheffer, SIGGRAPH 2015]

Digital representations of 3D shapes



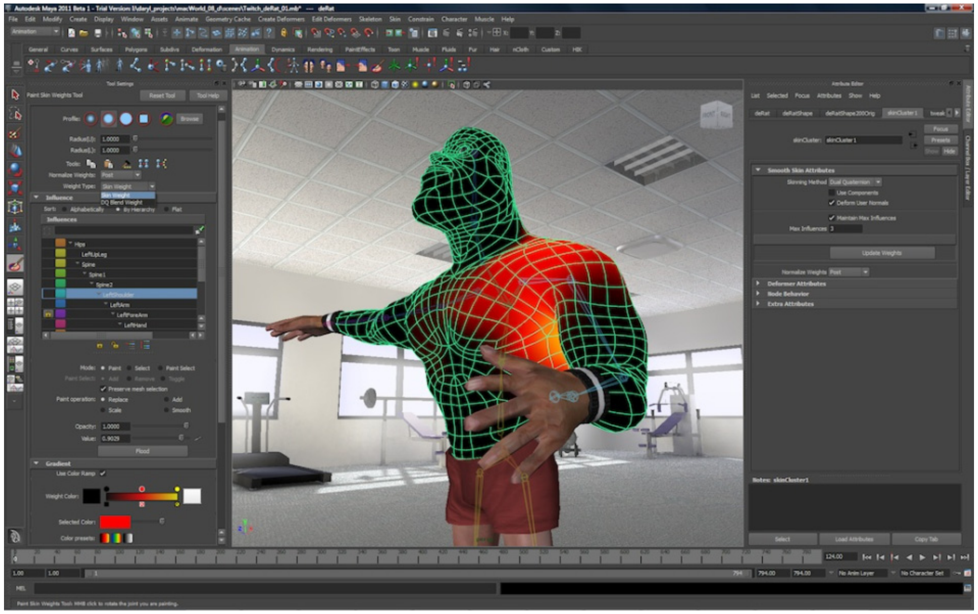
Polygon mesh



Analytic Surface

Models from 3D Warehouse &
FlyingArchitecture

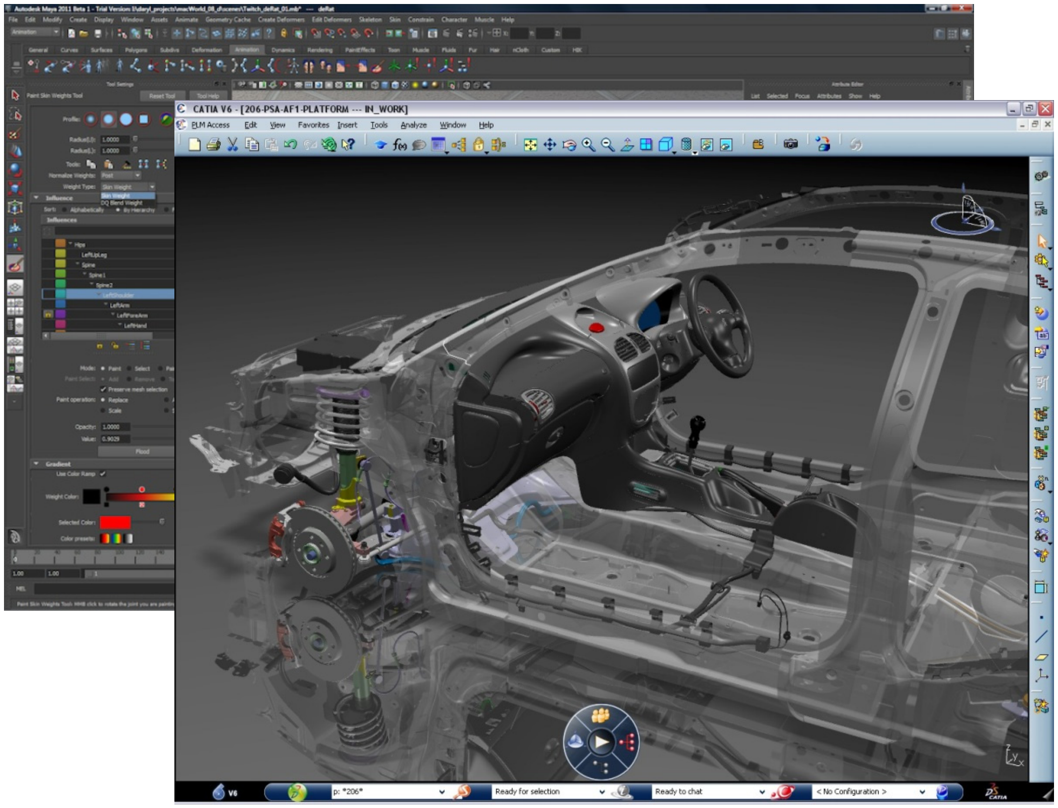
Digitizing our imagination



Professional 3D modeling tools

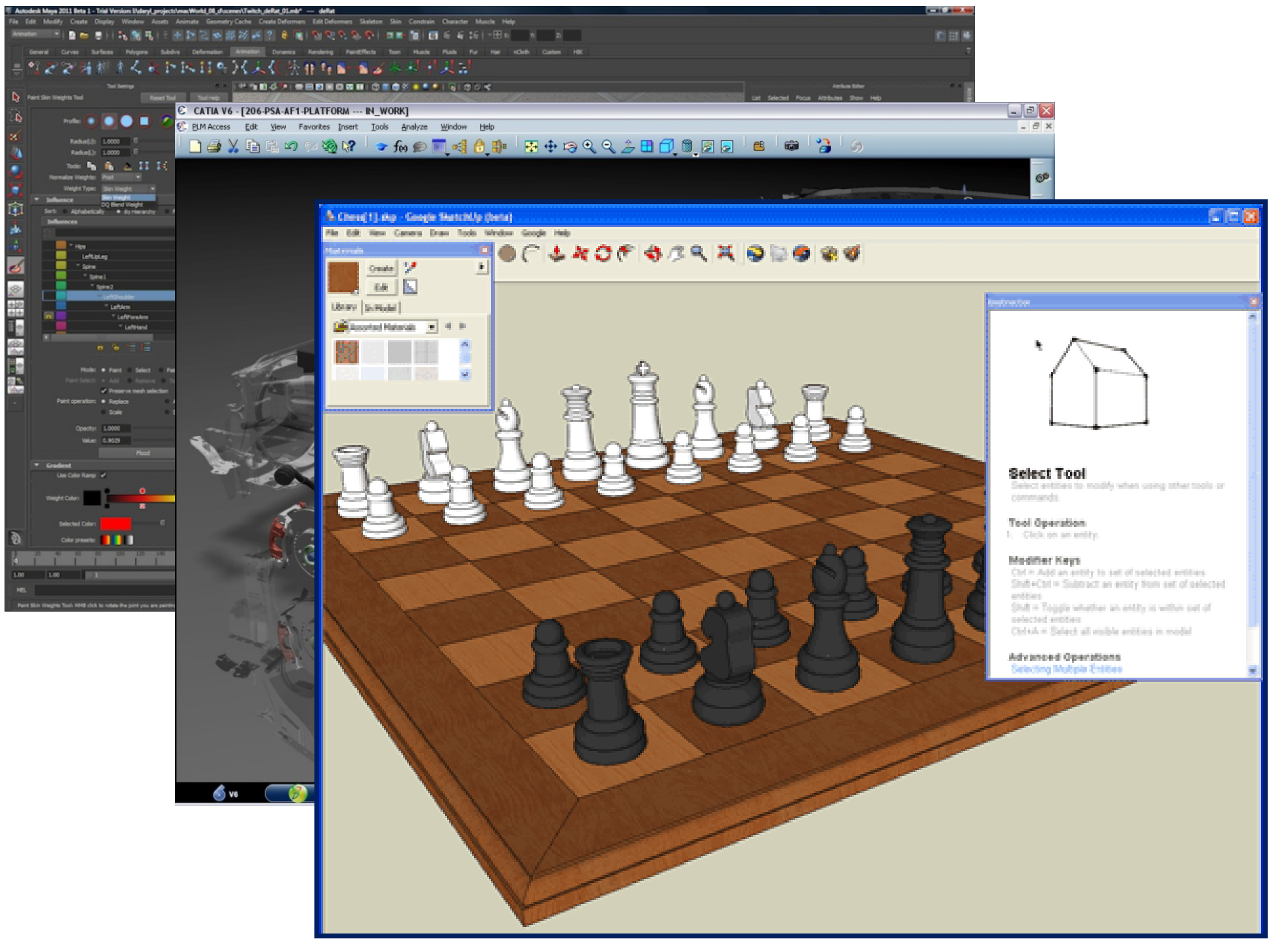
[Autodesk Maya]

Digitizing our imagination



Computer-Aided Design tools
[Catia]

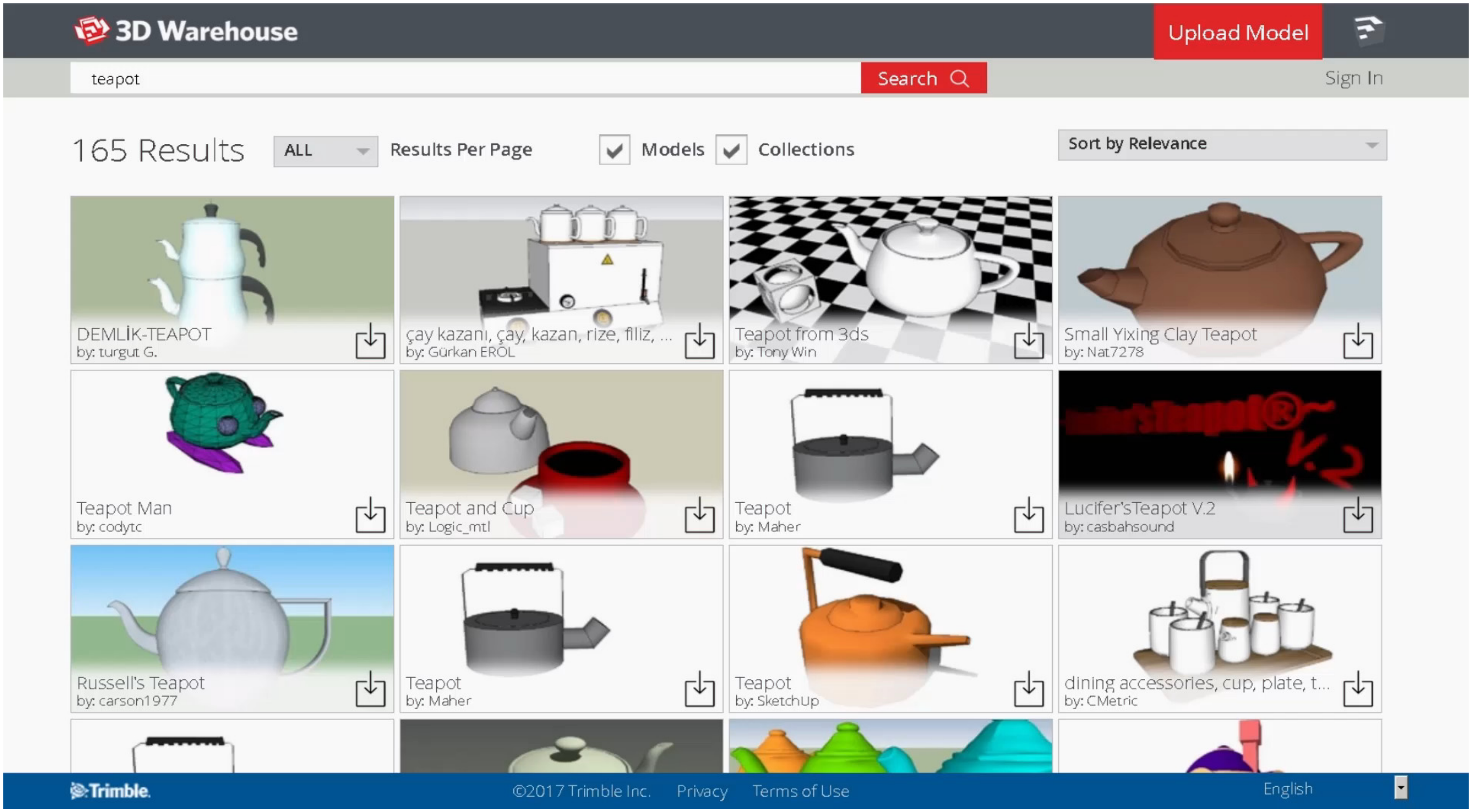
Digitizing our imagination



General-Purpose Modeling tools

[Trimble SketchUp]

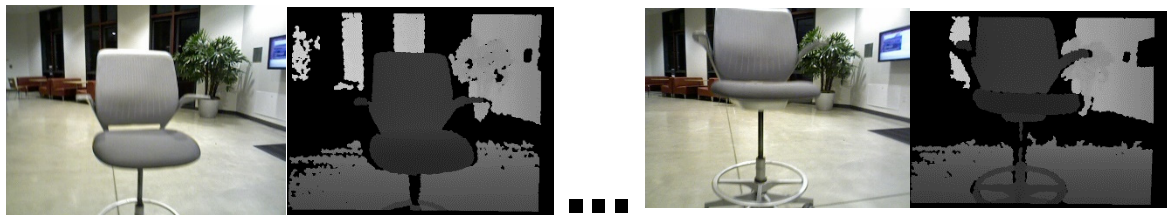
3D model repositories



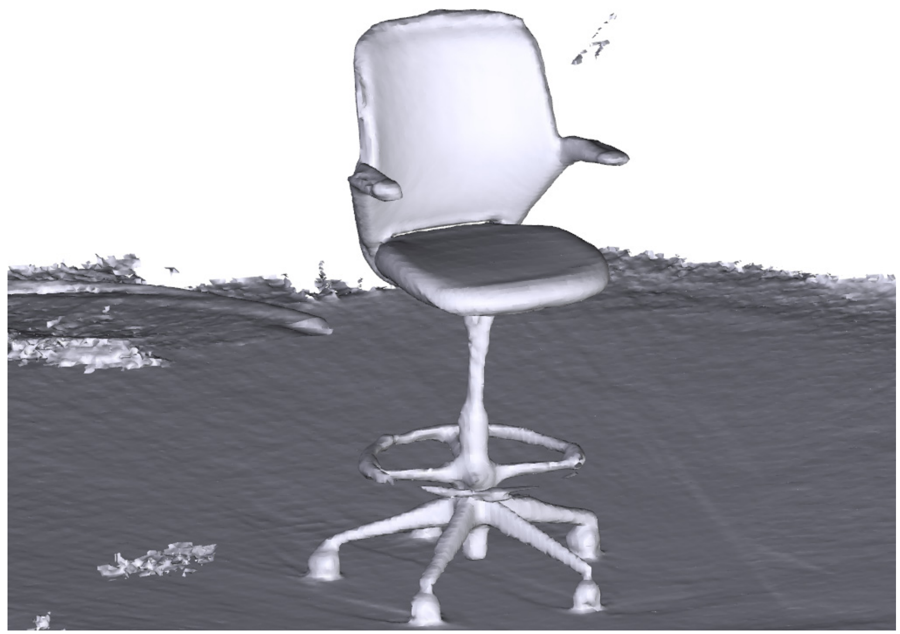
[Trimble 3D Warehouse]

3D geometry acquisition

RGB Image & depth data

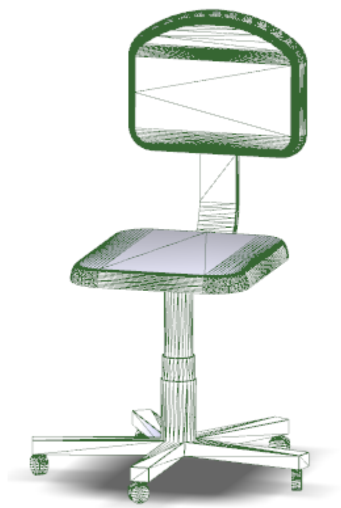


Resulting surface



"A Large Dataset of Object Scans"
Choi, Zhou, Miller, Koltun 2016

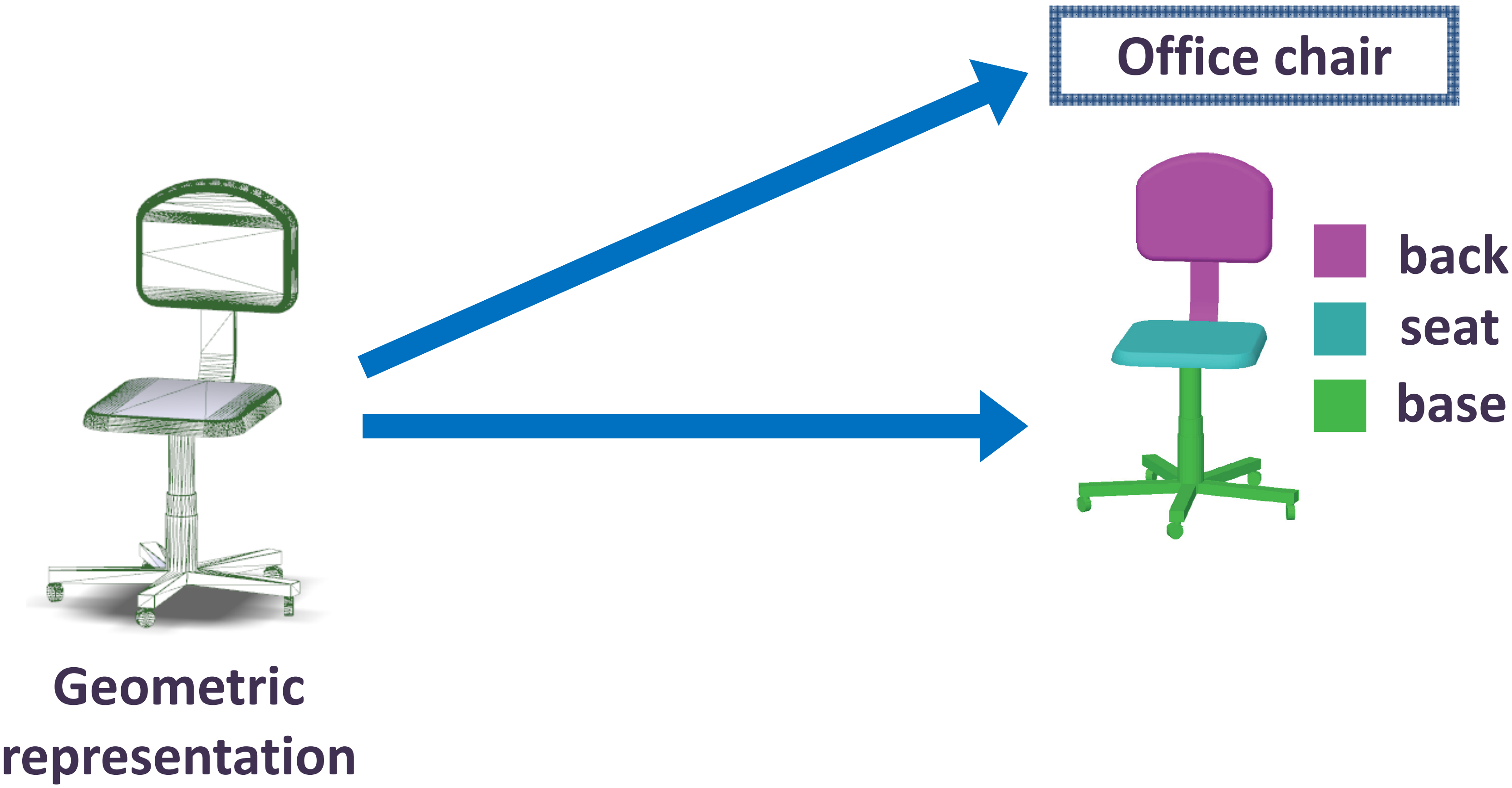
We need algorithms that “understand” shapes



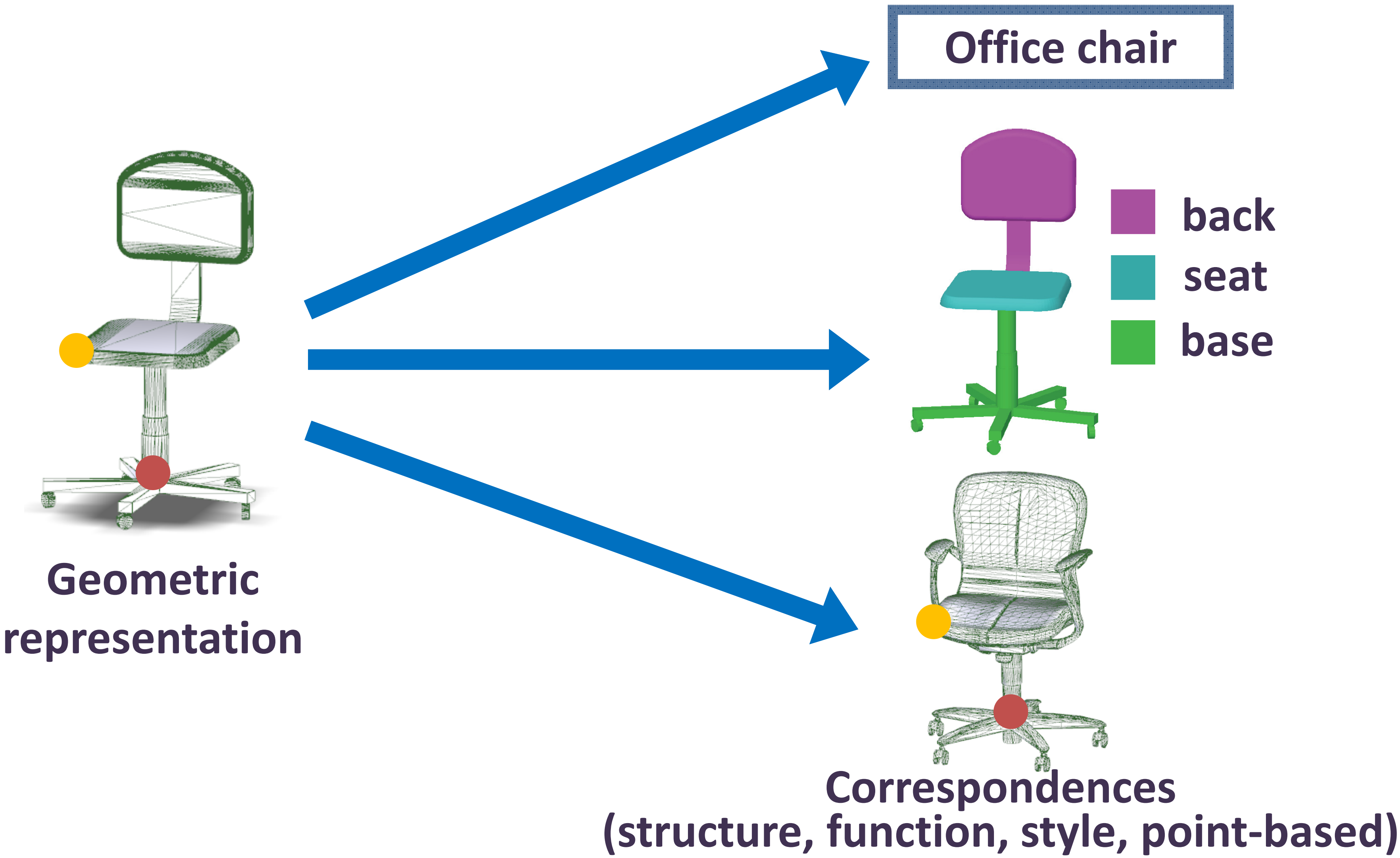
Geometric representation

Office chair

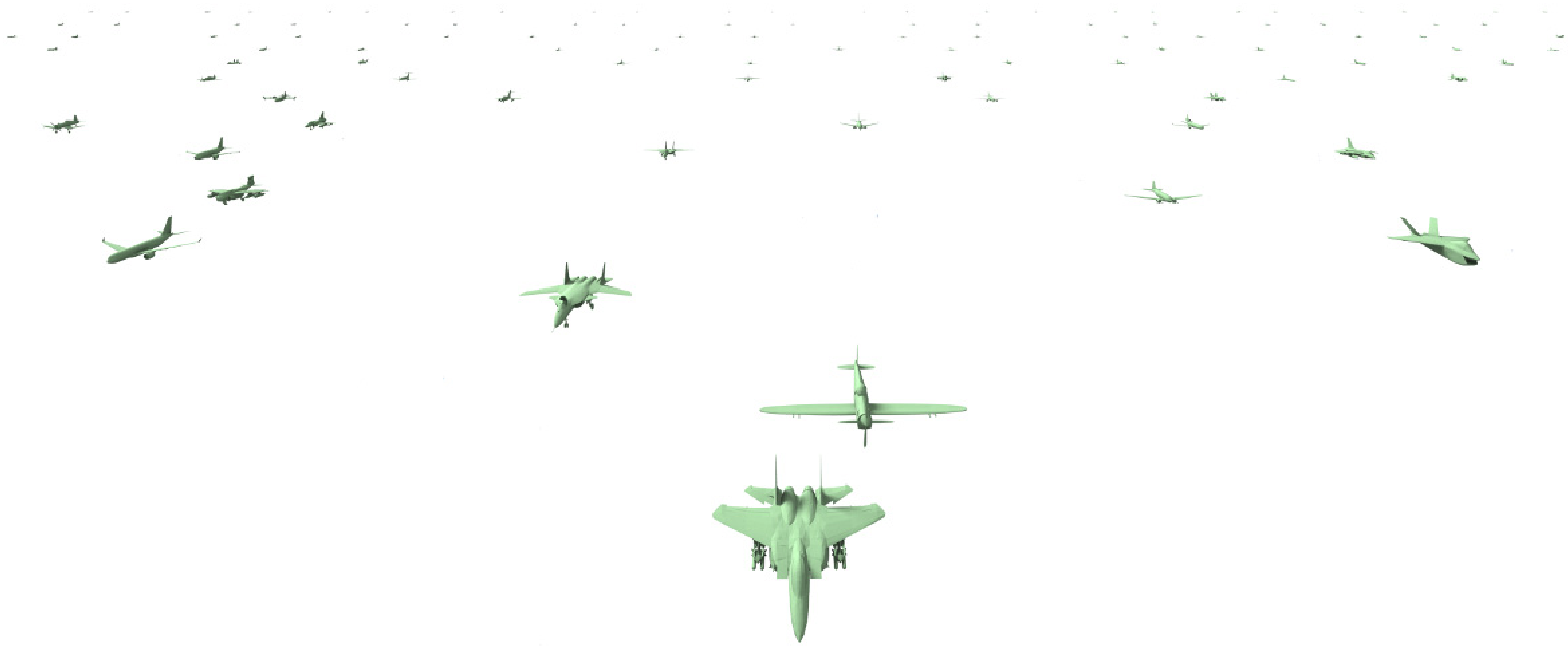
We need algorithms that "understand" shapes



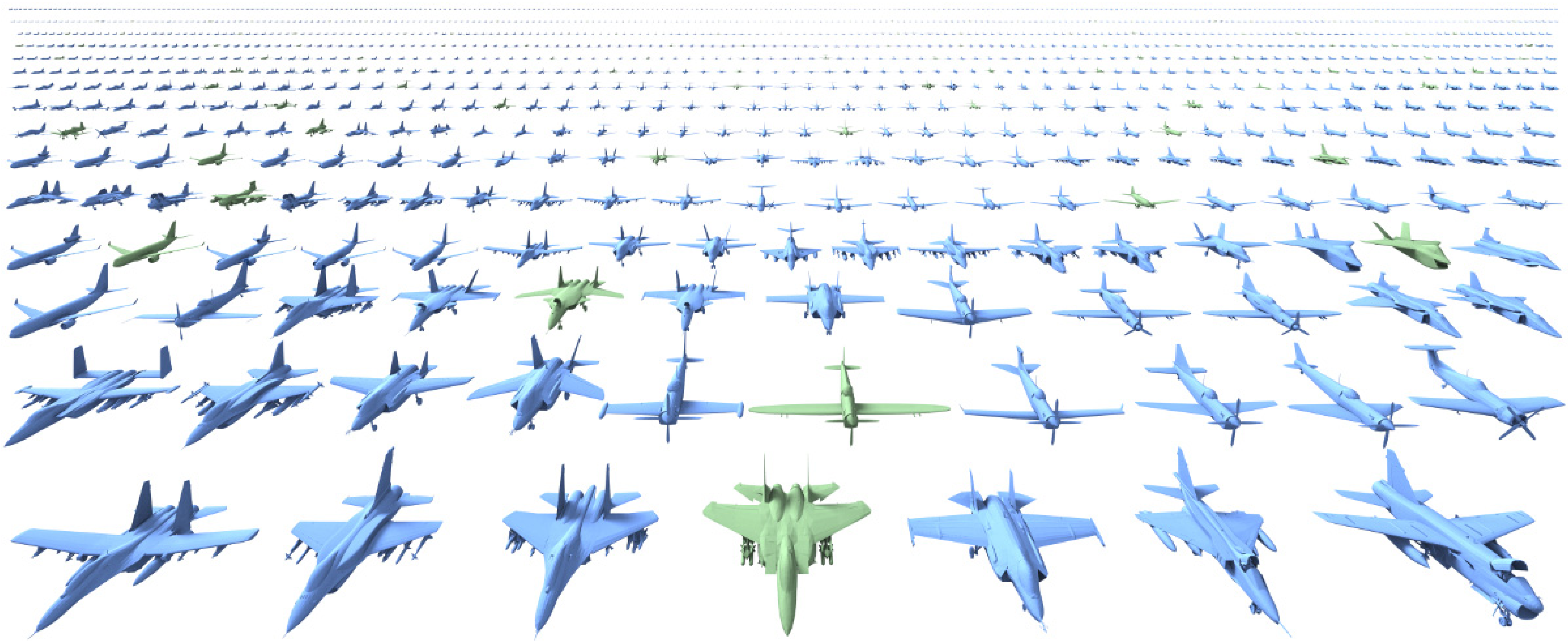
We need algorithms that “understand” shapes



Why shape understanding?
Generative models of shapes

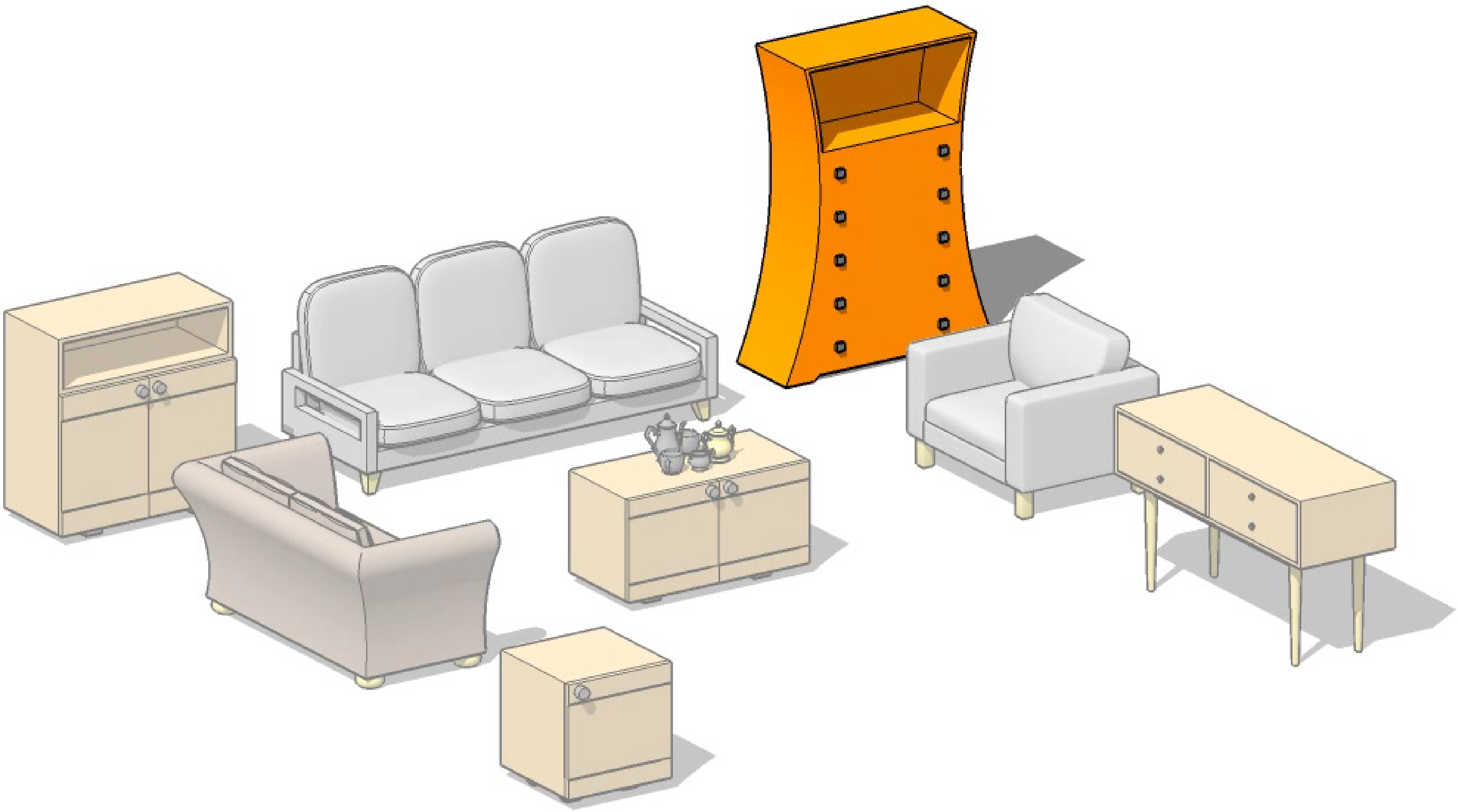


Why shape understanding?
Generative models of shapes



Kalogerakis, Chaudhuri, Koller, Koltun, SIGGRAPH 2012

Why shape understanding? Scene design

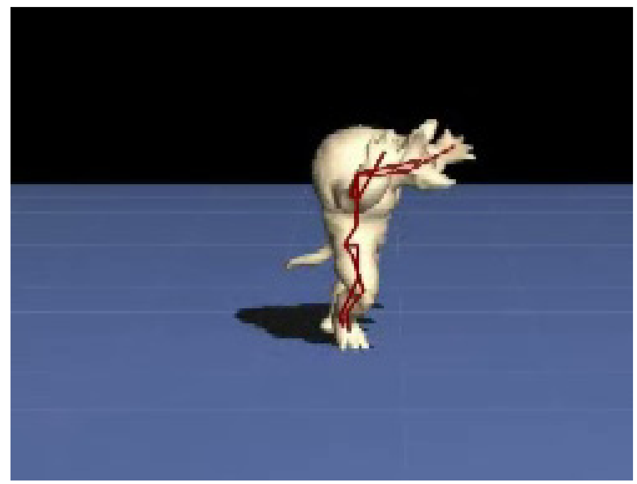


Lun, Kalogerakis, Wang, Sheffer, SIGGRAPH ASIA 2016

Why shape understanding?
Texturing

Kalogerakis, Hertzmann, Singh, SIGGRAPH 2010

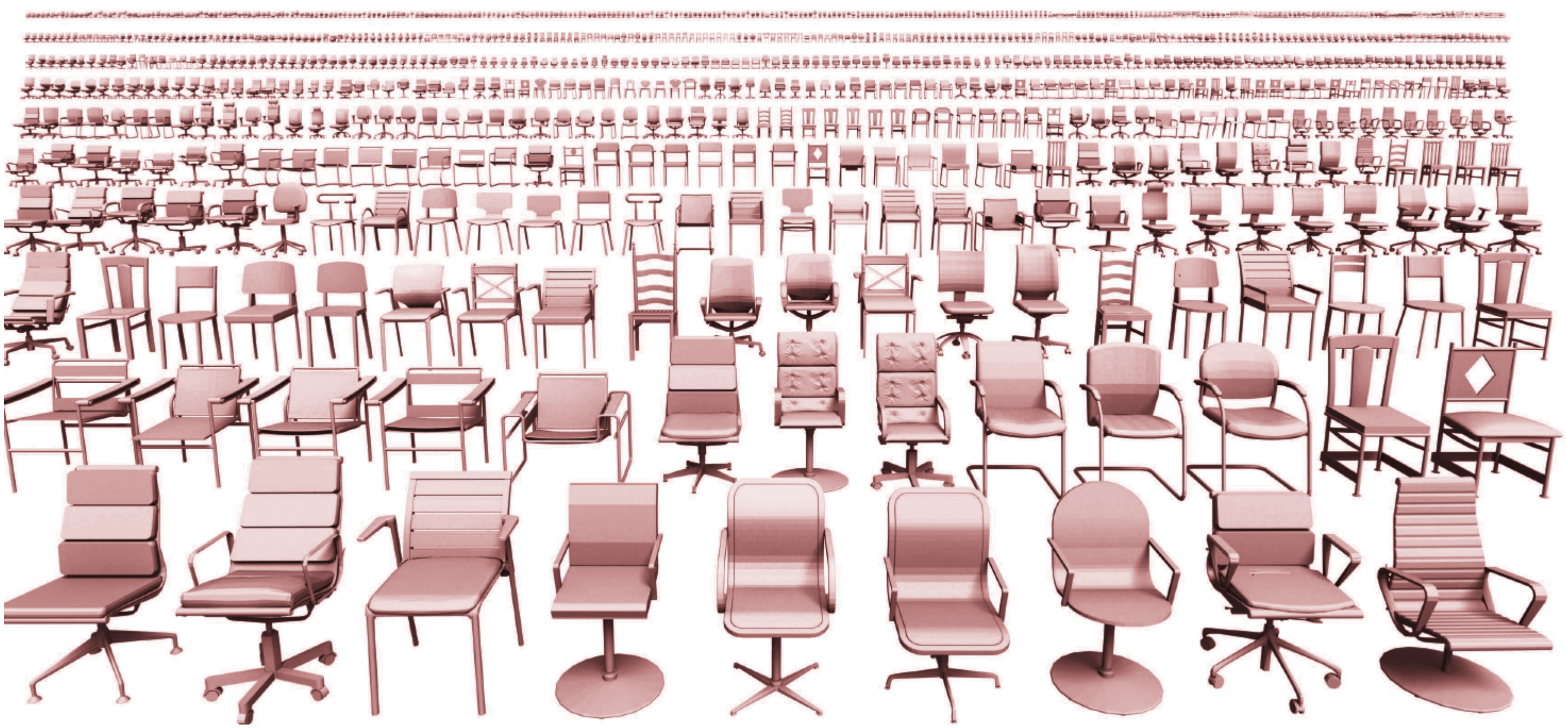
Why shape understanding?
Character Animation



Simari, Nowrouzezahrai, Kalogerakis, Singh, SGP 2009

How can we perform shape understanding?

It is very hard to perform shape understanding with **manually specified rules & hand-engineered descriptors.**

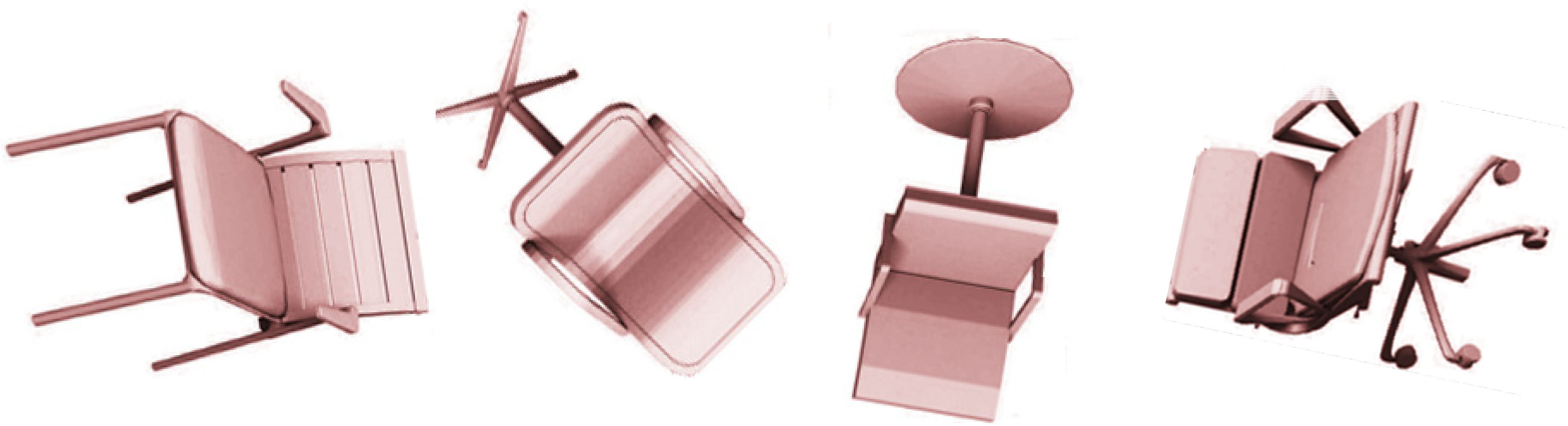


Models from Dosch Design

How can we perform shape understanding?

It is very hard to perform shape understanding with **manually specified rules & hand-engineered descriptors.**

One more complication: **arbitrary shape orientation**

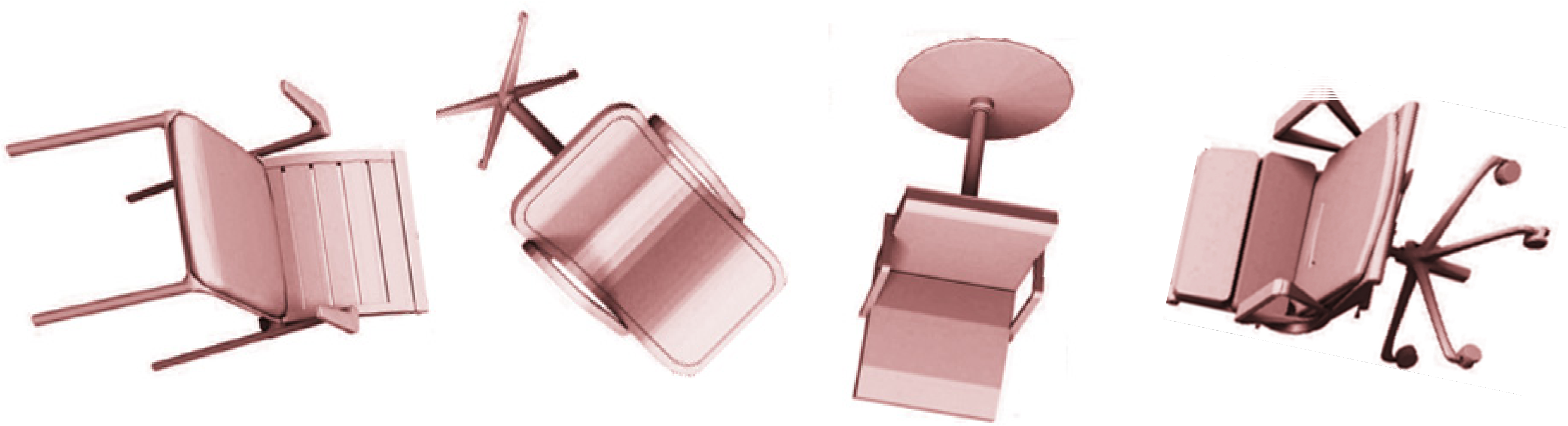


Models from Dosch Design

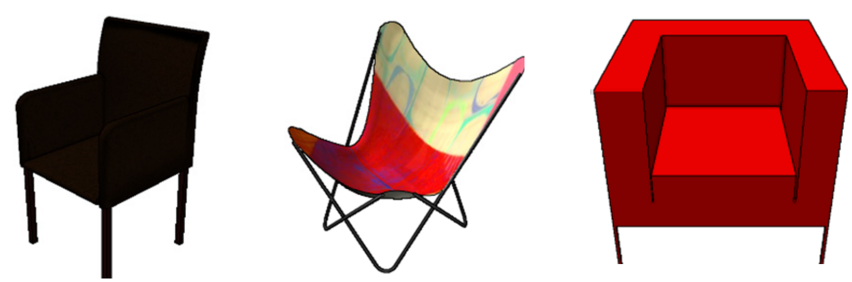
How can we perform shape understanding?

It is very hard to perform shape understanding with **manually specified rules & hand-engineered descriptors.**

One more complication: **arbitrary shape orientation**

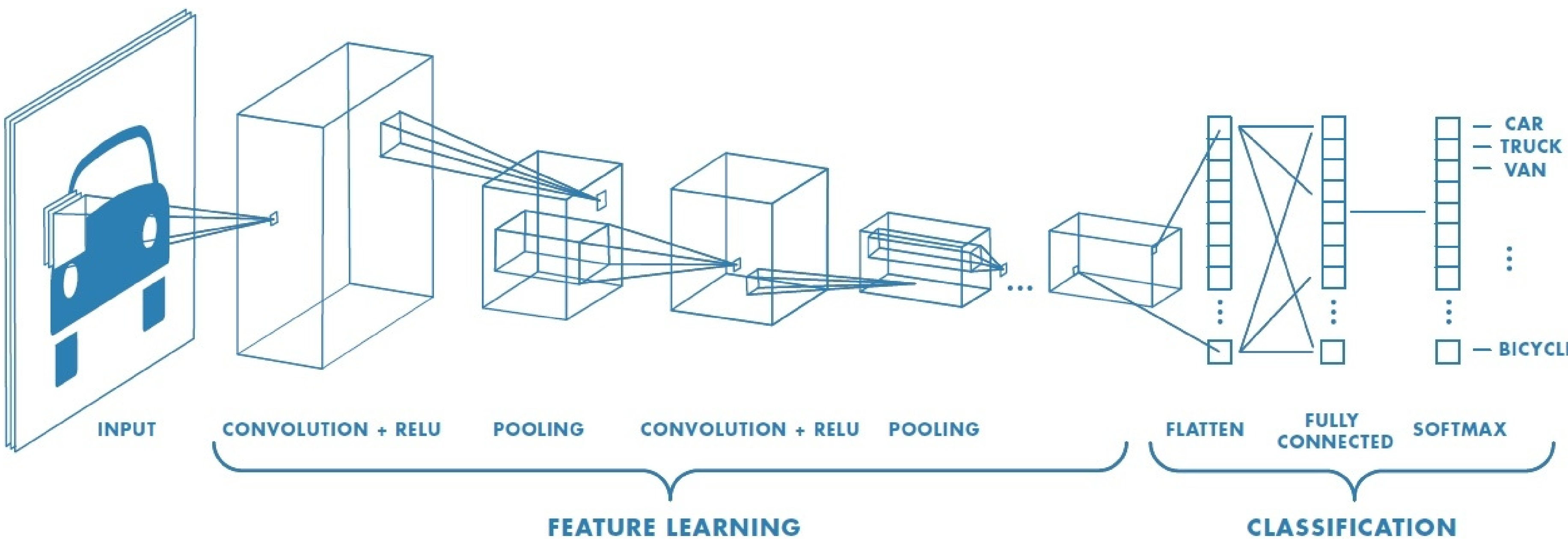


+ arbitrary or no texture



The image understanding "success story"

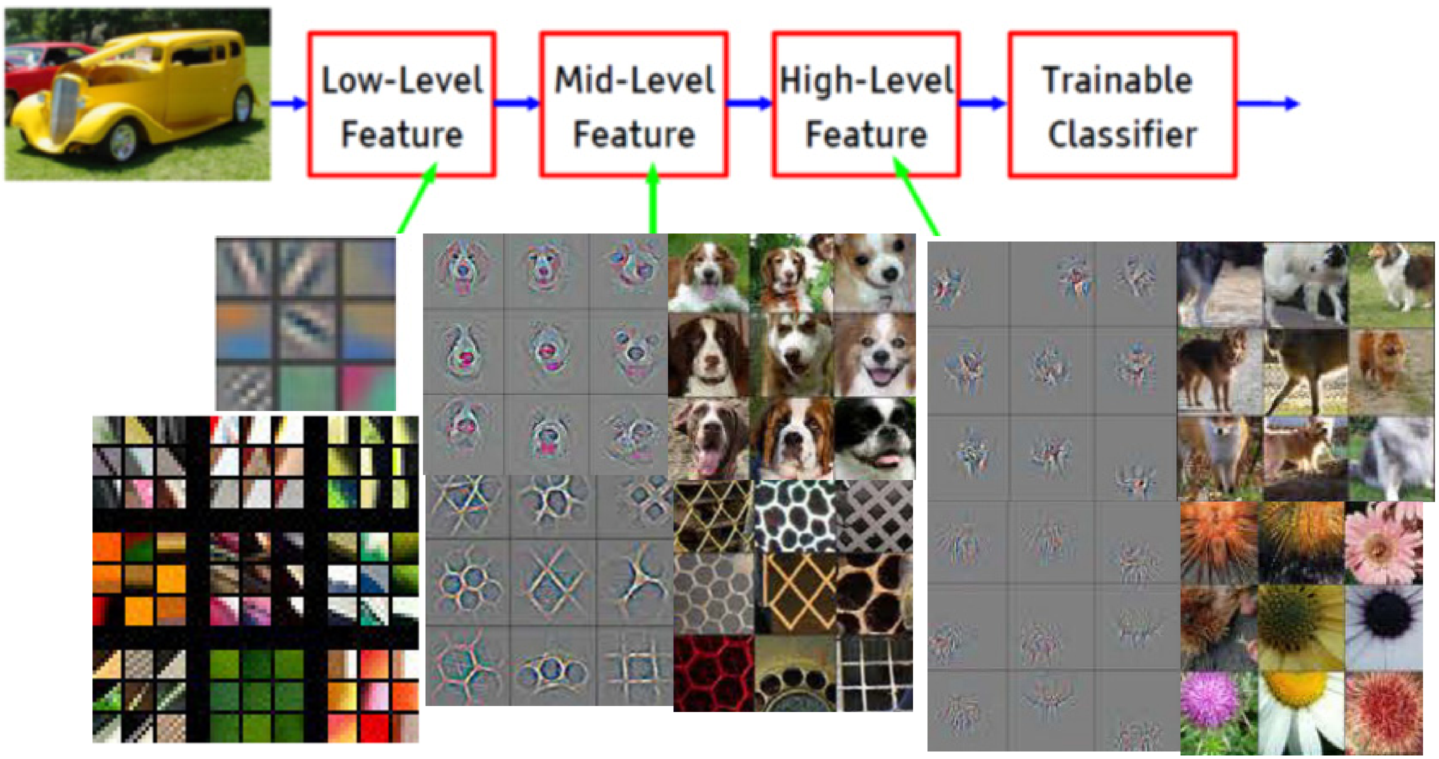
Layers of **convolutional filters** trained to extract descriptors + **learned functions** that map descriptors to high-level concepts.



