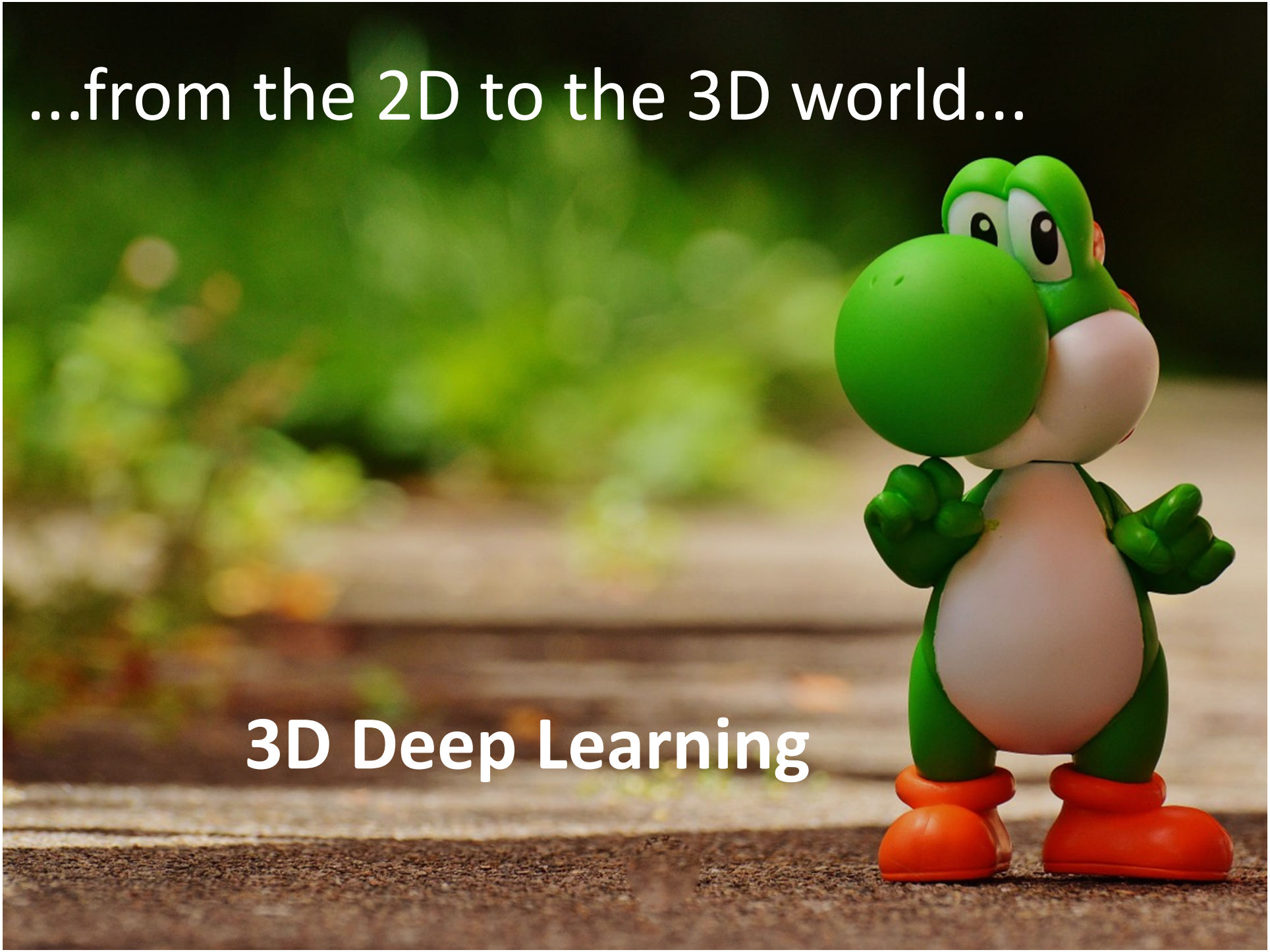
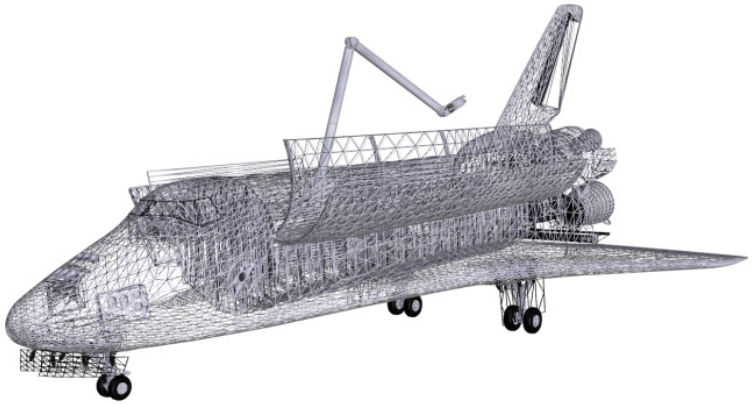
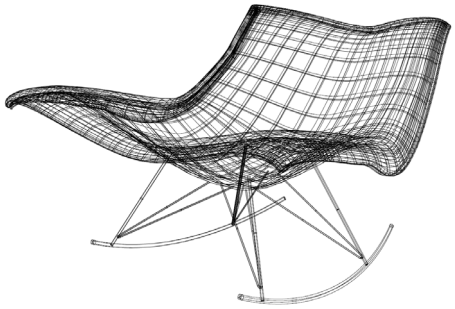