Source: Mathworks

The image understanding "success story"

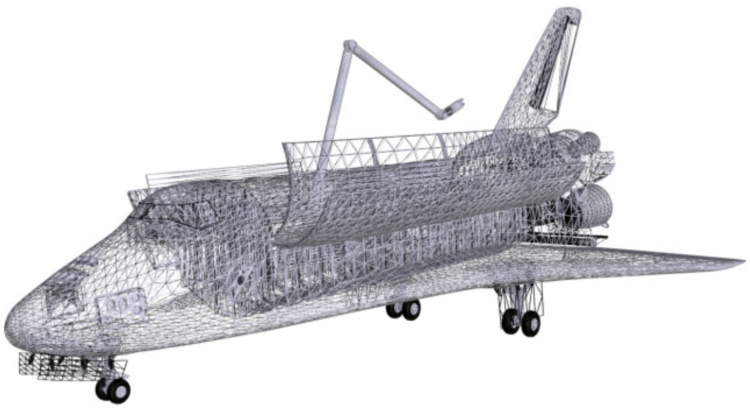
Convnet filters capture **various hierarchical patterns.**
Very high accuracy in image-processing tasks.



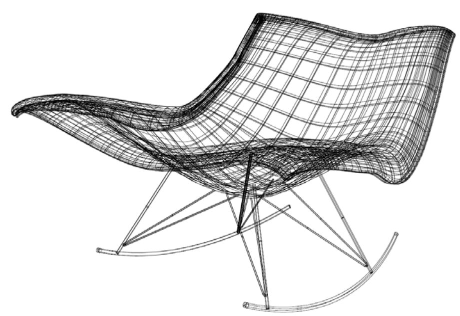
Matthew D. Zeiler and Rob Fergus,
Visualizing and Understanding
Convolutional Networks, 2014

How do we apply convnets for 3D shapes?

Geometric representations are **unordered**: arbitrary point order, different #points, different #neighbors per point...



Polygon mesh



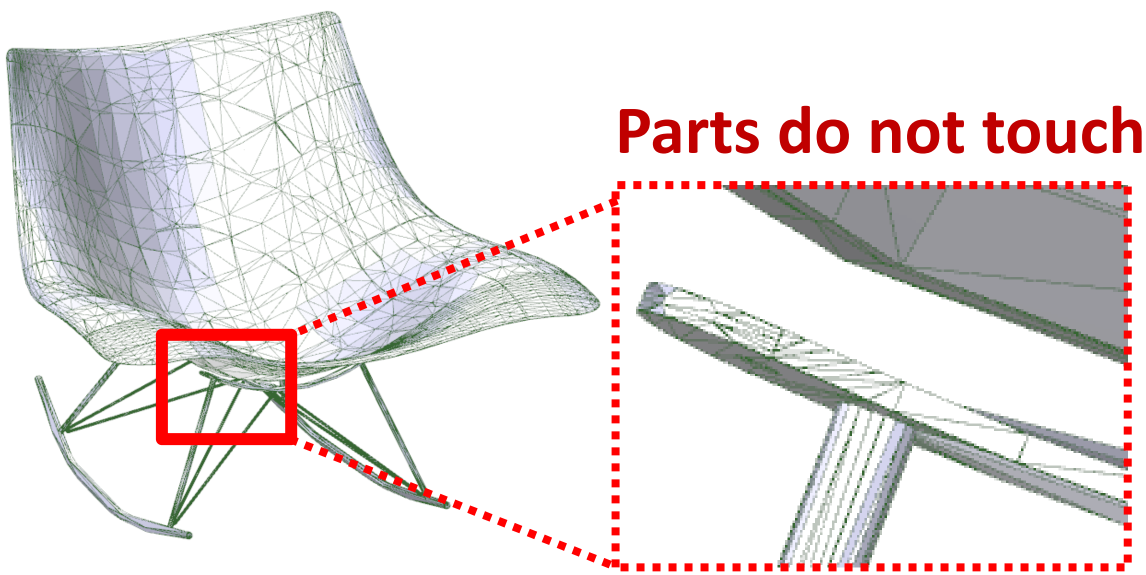
Analytic Surface

Point clouds

Models from 3D Warehouse & FlyingArchitecture

Key observations

Geometric representations have **artifacts**.



(not easily noticeable to the viewer,
yet geometric implications on topology, connectedness...)

Key observations

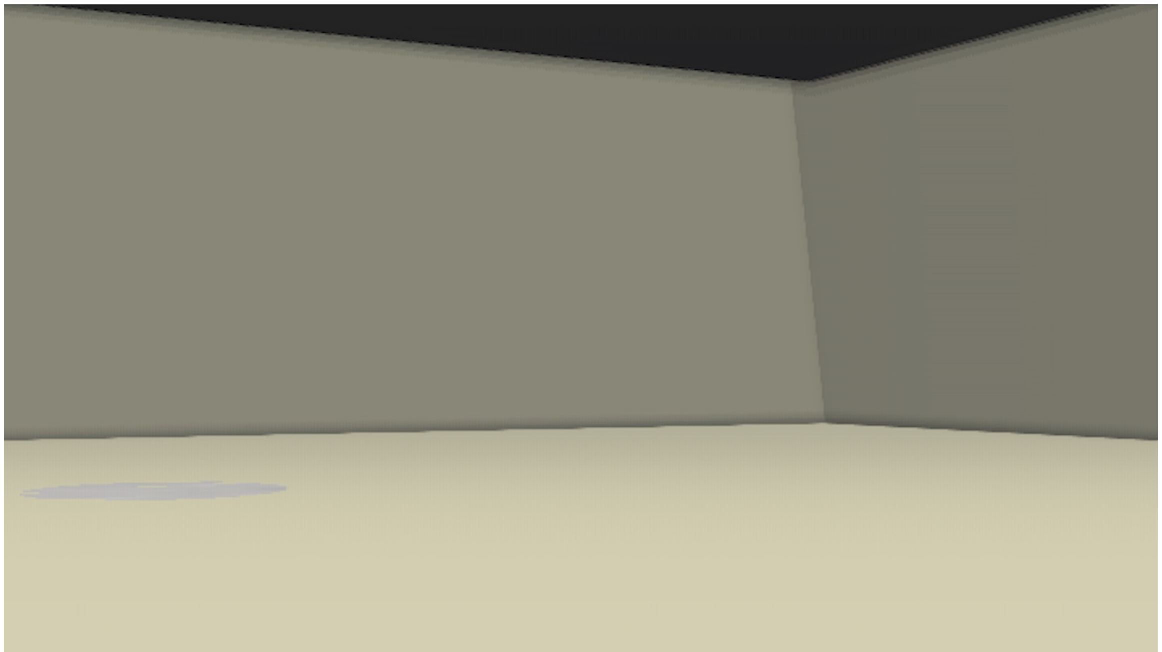
3D shapes are often **designed for viewing…**

Key observations

3D shapes are often **designed for viewing…**

Key observations

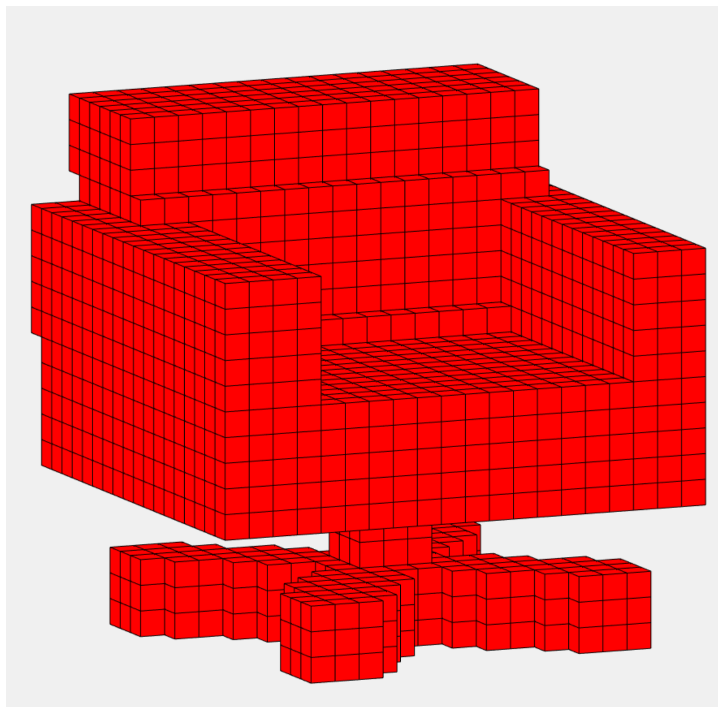
3D shapes are often **designed for viewing…**



Empty inside!

Key observations

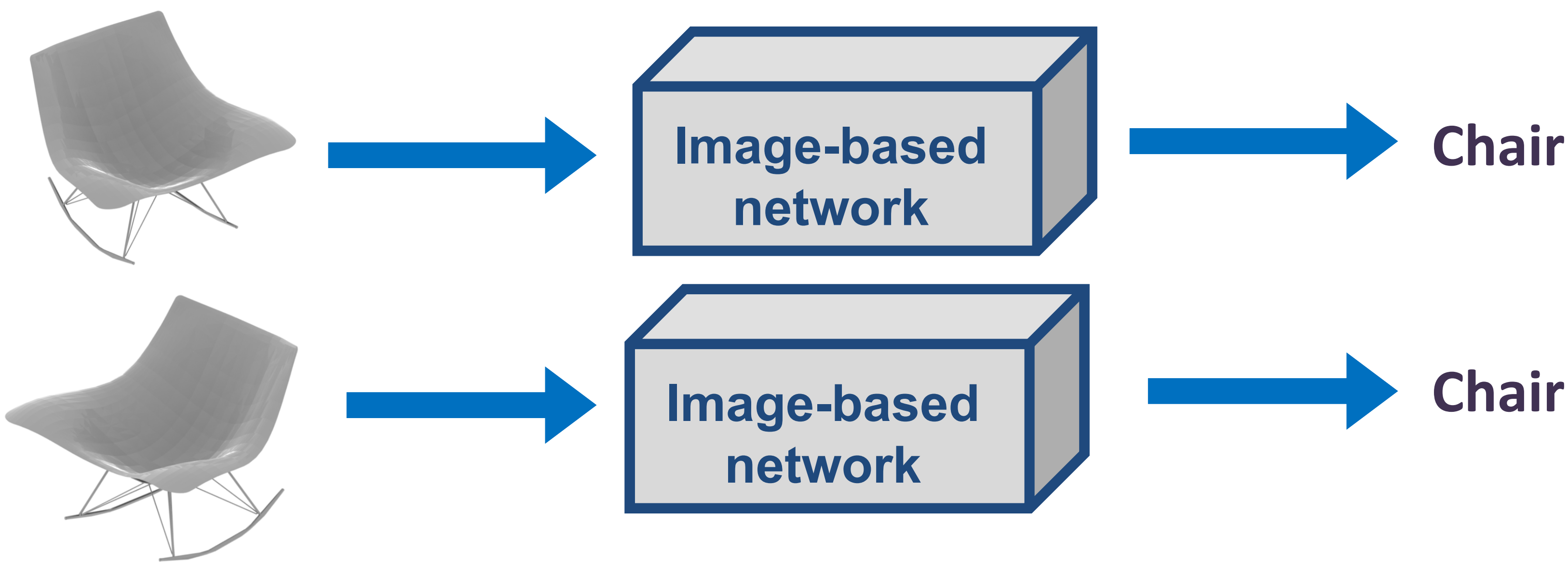
A natural extension of image-based convnets are **volumetric convnets** operating on **voxel shape representations**.



Voxel representation wasteful?

Key observations

Image-based nets can process individual shape renderings.

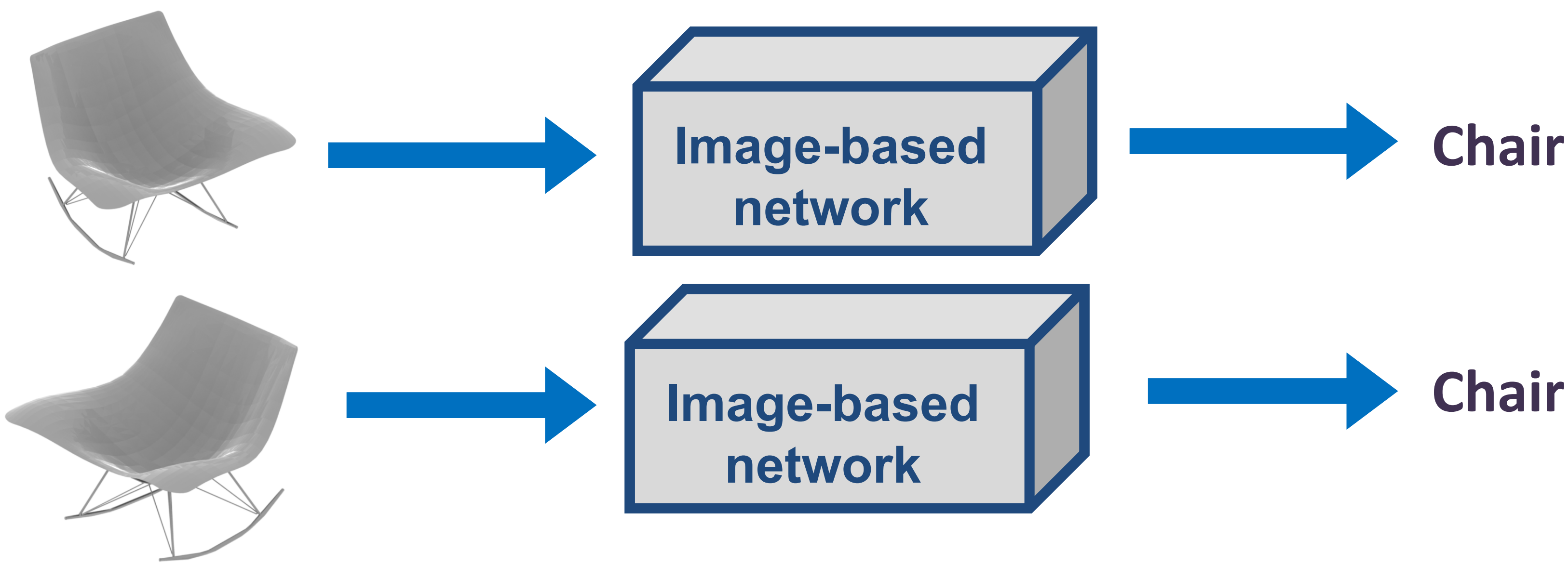


Su, Maji, Kalogerakis, Learned-Miller, ICCV 2015

Key observations

Image-based nets can process individual shape renderings.

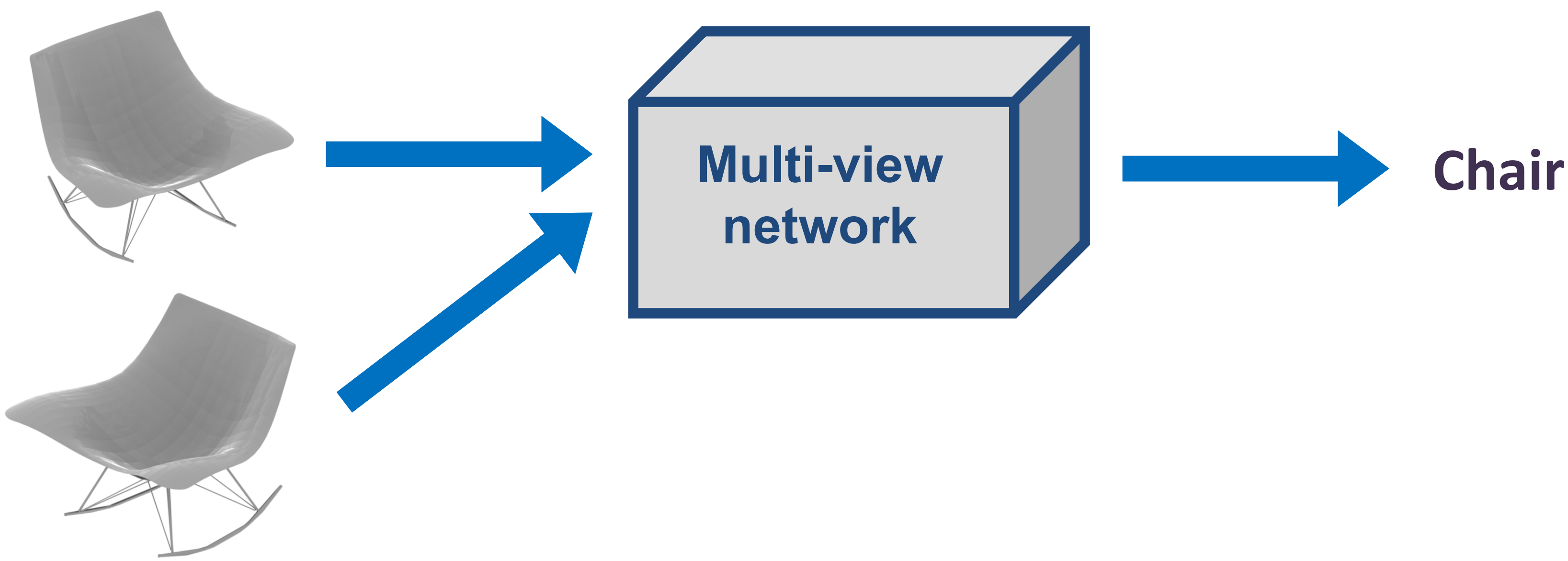
⇒ **83% shape classification accuracy in ModelNet40 (VGG net trained on ImageNet)**



Su, Maji, Kalogerakis, Learned-Miller, ICCV 2015

Can we do better? The multi-view approach

Deep architecture that combine **convolution layers for reasoning across multiple rendered shape views**

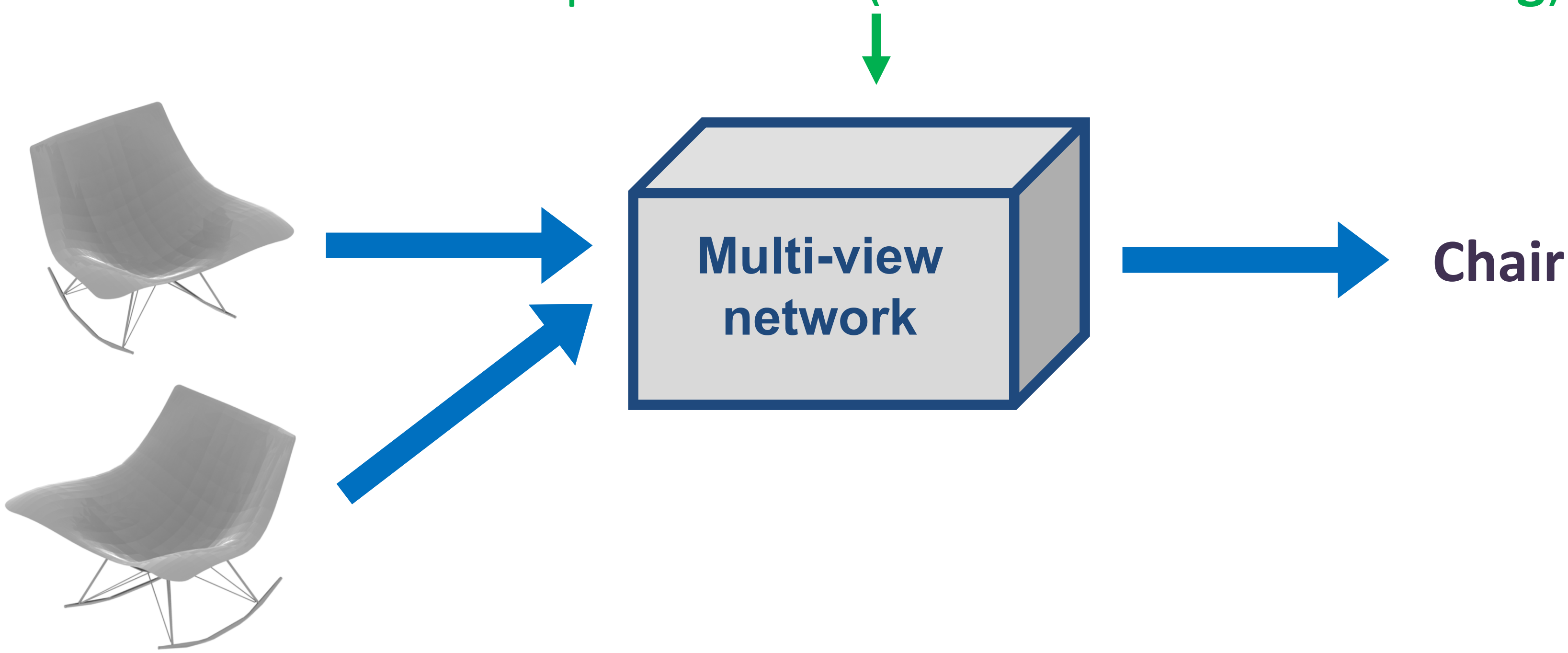


Su, Maji, Kalogerakis, Learned-Miller, ICCV 2015

Can we do better? The multi-view approach

Deep architecture that combine **convolution layers for reasoning across multiple rendered shape views**

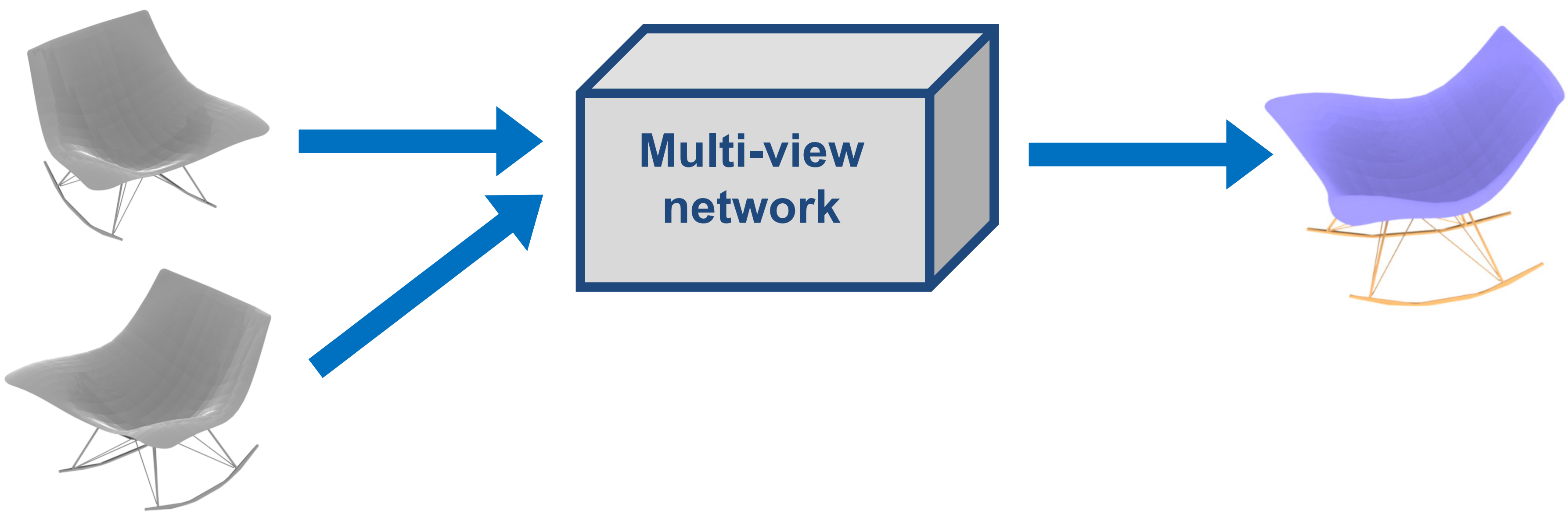
Fine-tune on shape datasets (a form of **transfer learning**)



Su, Maji, Kalogerakis, Learned-Miller, ICCV 2015

Can we do better? The multi-view approach

Deep architecture that combine **convolution layers for reasoning across multiple rendered shape views** + **surface-based probabilistic models for producing a coherent signal on the surface.**



Key challenges of multi-view convnets

Deep architecture that combine **convolution layers for reasoning across multiple rendered shape views** + **surface-based probabilistic models for producing a coherent signal on the surface.**

Key challenges:

- **Joint reasoning about parts across multiple views + surface**

Key challenges of multi-view convnets

Deep architecture that combine **convolution layers for reasoning across multiple rendered shape views** + **surface-based probabilistic models for producing a coherent signal on the surface.**

Key challenges:

- **Joint reasoning about parts across multiple views + surface**
- **Deal with self-occlusions / surface information loss**

Key challenges of multi-view convnets

Deep architecture that combine **convolution layers for reasoning across multiple rendered shape views** + **surface-based probabilistic models for producing a coherent signal on the surface.**

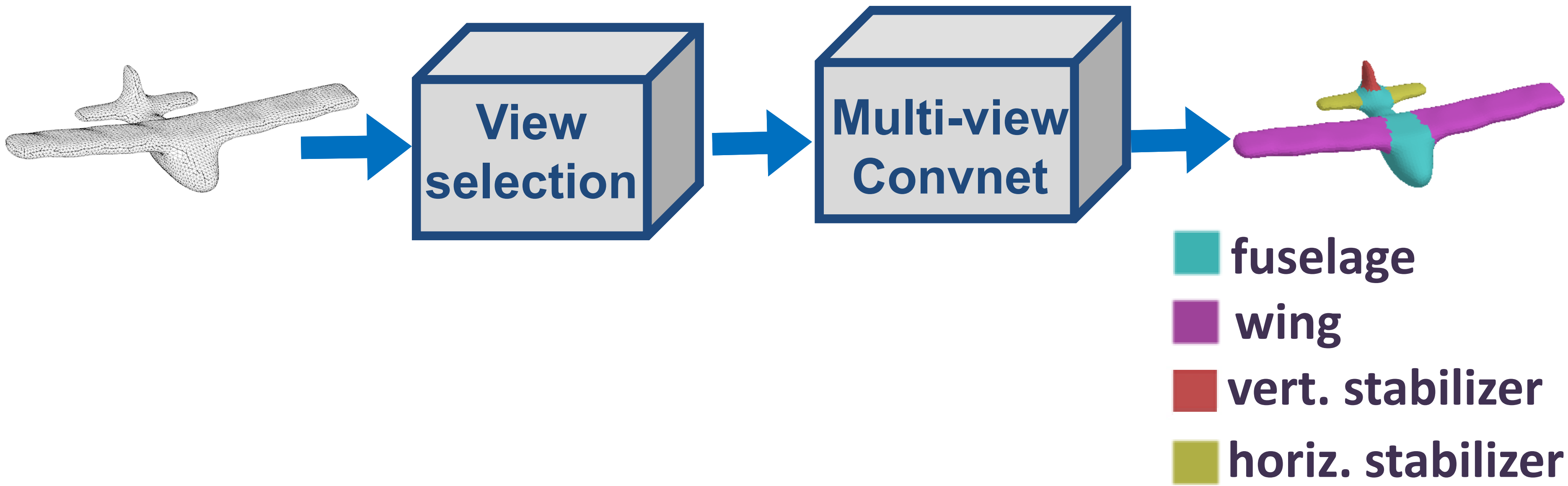
Key challenges:

- **Joint reasoning about parts across multiple views + surface**
- **Deal with self-occlusions / surface information loss**
- **Promote invariance over 3D shape rotations**

Outline

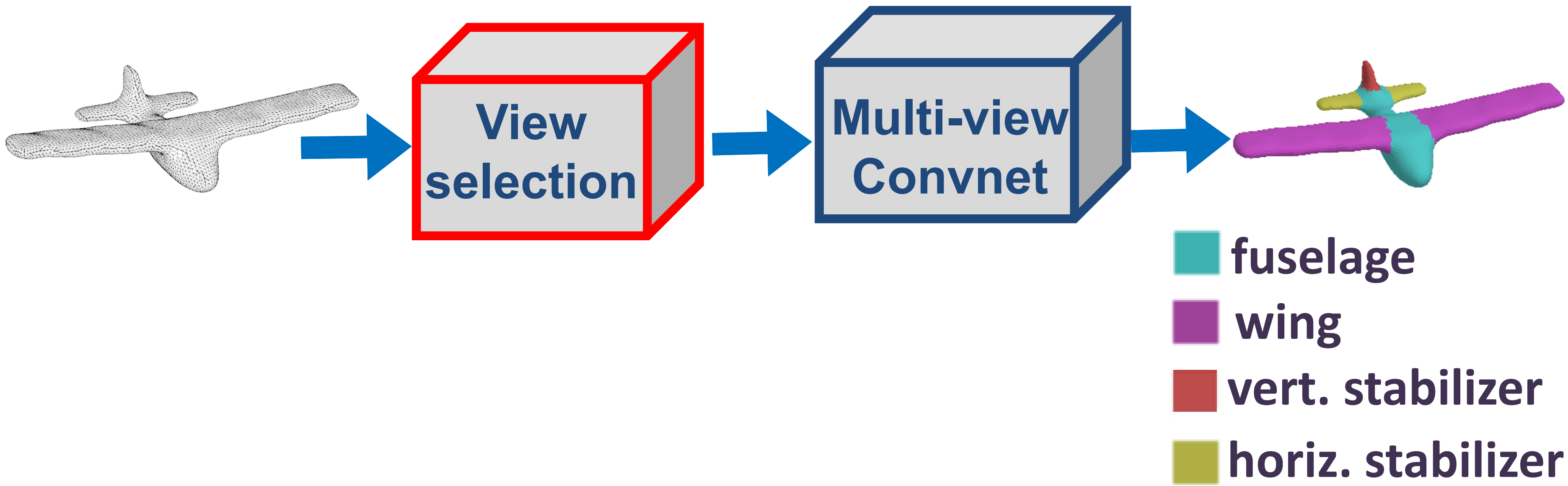
1. **Multi-view convnets for 3D shape analysis**
   - **Shape Segmentation**
   - **Shape Classification & Retrieval**
   - **Shape Correspondences**
2. **Multi-view convnets for 3D shape synthesis**
3. **Discussion / Future work**

View-based convnets for 3D shapes
- Segmentation Pipeline



Kalogerakis, Averkiou, Maji, Chaudhuri, CVPR 2017 (oral)

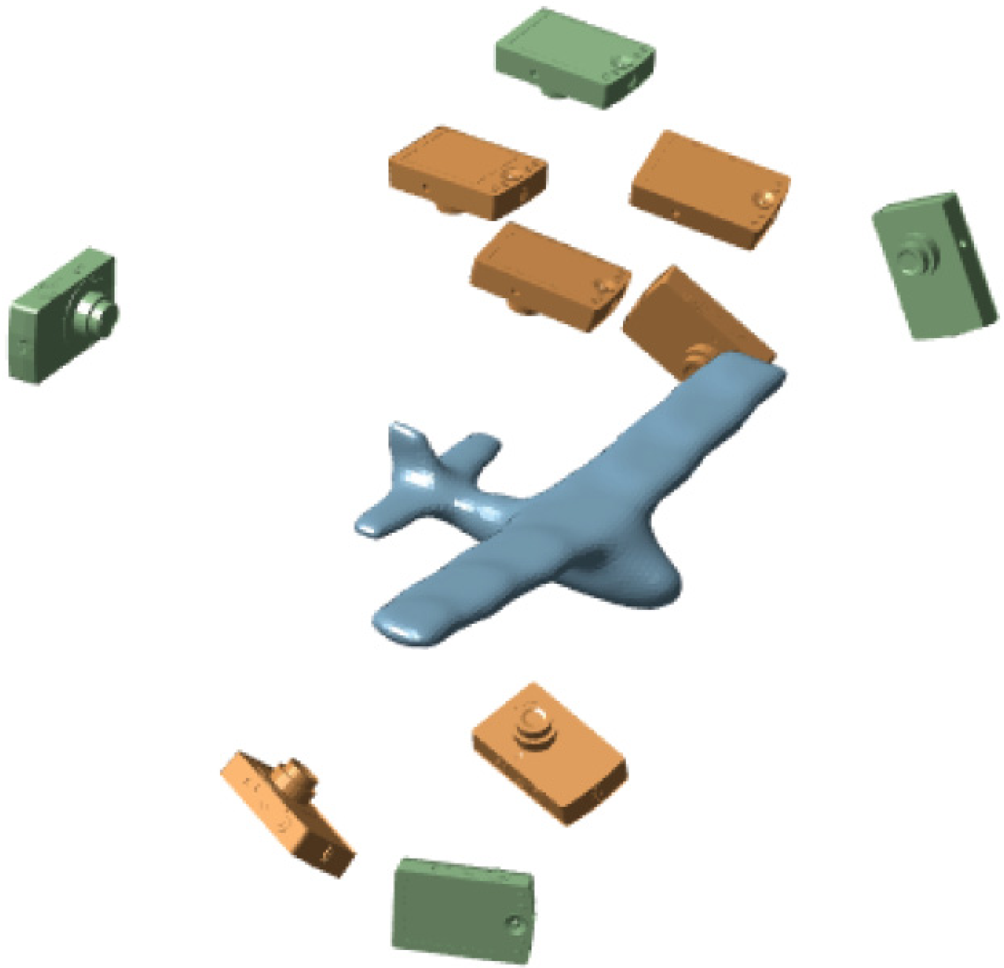
View-based convnets for 3D shapes - Segmentation Pipeline



Kalogerakis, Averkiou, Maji, Chaudhuri, CVPR 2017 (oral)

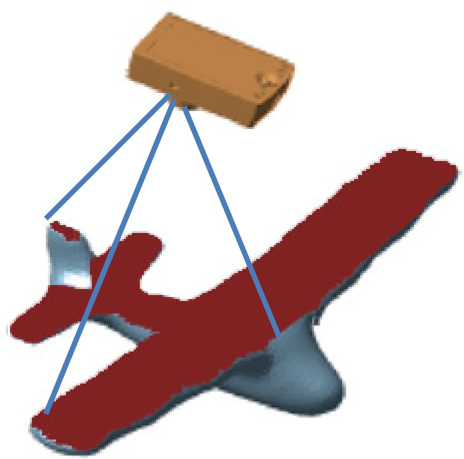
Input: shape as a collection of rendered views

For each input shape, infer a set of viewpoints that **maximally cover its surface**.



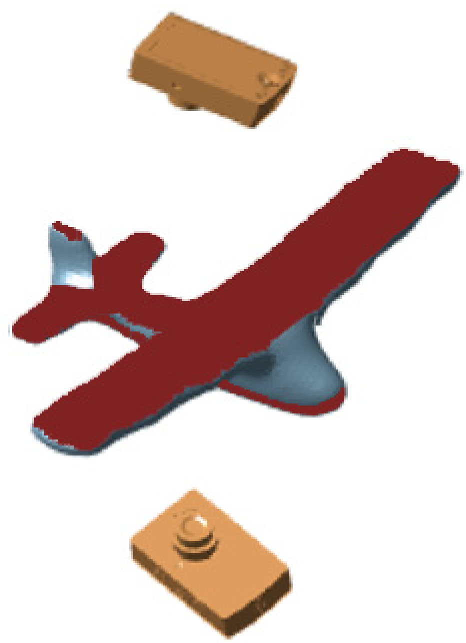
Input: shape as a collection of rendered views

For each input shape, infer a set of viewpoints that **maximally cover its surface**.



Input: shape as a collection of rendered views

For each input shape, infer a set of viewpoints that **maximally cover its surface**.

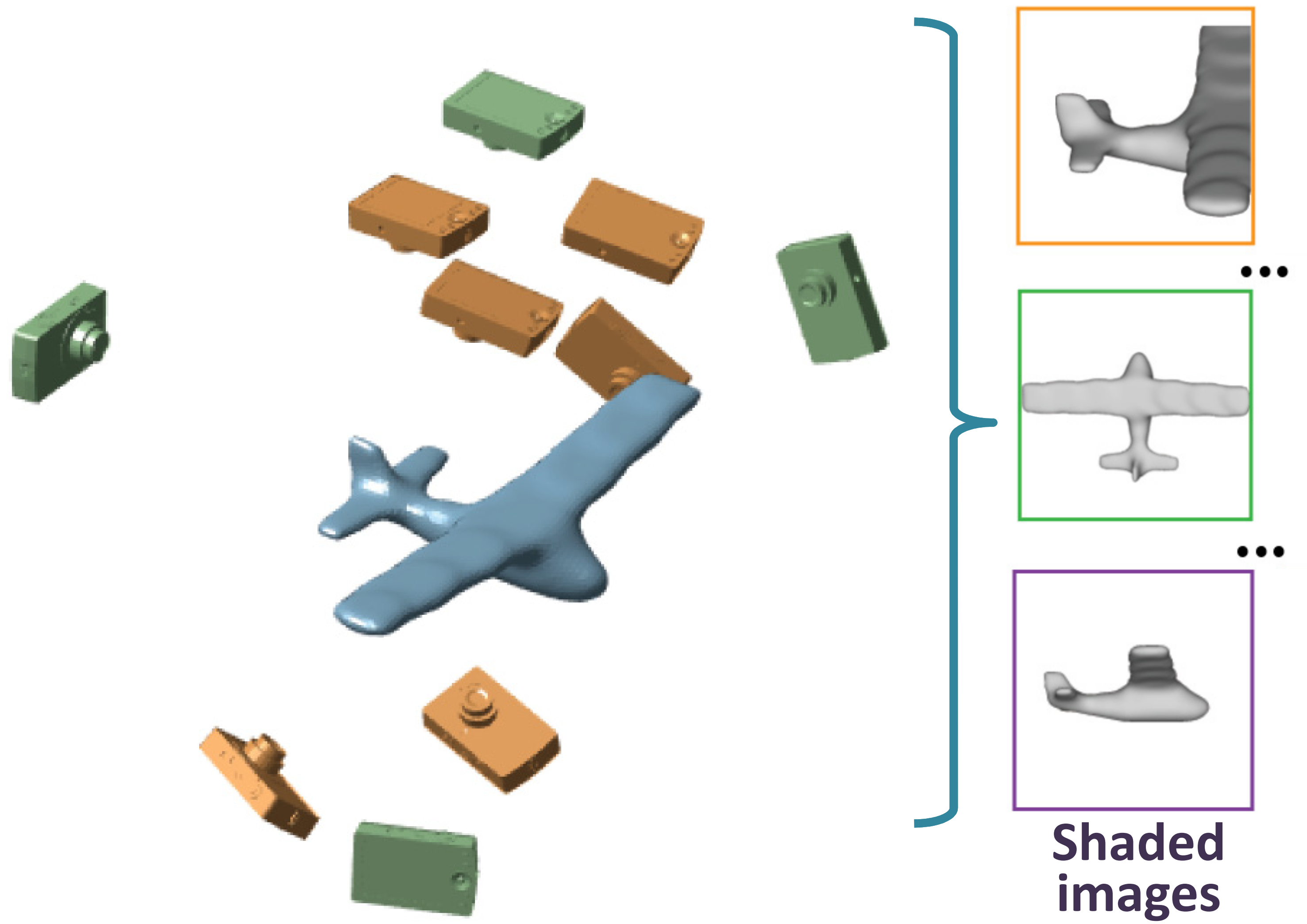


Input: shape as a collection of rendered views

For each input shape, infer a set of viewpoints that **maximally cover its surface**.

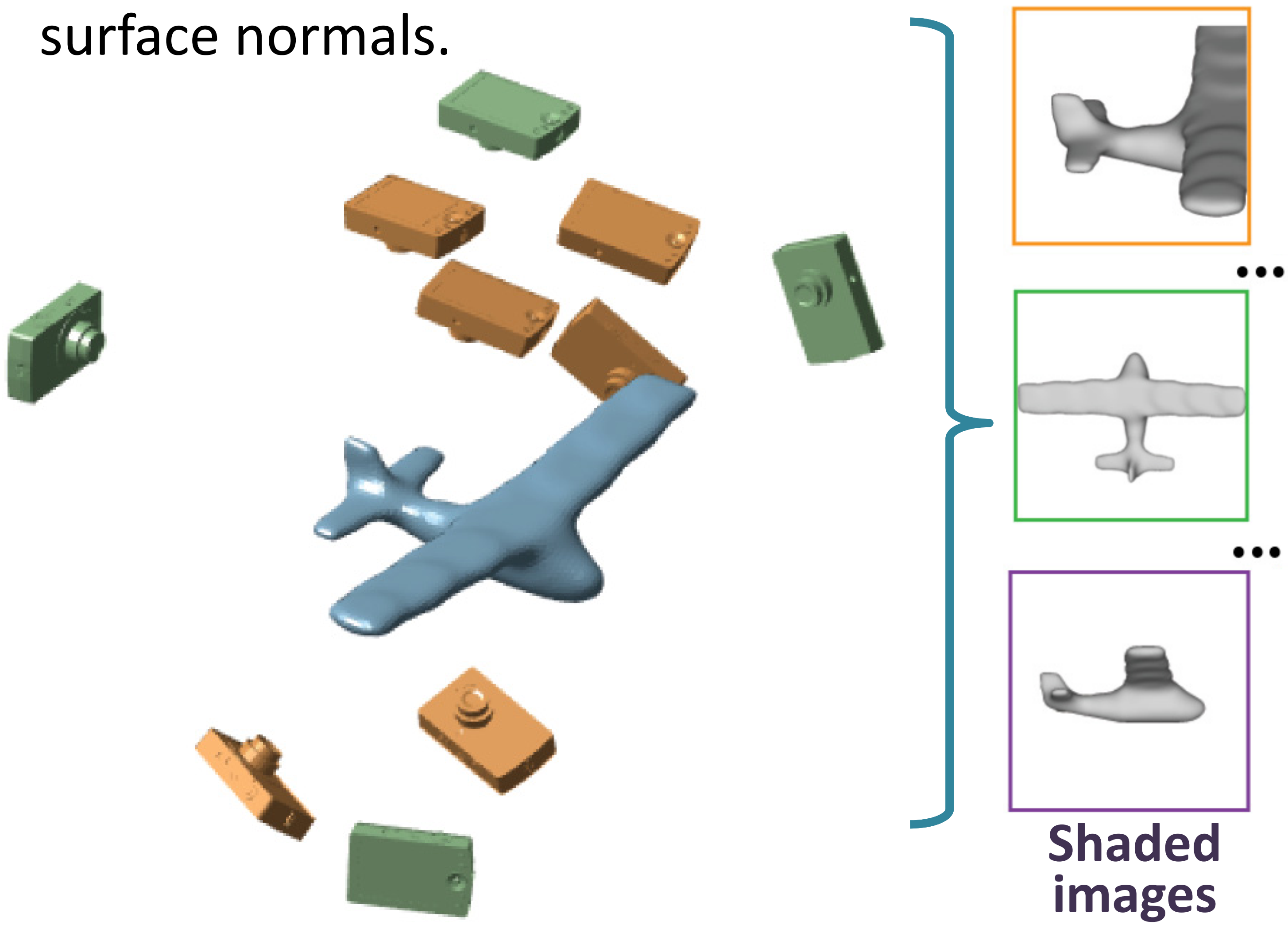
Input: shape as a collection of rendered views

… and **across multiple distances from the surface.**



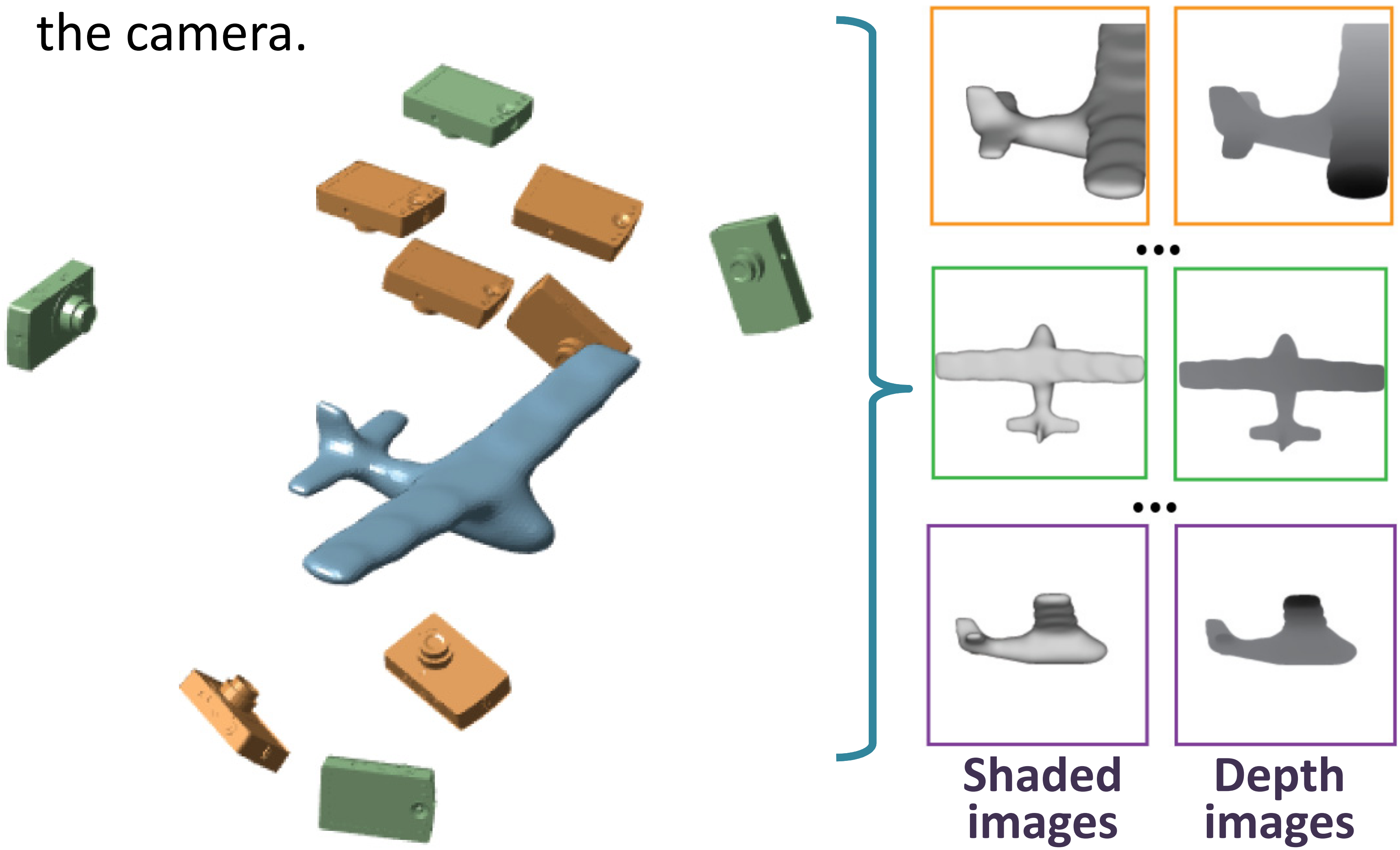
Input: shape as a collection of rendered views

Render **shaded images** (normal dot view vector) encoding surface normals.



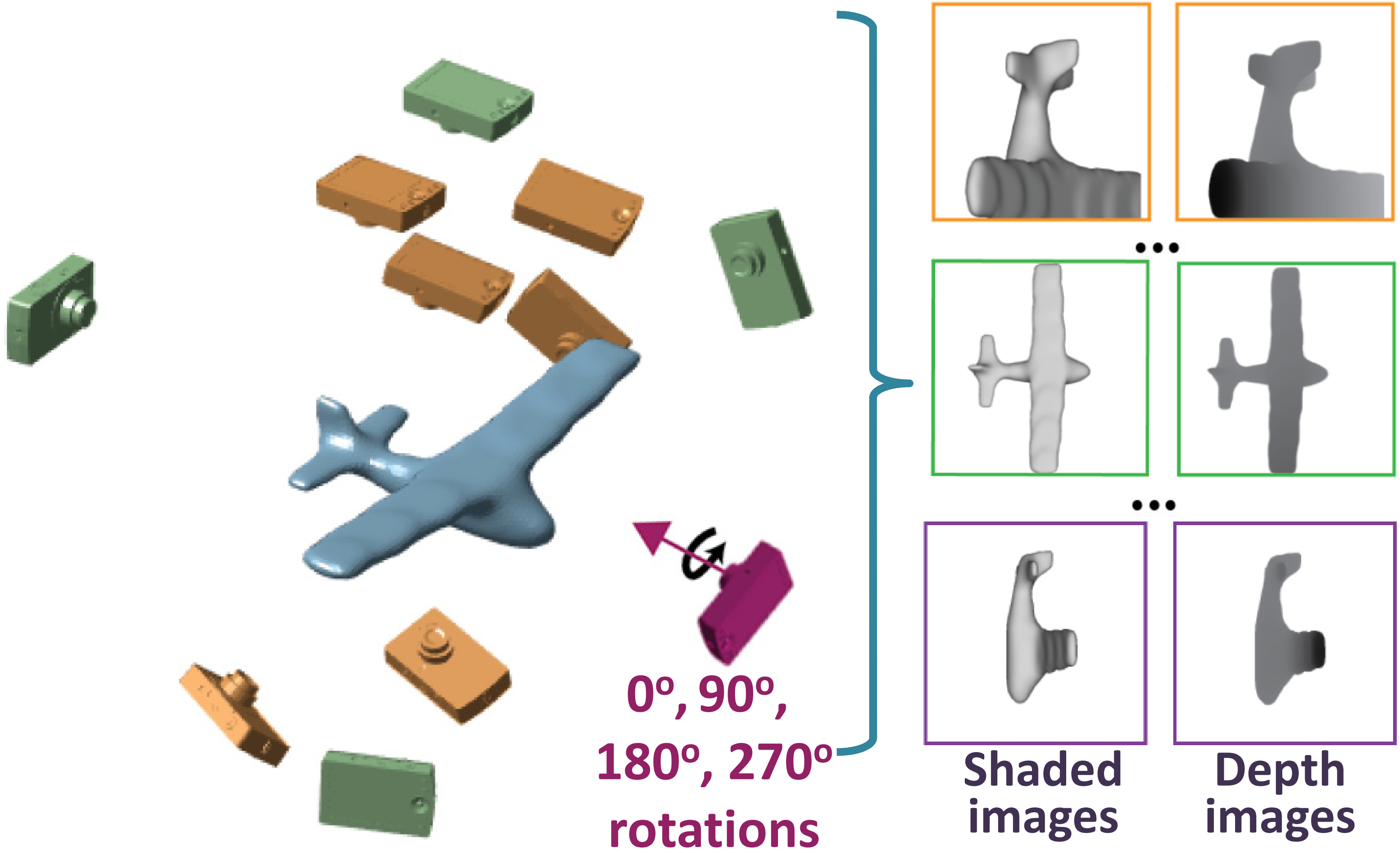
Input: shape as a collection of rendered views

Render also **depth images** encoding surface position relative to the camera.



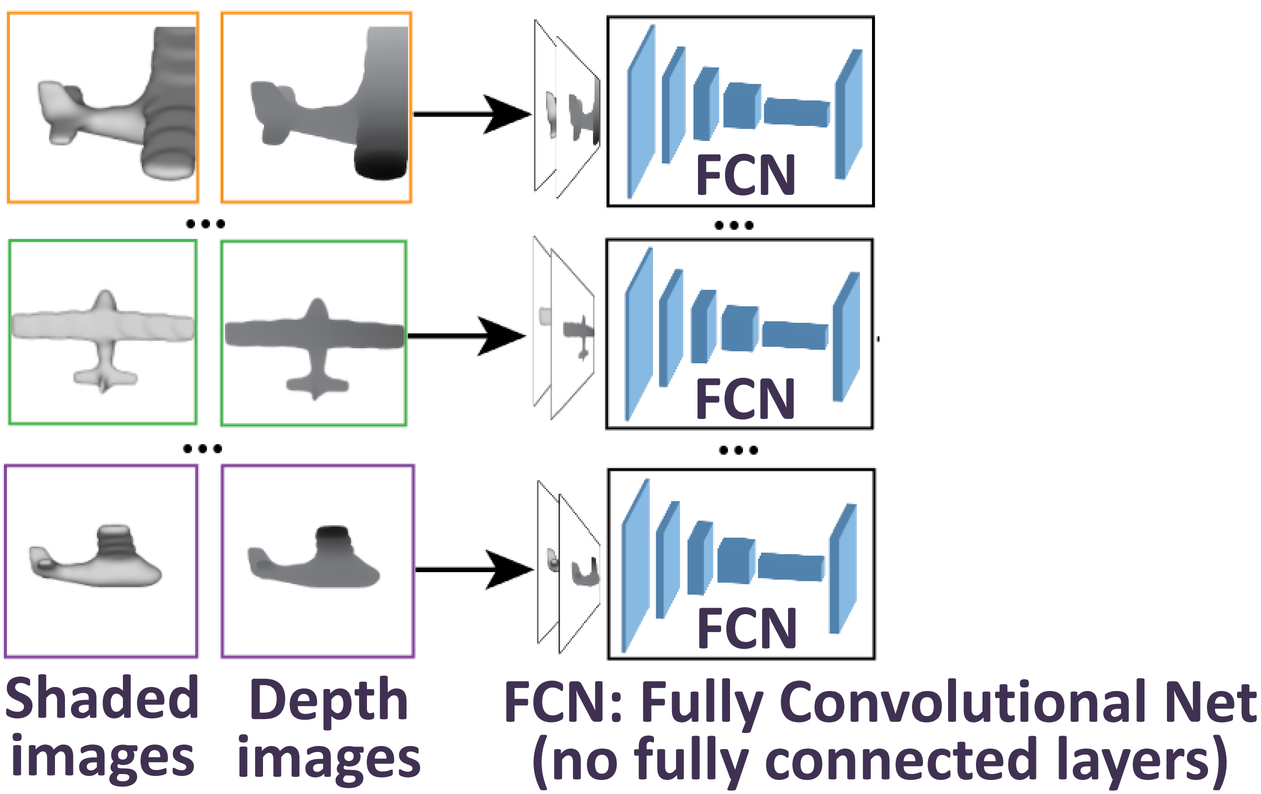
Input: shape as a collection of rendered views

Perform in-plane camera rotations for **rotational invariance.**



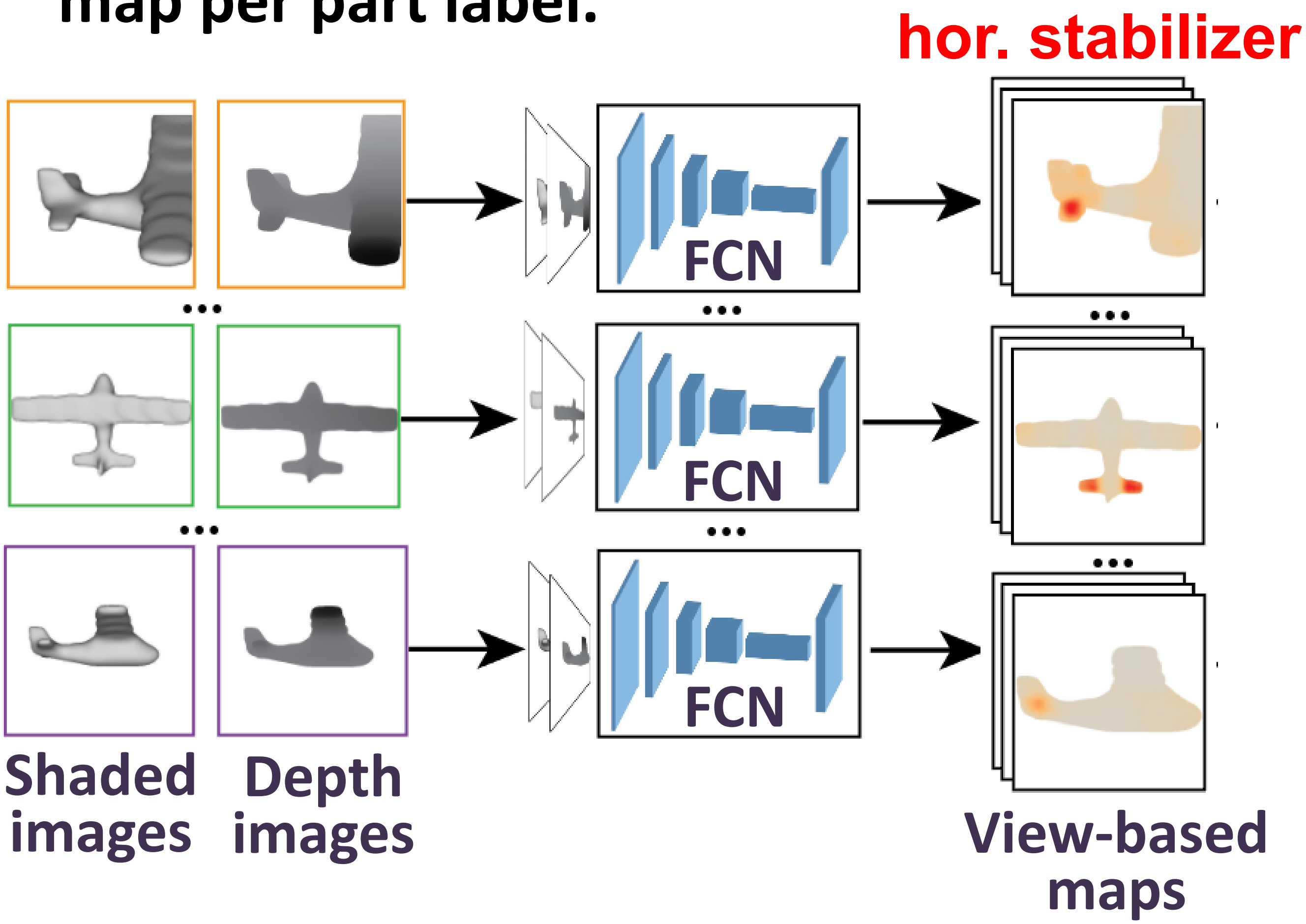
Projective convnet architecture

Each **pair of depth & shaded images** is processed by a FCN.



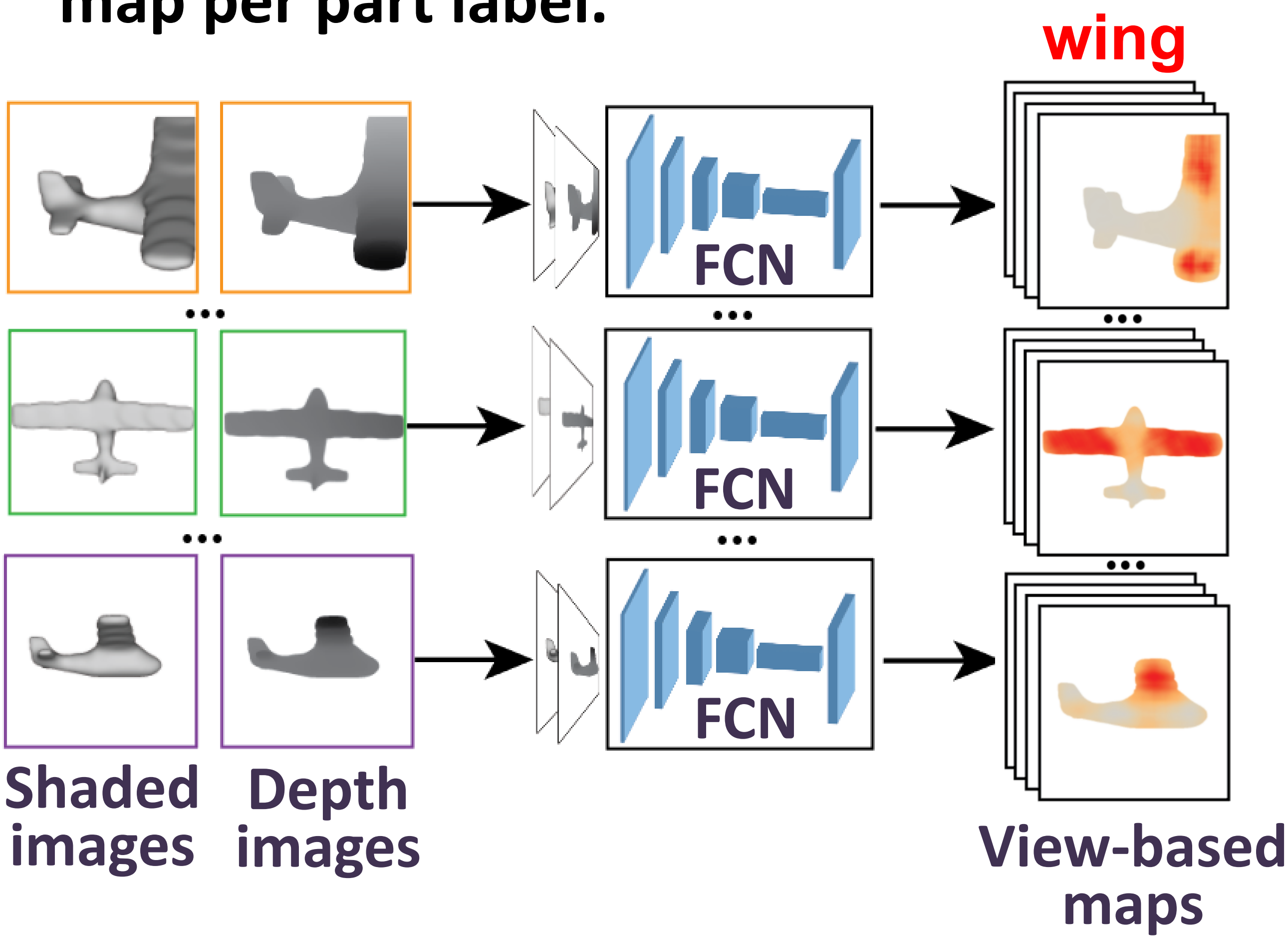
Projective convnet architecture

The output of each FCN branch is a view-based **confidence map per part label.**



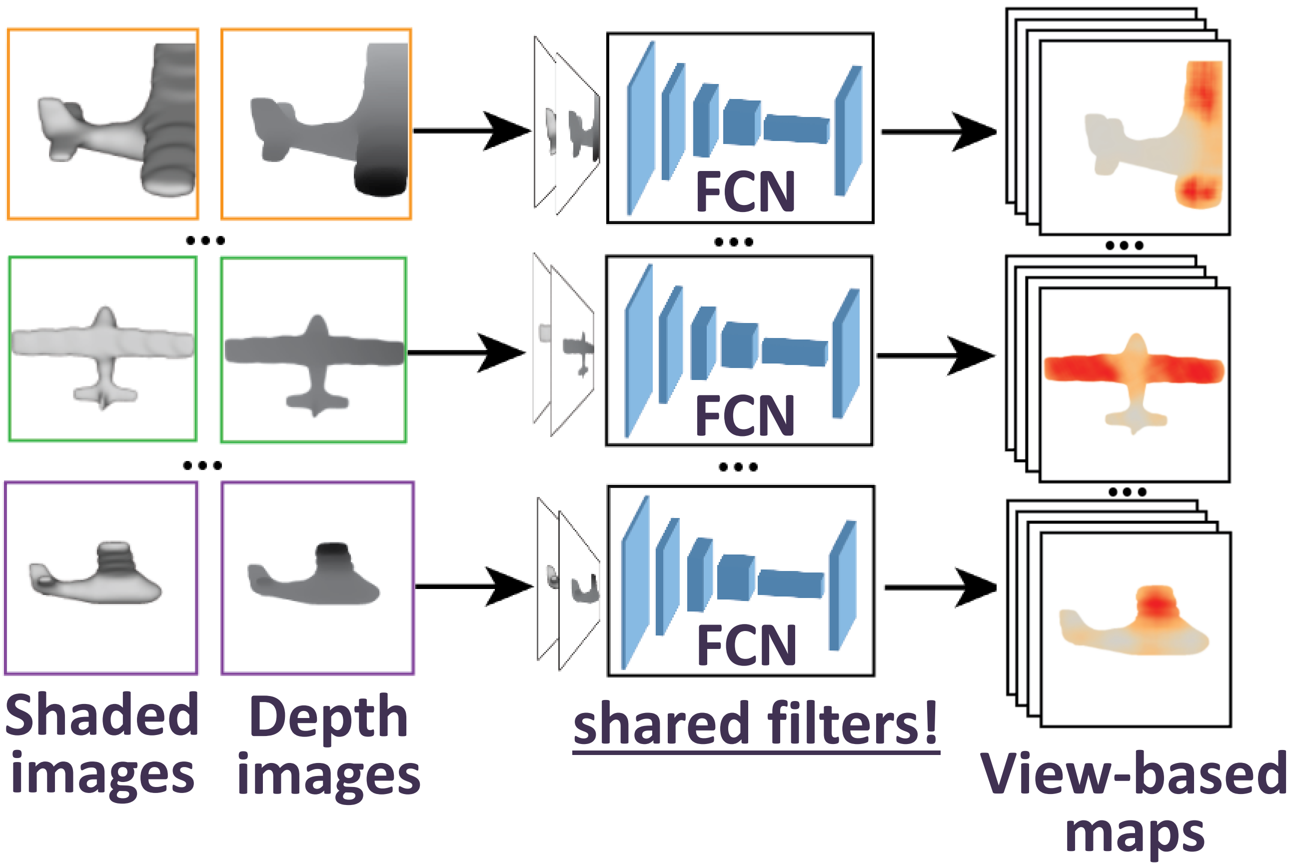
Projective convnet architecture

The output of each FCN branch is a view-based **confidence map per part label.**



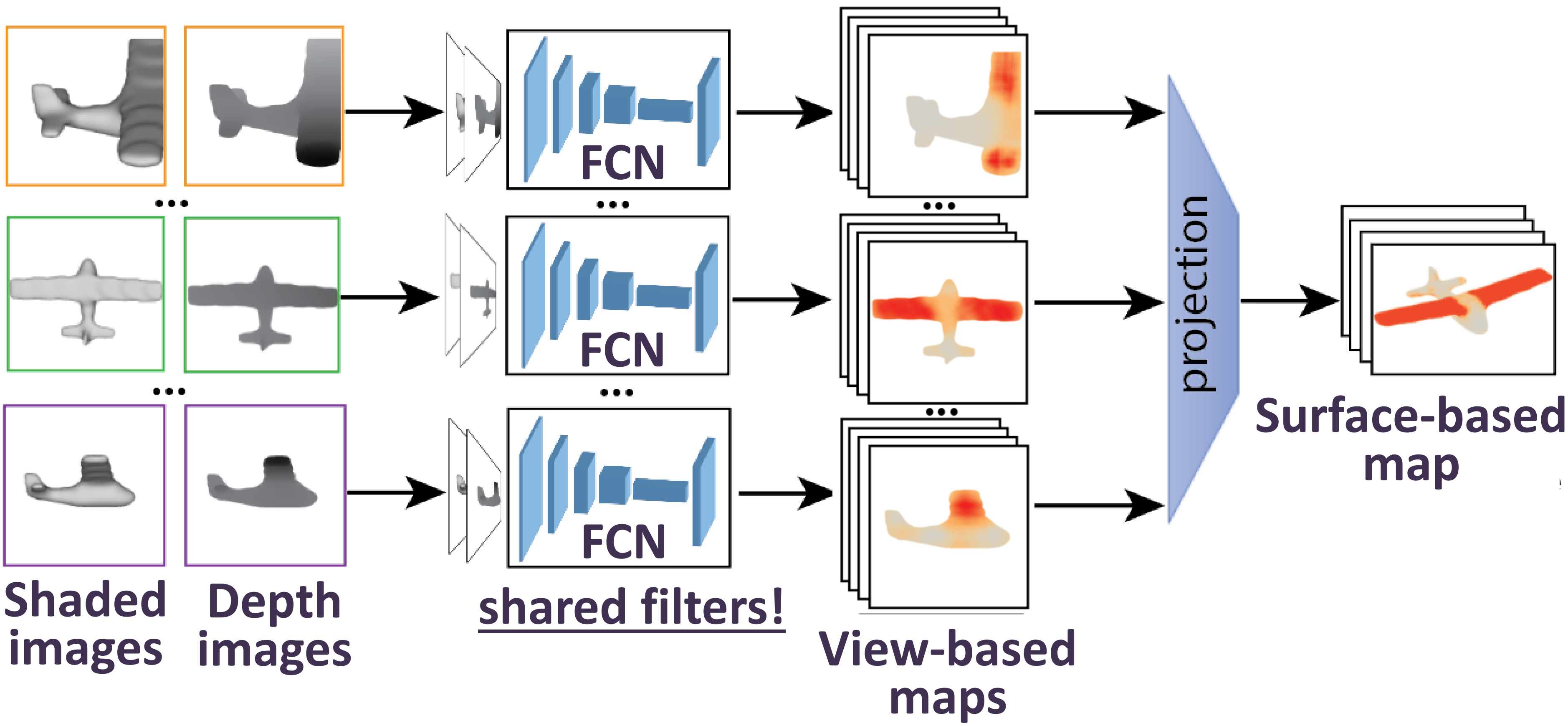
Projective convnet architecture

Views **not ordered** (**no view correspondence** across shapes), thus the **FCN branches share the same parameters**.



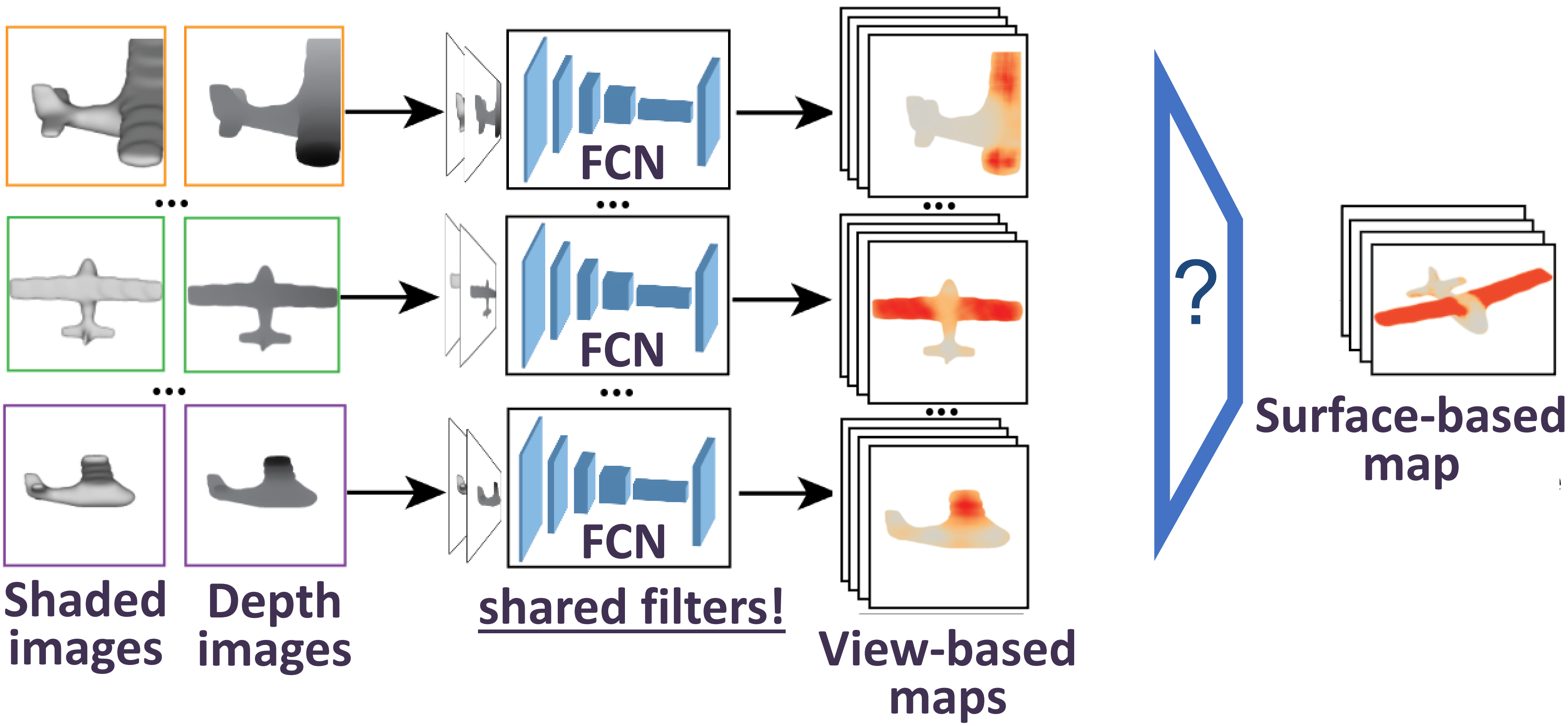
Projective convnet architecture

Aggregate & **project** the image confidence maps from all views **on the surface.**



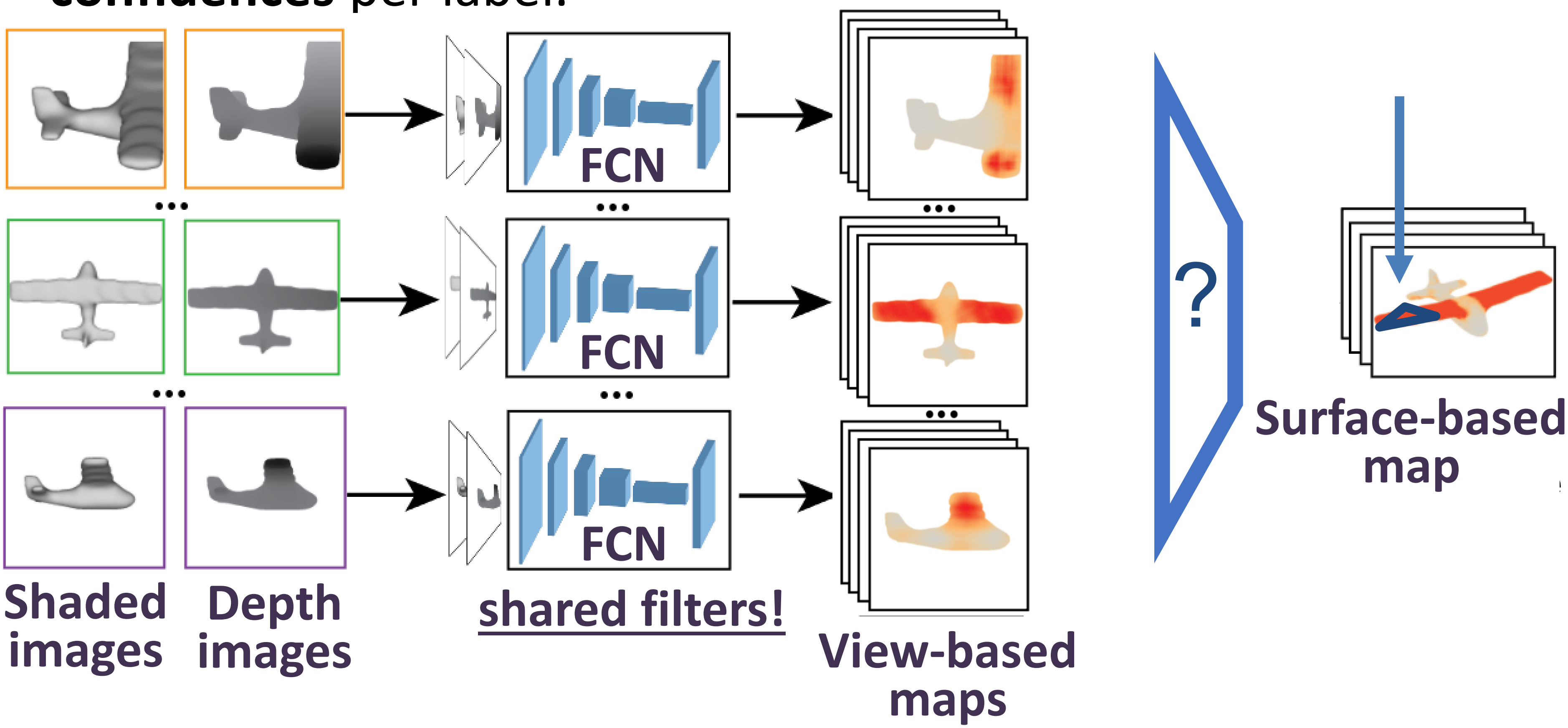
Projective convnet architecture

For each surface element, find all pixels painted by it in all views.
Surface confidence: **max of these pixel confidences** per label.



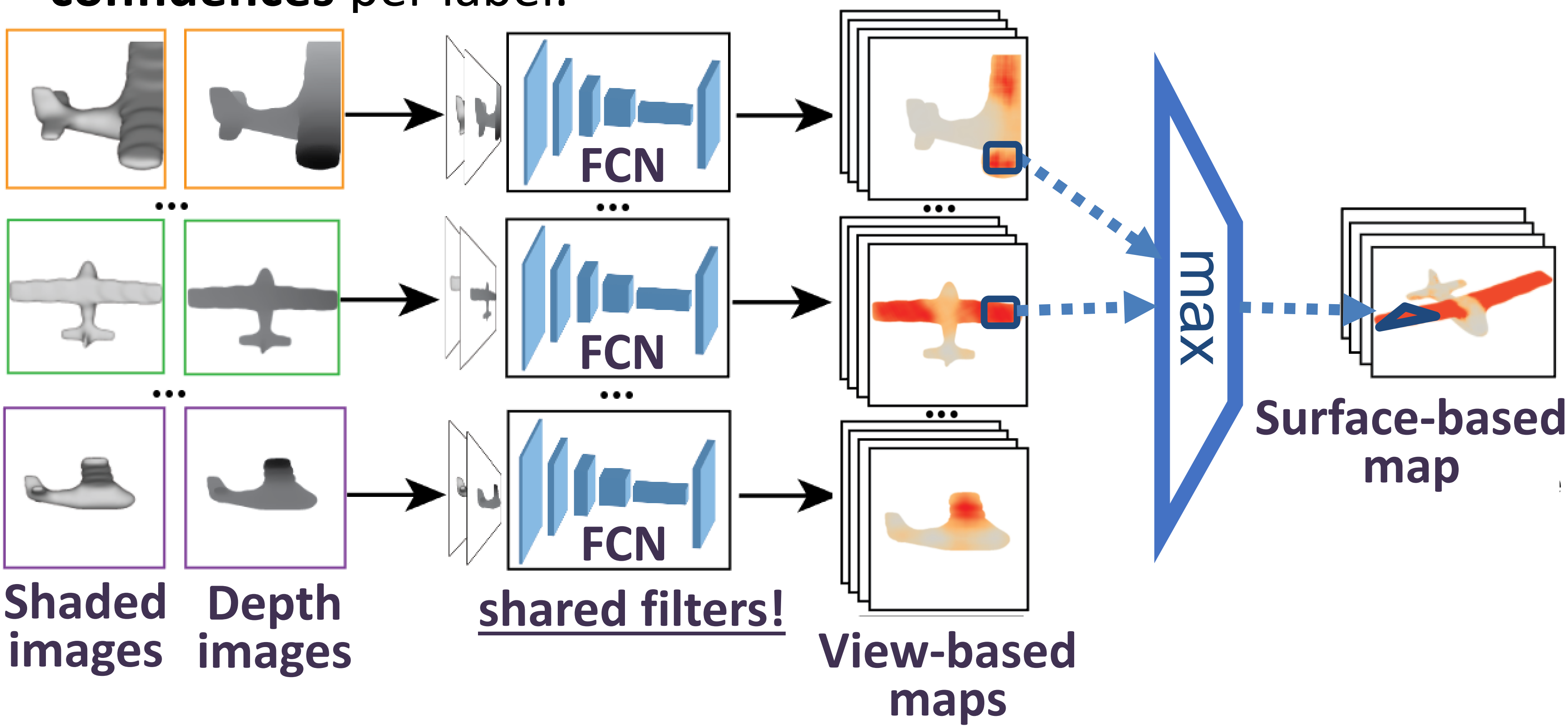
Projective convnet architecture

For each surface element (triangle), find all pixels that include it in all views. **Surface confidence**: use **max of these pixel confidences** per label.



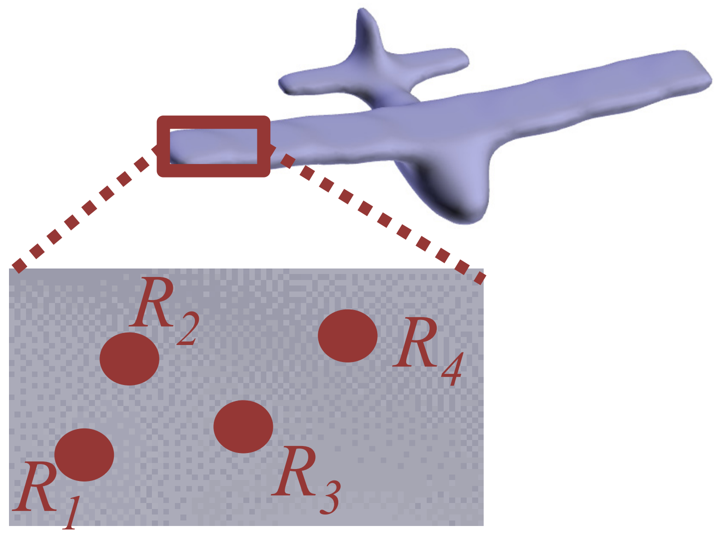
Projective convnet architecture

For each surface element (triangle), find all pixels that include it in all views. **Surface confidence**: use **max of these pixel confidences** per label.



Surface model for spatially coherent labeling

Last layer performs **inference in a probabilistic model defined on the surface.**

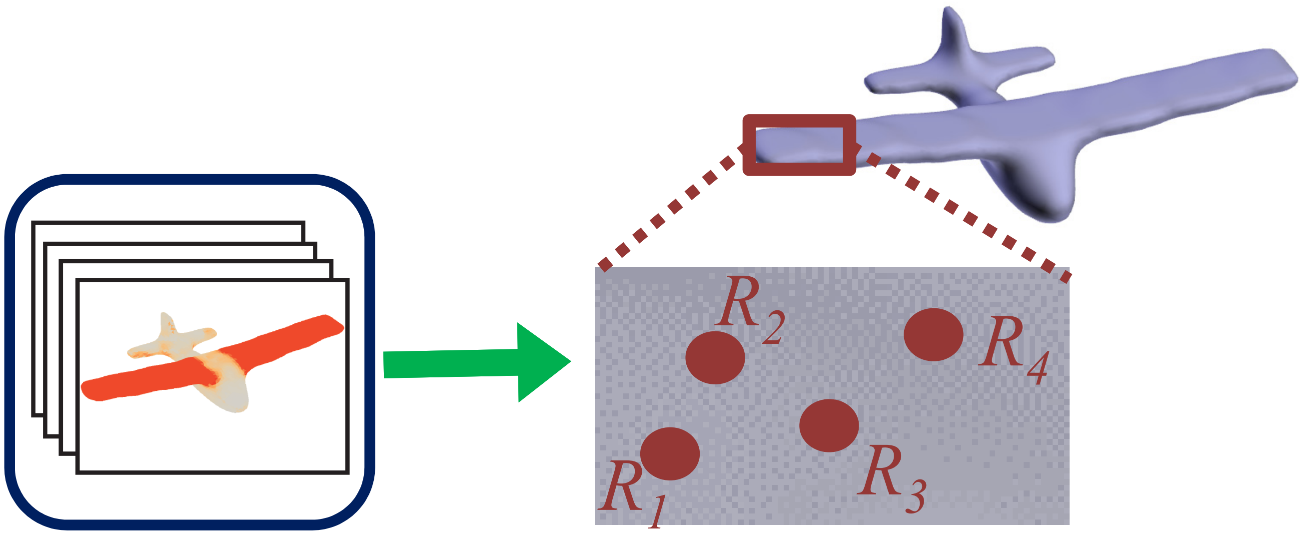


$R_1, R_2, R_3, R_4 \ldots$
random variables
taking values:

- fuselage
- wing
- vert. stabilizer
- horiz. stabilizer

Surface model for spatially coherent labeling

Probabilistic model consists of unary factors based on **surface-based confidences**

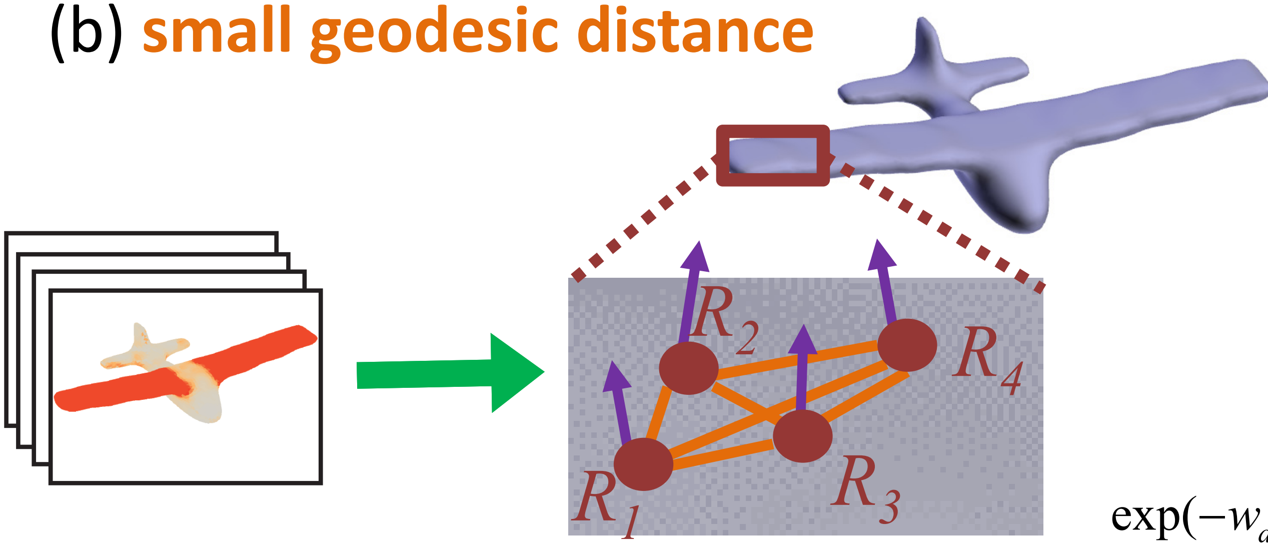


$$P(R_1, R_2, R_3, R_4 \ldots \mid \textbf{shape}) = \frac{1}{Z} \boxed{\prod_{f=1..n} P(R_f \mid \textbf{views})} \prod_{f,f'} P(R_f, R_{f'} \mid \textbf{surface})$$

Unary factors (FCN confidences)

Surface model for spatially coherent labeling

Pairwise terms **favor same label** for triangles with:
(a) **similar surface normals**
(b) **small geodesic distance**



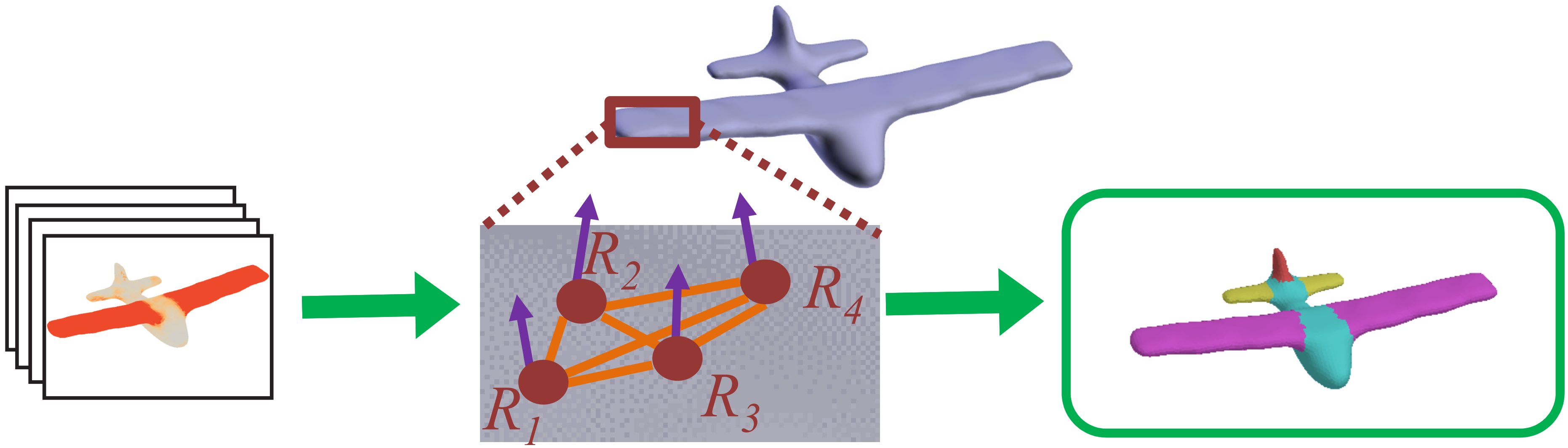
$$\exp(-w_{dist} \cdot \mathbf{d}_{\mathbf{f,f'}}^{2}) \qquad \exp(-w_{angle} \cdot \boldsymbol{\omega}_{\mathbf{f,f'}}^{2})$$

$$P(R_1, R_2, R_3, R_4 \ldots \mid \textbf{shape}) = \frac{1}{Z} \prod_{f=1..n} P(R_f \mid \textbf{views}) \boxed{\prod_{f,f'} P(R_f, R_{f'} \mid \textbf{surface})}$$

Pairwise factors
(geodesic+normal distance)

Inference

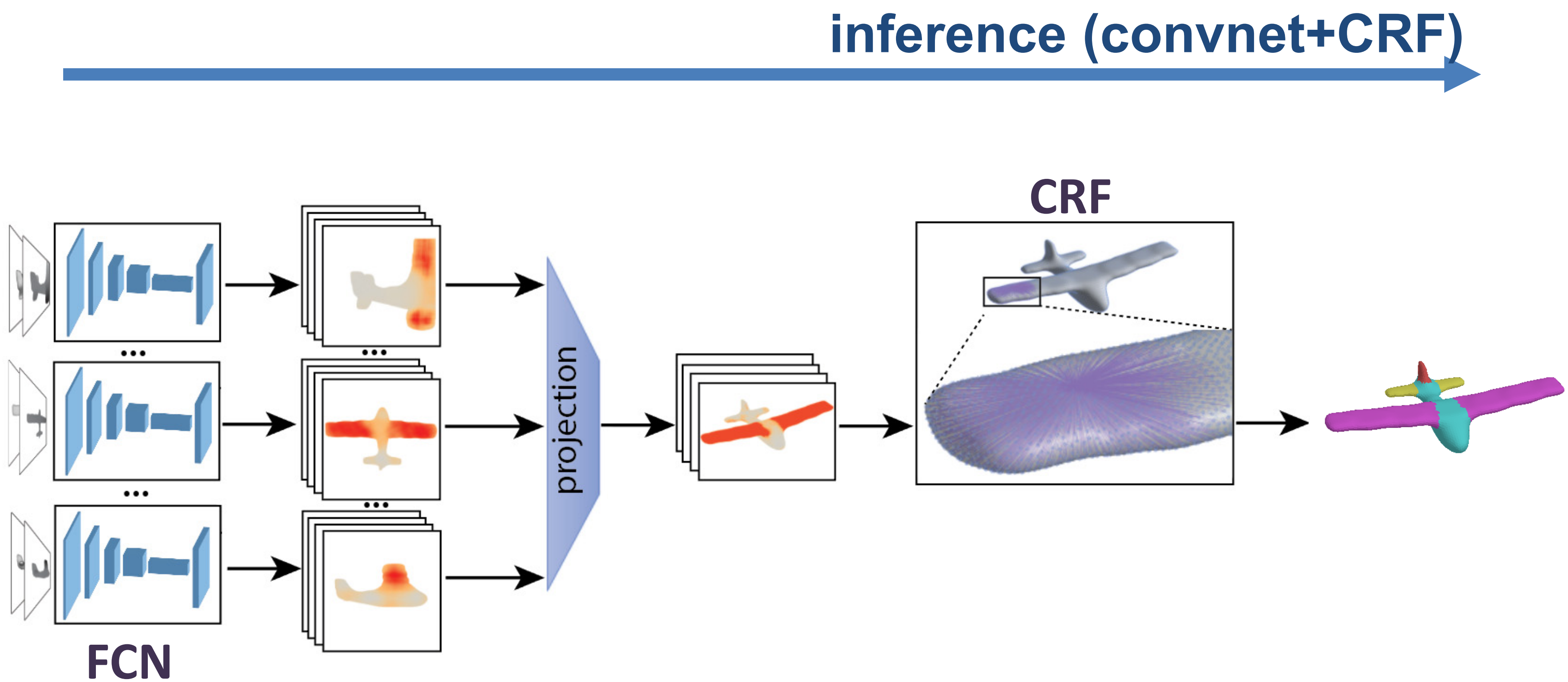
Infer **most likely joint assignment** to all surface random variables of the probabilistic model (Conditional Random Field)



$$\boxed{\max_{} P(R_1, R_2, R_3, R_4 \ldots \mid \textbf{shape})} = \frac{1}{Z} \prod_{f=1..n} P(R_f \mid \textbf{views}) \prod_{f,f'} P(R_f, R_{f'} \mid \textbf{surface})$$

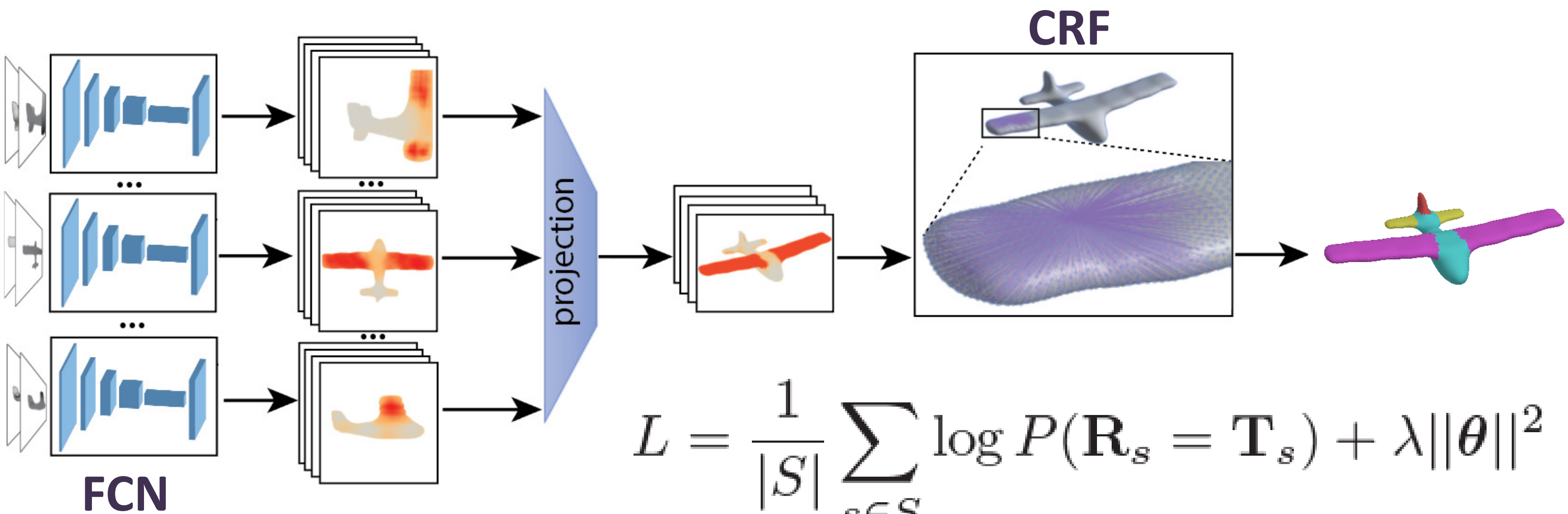
MAP assignment (mean–field inference)

Forward pass



Training

The architecture is trained **end-to-end** with analytic gradients.