...from the 2D to the 3D world...
3D Deep Learning

How do we apply convnets for 3D shapes?

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



Polygon mesh



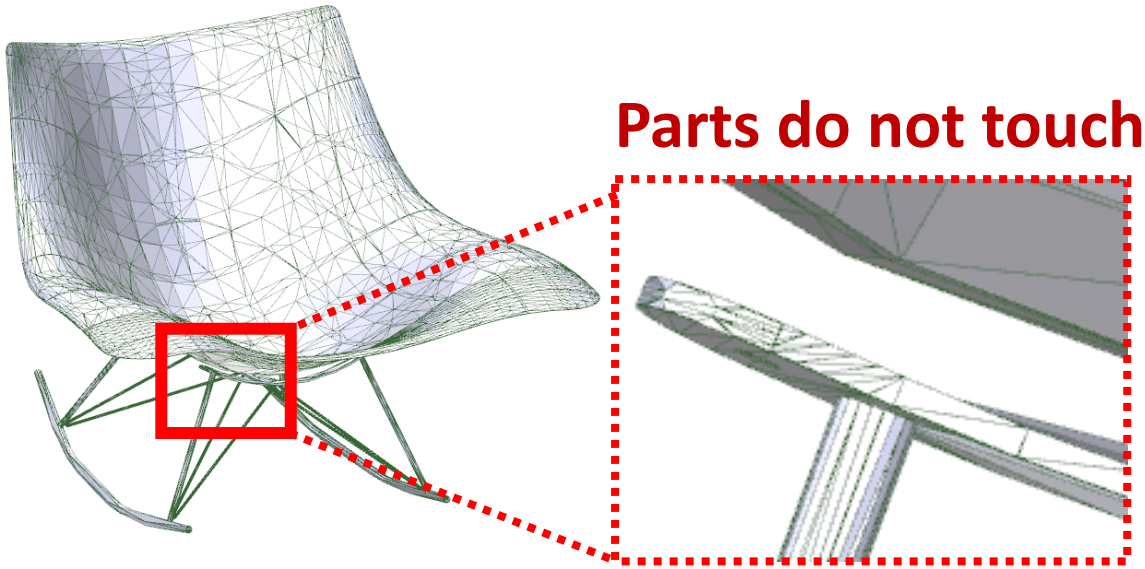
Analytic Surface

Point clouds

Models from 3D Warehouse & FlyingArchitecture

3D Deep Learning Challenges

3D models have **artifacts**.