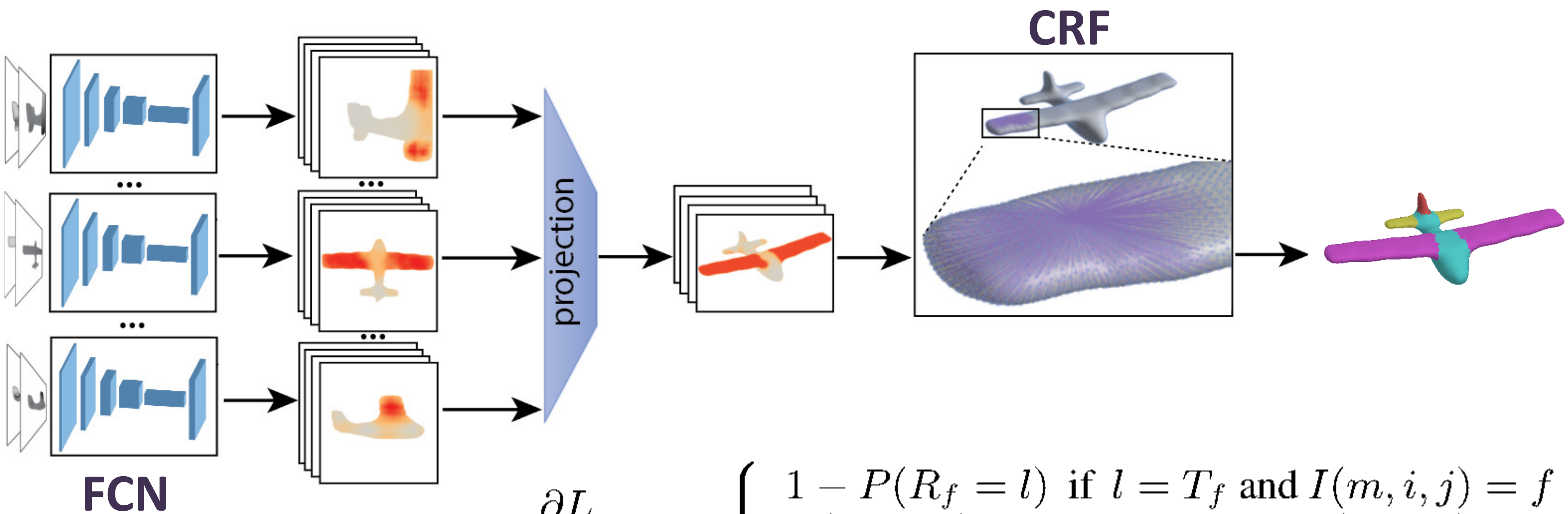


$$L = \frac{1}{|S|} \sum_{s \in S} \log P(\mathbf{R}_s = \mathbf{T}_s) + \lambda ||\boldsymbol{\theta}||^2$$

Backpropagation / joint training (convnet+CRF)

Training

The architecture is trained **end-to-end** with analytic gradients.

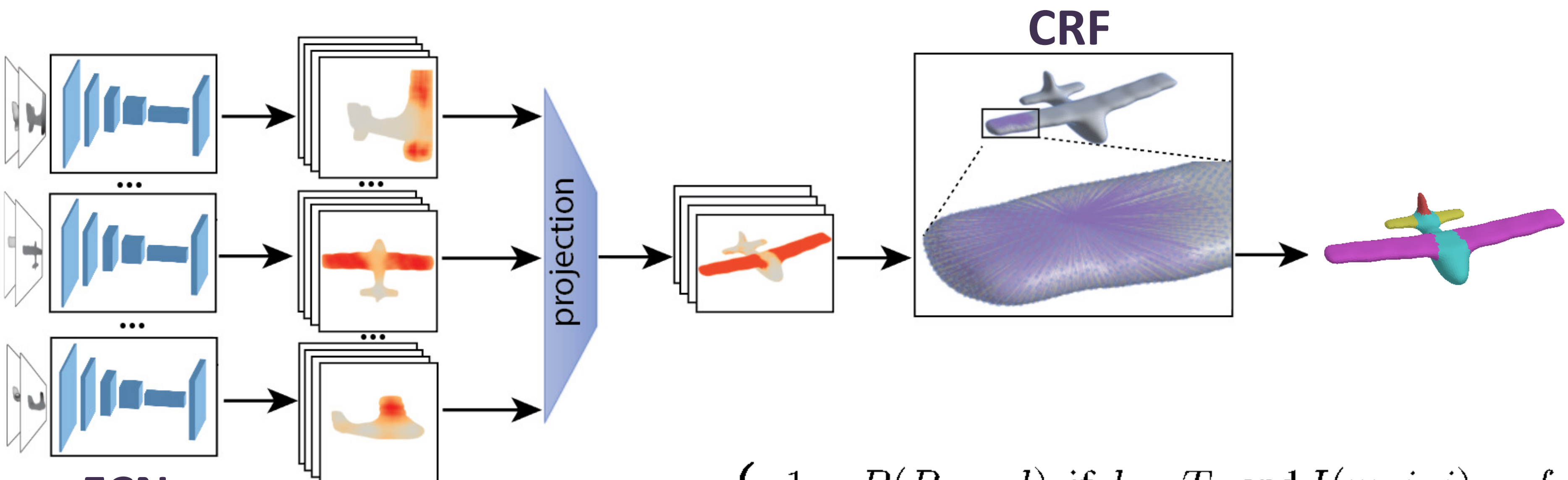


$$\frac{\partial L}{\partial C(m,i,j,l)} = \begin{cases} 1 - P(R_f = l) & \text{if } l = T_f \text{ and } I(m,i,j) = f \\ P(R_f = l) & \text{if } l \neq T_f \text{ and } I(m,i,j) = f \\ 0 & \text{otherwise} \end{cases}$$

Backpropagation / joint training (convnet+CRF)

Training

The architecture is trained **end-to-end** with analytic gradients.
Training starts from a **pretrained image-based net** (VGG16).



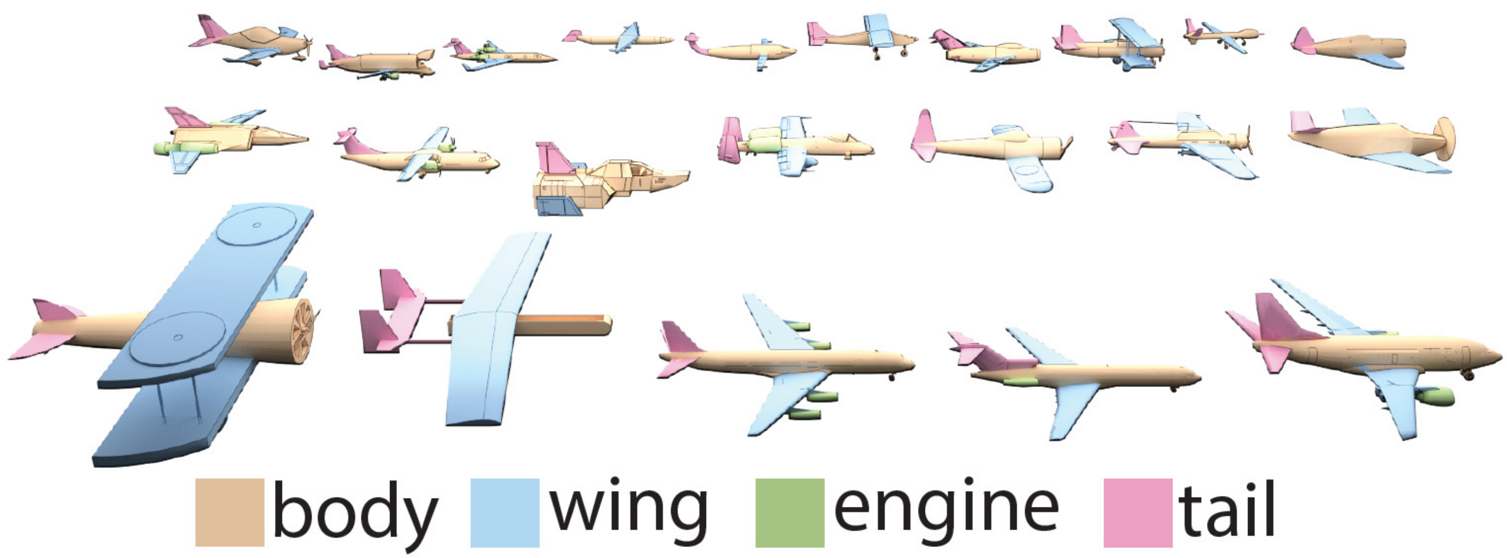
FCN
Pre-trained

$$\frac{\partial L}{\partial C(m,i,j,l)} = \begin{cases} 1 - P(R_f = l) & \text{if } l = T_f \text{ and } I(m,i,j) = f \\ P(R_f = l) & \text{if } l \neq T_f \text{ and } I(m,i,j) = f \\ 0 & \text{otherwise} \end{cases}$$

Backpropagation / joint training (convnet+CRF)

Training

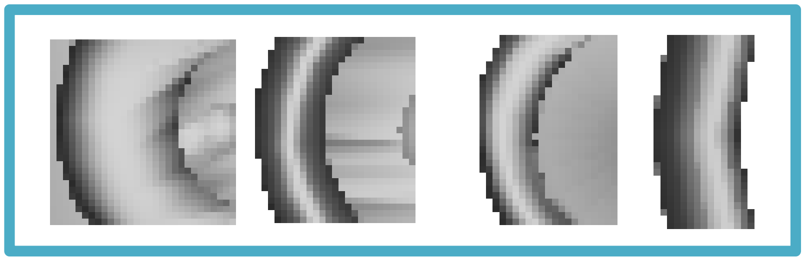
The architecture is trained **end-to-end** with analytic gradients. Training starts from a **pretrained image-based net** (VGG16), then **fine-tune on segmented shape datasets.**



[Yi et al. 2016]

What are the learned filters doing?

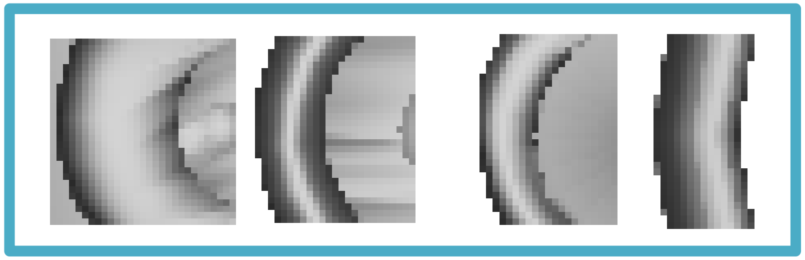
Activated in the presence of **certain surface patterns / patches**



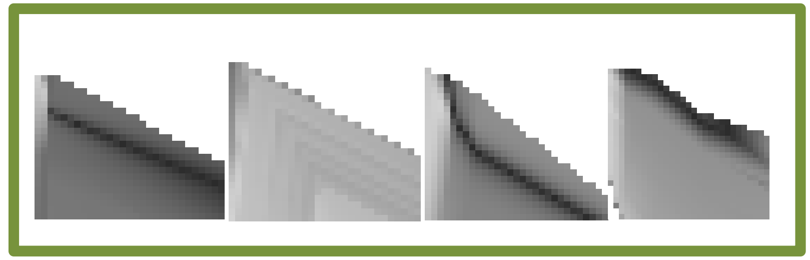
conv4

What are the learned filters doing?

Activated in the presence of **certain surface patterns / patches**

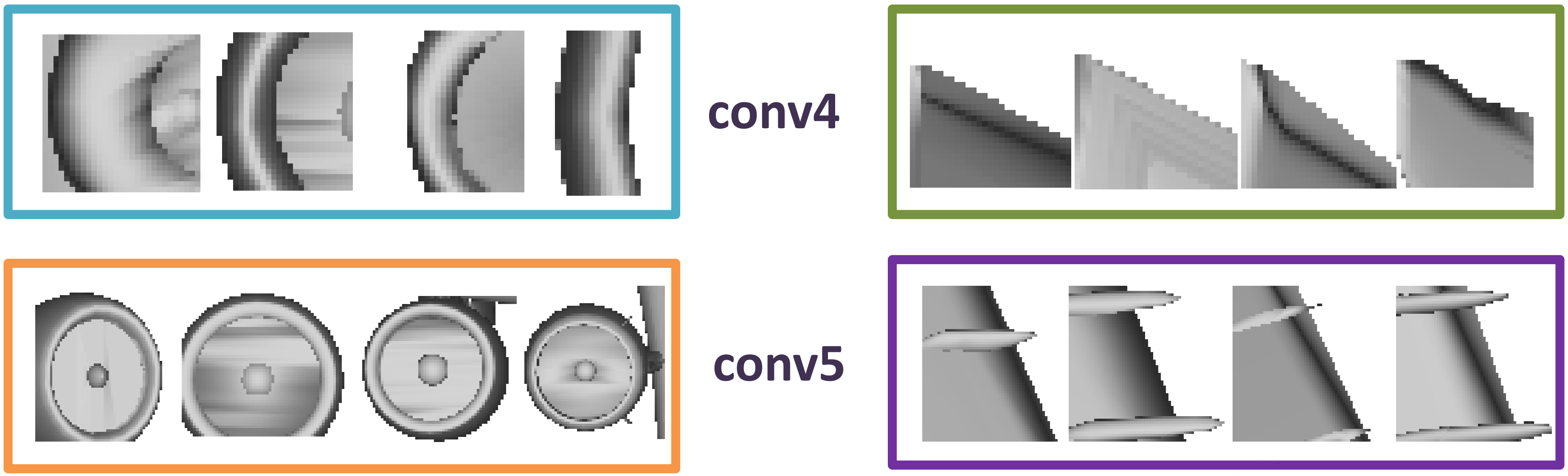


conv4



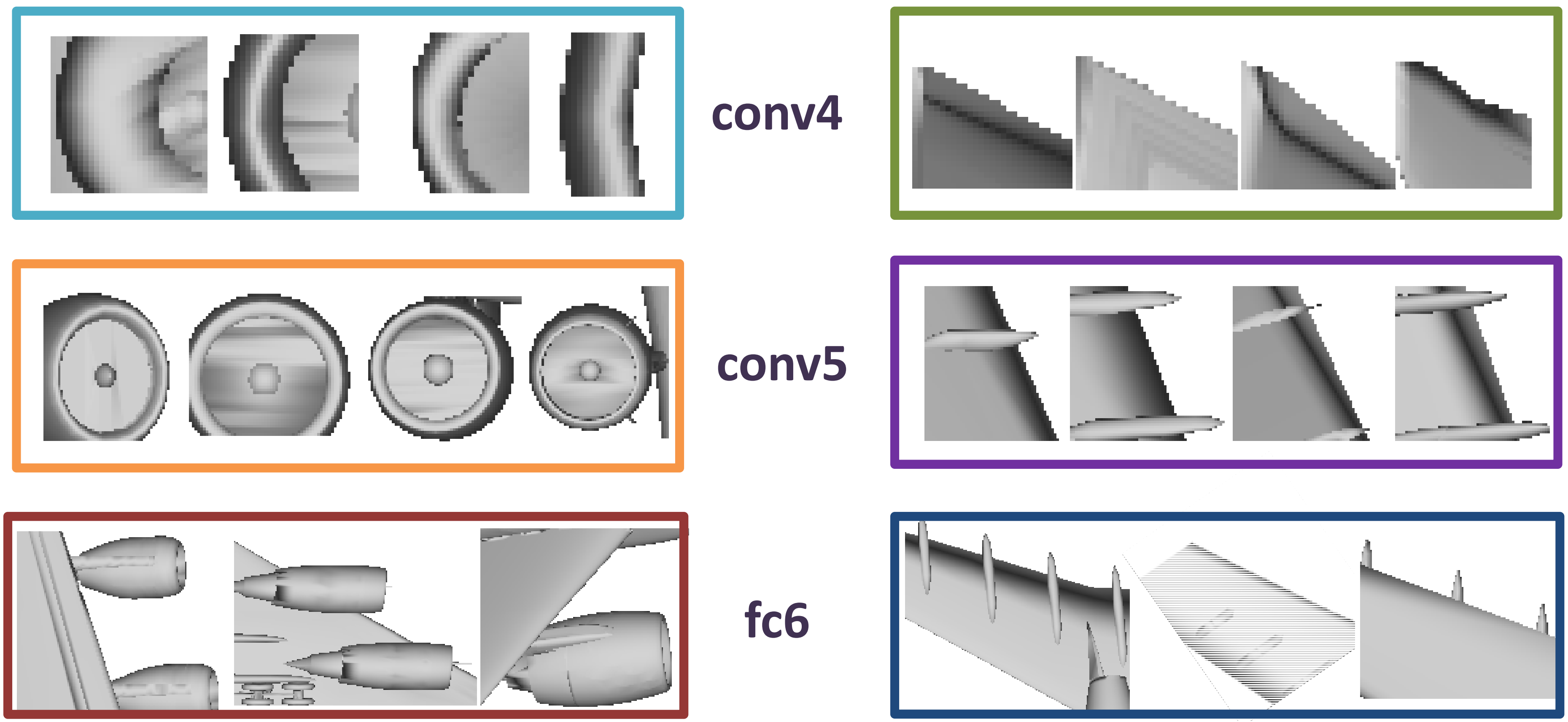
What are the learned filters doing?

Activated in the presence of **certain surface patterns / patches**



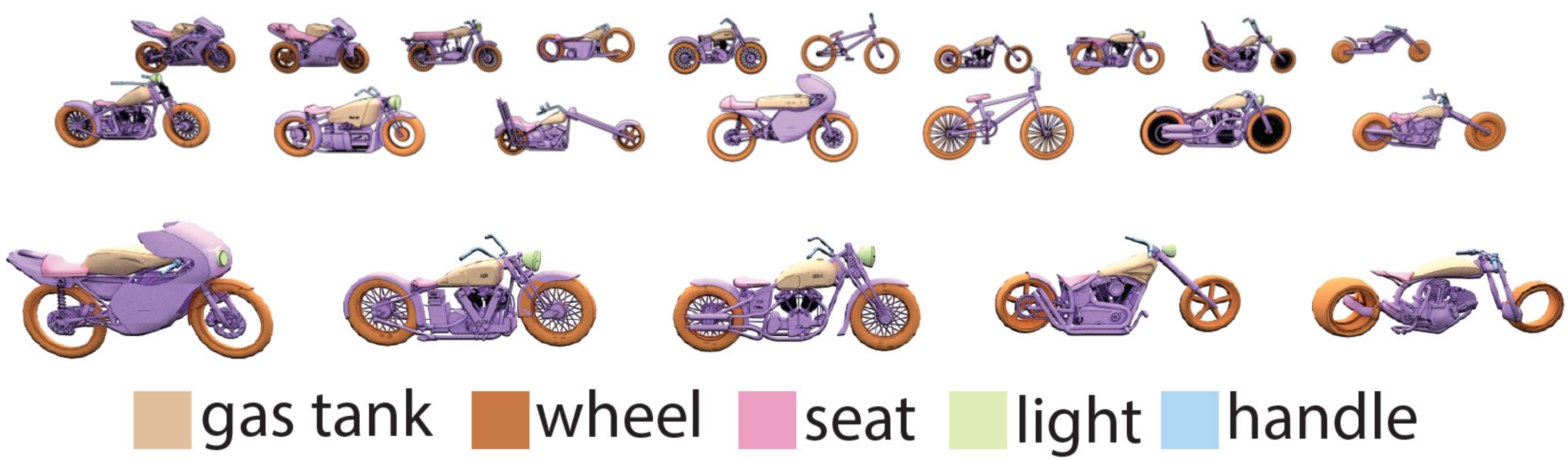
What are the learned filters doing?

Activated in the presence of **certain surface patterns / patches**



Dataset used in experiments

Evaluation on **ShapeNet + LPSB + COSEG** (46 classes of shapes). **50%** used for training / **50%** used for test split **per Shapenet category**. No assumption on shape orientation.



[Yi et al. 2016]

Results

Labeling accuracy on **ShapeNet** test dataset:
(no assumption on shape orientation)

ShapeBoost	Guo et al.	ShapePFCN
81.2	80.6	**87.5**

ShapeBoost: JointBoost on geometric descriptors [Kalogerakis et al. 2010]
Guo et al.: Convnet on geometric descriptors
ShapePFCN: Shape Projective Fully Convolutional Network

Results

Labeling accuracy on **ShapeNet** test dataset:
(no assumption on shape orientation)

Ignore easy classes (2 or 3 part labels)

ShapeBoost	Guo et al.	ShapePFCN
81.2	80.6	**87.5**
76.8	76.8	**84.7**

~8% improvement in labeling accuracy for complex categories (vehicles, furniture)

Results

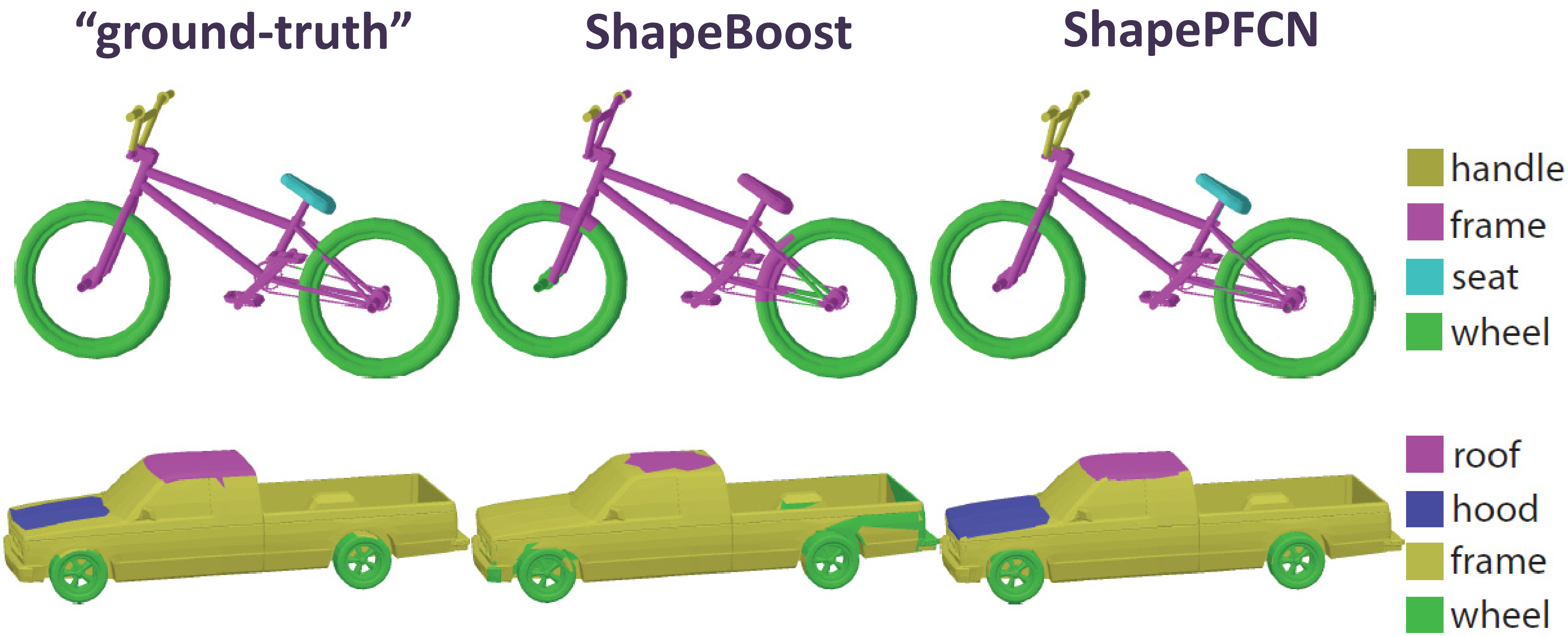
Labeling accuracy on **ShapeNet** test dataset:

(assume consistent upright orientation + render y-coords)

Ignore easy classes (2 or 3 part labels)

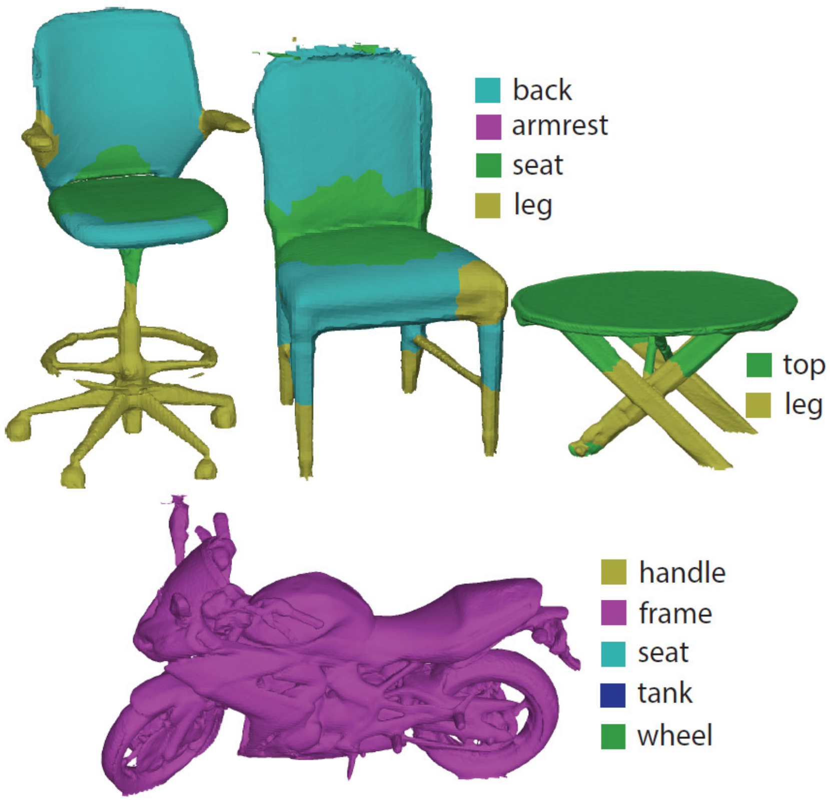
ShapeBoost	Guo et al.	ShapePFCN
81.2	80.6	**89.4**
76.8	76.8	**86.6**

~10% improvement in labeling accuracy for complex categories (vehicles, furniture)

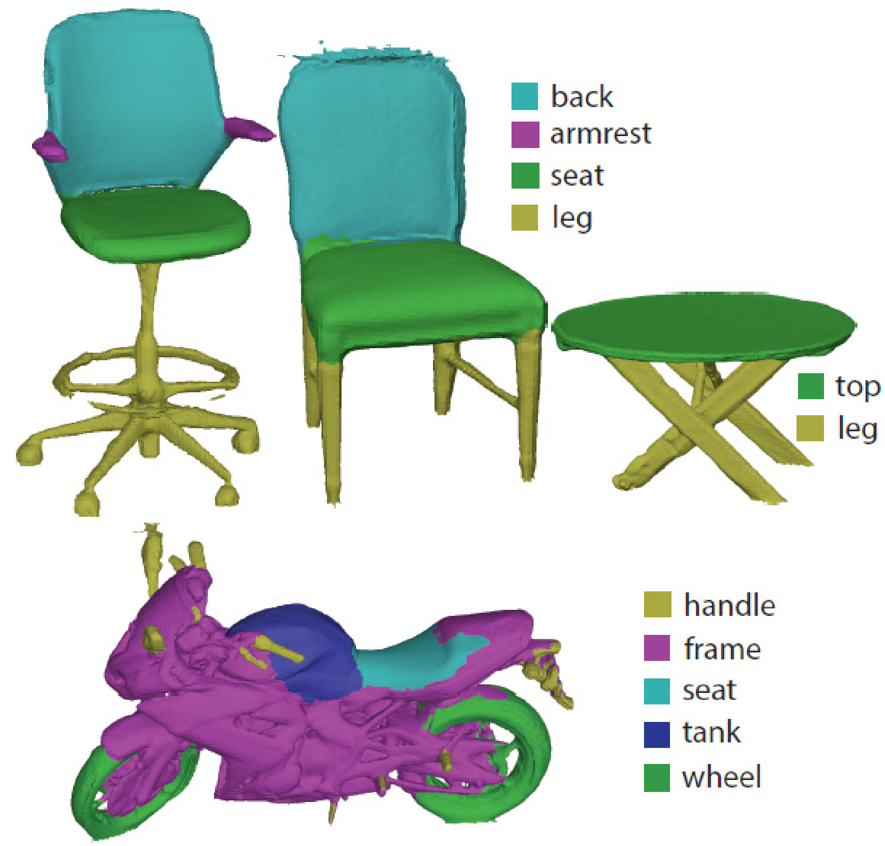


“ground-truth”
ShapeBoost
ShapePFCN
handle
frame
seat
wheel
roof
hood
frame
wheel

ShapeBoost



ShapePFCN

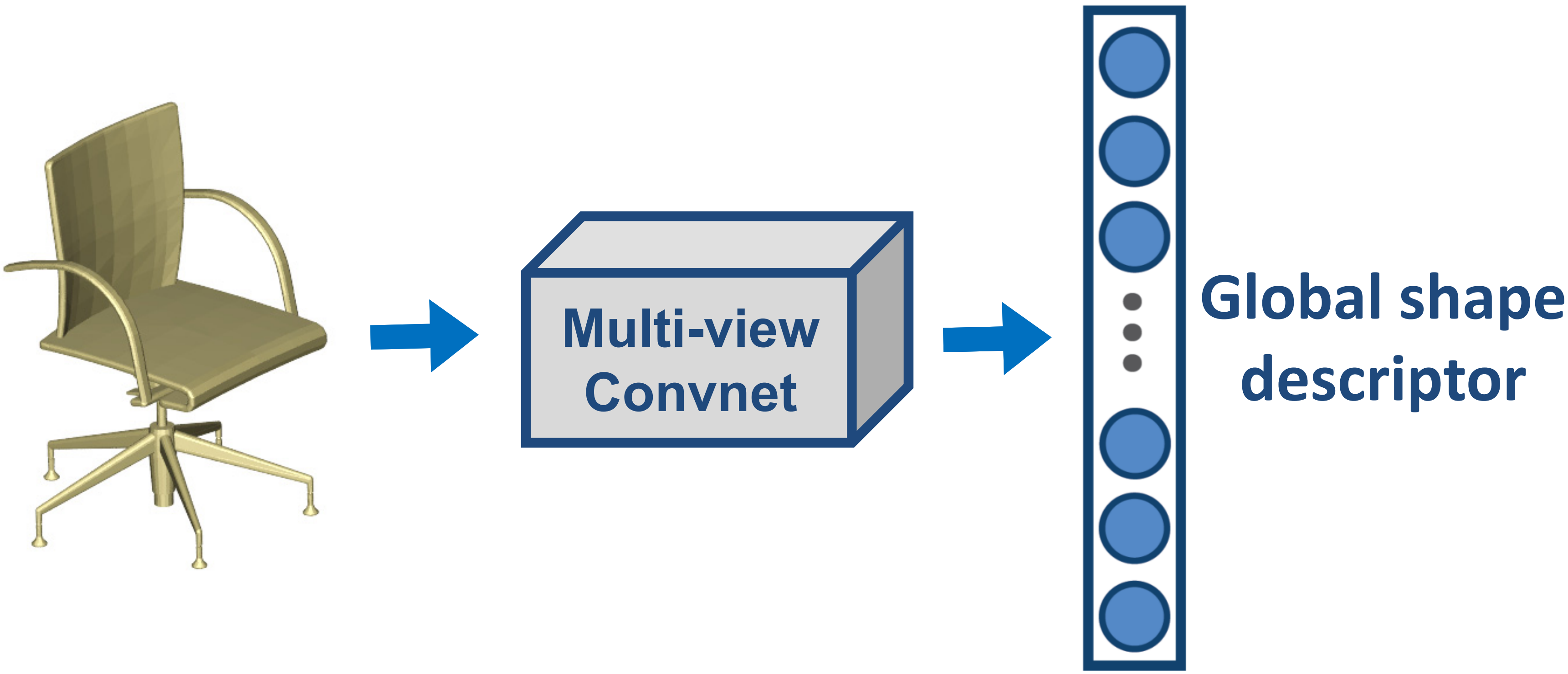


Object scans from “A Large Dataset of Object Scans” Choi et al. 2016

Outline

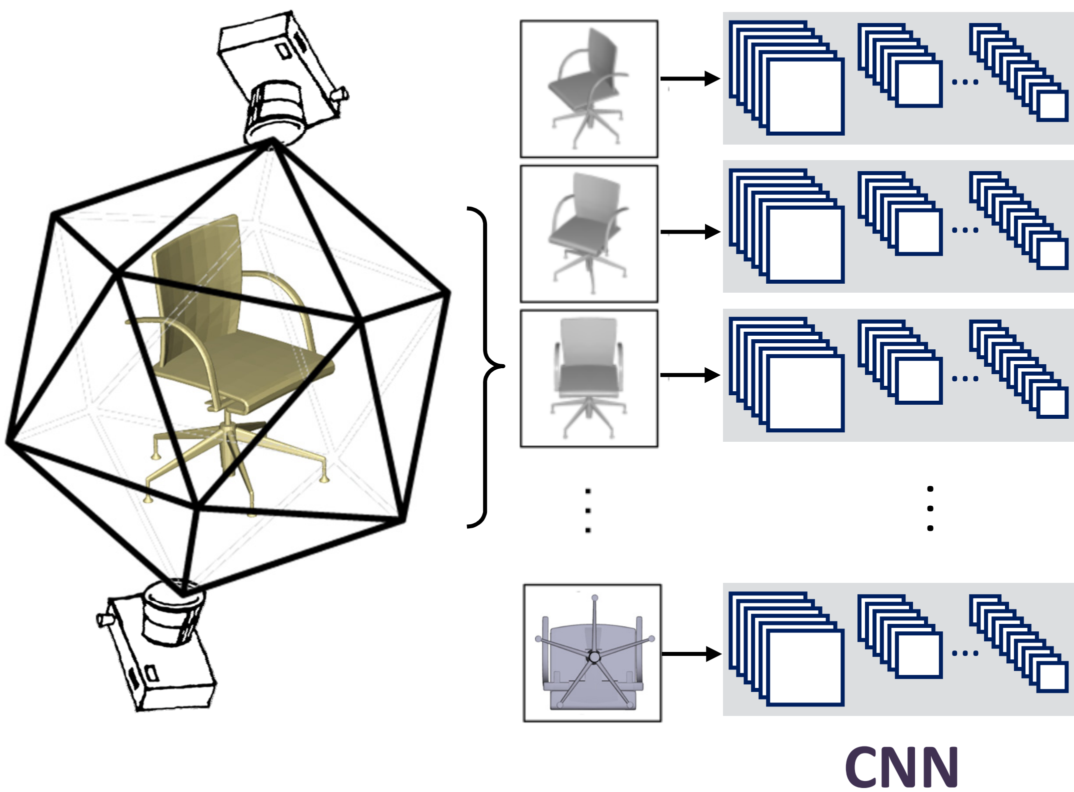
1. **Multi-view convnets for 3D shape analysis**
   - **Shape Segmentation**
   - **Shape Classification & Retrieval**
   - **Shape Correspondences**
2. **Multi-view convnets for 3D shape synthesis**
3. **Discussion / Future work**

Goal

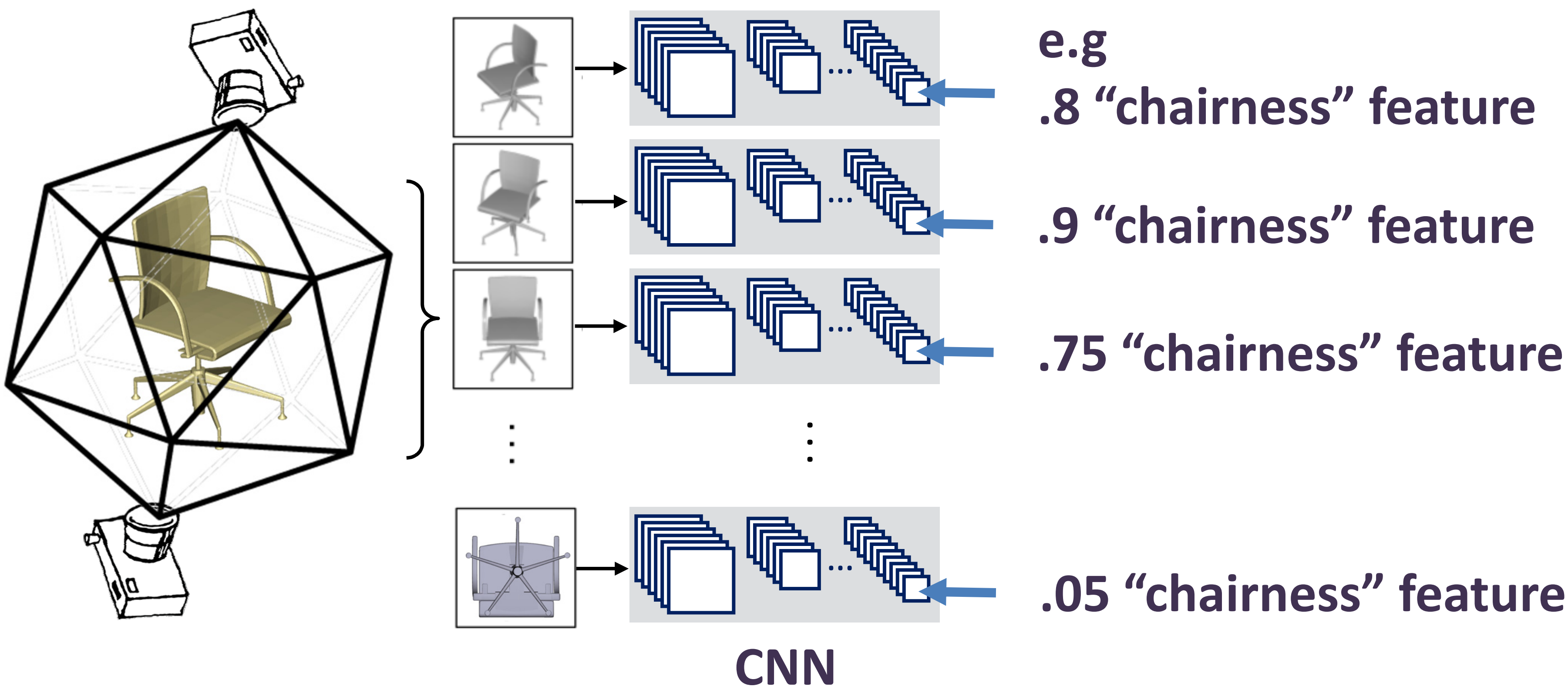


Su, Maji, Kalogerakis, Learned-Miller, ICCV 2015

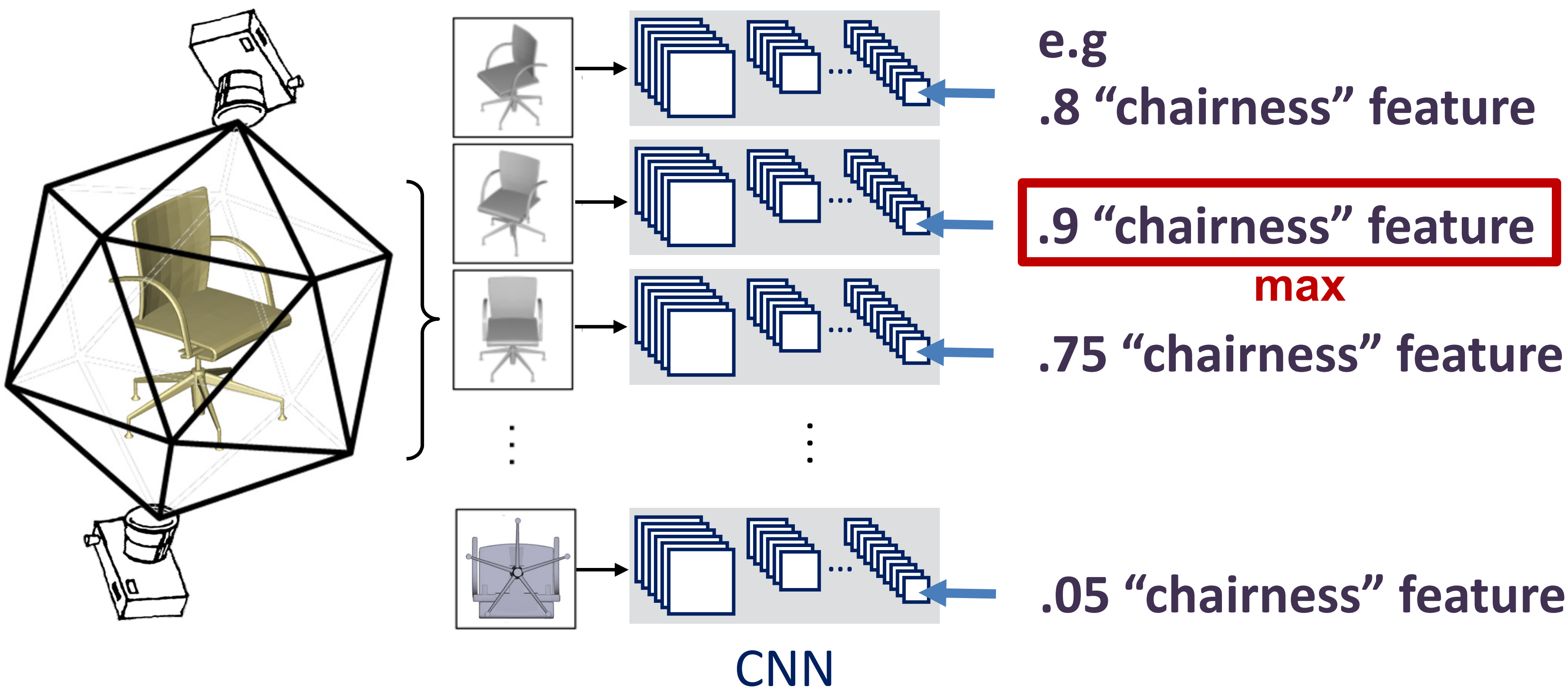
Shape recognition with multi-view CNNs



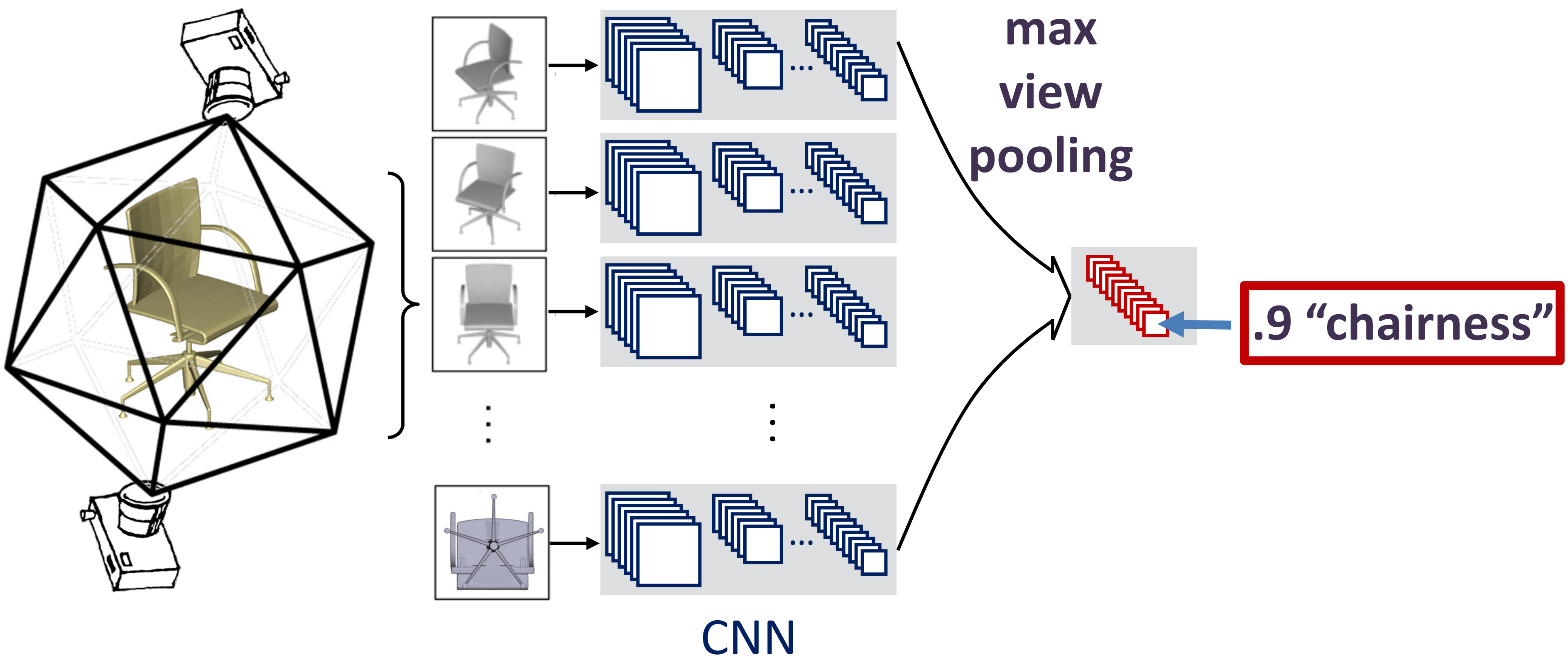
View Pooling



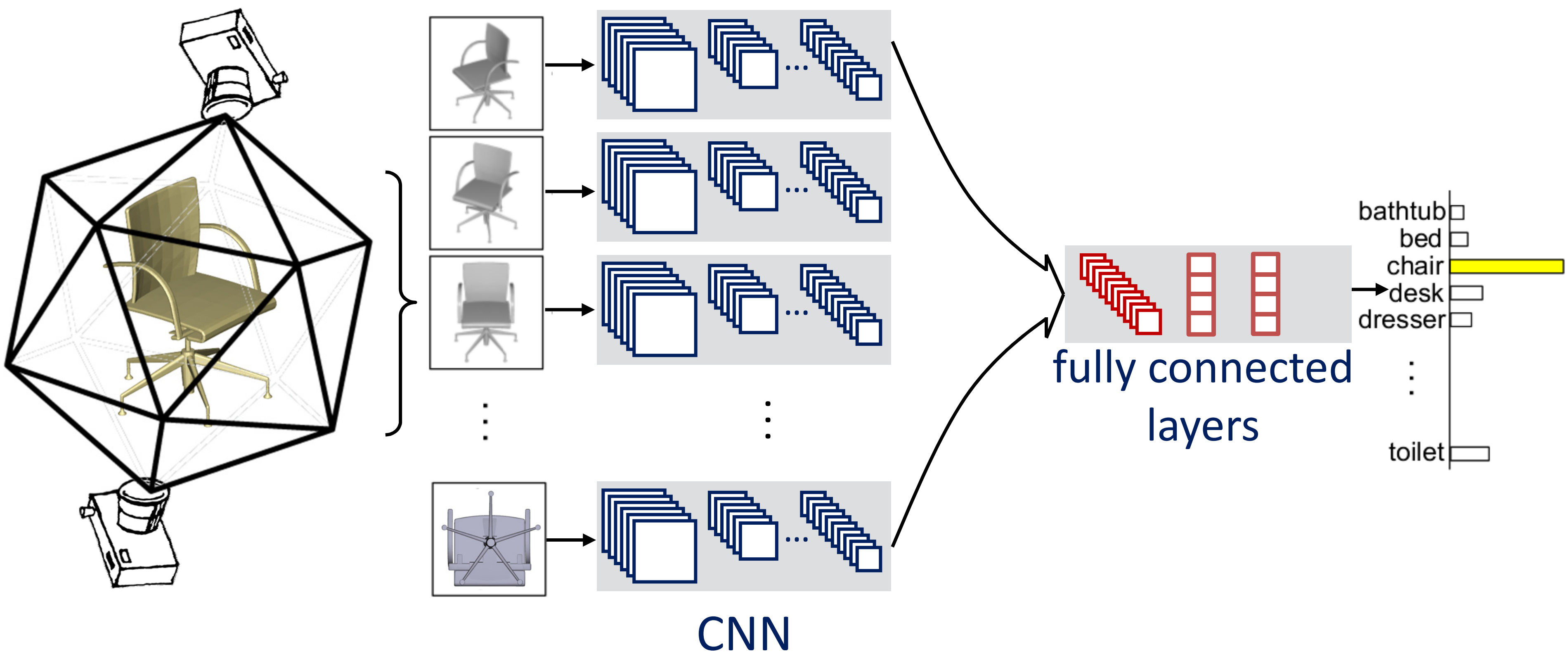
View Pooling



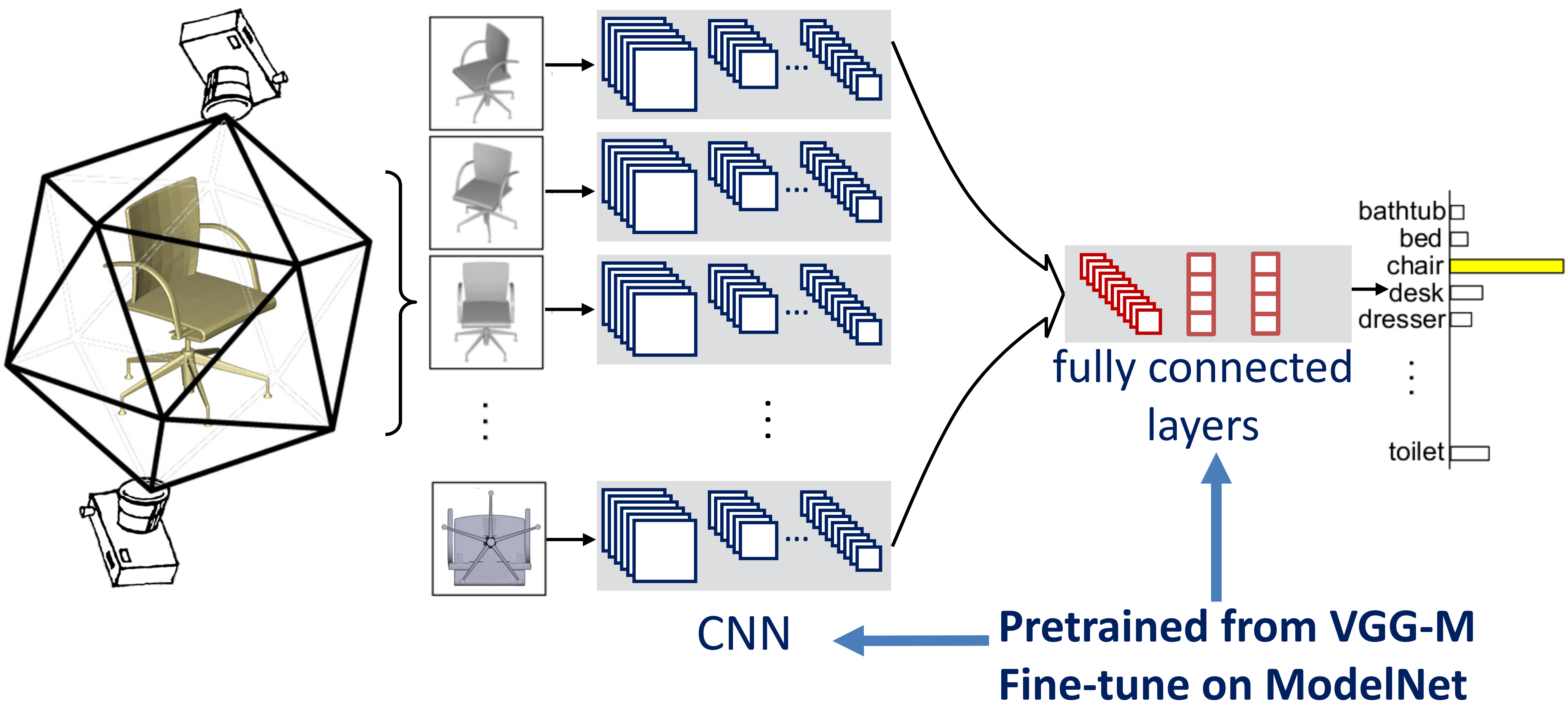
View Pooling



Classification

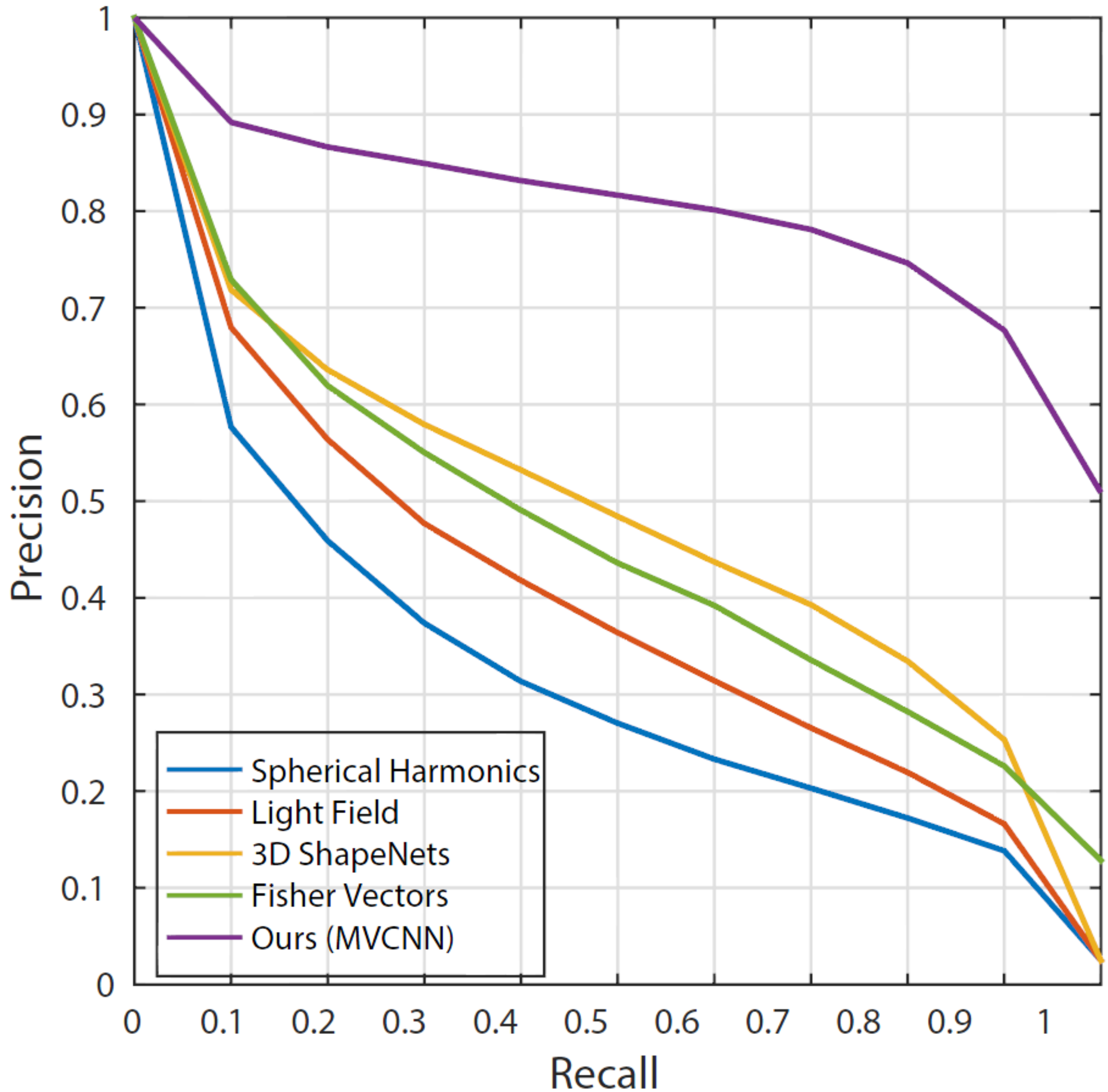


Training



bathtub
bed
chair
desk
dresser
toilet
fully connected layers
CNN
Pretrained from VGG-M
Fine-tune on ModelNet

ModelNet40: Classification & Retrieval



ModelNet40: Classification & Retrieval

Method	Classification (Accuracy)	Retrieval (mAP)
Spherical Harmonics **[Kazhdan et al.]**	68.2%	33.3%
LightField **[Chen et al.]**	75.5%	40.9%
Volumetric Net **[Wu et al.]**	77.3%	49.2%
ImageNet-trained CNN (VGG-M, 1 view)	83.0%	44.1%
Multi-view convnet (MVCNN)	**90.1%**	**79.5%**

ModelNet40: Classification & Retrieval

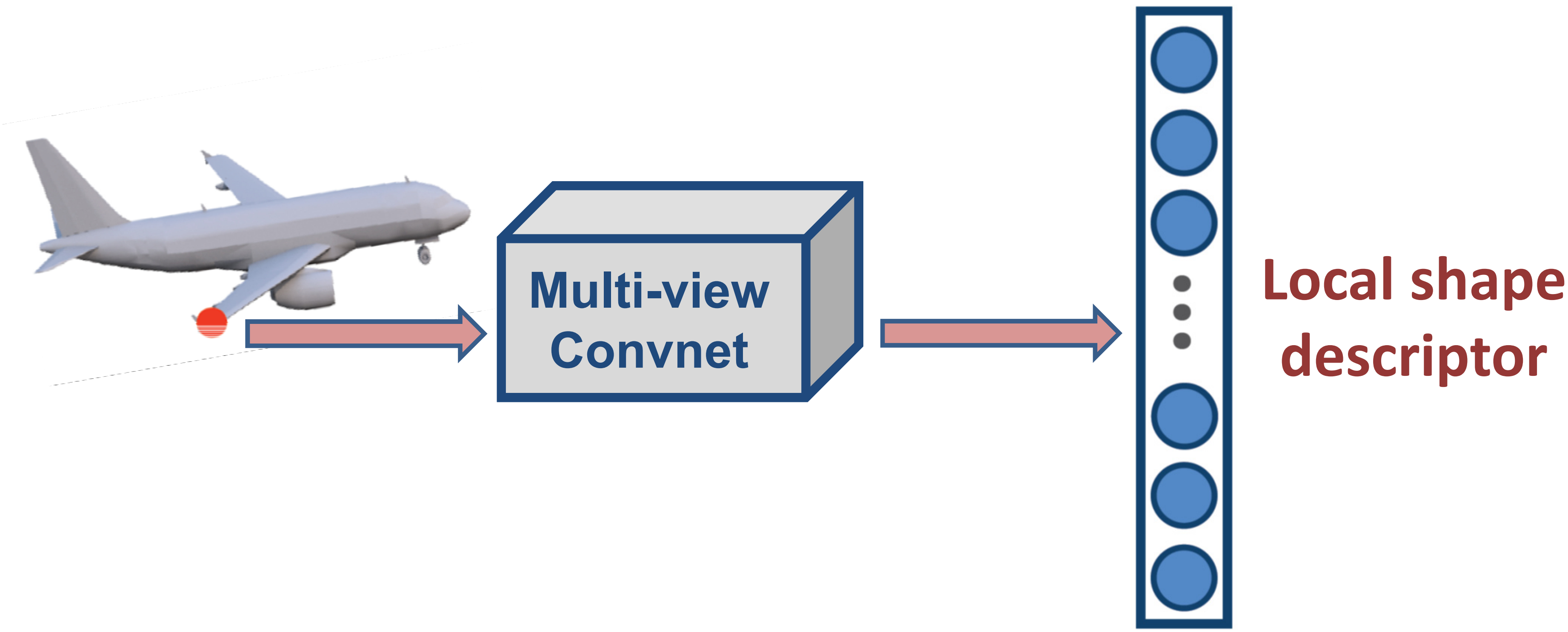
Method	Classification (Accuracy)	Retrieval (mAP)
Spherical Harmonics **[Kazhdan et al.]**	68.2%	33.3%
LightField **[Chen et al.]**	75.5%	40.9%
Volumetric Net **[Wu et al.]**	77.3%	49.2%
ImageNet-trained CNN (VGG-M, 1 view)	83.0%	44.1%
Multi-view convnet (MVCNN)	**90.1%**	**79.5%**

Updates since 2015: new pooling strategies & using depth+normal renderings yield 93.8% classification accuracy for MVCNNs **[Wang et al. 17]** vs 91.3% for the best volumetric net **[Brock et al. 2016, no ensemble]**

Outline

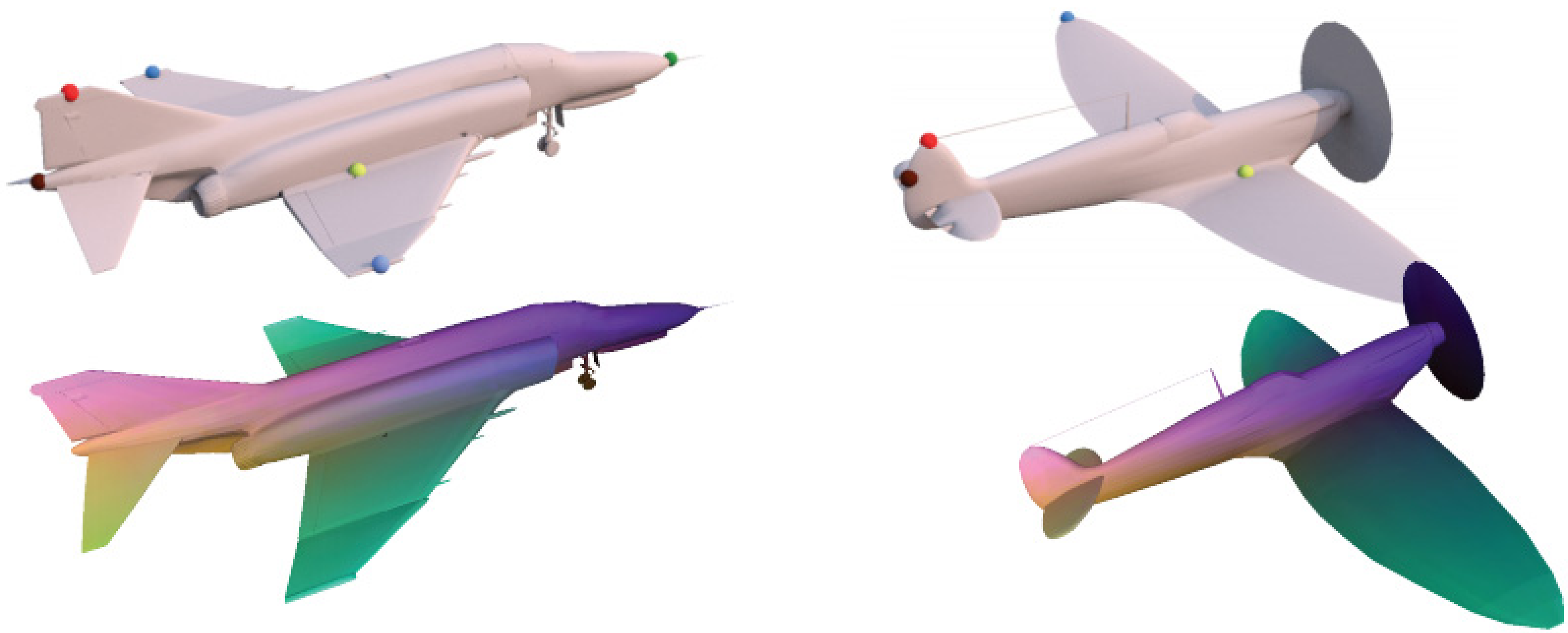
1. **Multi-view convnets for 3D shape analysis**
   - **Shape Segmentation**
   - **Shape Classification & Retrieval**
   - **Shape Correspondences**
2. **Multi-view convnets for 3D shape synthesis**
3. **Discussion / Future work**

Goal



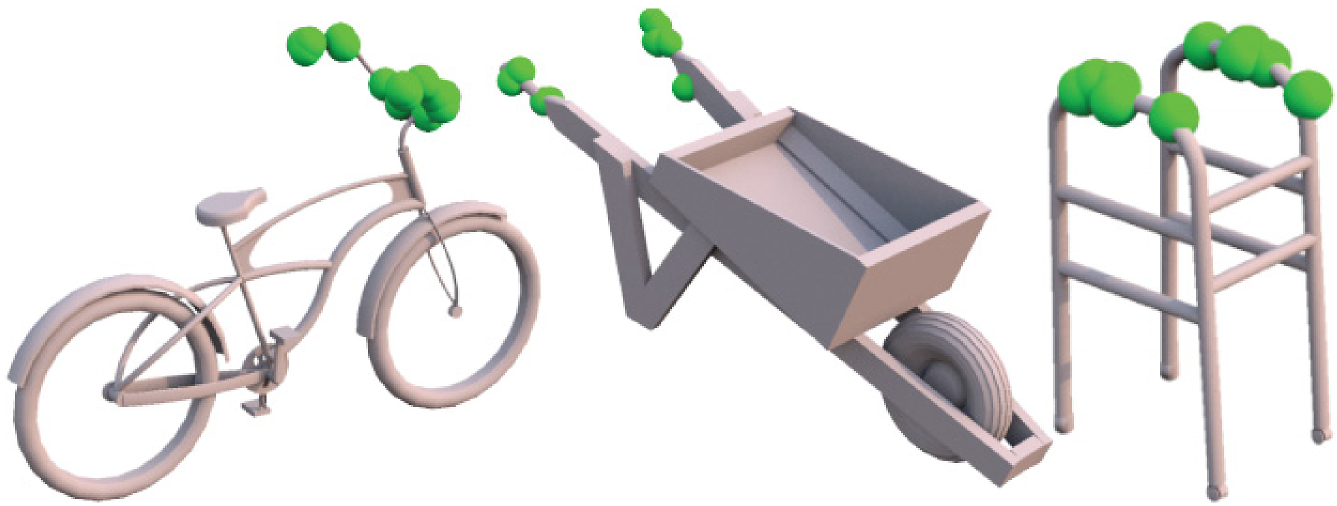
Huang, Kalogerakis, Chaudhuri, Ceylan, Kim, Yumer (TOG, to appear)

Applications of local descriptors: keypoint prediction & correspondences



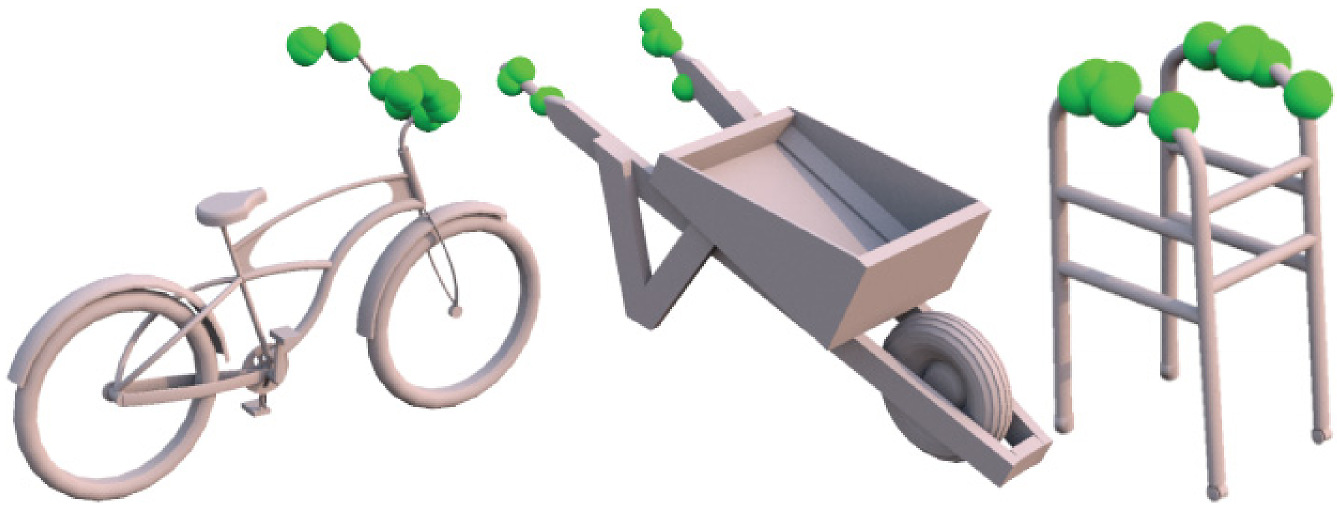
(similar colors correspond to points with similar descriptors)

Applications of local descriptors: affordance prediction

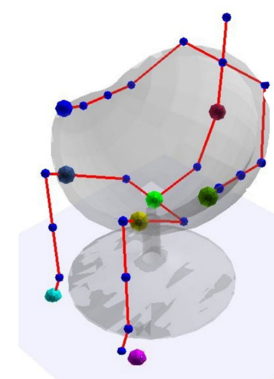


Where humans tend to place their palms
when they interact with these objects?

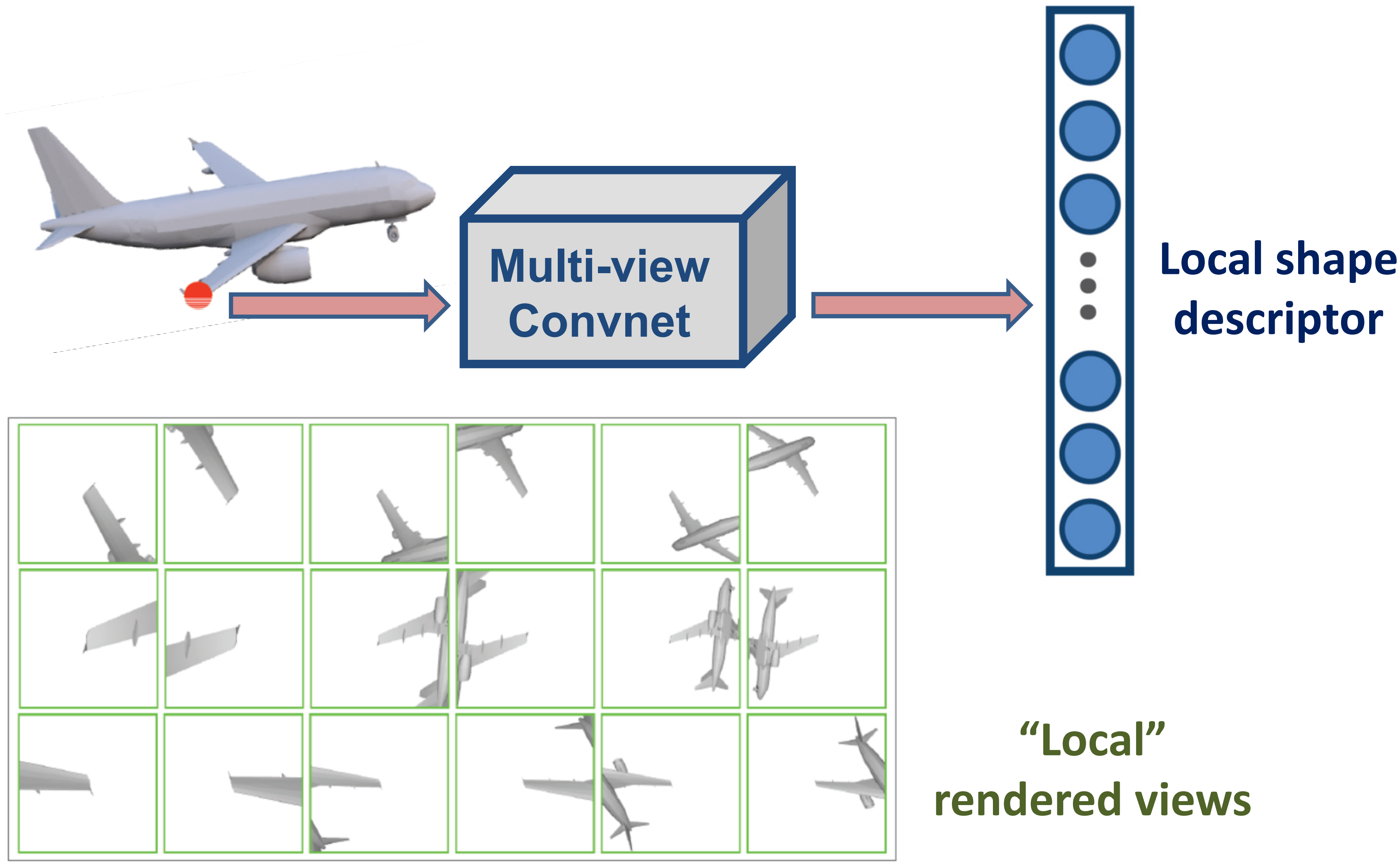
Applications of local descriptors: affordance prediction



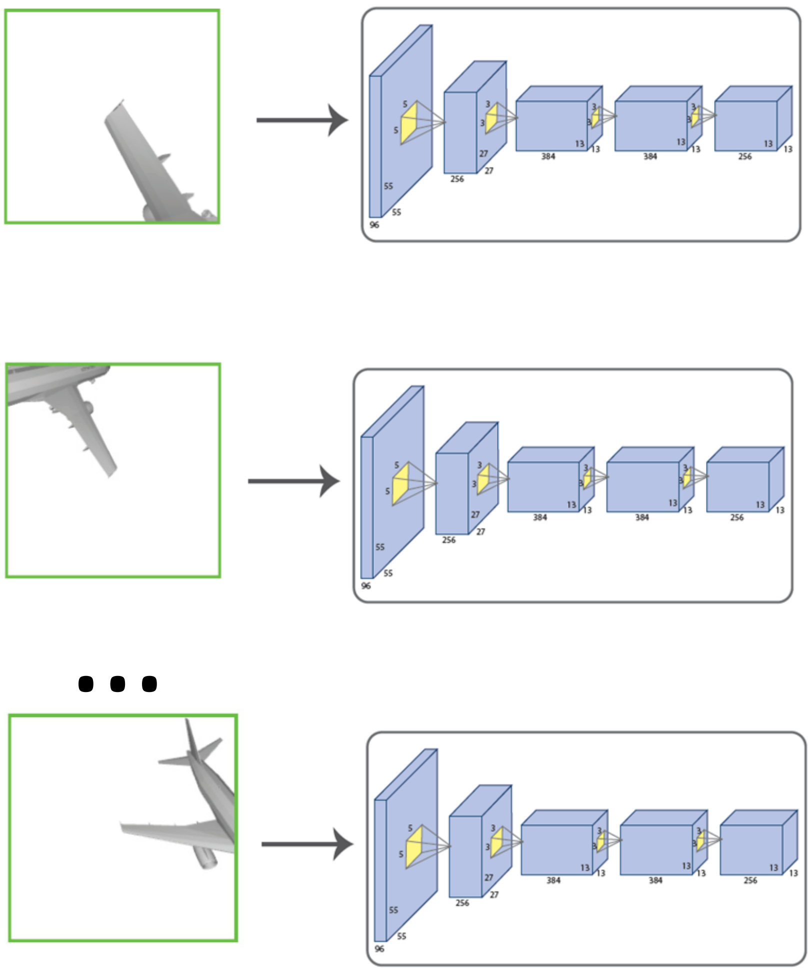
How would you place a human body relative to this object?



Goal

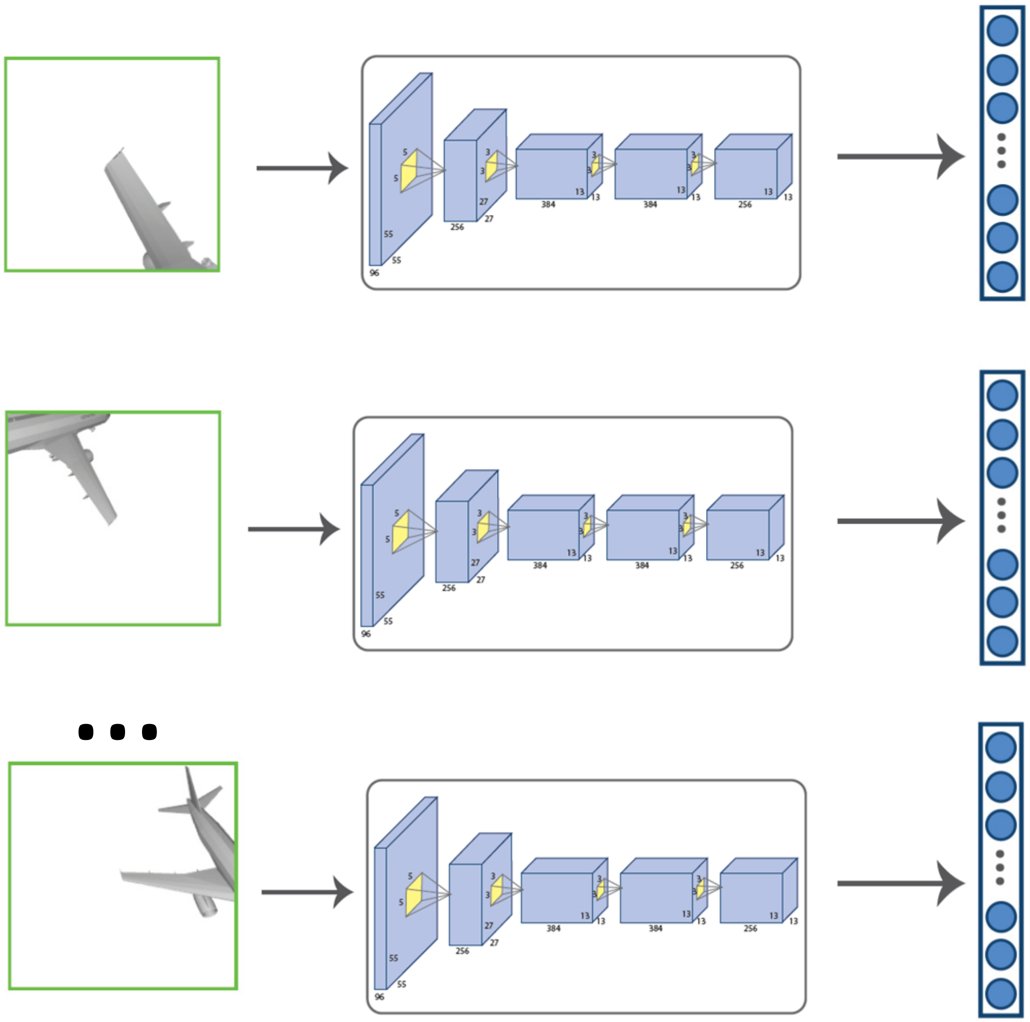

Multi-view
Convnet
Local shape
descriptor
“Local”
rendered views

Local MVCNN



CNN branches (shared parameters)

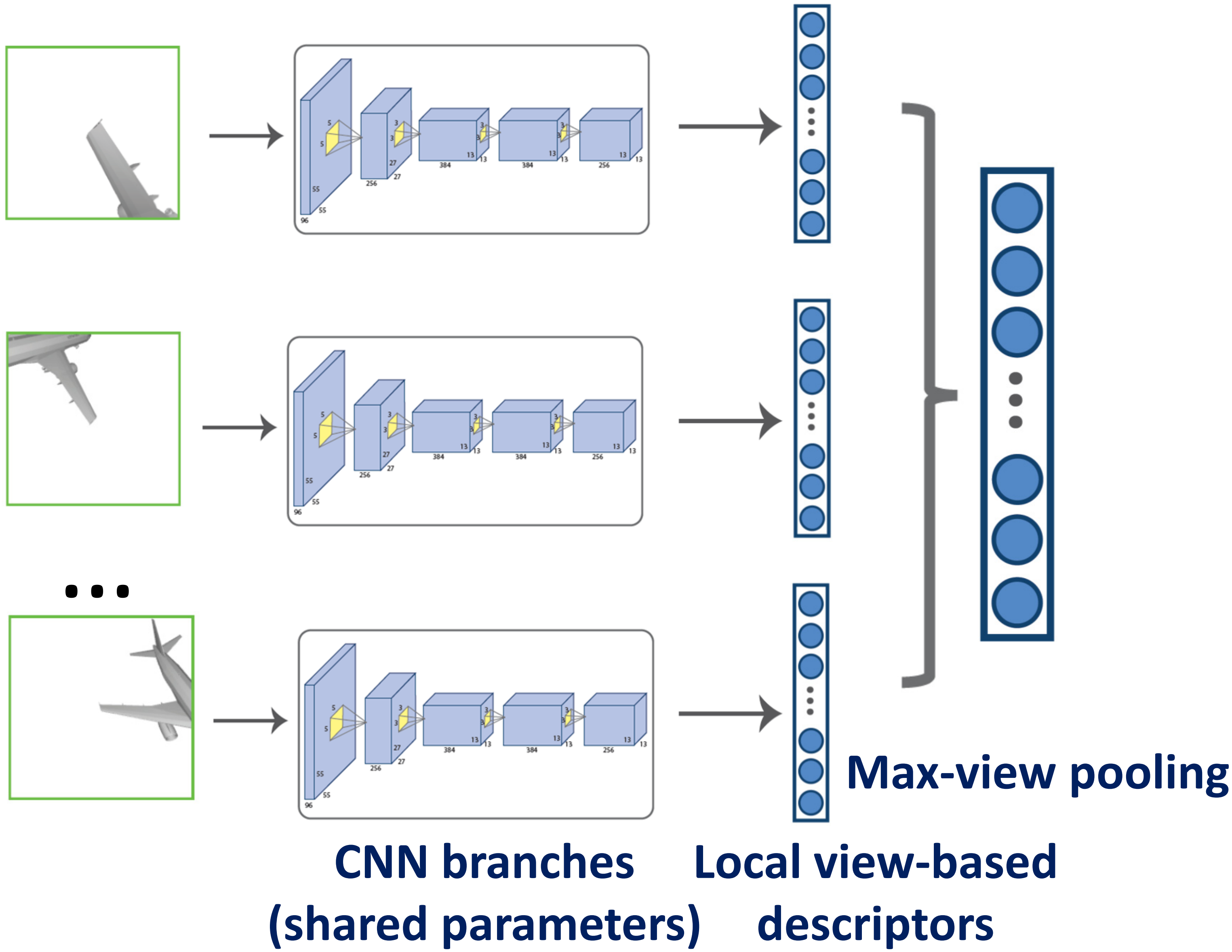
Local MVCNN



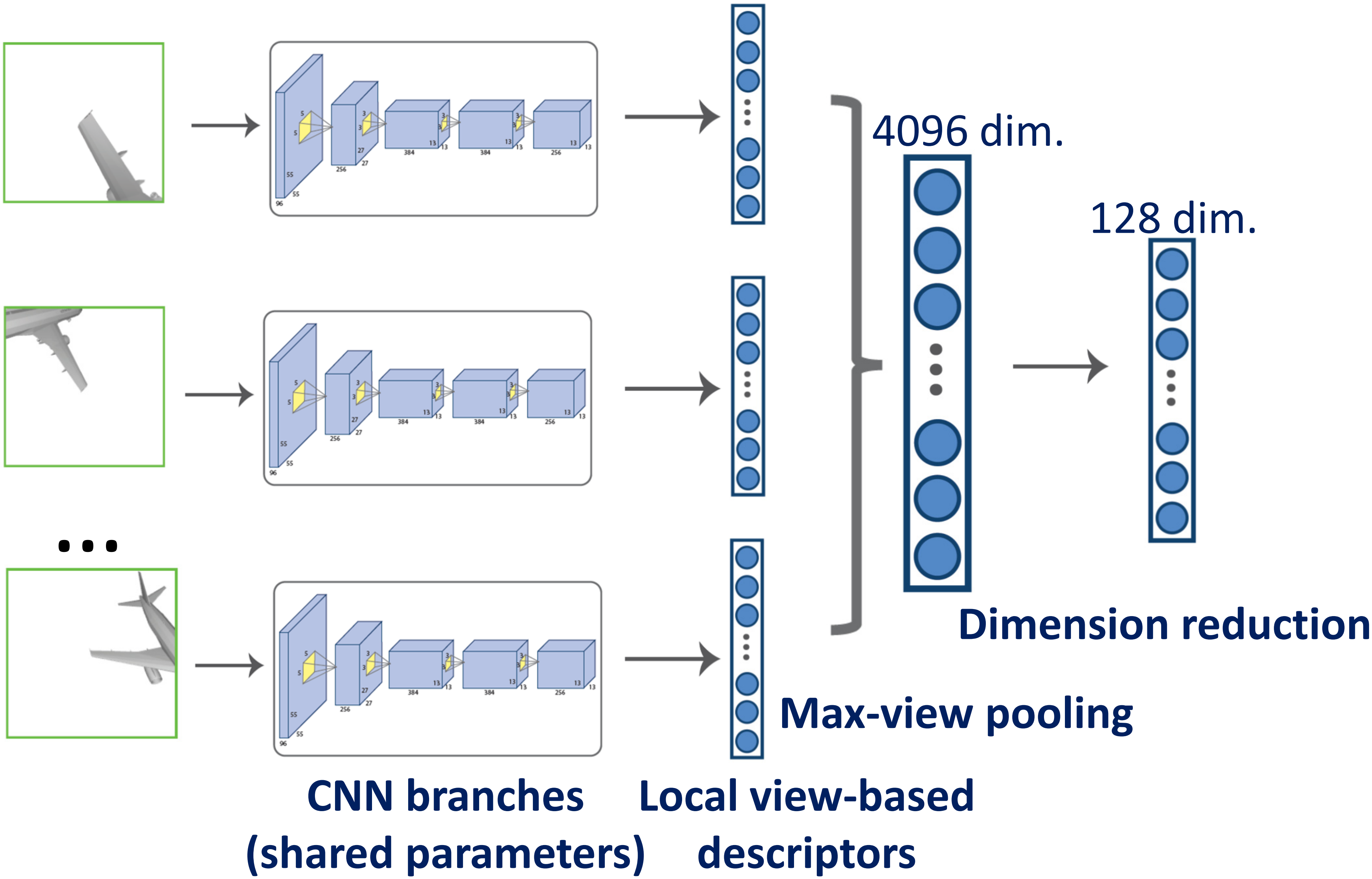
CNN branches (shared parameters)

Local view-based descriptors

Local MVCNN

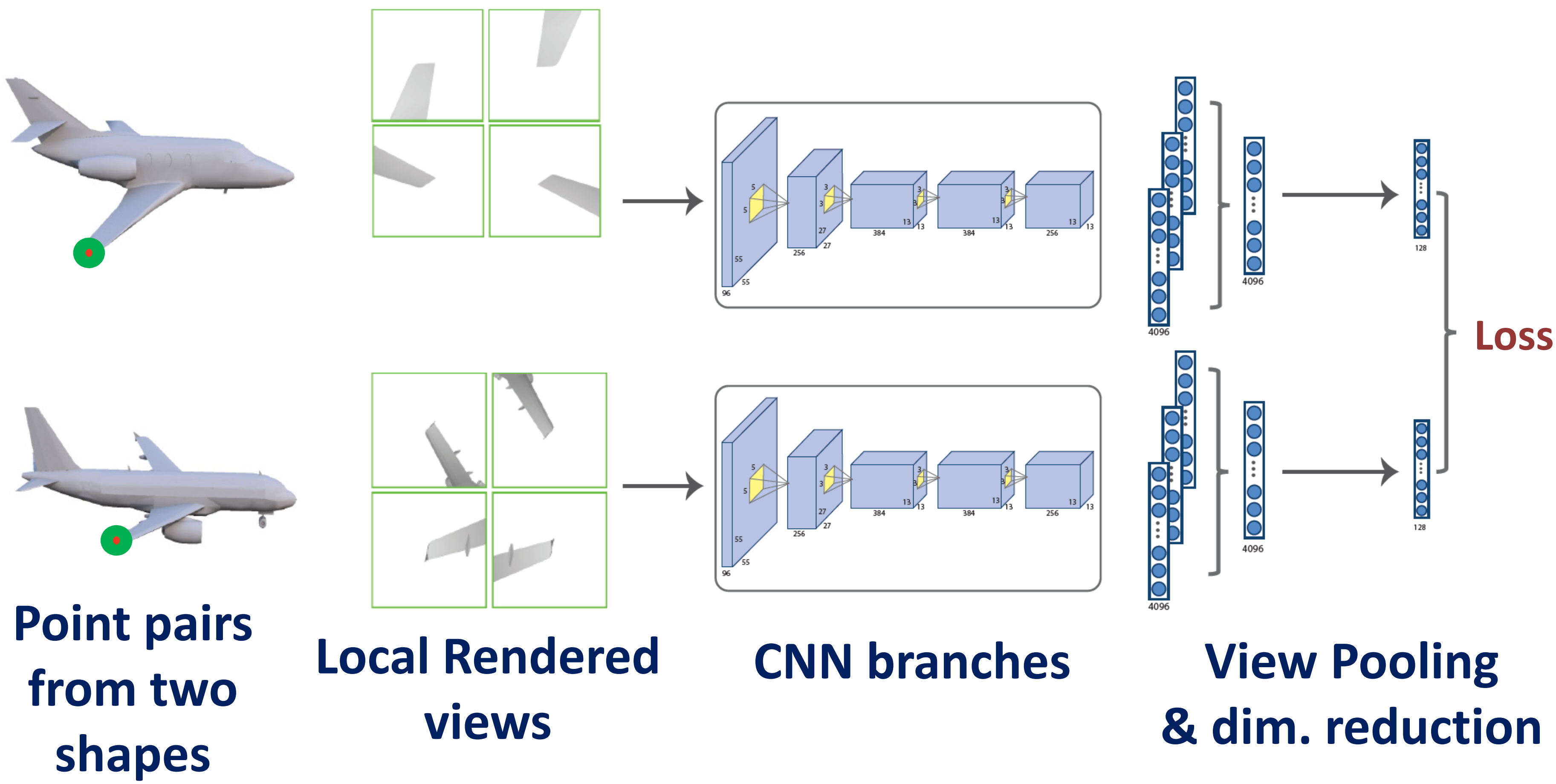

Max-view pooling
CNN branches
(shared parameters)
Local view-based
descriptors

Local MVCNN



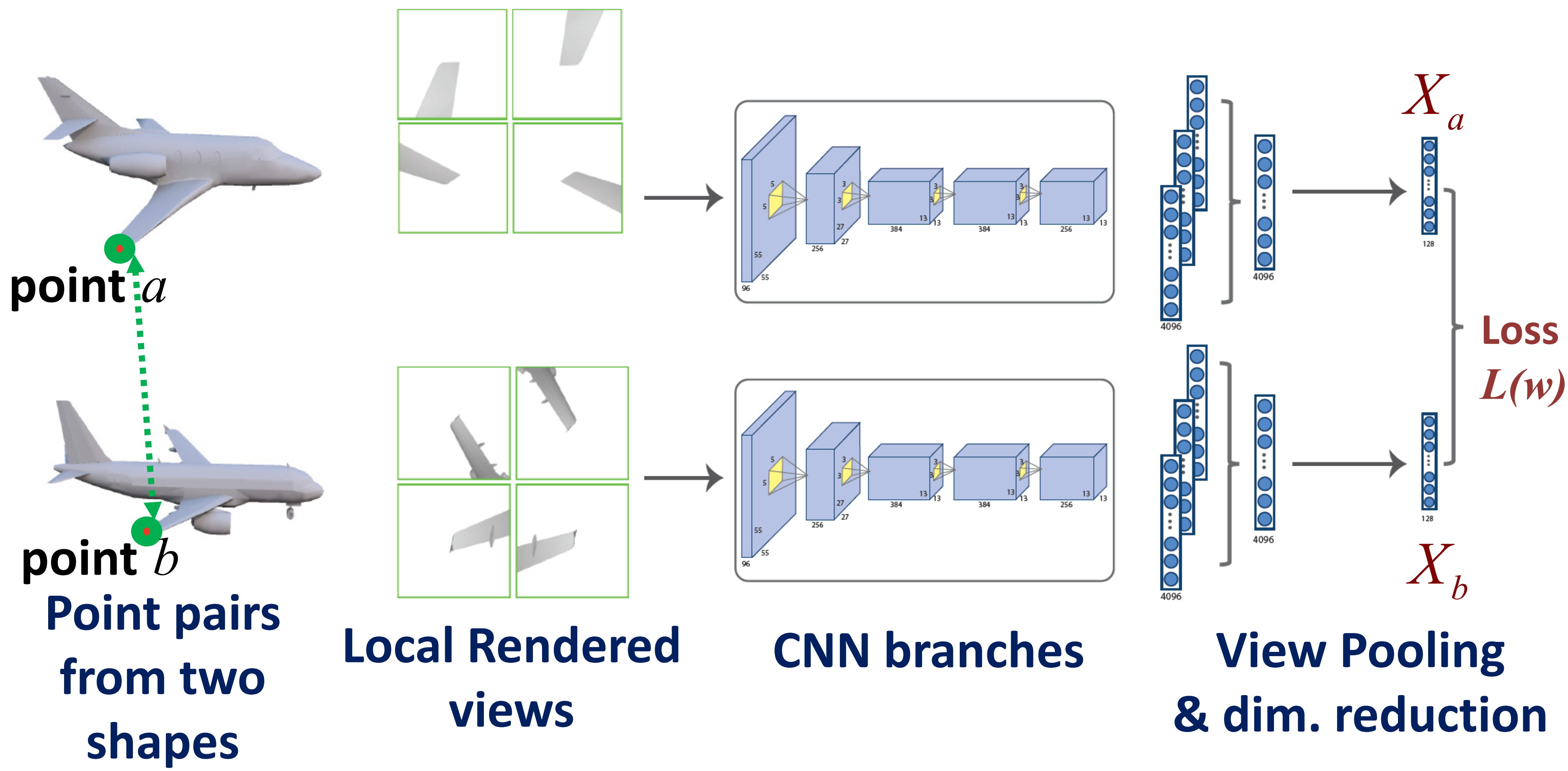
4096 dim.
128 dim.
Dimension reduction
Max-view pooling
CNN branches
(shared parameters)
Local view-based
descriptors