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

3D Deep Learning Challenges

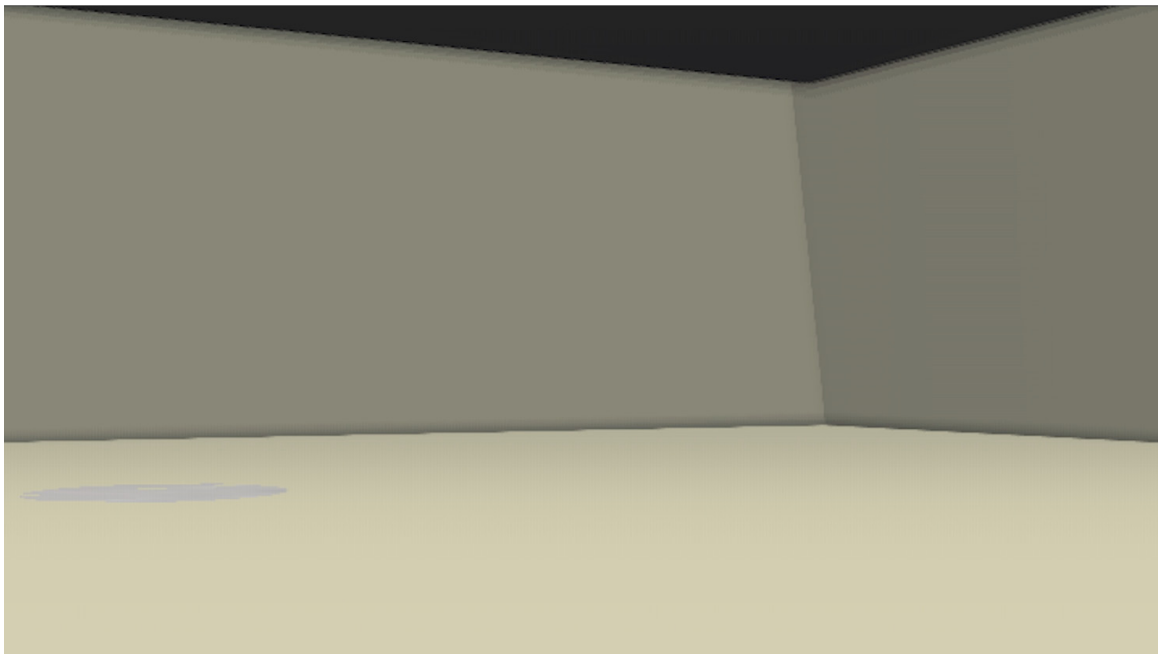
3D shapes are often **designed for viewing…**

3D Deep Learning Challenges

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

3D Deep Learning Challenges

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



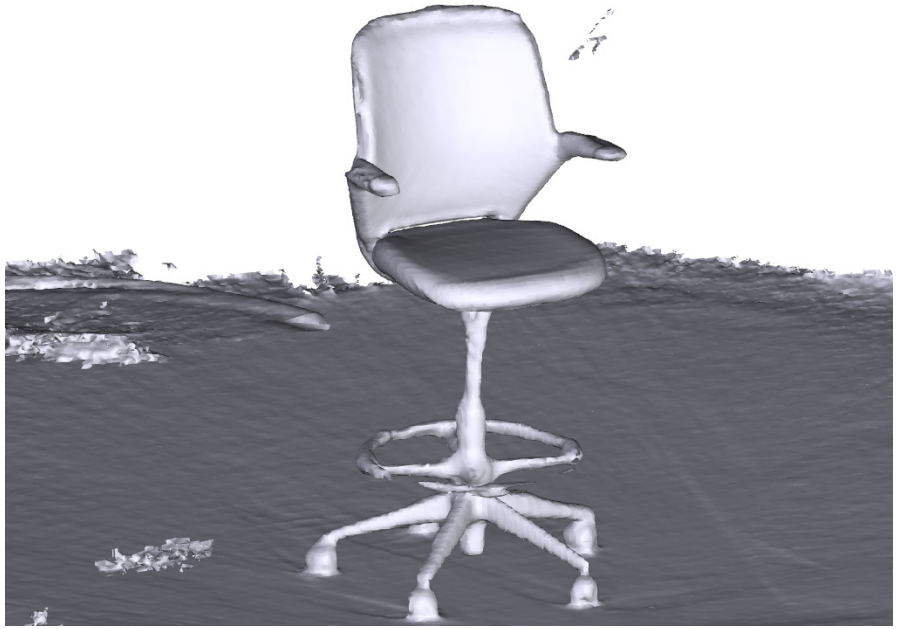
Empty inside!

3D Deep Learning Challenges

Scanned surfaces have **noise** & **missing parts**.