Training



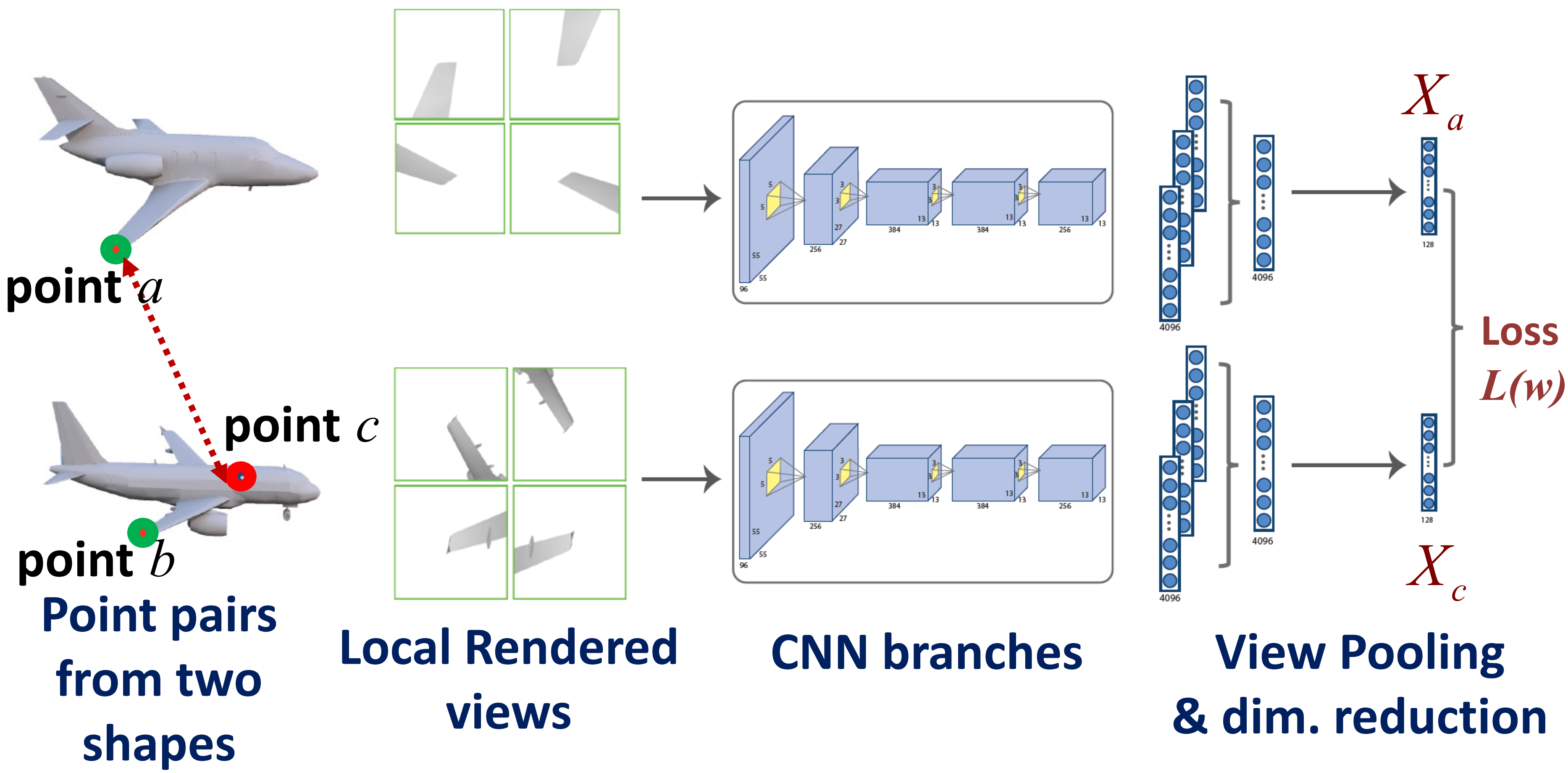
Loss
Point pairs from two shapes
Local Rendered views
CNN branches
View Pooling & dim. reduction

Training



$$L(w) = \sum_{\substack{\text{similar} \\ \text{point pairs} \\ \text{(a,b)}}} D^2(X_a, X_b)$$

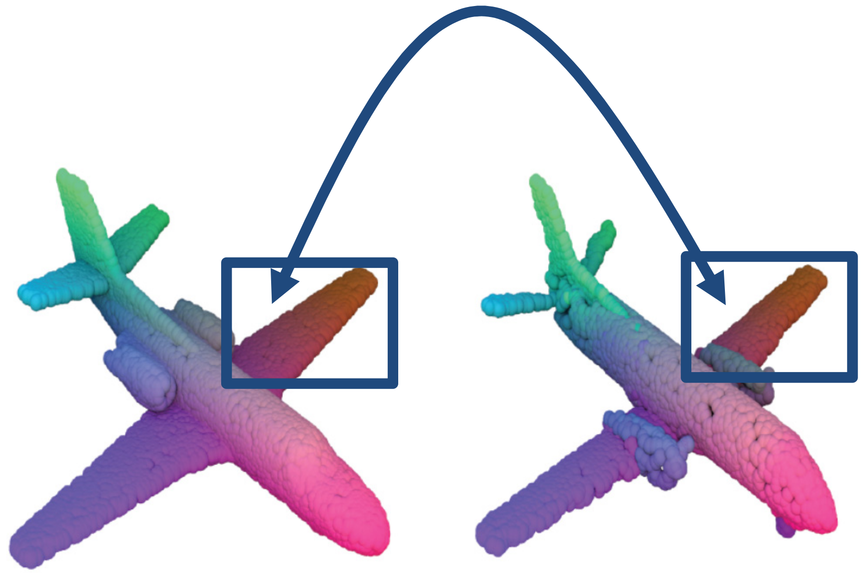
Training



$$L(w) = \sum_{\substack{\text{similar}\\\text{point pairs}\\\text{(a,b)}}} D^2(X_a, X_b) + \sum_{\substack{\text{dissimilar}\\\text{point pairs}\\\text{(a,c)}}} \max(\text{margin} - D(X_a, X_c), 0)^2$$

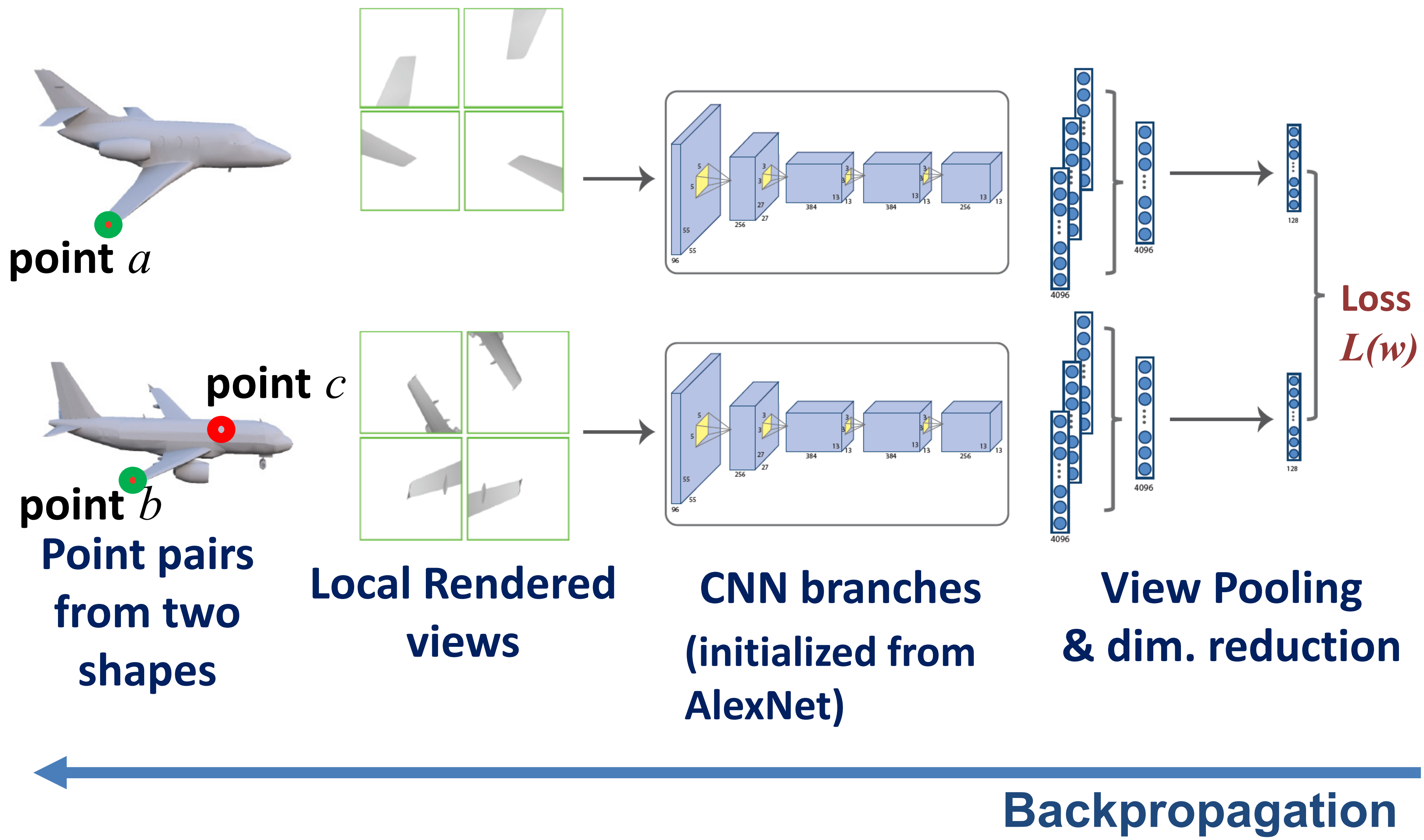
Training dataset generation

Non-rigid alignment **per part**
from segmented ShapeNetCore



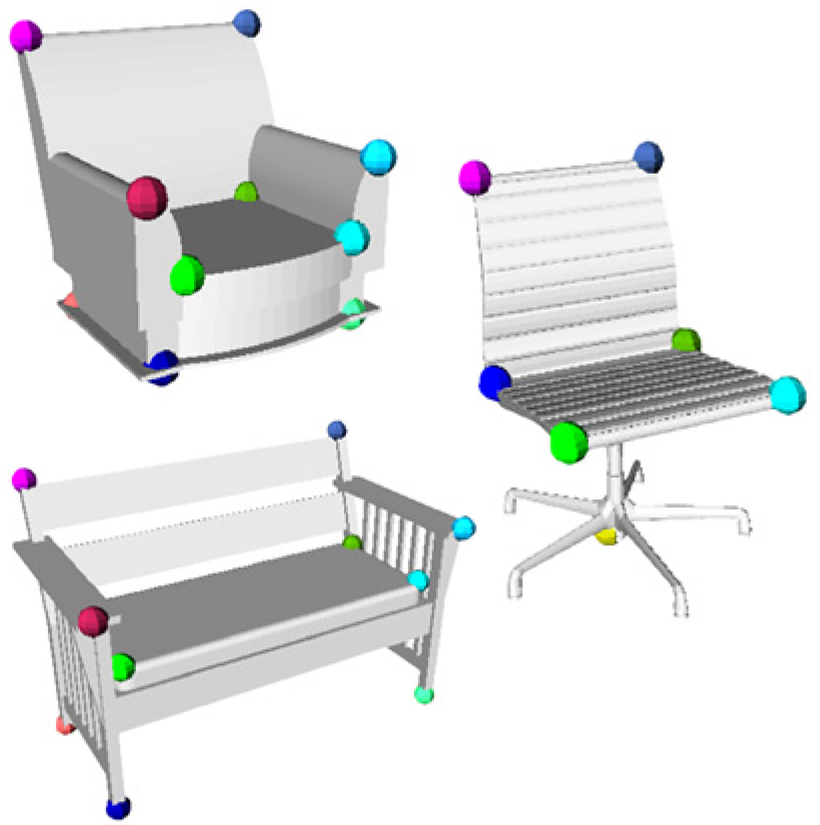
(corresponding points have same color)

Training



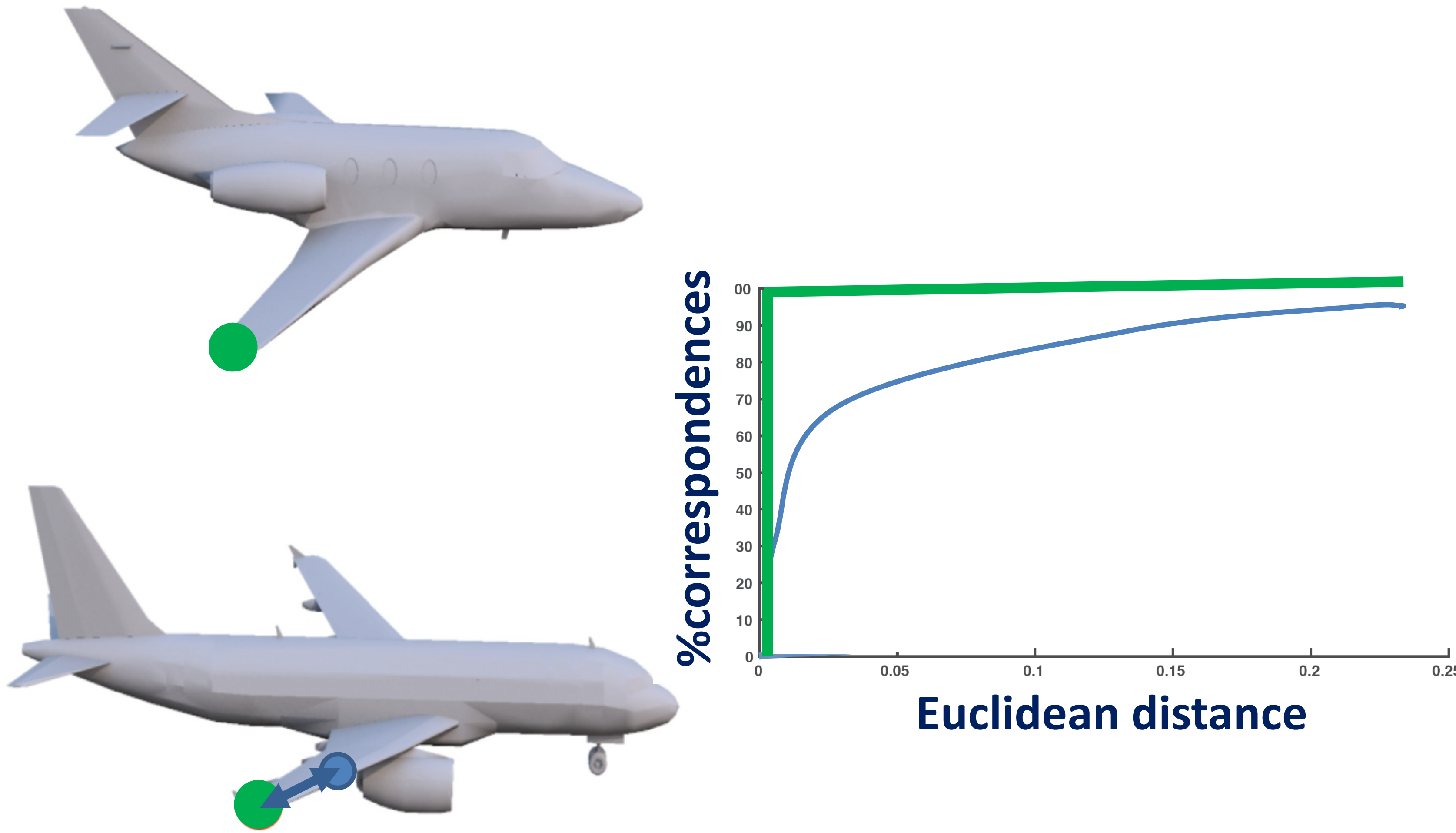
point a
point c
point b
Point pairs from two shapes
Local Rendered views
CNN branches (initialized from AlexNet)
View Pooling & dim. reduction
Loss $L(w)$
Backpropagation

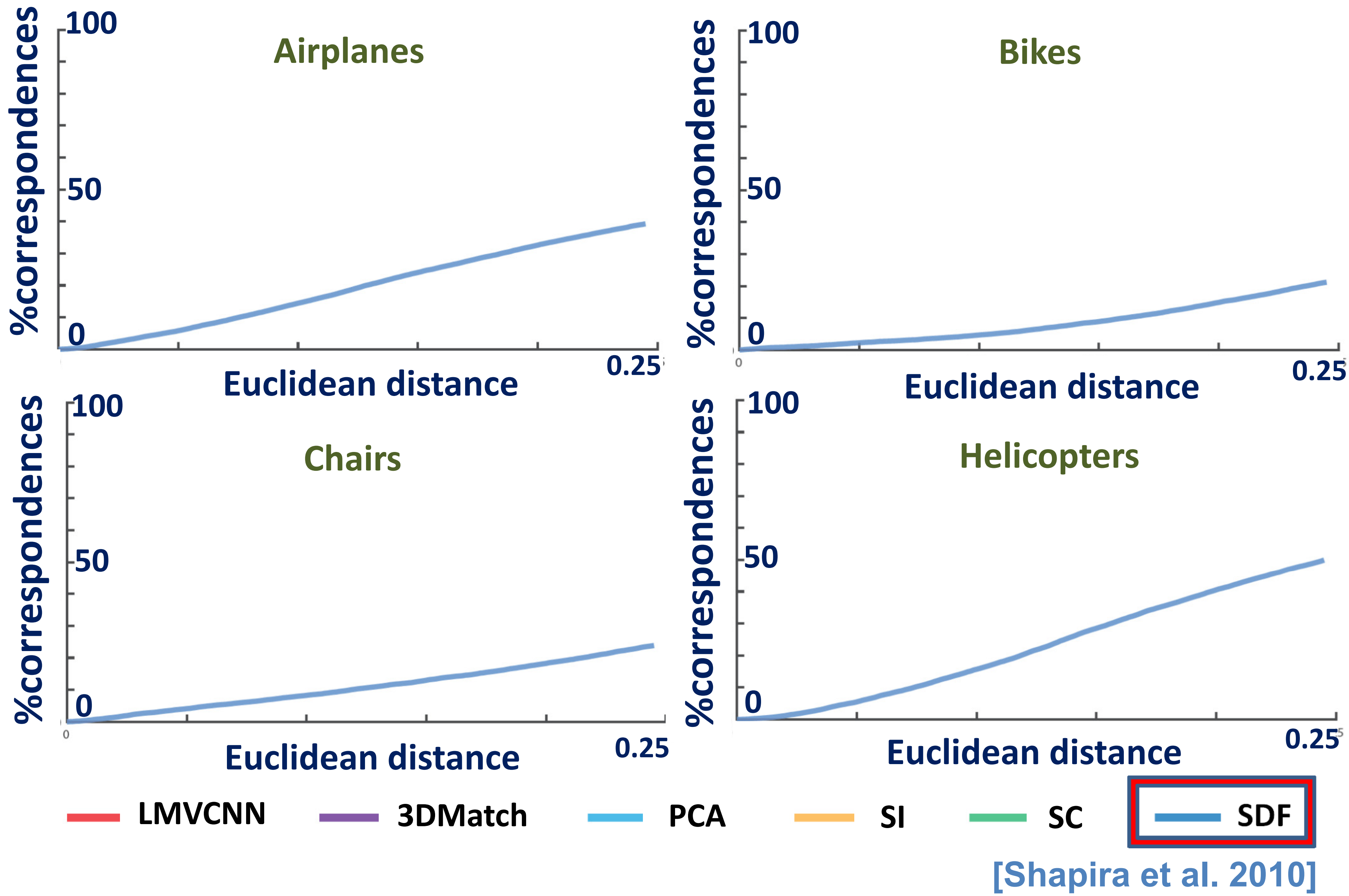
Evaluation: correspondence accuracy

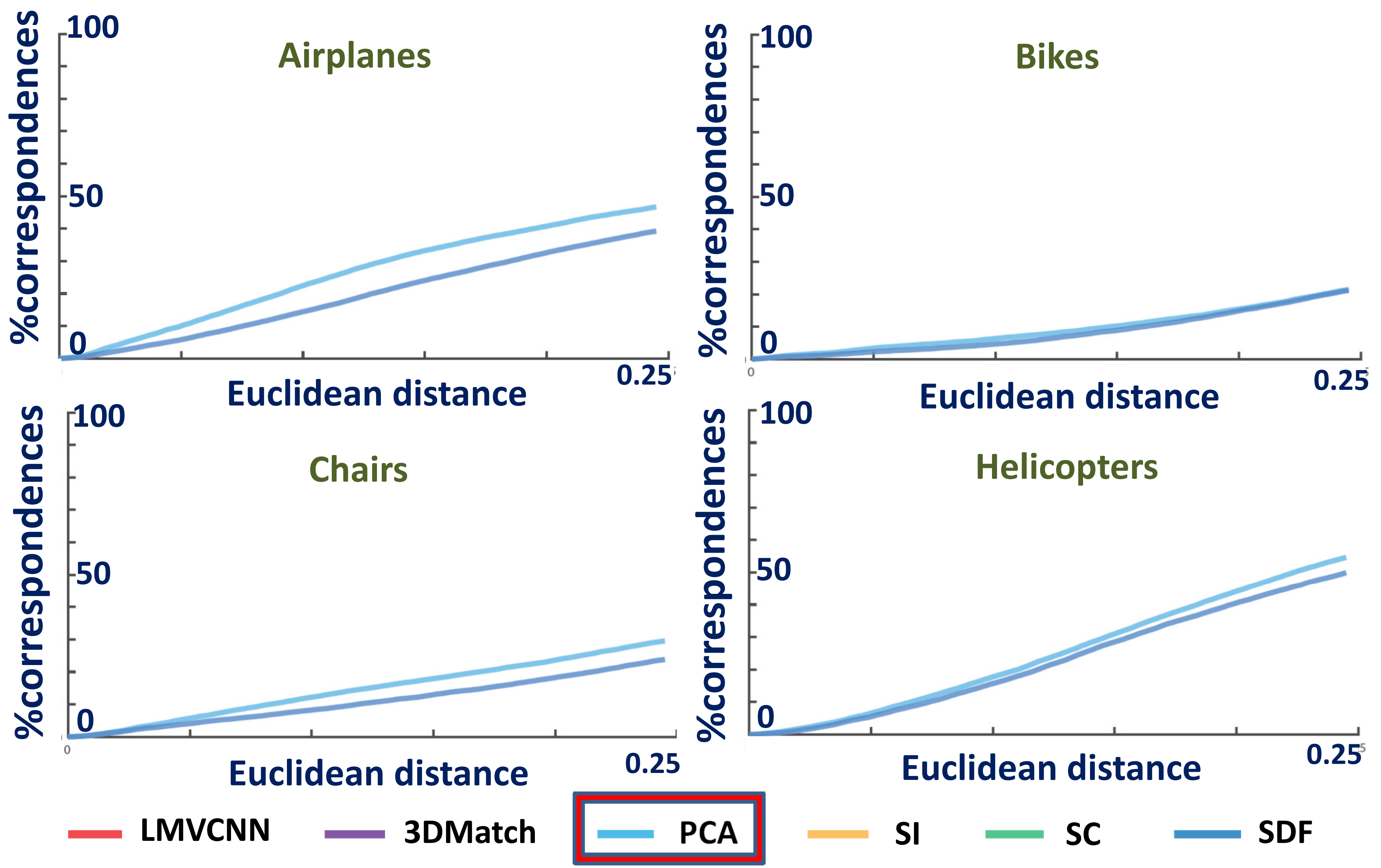


[Kim et al. 2013]

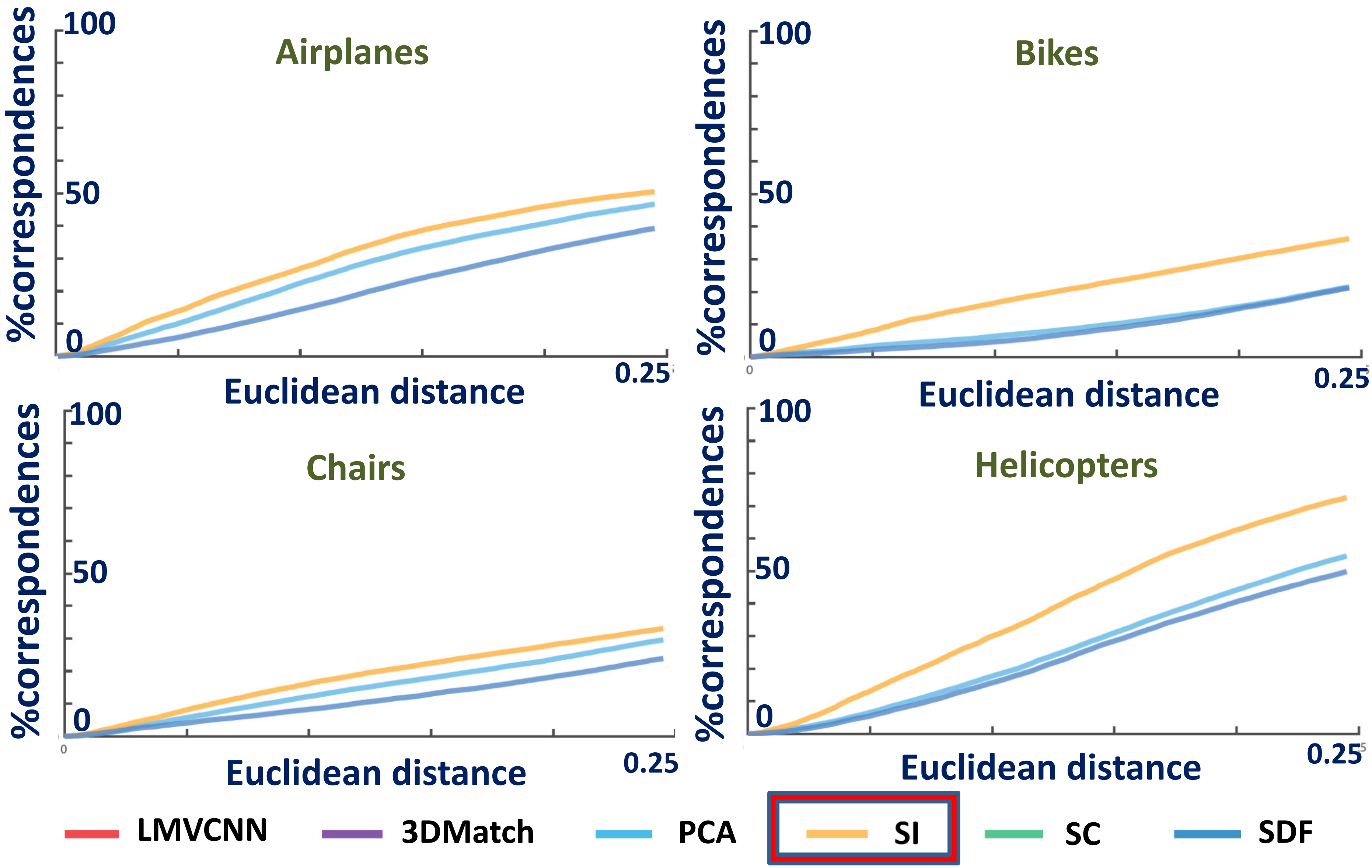
Evaluation: correspondence accuracy



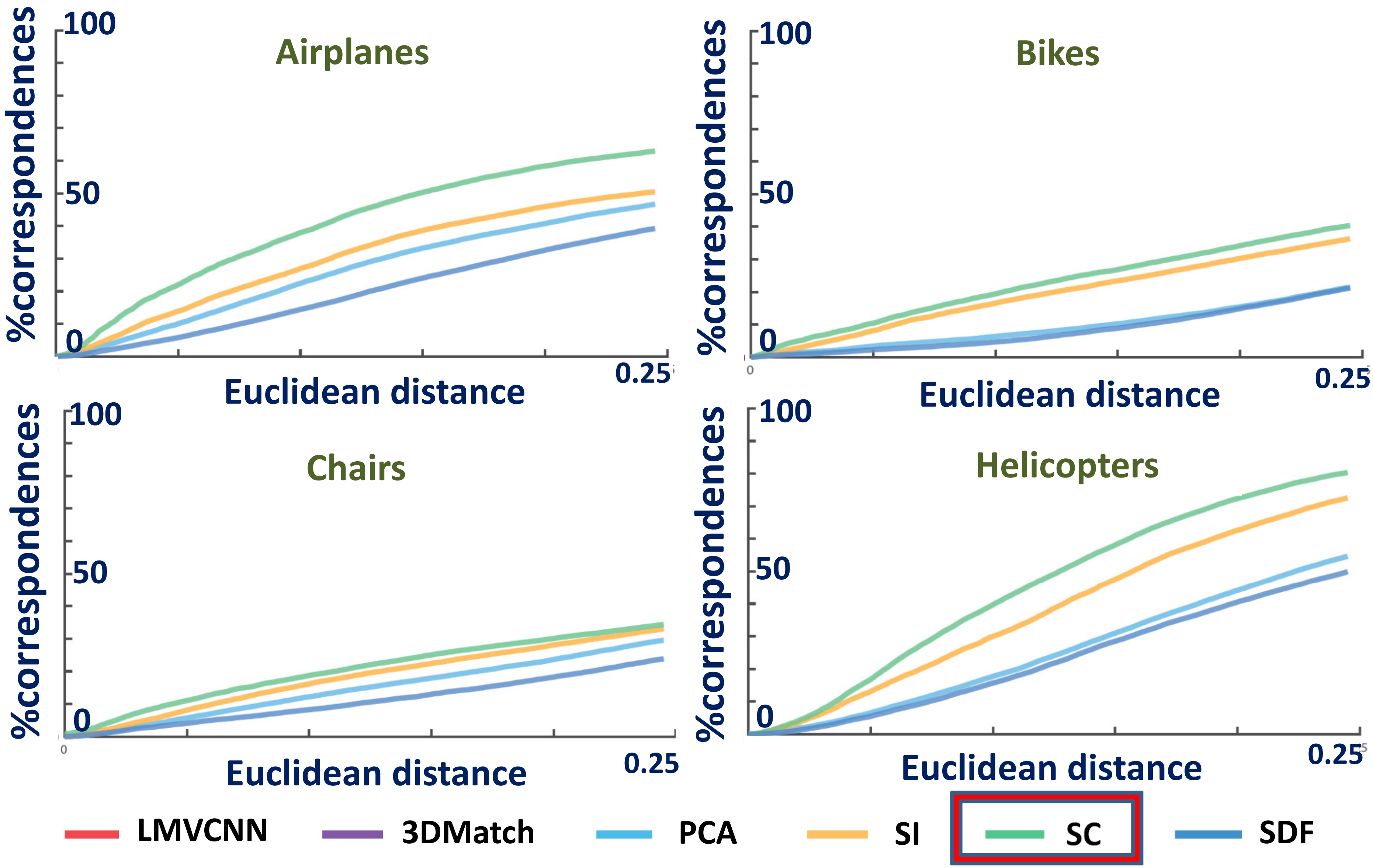

Airplanes
Bikes
Chairs
Helicopters
%correspondences
100
50
0
Euclidean distance
0.25
LMVCNN
3DMatch
PCA
SI
SC
SDF
[Shapira et al. 2010]



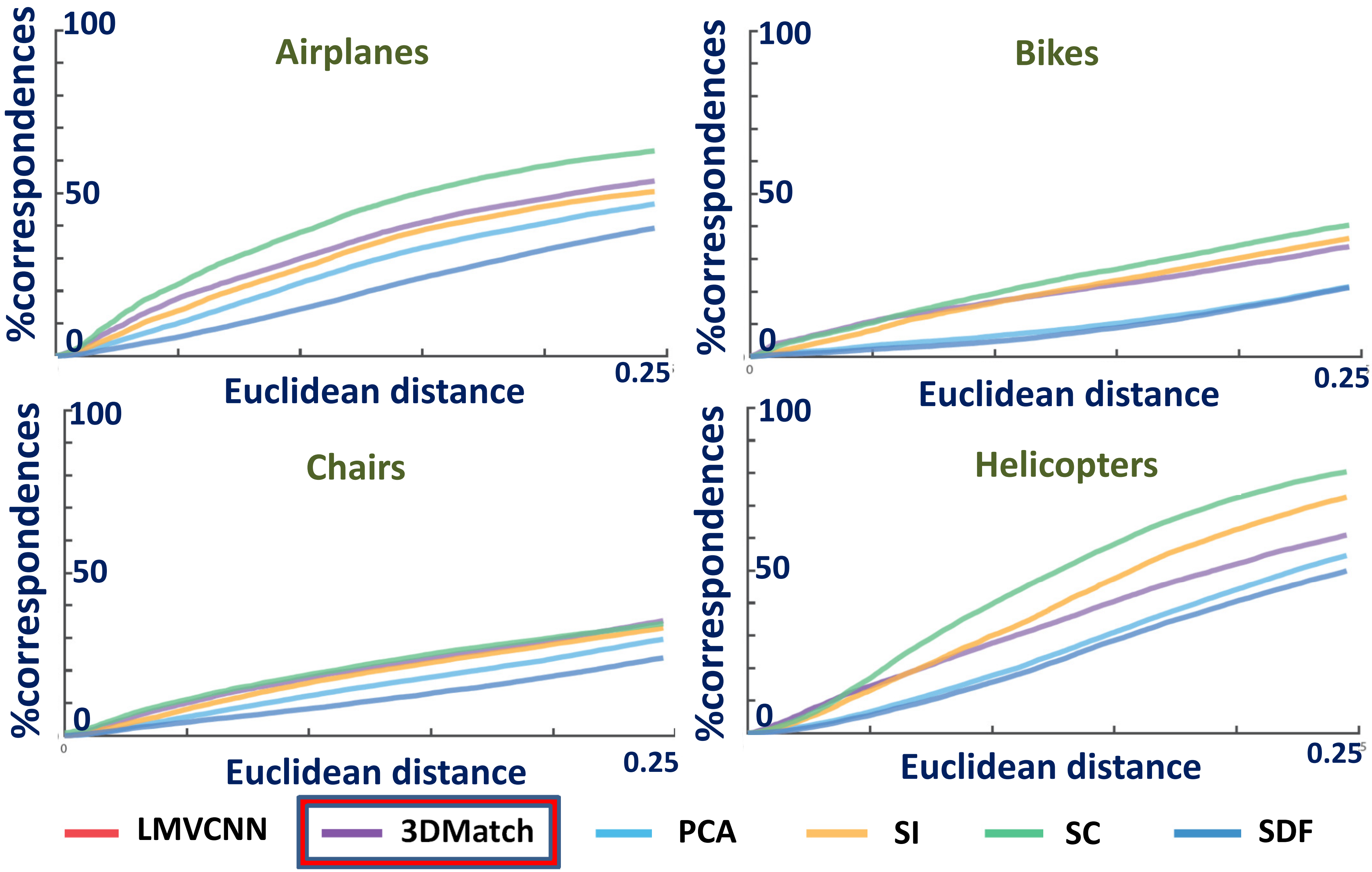
[Kalogerakis et al. 2010]



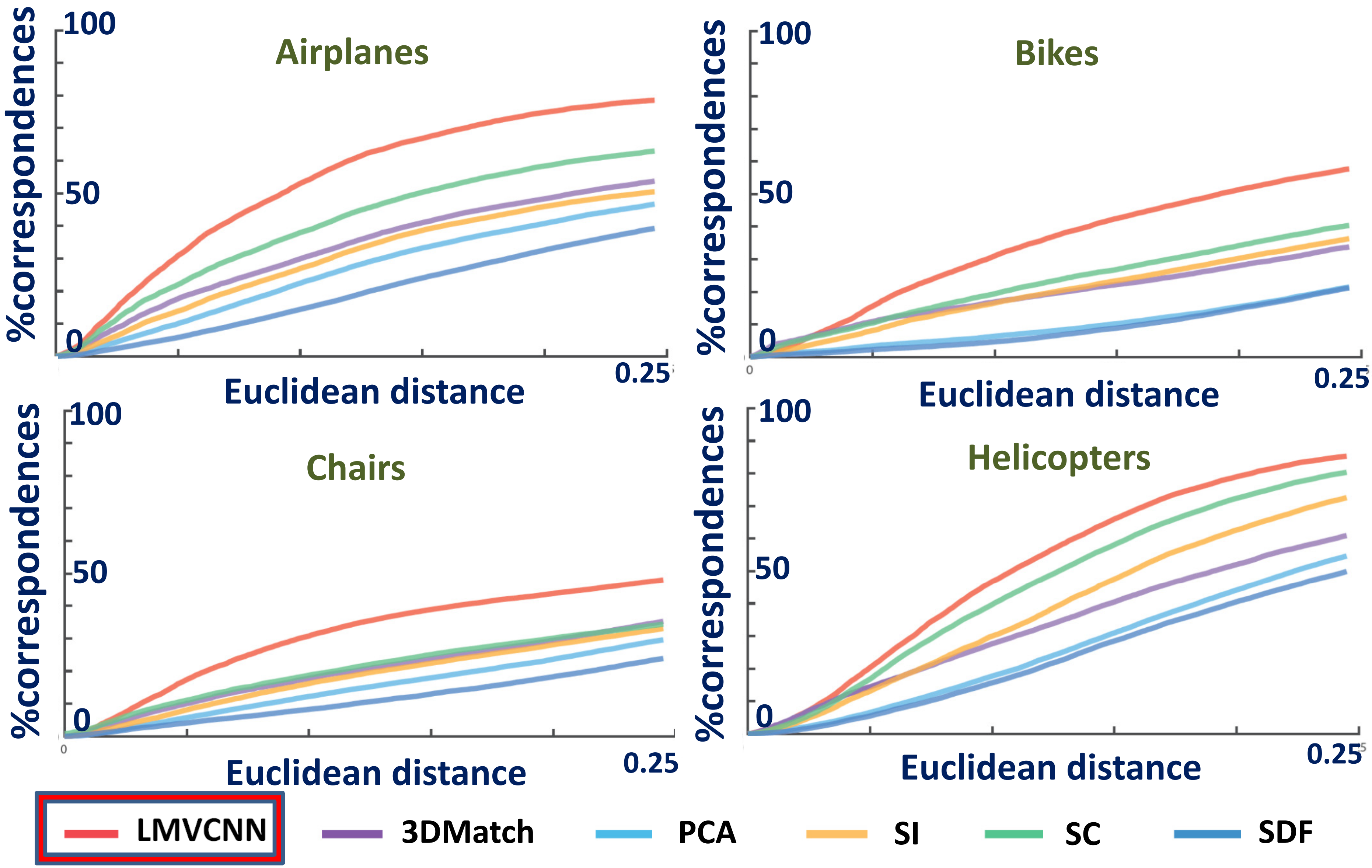
[Johnson and Hebert 1999]



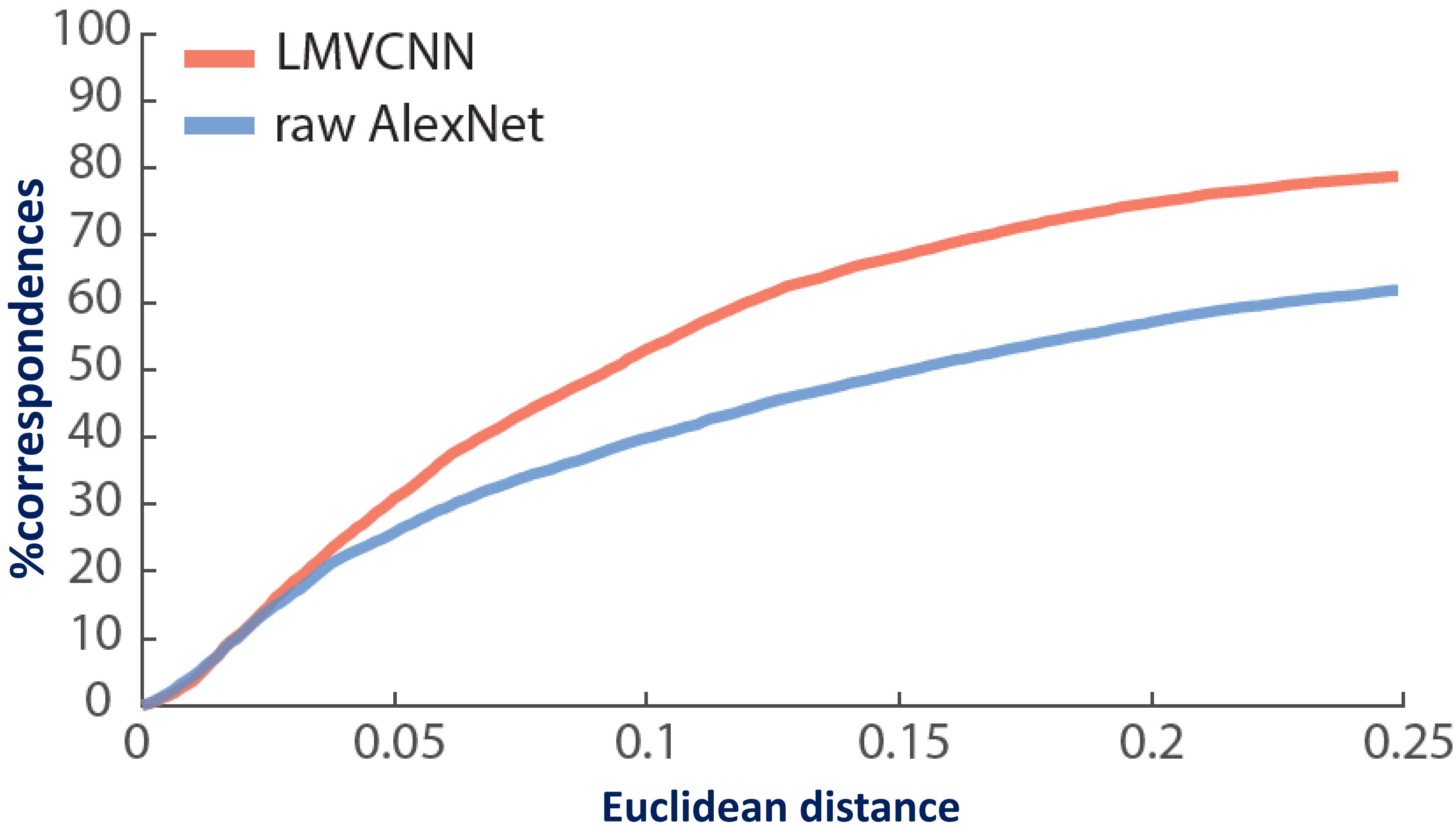
[Belongie and Malik 2000, Kalogerakis et al. 2010]



[Zeng et al. 2017 - volumetric net]



Our method – Local Multi-view CNN (LMVCNN)



LMVCNN
raw AlexNet
%correspondences
Euclidean distance
100
90
80
70
60
50
40
30
20
10
0
0
0.05
0.1
0.15
0.2
0.25

Matching 3D point clouds to 3D models

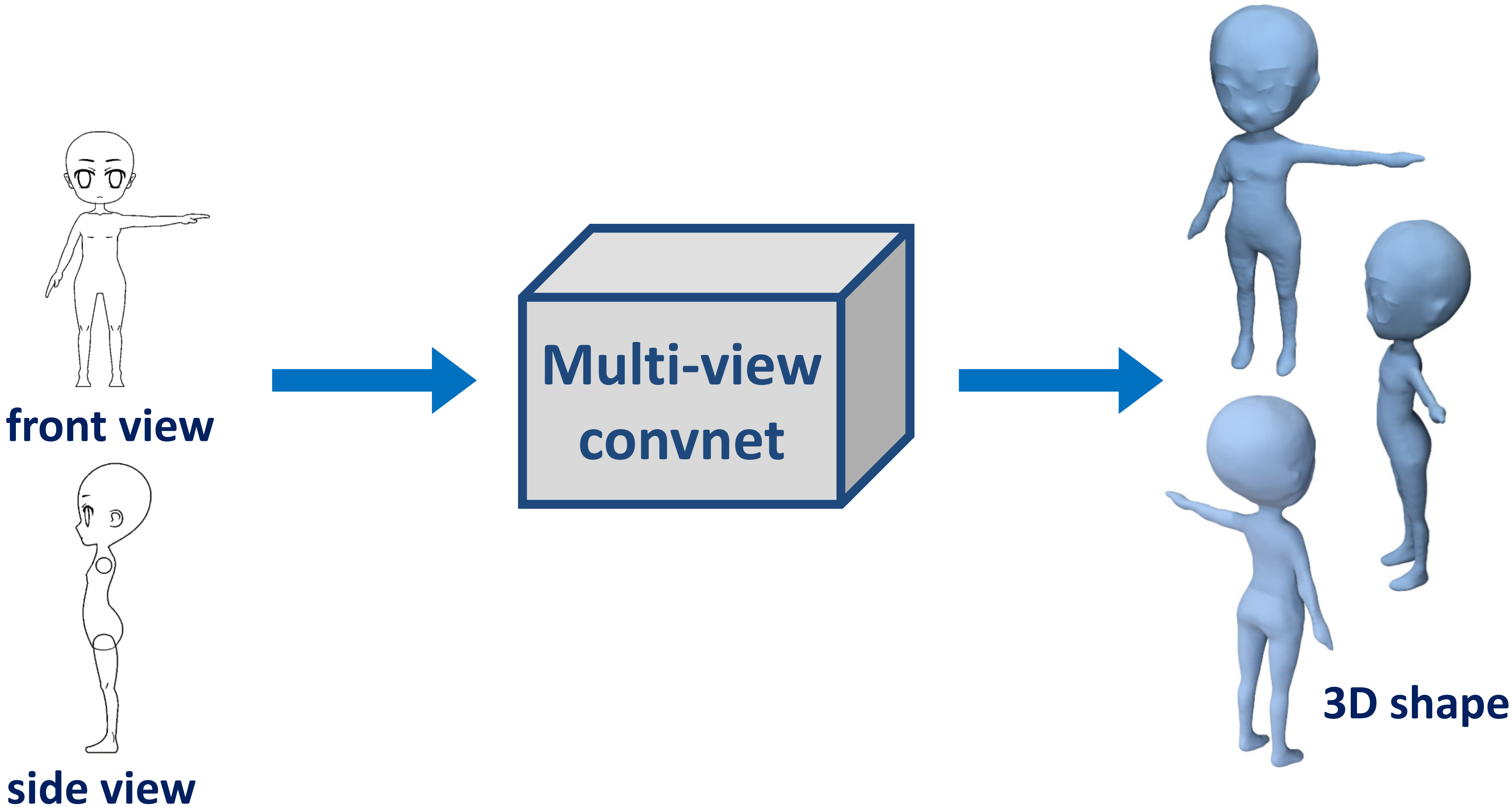
(similar colors correspond to points with similar descriptors)