RGB Image & depth data

Resulting surface

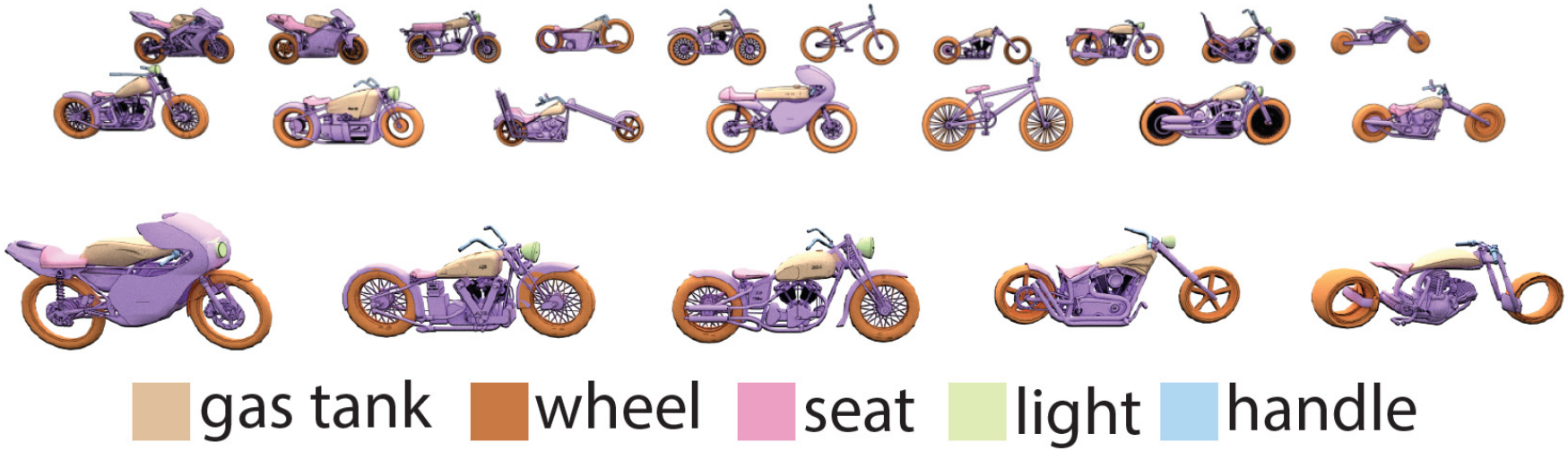


"A Large Dataset of Object Scans"
Choi, Zhou, Miller, Koltun 2016

3D Deep Learning approaches

- The Multi-View approach
- The Voxel approach
- The Point approach
- The Graph approach

All approaches need *3D Datasets*!



Slides from Hao Su, He Wang, Jiajun Wu

Datasets for 3D Object Classification

ShapeNet

Large-scale Synthetic Objects: 3M models
ModelNet: absorbed by ShapeNet
ShapeNetCore: 51.3K models in 55 categories

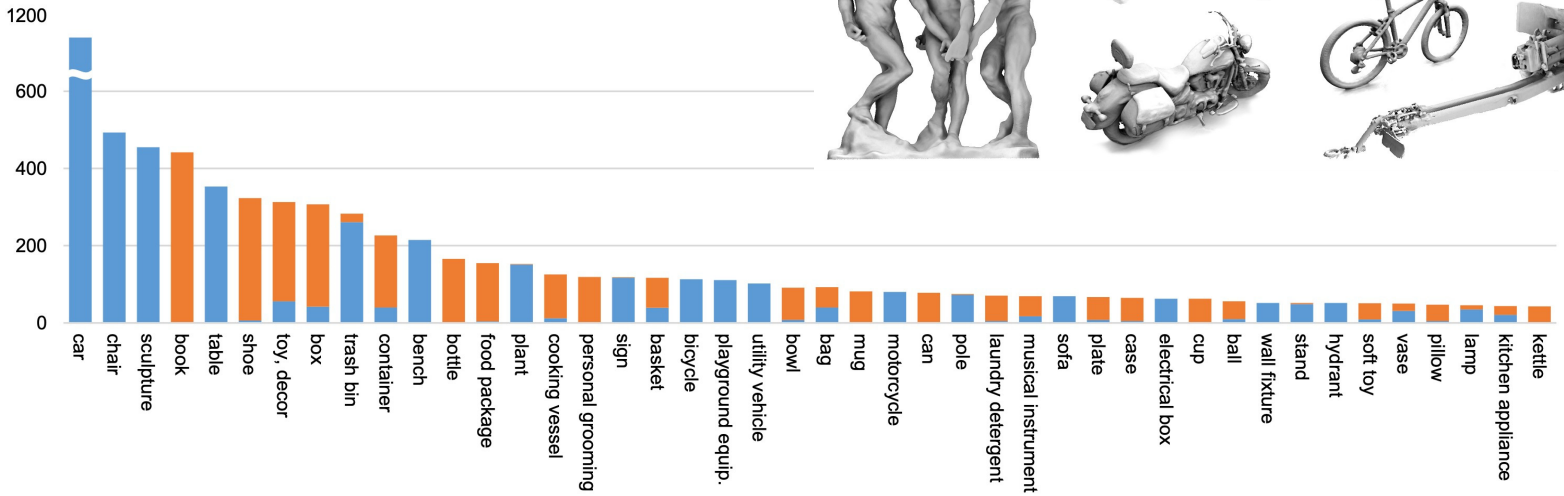
Chang et al. ShapeNet. arXiv 2015
Wu et al. 3D ShapeNets. CVPR 2015

Datasets for 3D Scanned Object Classification

Redwood-3D Scan

10,933 RGBD scans

441 models



Choi et al, arXiv 2016

Image + Reference 3D Models

Pascal 3D+

Retrieve a nearest-neighbor 3D model for objects in real images

8,505 PASCAL images (13,898 instances) + 22,394 ImageNet images

12 rigid categories, 3,000+ instances per category on average

Xiang et al, WACV 2014

Image + Reference 3D Models
Pix3D

10,069 images
395 shapes (IKEA furniture + 3D scan)

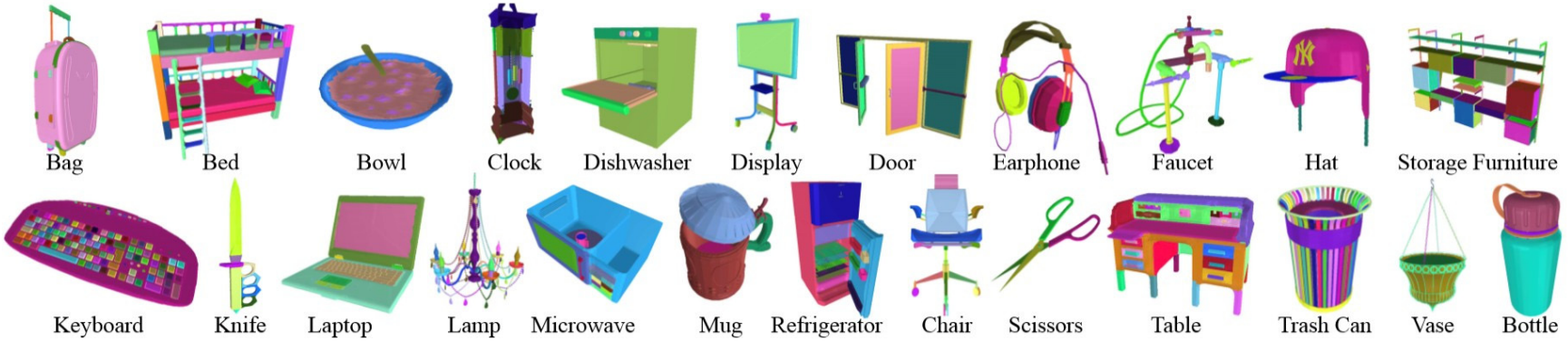



Sun et al. CVPR 2018, building upon Lim et al. ICCV 2013

Labeled Parts of 3D objects
PartNet

26K models, 574K labeled parts

Fine-grained and Hierarchical



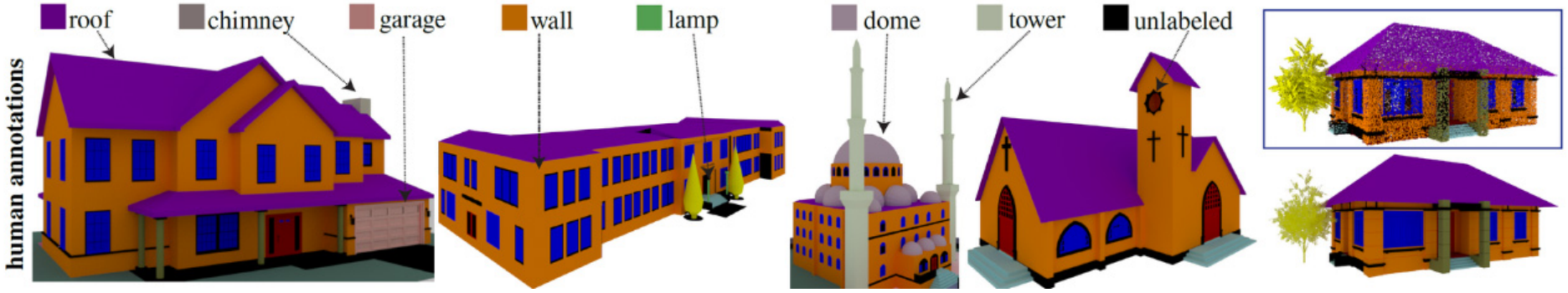
Mo et al. CVPR 2019

Slide credit: Hao Su

Labeled Parts of 3D buildings
BuildingNet

2K models, 513K labeled parts

Focuses on large-scale structures (buildings)