Note: point clouds are rendered using a sphere per point

Outline

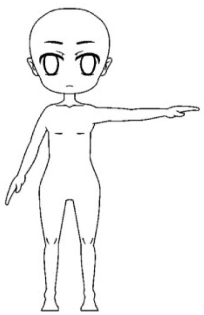
1. Multi-view convnets for 3D shape analysis
   - Shape Segmentation
   - Shape Classification & Retrieval
   - Shape Correspondences
2. **Multi-view convnets for 3D shape synthesis**
3. Discussion / Future work

Sketch-based shape synthesis

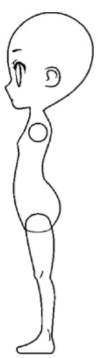


Lun, Gadelha, Kalogerakis, Maji, Wang (3DV 2017 oral)

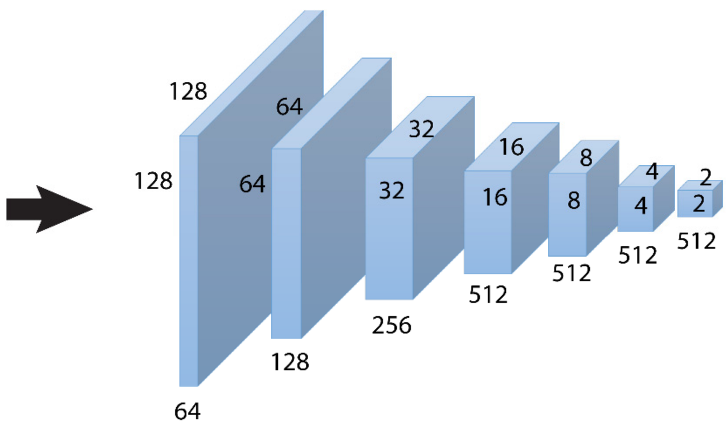
Encoder



front view



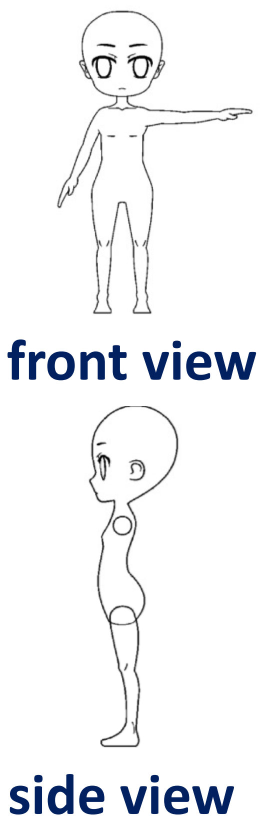
side view

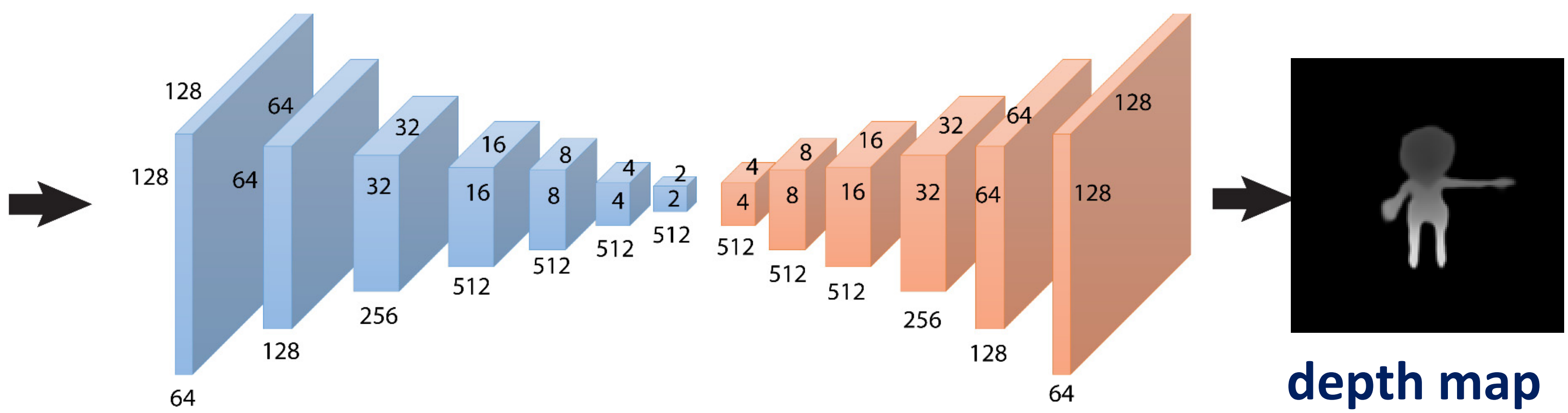


Feature representations capturing **increasingly larger context** in the sketches

Decoder

Infer **depth and normal maps**

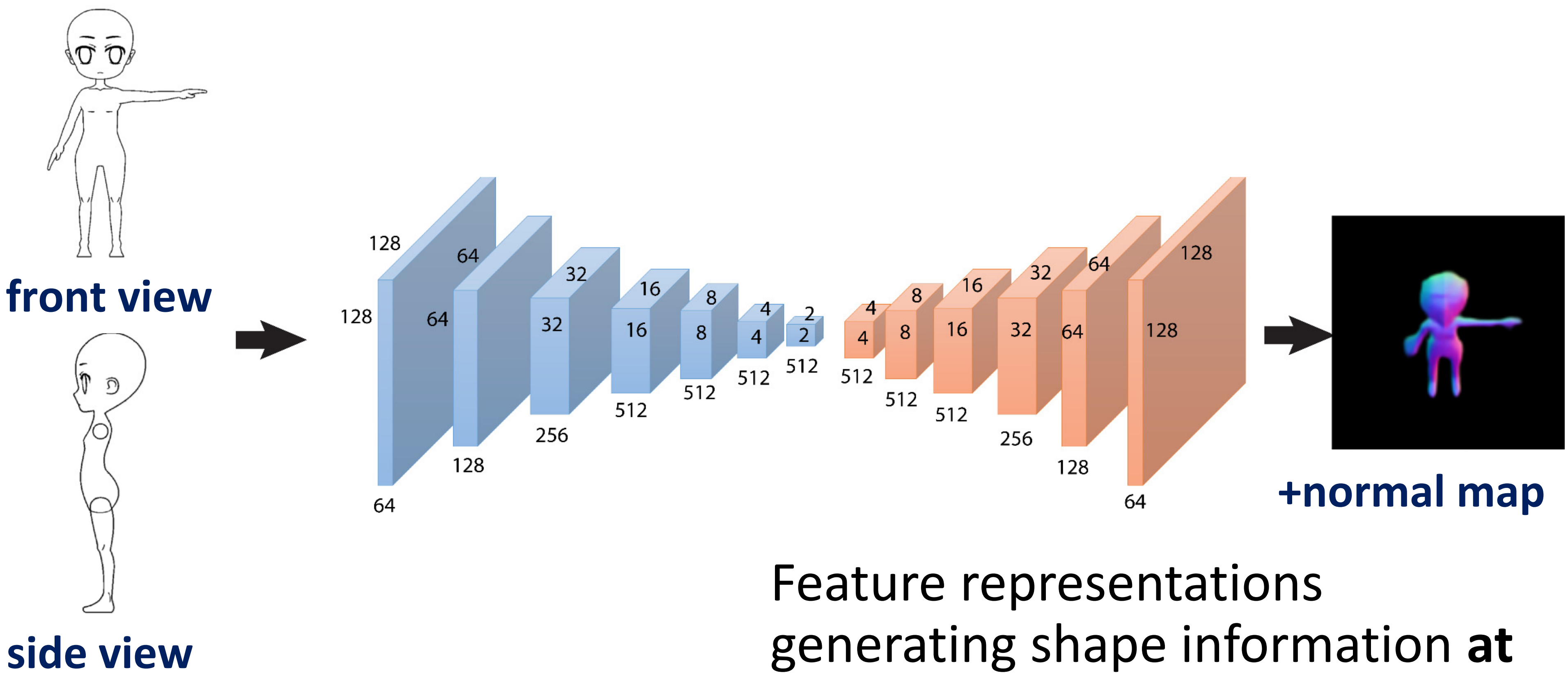




Feature representations
generating shape representations
at increasingly finer scales

Decoder

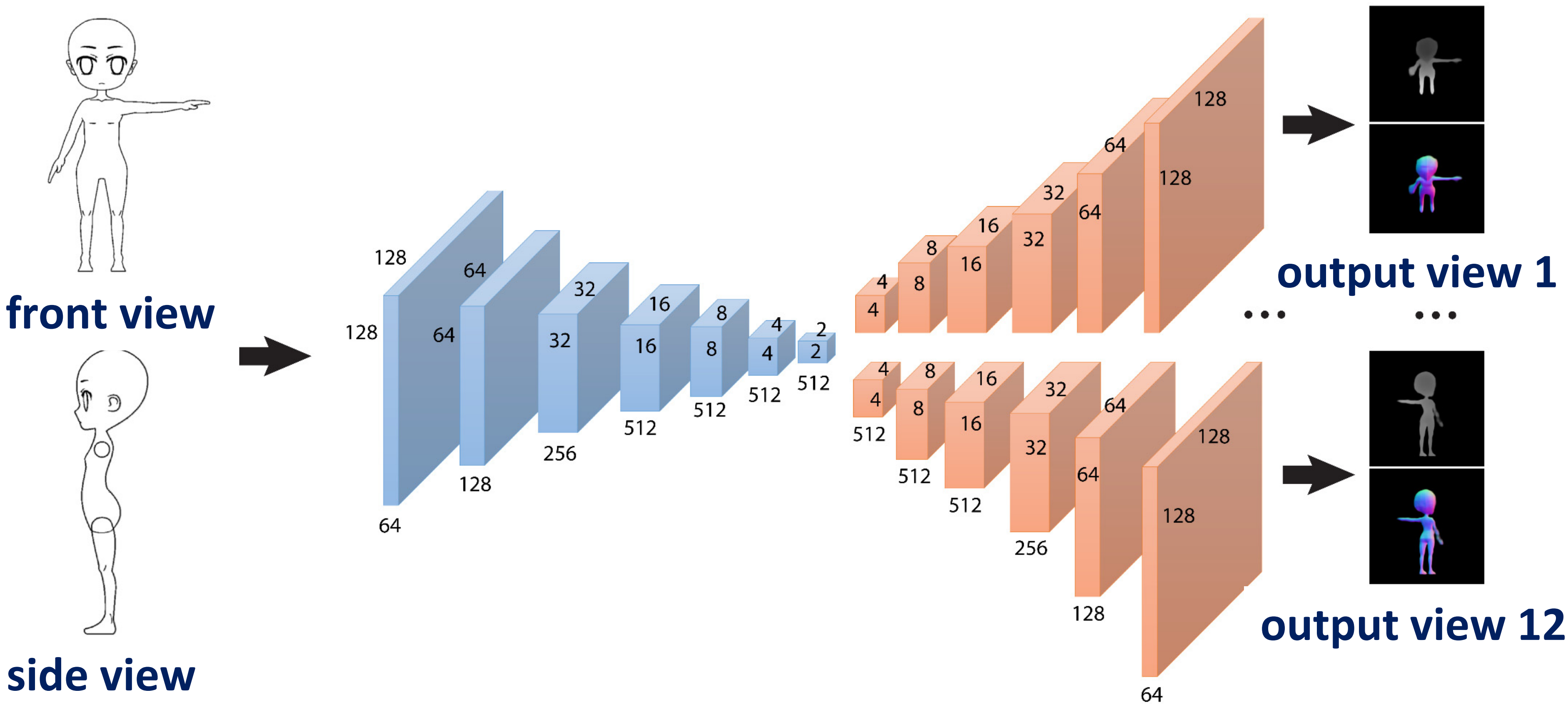
Infer **depth and normal maps**



Feature representations generating shape information **at increasingly finer scales**

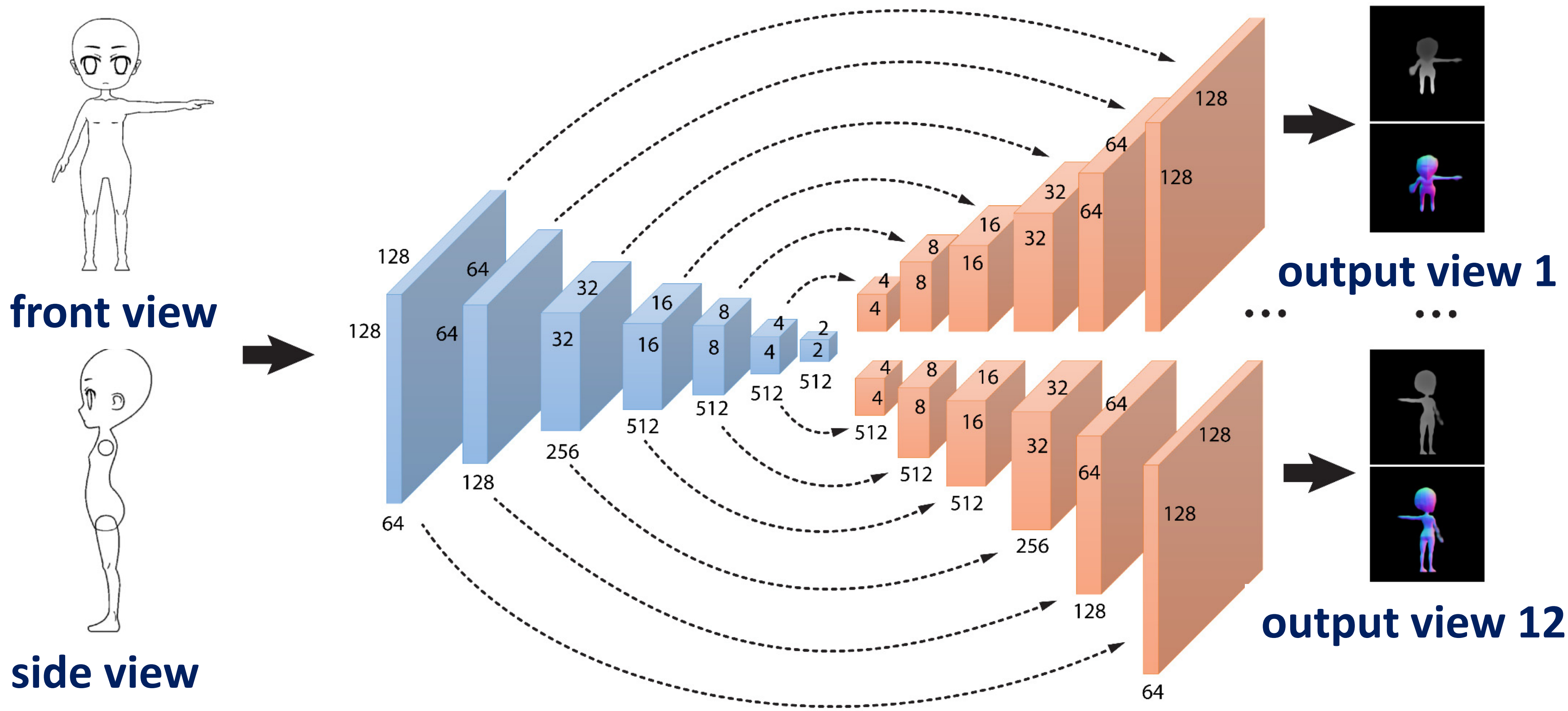
Multi-view decoder

Infer depth and normal maps **for several views**



“U-Net”

Feature representations in the decoder depend on **previous layer & encoder’s corresponding layer**

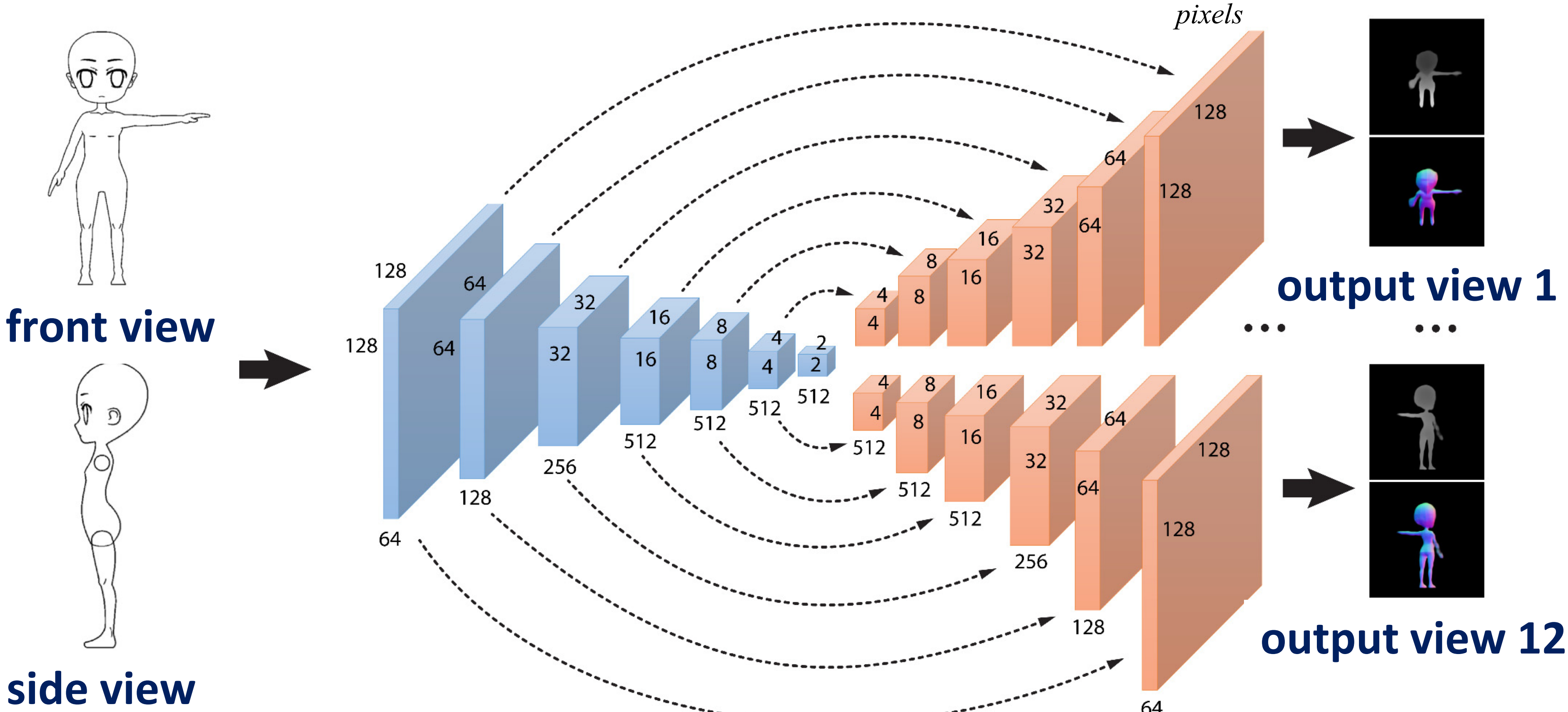


U-net: Ronneberger et al. 2015, Isola et al. 2016

Initial training loss

Penalize per-pixel depth reconstruction error: $\sum_{pixels} | d_{pred} - d_{gt} |$

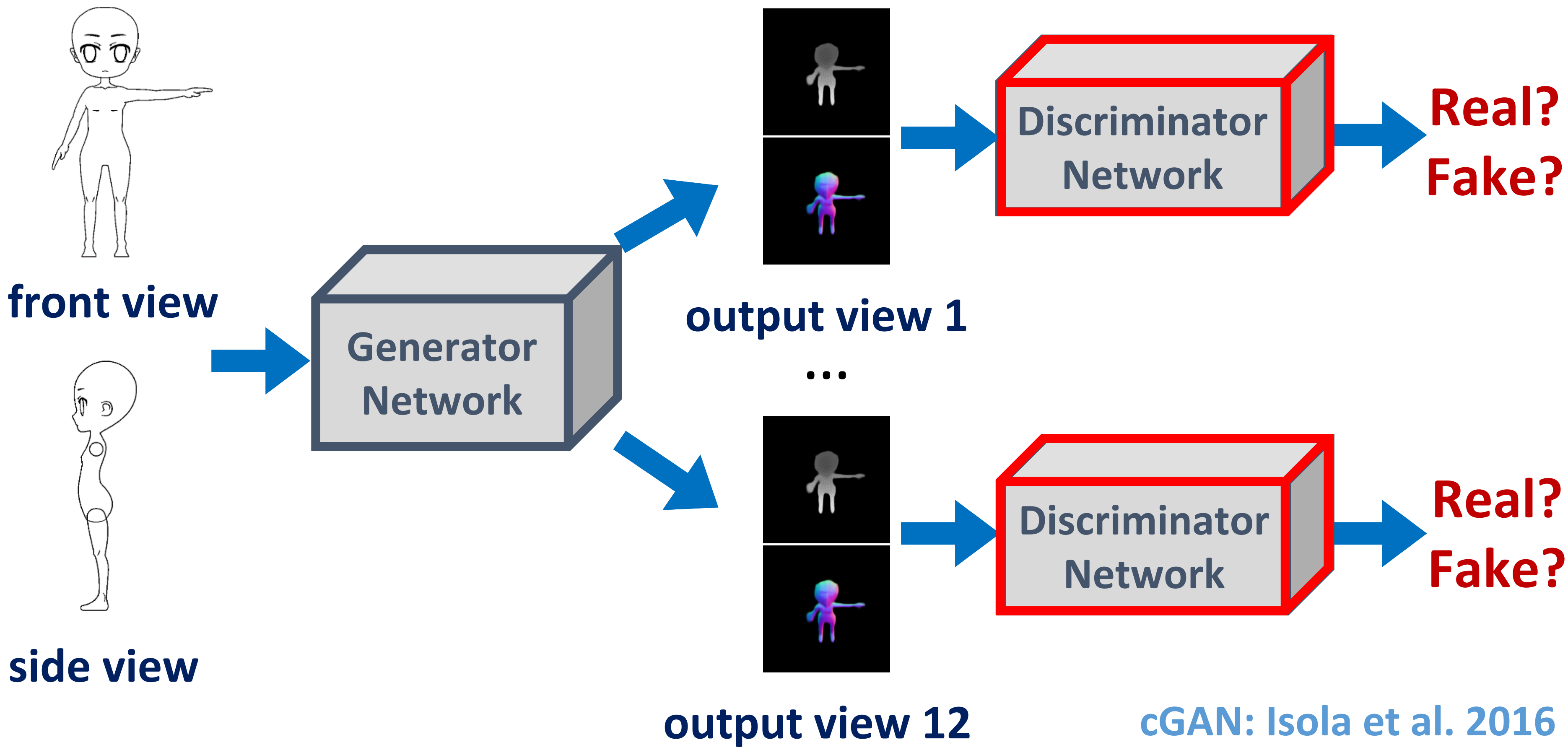
& per-pixel normal reconstruction error: $\sum_{pixels} (1 - n_{pred} \cdot n_{gt})$



U-net: Ronneberger et al. 2015,
Isola et al. 2016

Training: discriminator network

Checks whether the output depth & normals look **real** or **fake**.
Trained by treating ground-truth as **real**, generated maps as **fake**.



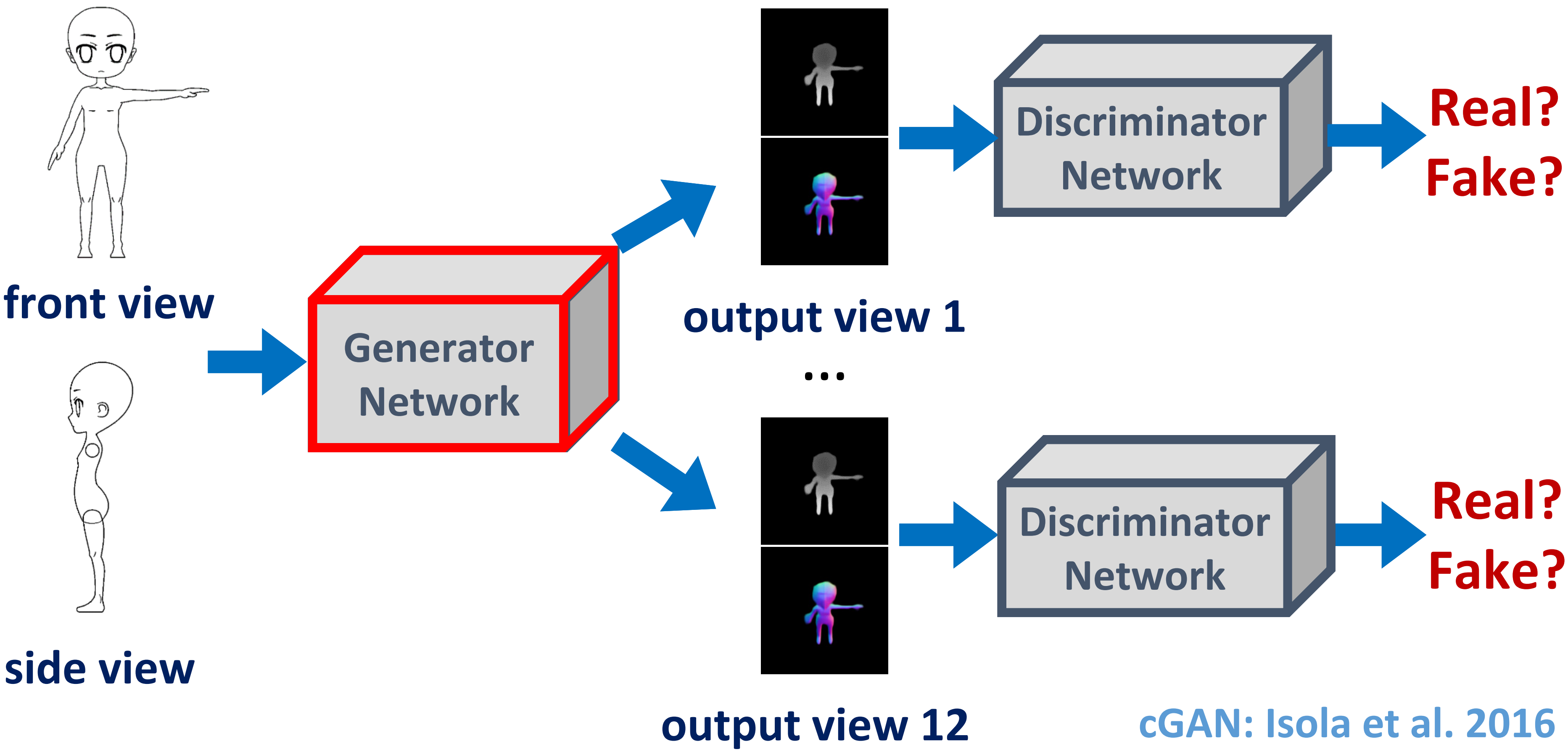
cGAN: Isola et al. 2016

Generator: Full training loss

Penalize per-pixel depth reconstruction error: $\sum_{pixels} | d_{pred} - d_{gt} |$

& per-pixel normal reconstruction error: $\sum_{pixels} (1 - n_{pred} \cdot n_{gt})$

& "unreal" outputs: $-\log P(real)$



cGAN: Isola et al. 2016

Training data

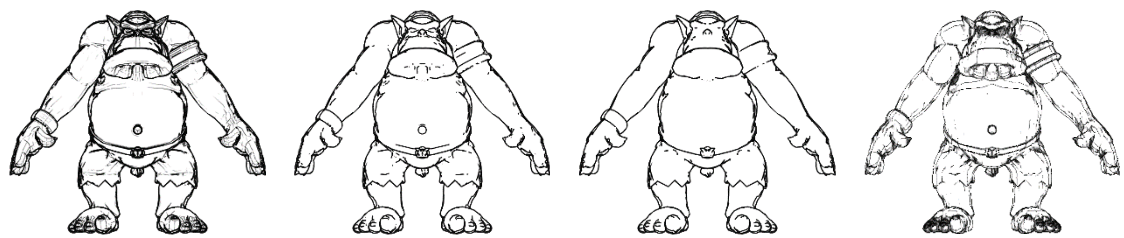
Character
10K models

Chair
10K models

Airplane
3K models

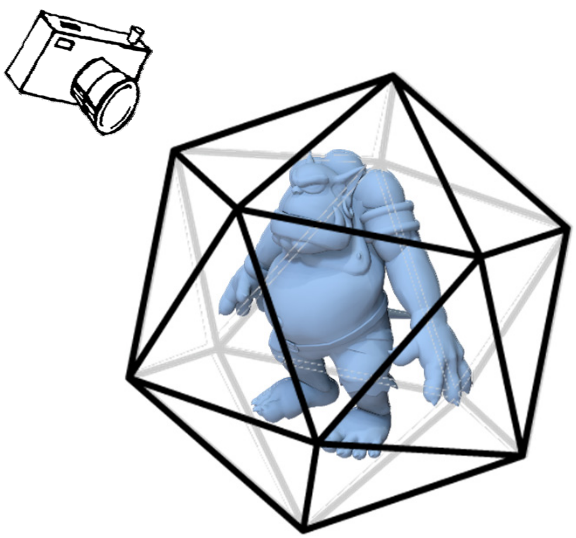
Models from “The Models Resource” & 3D Warehouse

Training data

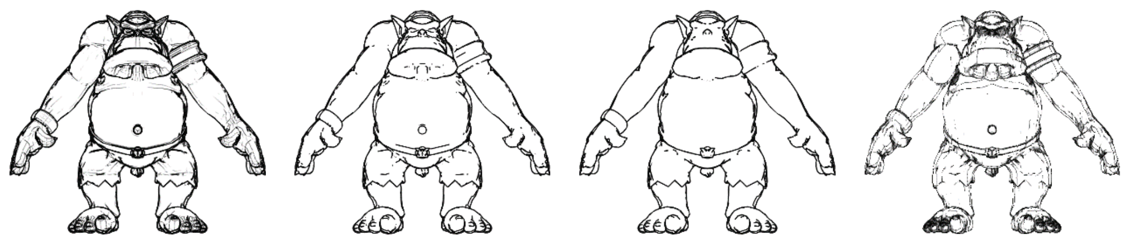


Synthetic line drawings

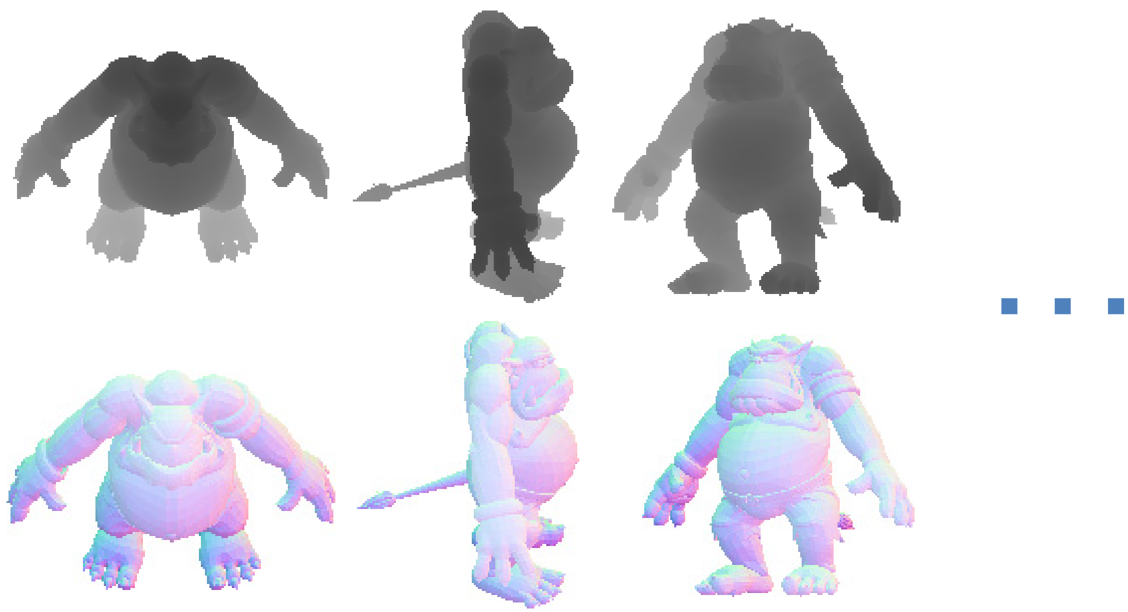
Training data



12 views



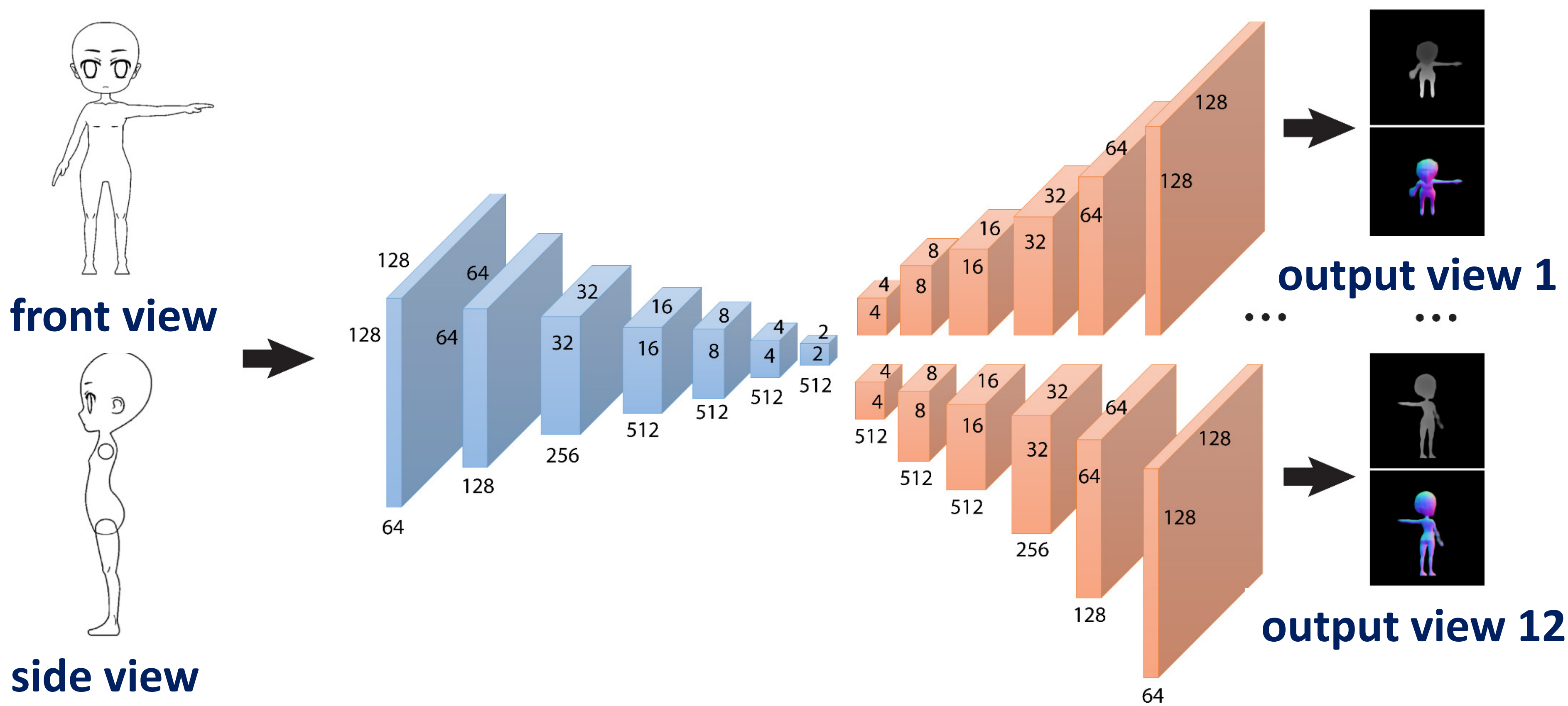
Synthetic line drawings



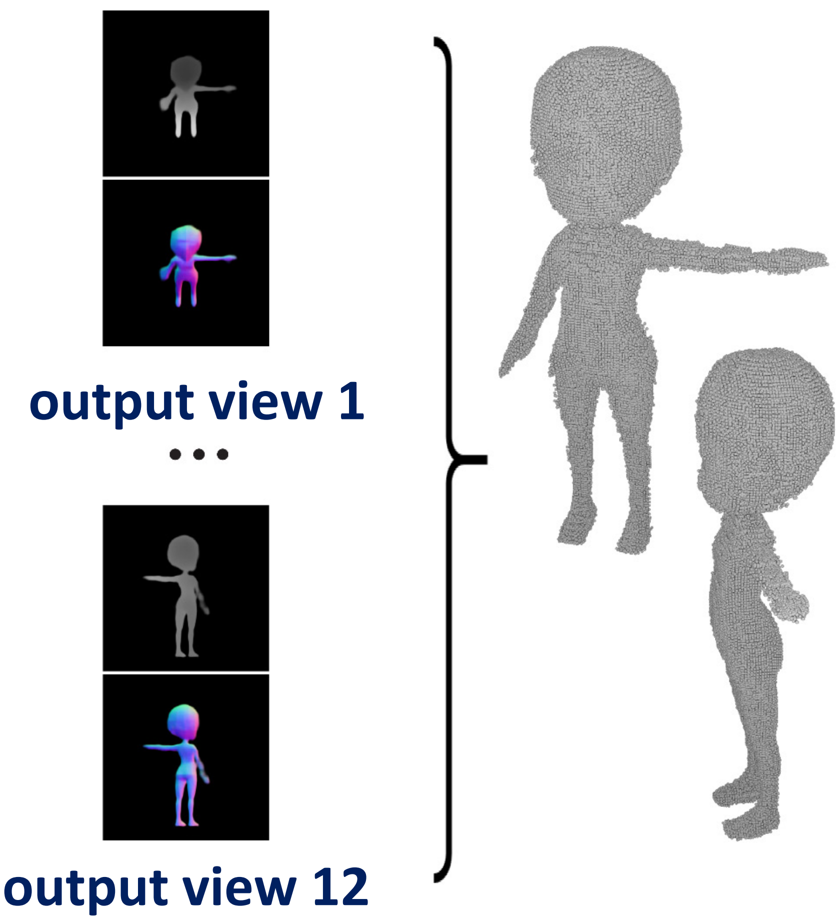
Training depth and normal maps

Test time

Predict **multi-view depth and normal maps!**



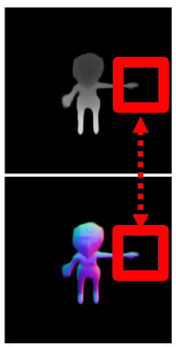
Multi-view depth & normal map fusion



Multi-view depth & normal maps

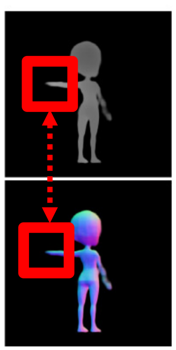
Consolidated point cloud

Multi-view depth & normal map fusion



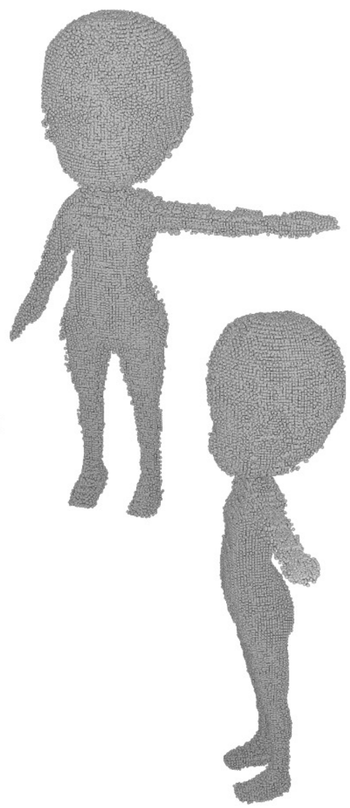
output view 1

• • •



output view 12

Multi-view depth & normal maps

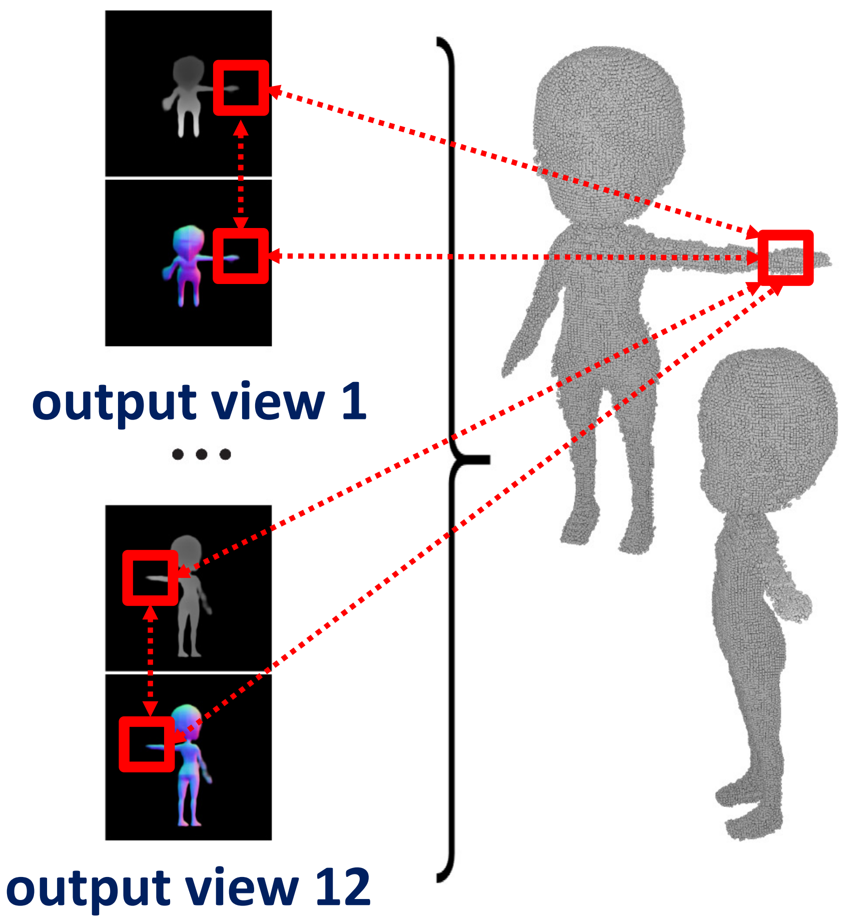


Consolidated point cloud

Optimization problem

- **Depth derivatives should be consistent with normals**

Multi-view depth & normal map fusion



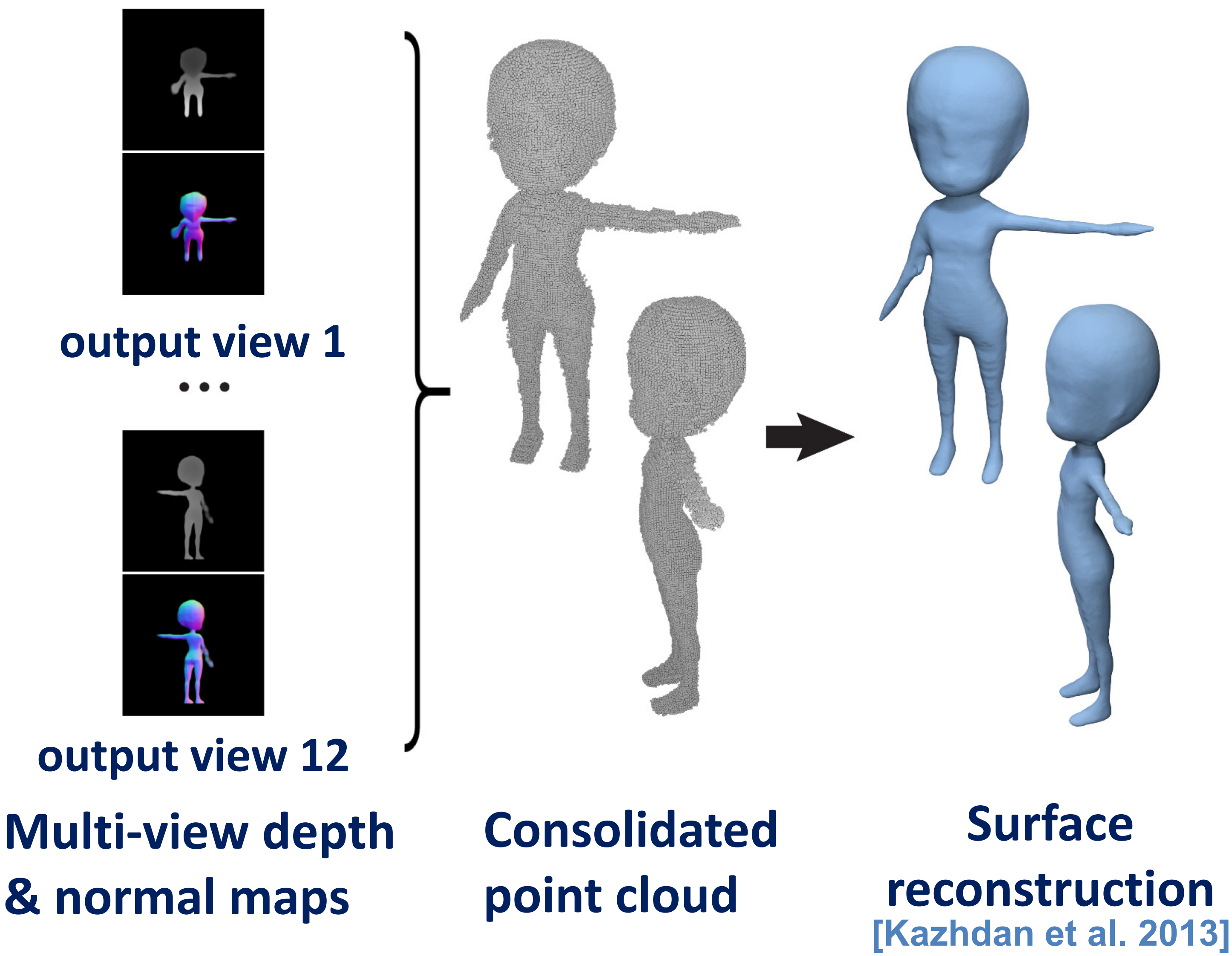
Multi-view depth & normal maps

Consolidated point cloud

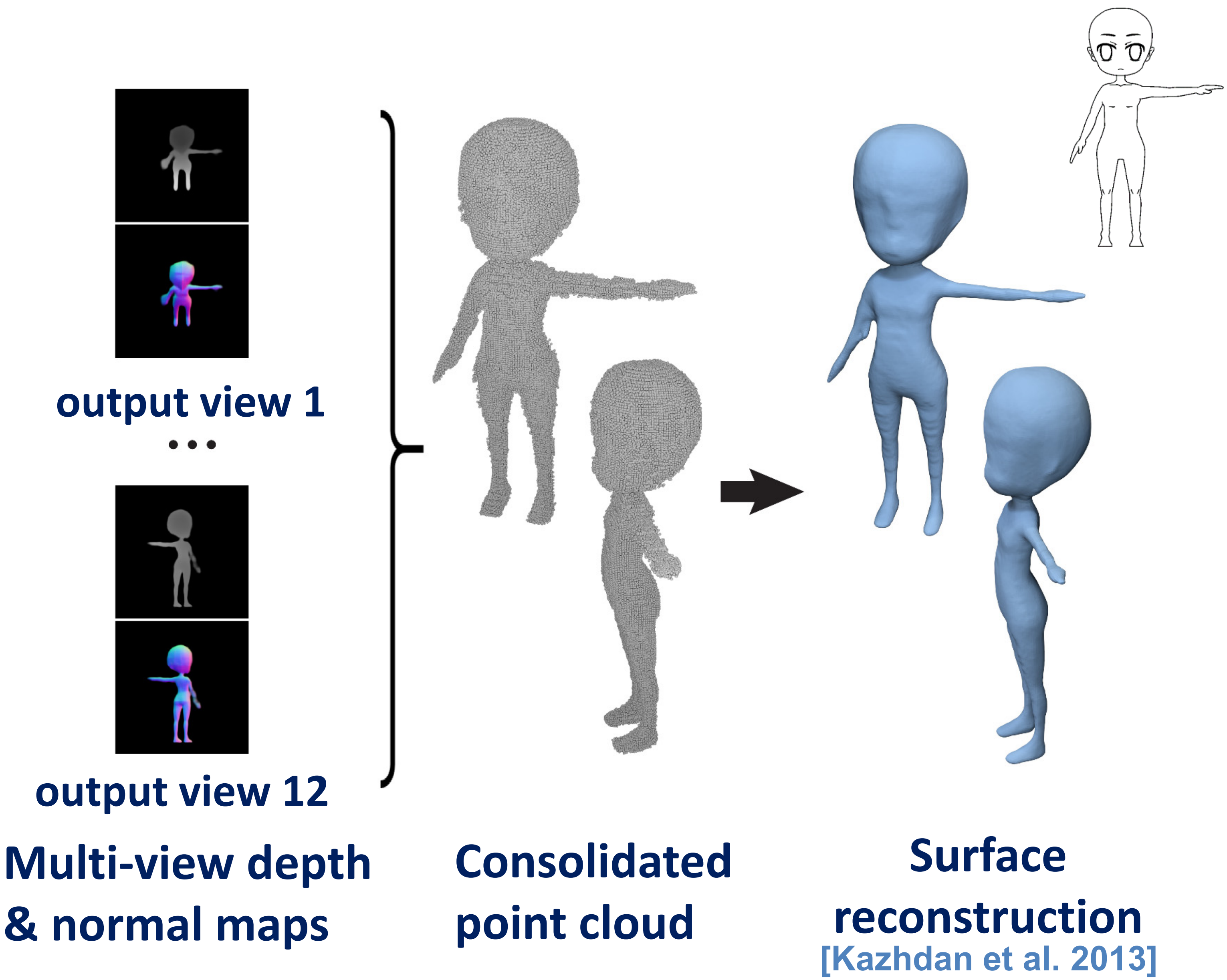
Optimization problem

- Depth derivatives should be consistent with normals
- Corresponding depths and normals across different views should agree