Selvaraju et al.,
ICCV 2021

Datasets of Indoor Scanned 3D Scenes

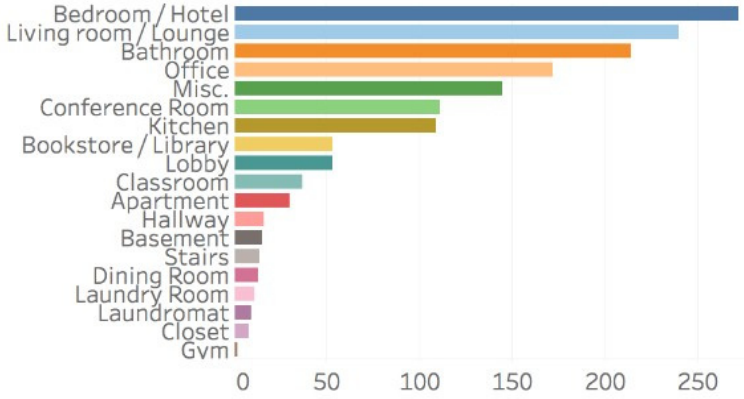
ScanNet

2.5M Views in 1,500 RGBD scans

3D camera poses

Surface reconstructions

Semantic segmentations



Dai et al. CVPR 2017. Slide credit: Hao Su

Datasets of Indoor Scanned 3D Scenes

SceneNet

Large-scale Synthetic Scenes3D meshes

- 5M Photorealistic Images

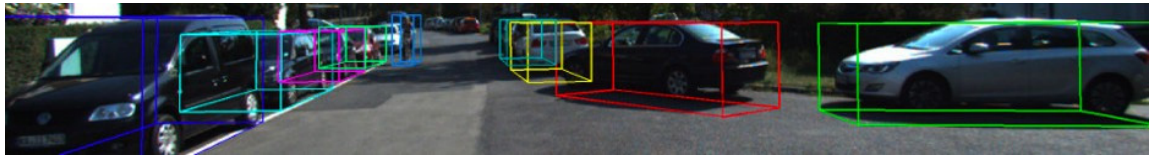
McCormac et al. ICCV 2017