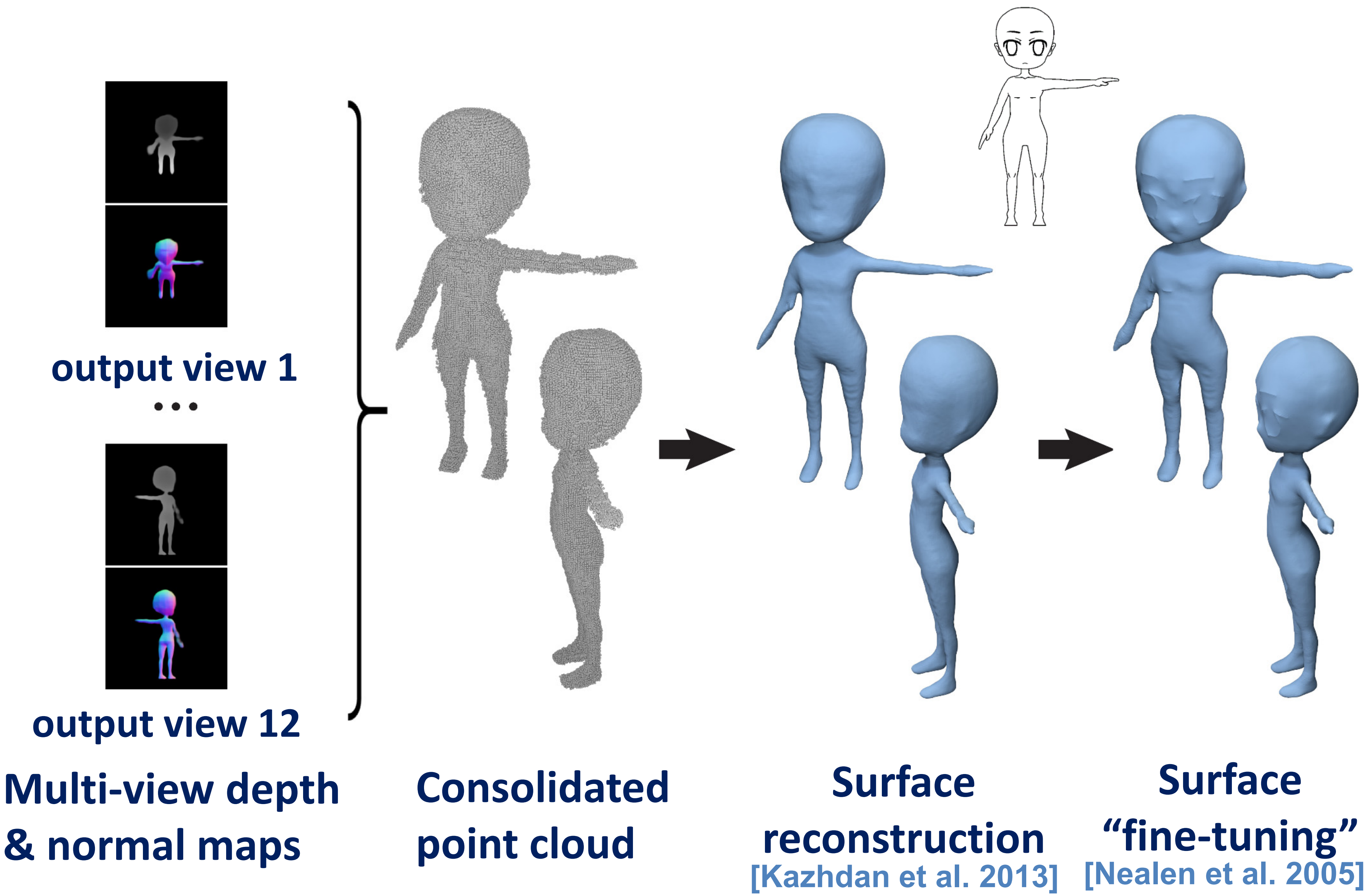
Surface reconstruction



Surface deformation



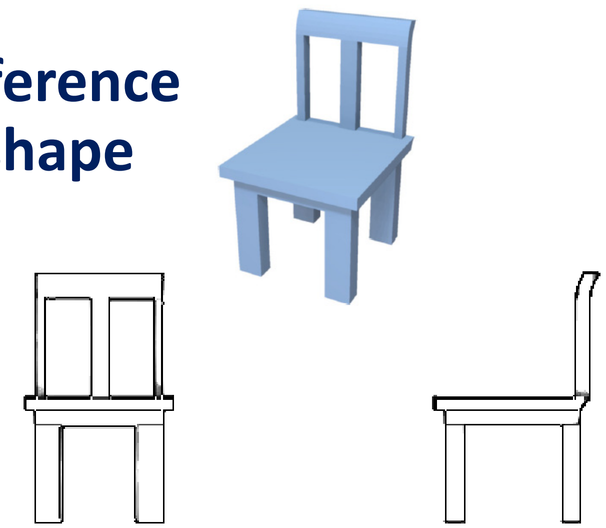
Surface deformation



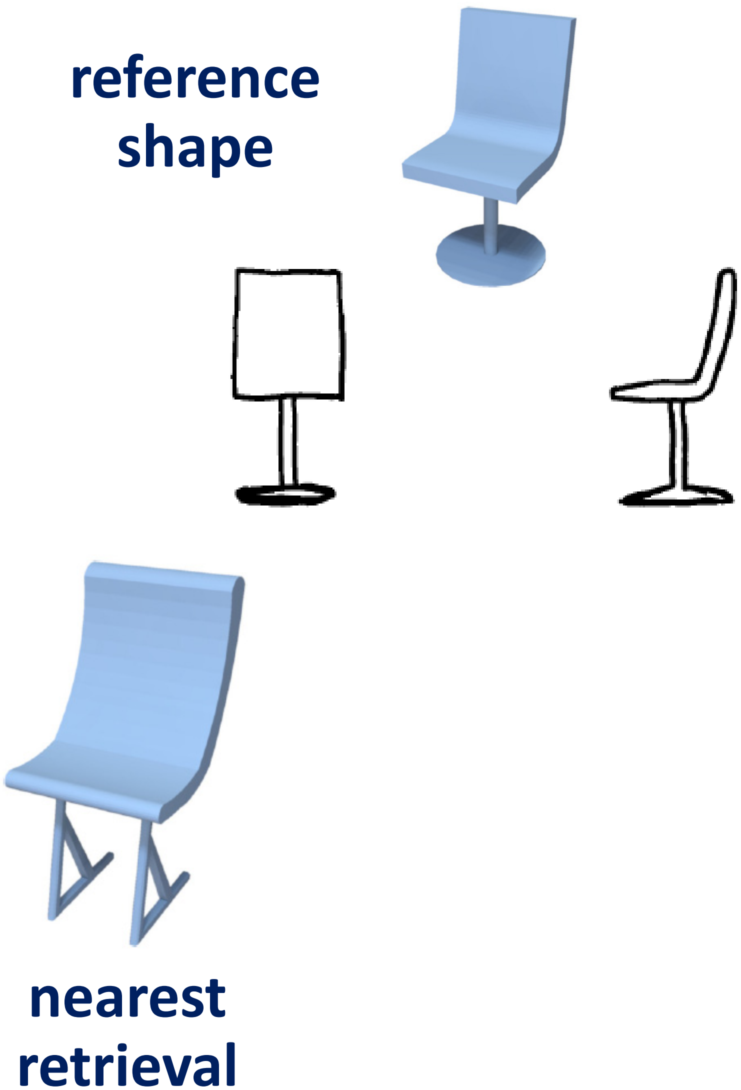
Quantitative results

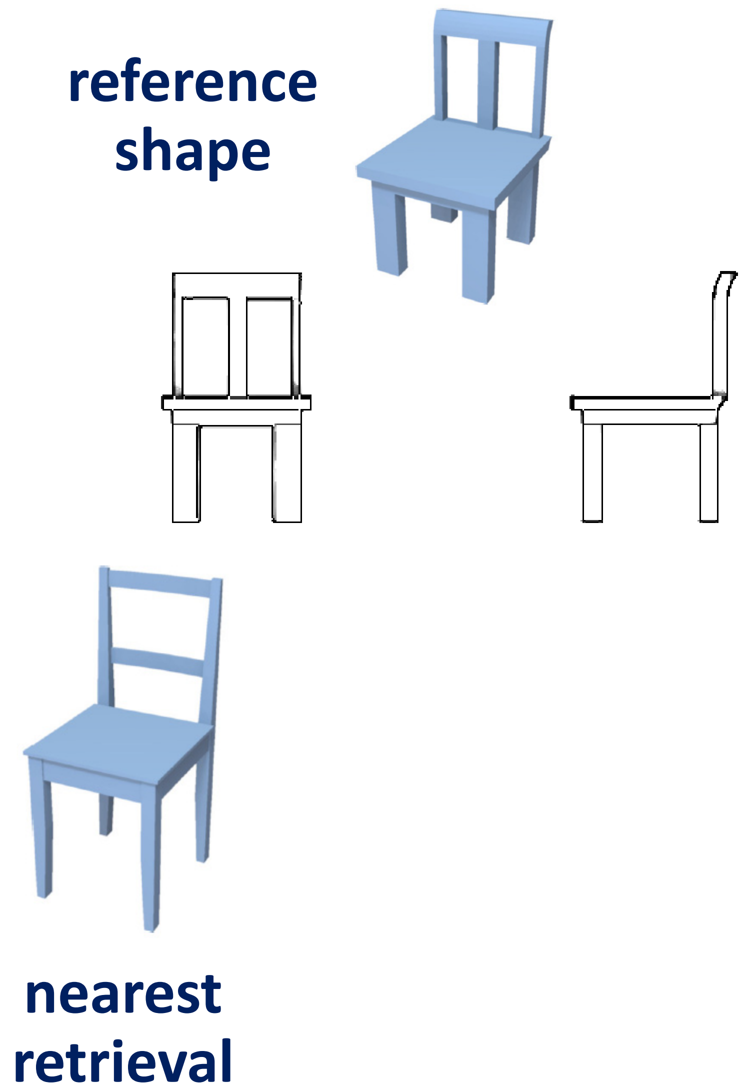
reference shape

reference shape

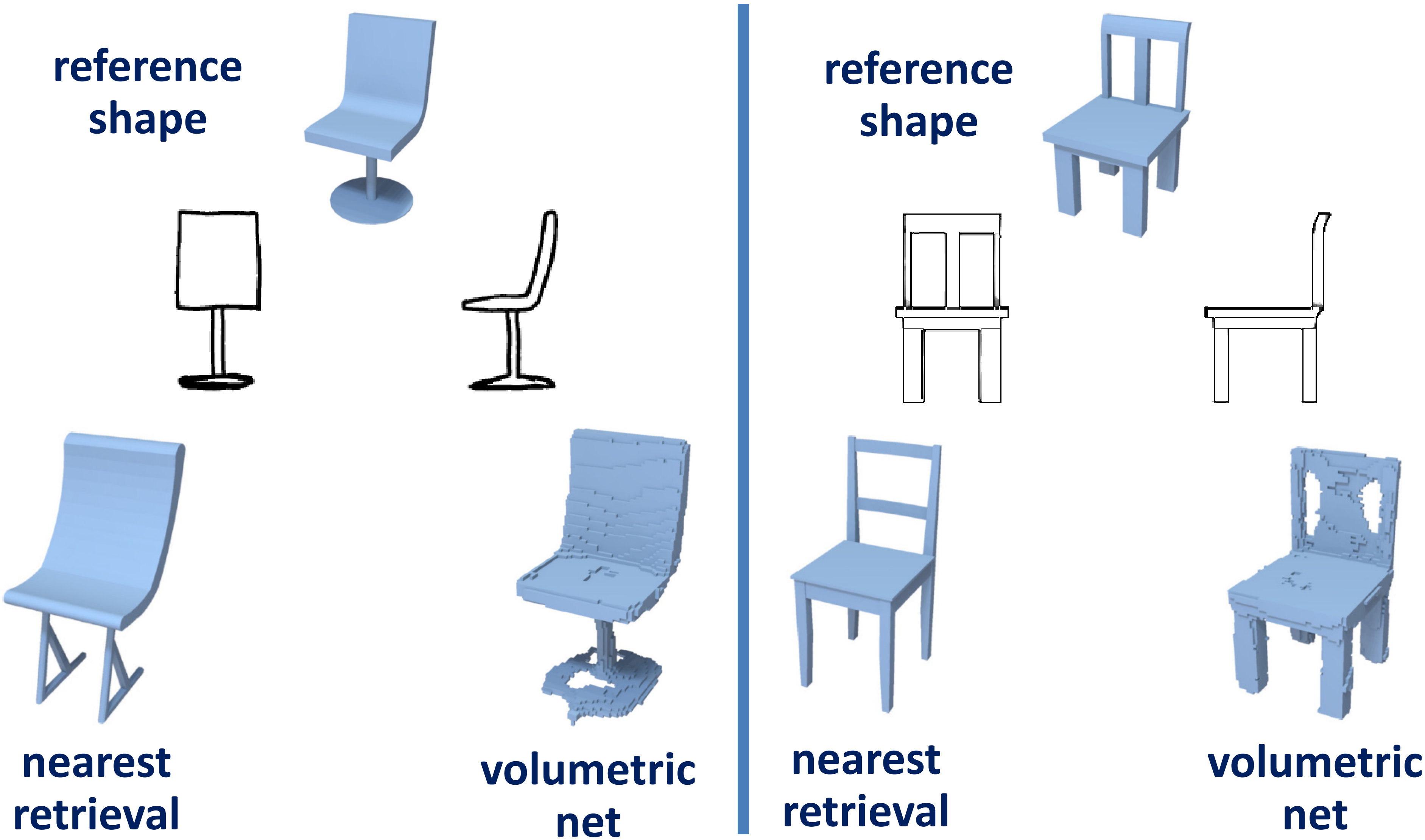


Quantitative results

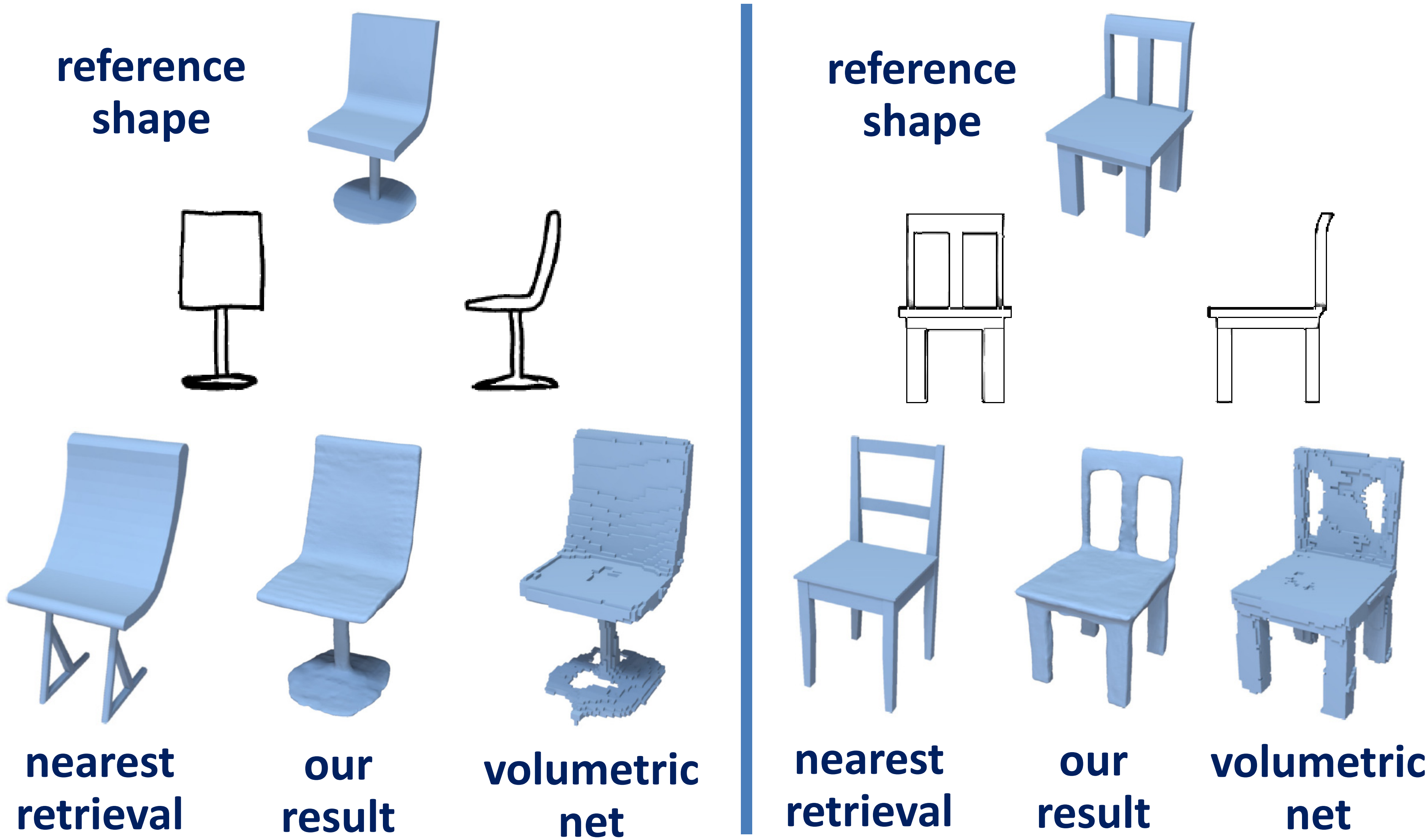




Quantitative results

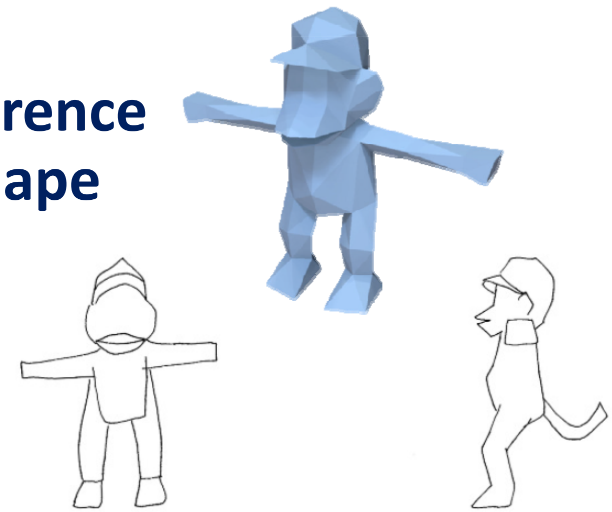


Quantitative results



Quantitative results

reference shape

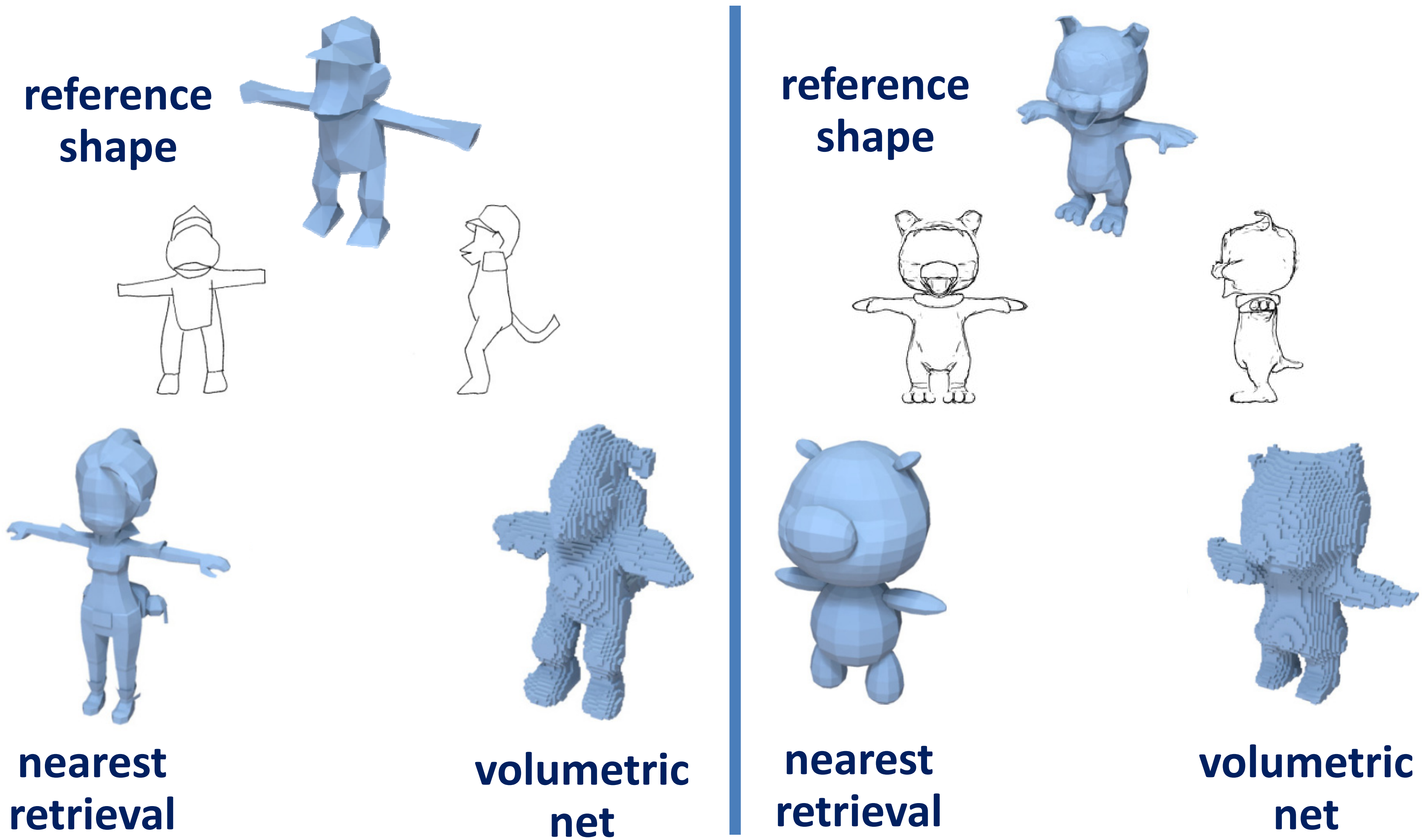


nearest retrieval

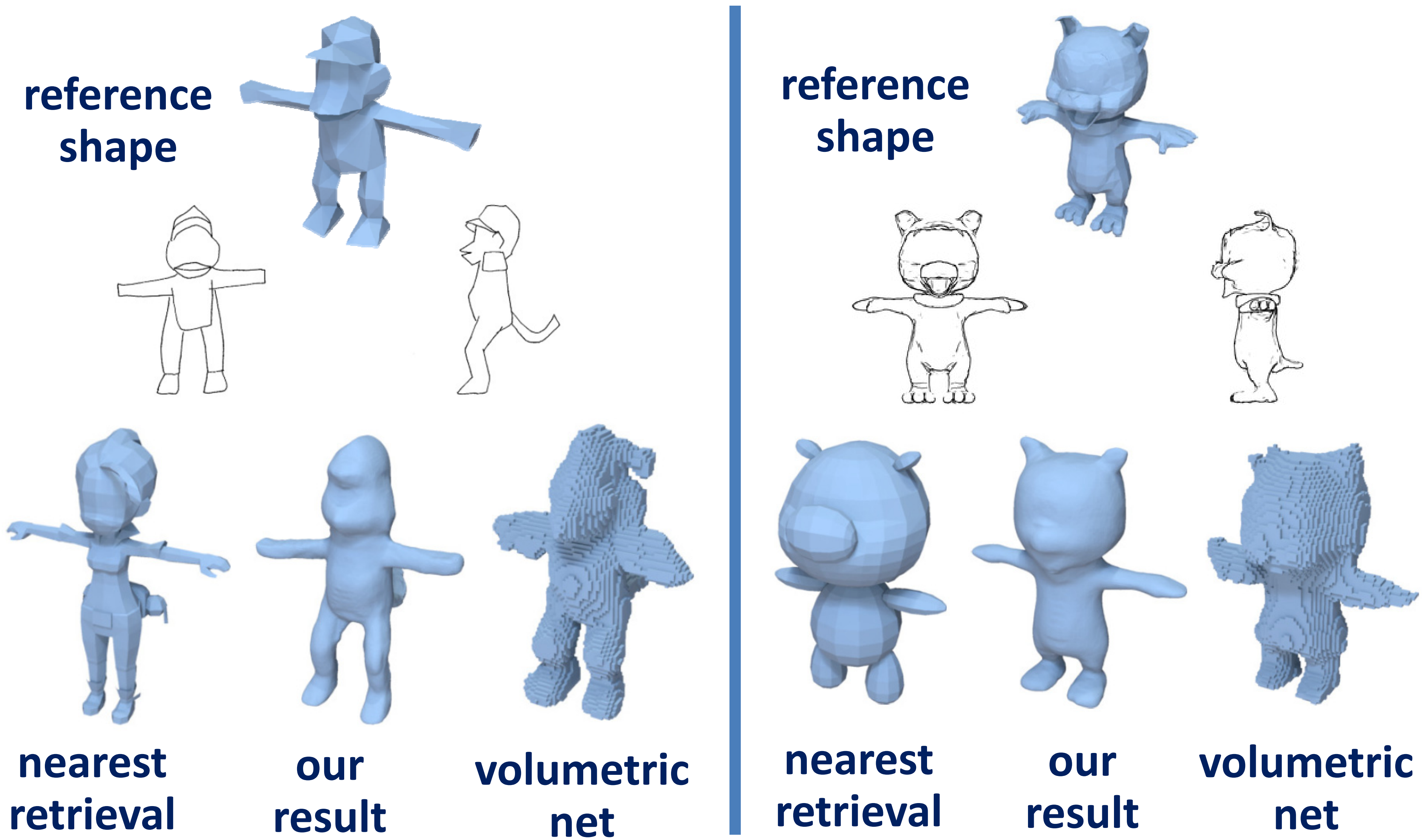
reference shape

nearest retrieval

Quantitative results



Quantitative results



Quantitative results

Characters (human drawing)

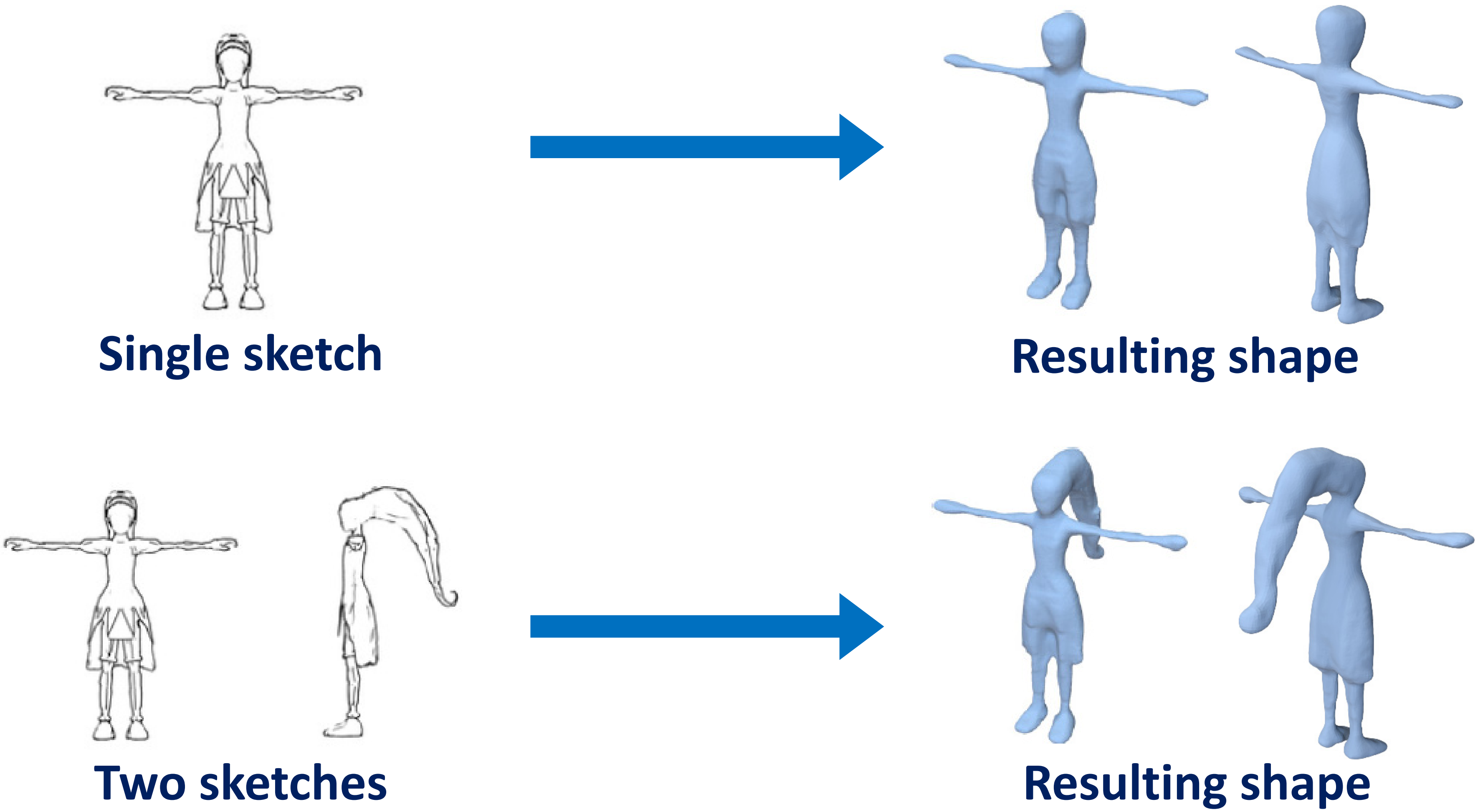
	Our method	**Volumetric decoder**	**Nearest retrieval**
Hausdorff distance	**0.120**	0.638	0.242
Chamfer distance	**0.023**	0.052	0.045
normal distance	**34.27**	56.97	47.94
depth map error	**0.028**	0.048	0.049
volumetric distance	**0.309**	0.497	0.550

Quantitative results

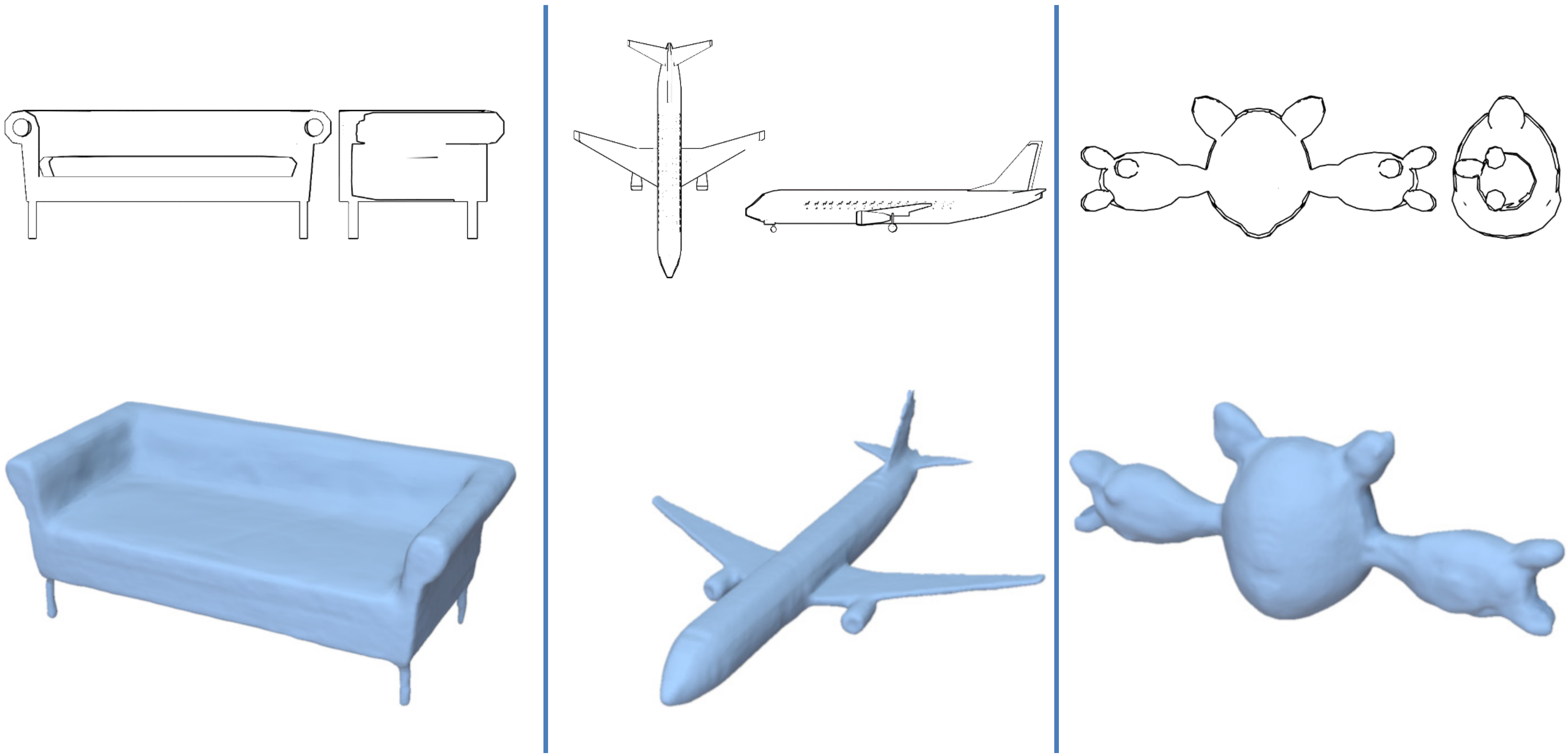
Man-made objects (human drawing)

	Our method	Volumetric decoder	Nearest retrieval
Hausdorff distance	**0.171**	0.211	0.228
Chamfer distance	**0.028**	0.032	0.038
normal distance	**34.19**	48.81	43.75
depth map error	**0.037**	0.046	0.059
volumetric distance	**0.439**	0.530	0.560

Single vs two input line drawings



More results



Outline

1. **Multi-view convnets for 3D shape analysis**
   - **Shape Segmentation**
   - **Shape Classification & Retrieval**
   - **Shape Correspondences**
2. **Multi-view convnets for 3D shape synthesis**
3. **Discussion / Future work**

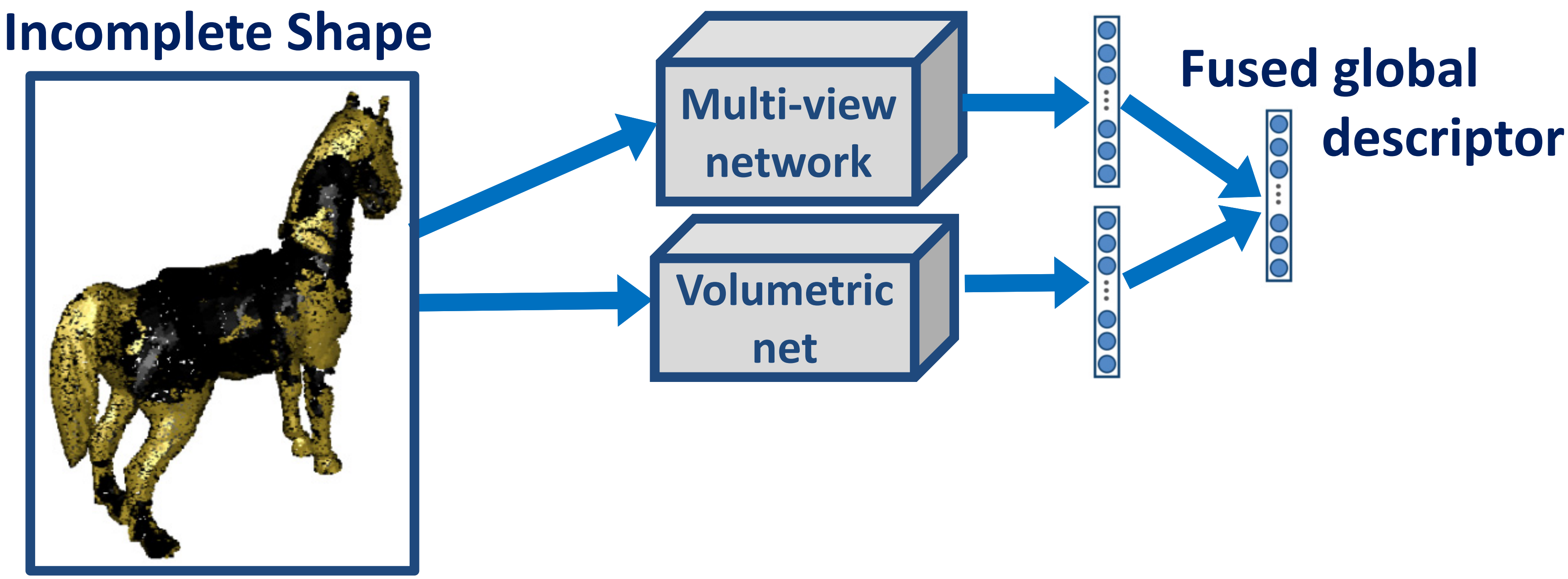
Summary: multi-view architectures

- Motivated by human vision: view-based convnets analyze **what can be seen** under view projections
- Aggregate information from **multiple views selected to maximally cover the surface**
- Combine with surface models to **deal with occlusions**
- **Robust** to input geometric representation artifacts (e.g., irregular tessellation, polygon soups, etc)
- Initialized from image-based architectures **pretrained on massive image datasets**

Limitations

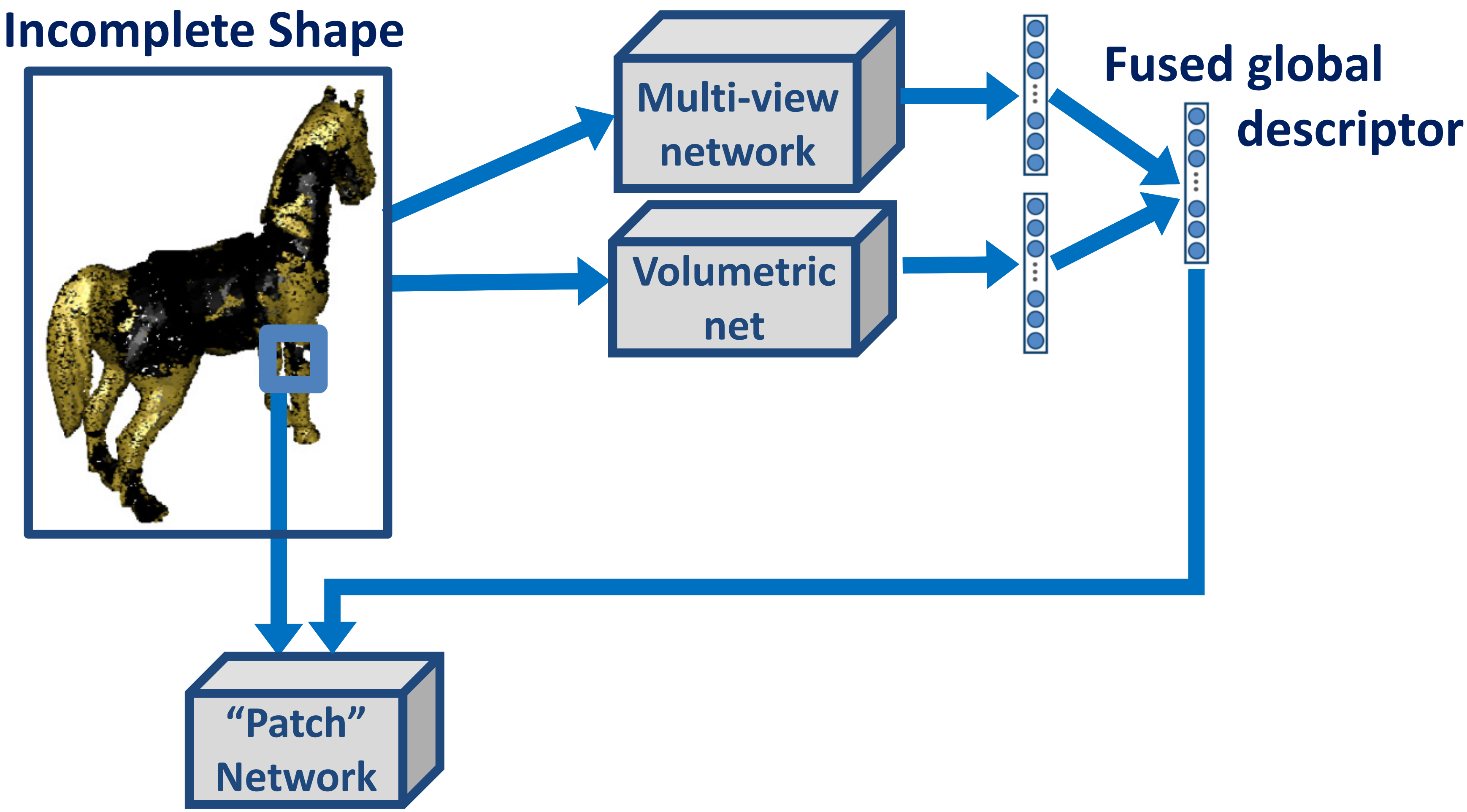
- **Volumetric/interior properties** of objects cannot be handled with MVCNNs
- **Some redundancy in processing** (same surface is visible from multiple views)
- View pooling might cause **some information loss**
- **Combine volumetric & multi-view nets, point-based nets?**

Combining volumetric & multi-view nets



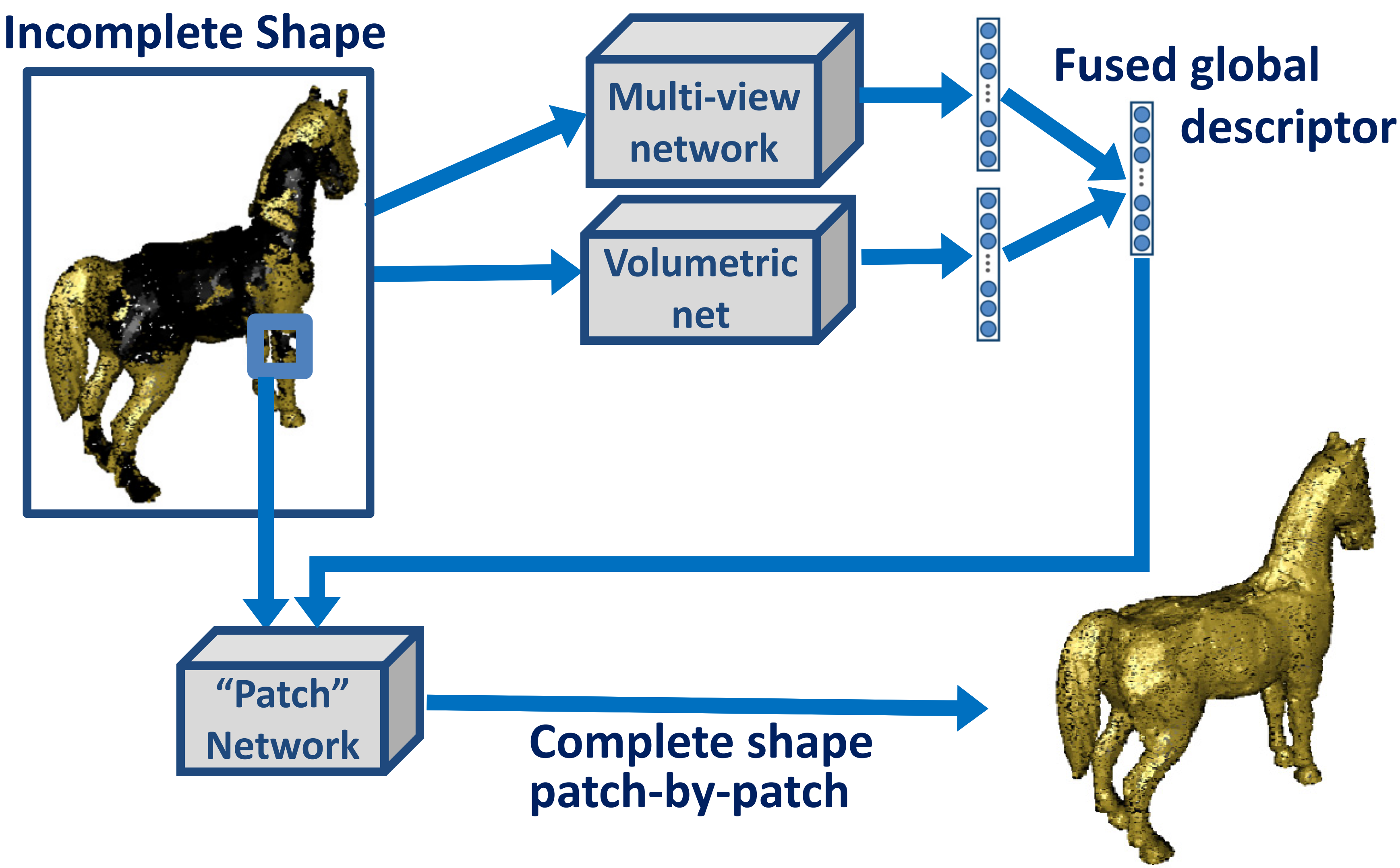
Han, Li, Huang, Kalogerakis, Yu, ICCV 2017

Combining volumetric & multi-view nets



Han, Li, Huang, Kalogerakis, Yu, ICCV 2017

Combining volumetric & multi-view nets



Han, Li, Huang, Kalogerakis, Yu, ICCV 2017

Thank you!

Acknowledgements: NSF (CHS-1422441, CHS-1617333), NVidia, MassTech Collaborative, Adobe.

Web page with papers, project data, source code, results:

http://people.cs.umass.edu/~kalo/