Slide credit: Hao Su

Datasets of Outdoor 3D Scenes

KITTI

- 3D bounding boxes

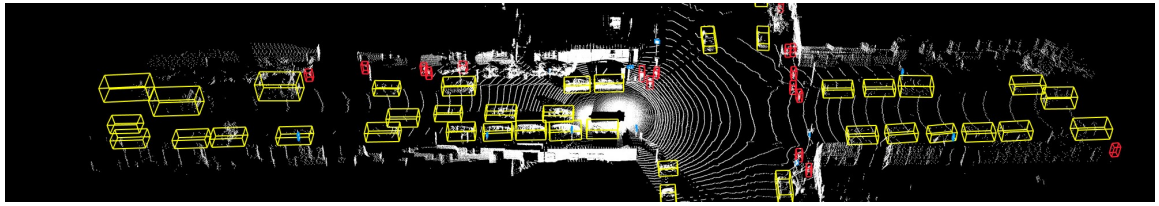


Semantic KITTI

- Point-level label

Waymo Open Dataset

- 3D bounding boxes

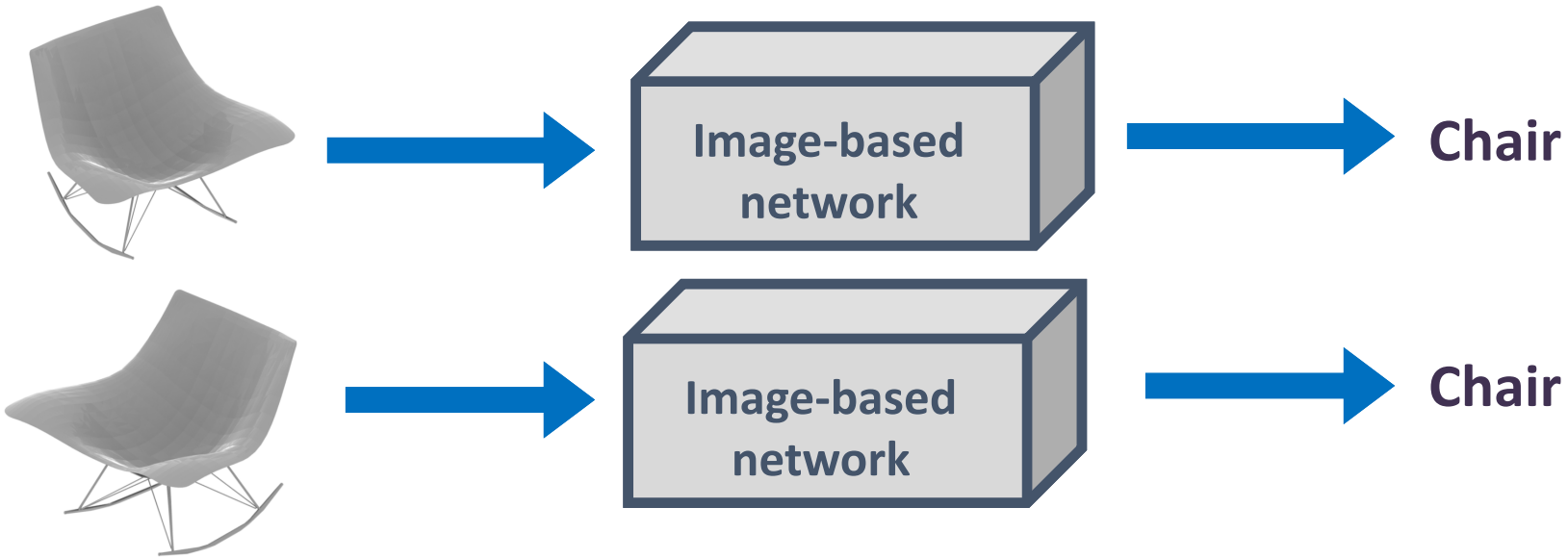


3D Deep Learning approaches

- **The Multi-View approach**
  - **Recognition**
  - Segmentation
  - Correspondences
- The Voxel approach
- The Point approach
- The Graph approach

Motivation

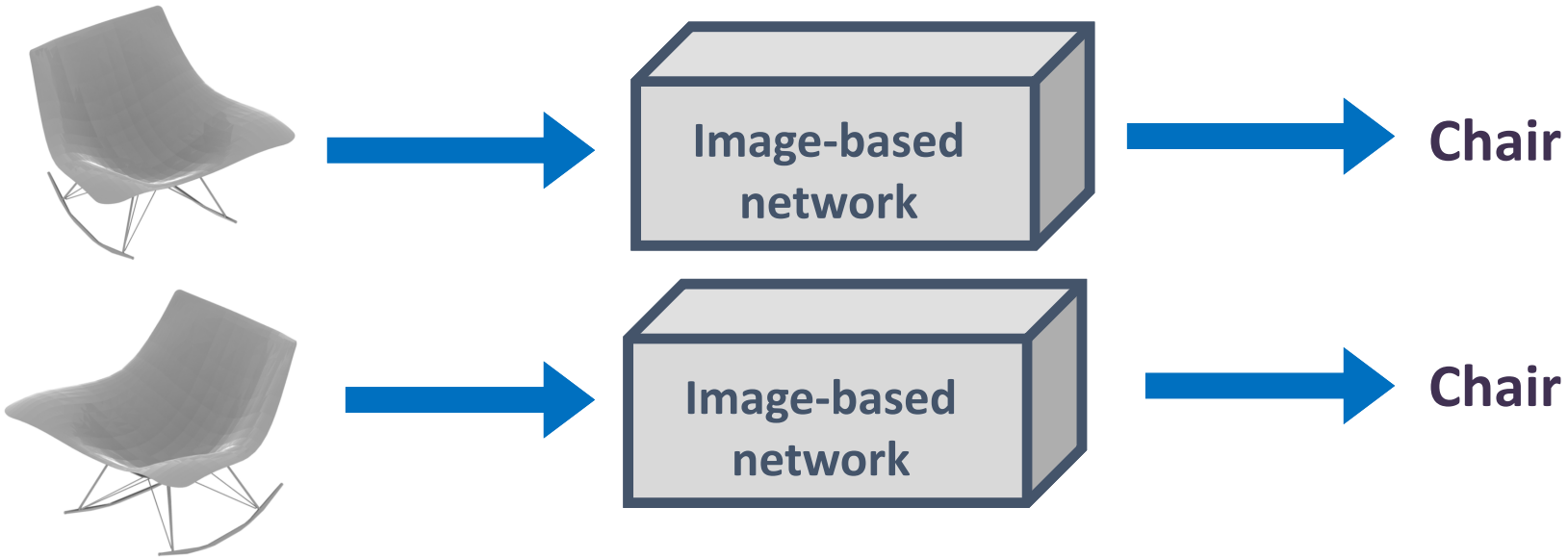
Image-based nets can process individual shape renderings.



Motivation

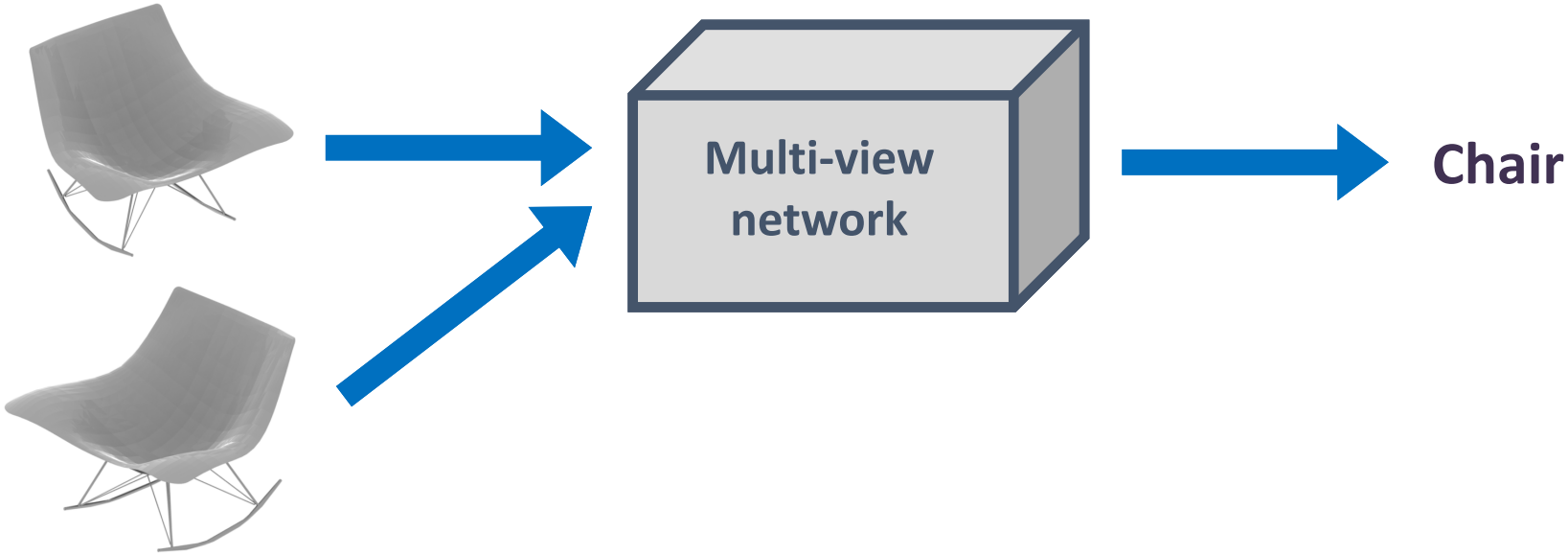
Image-based nets can process individual shape renderings.

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



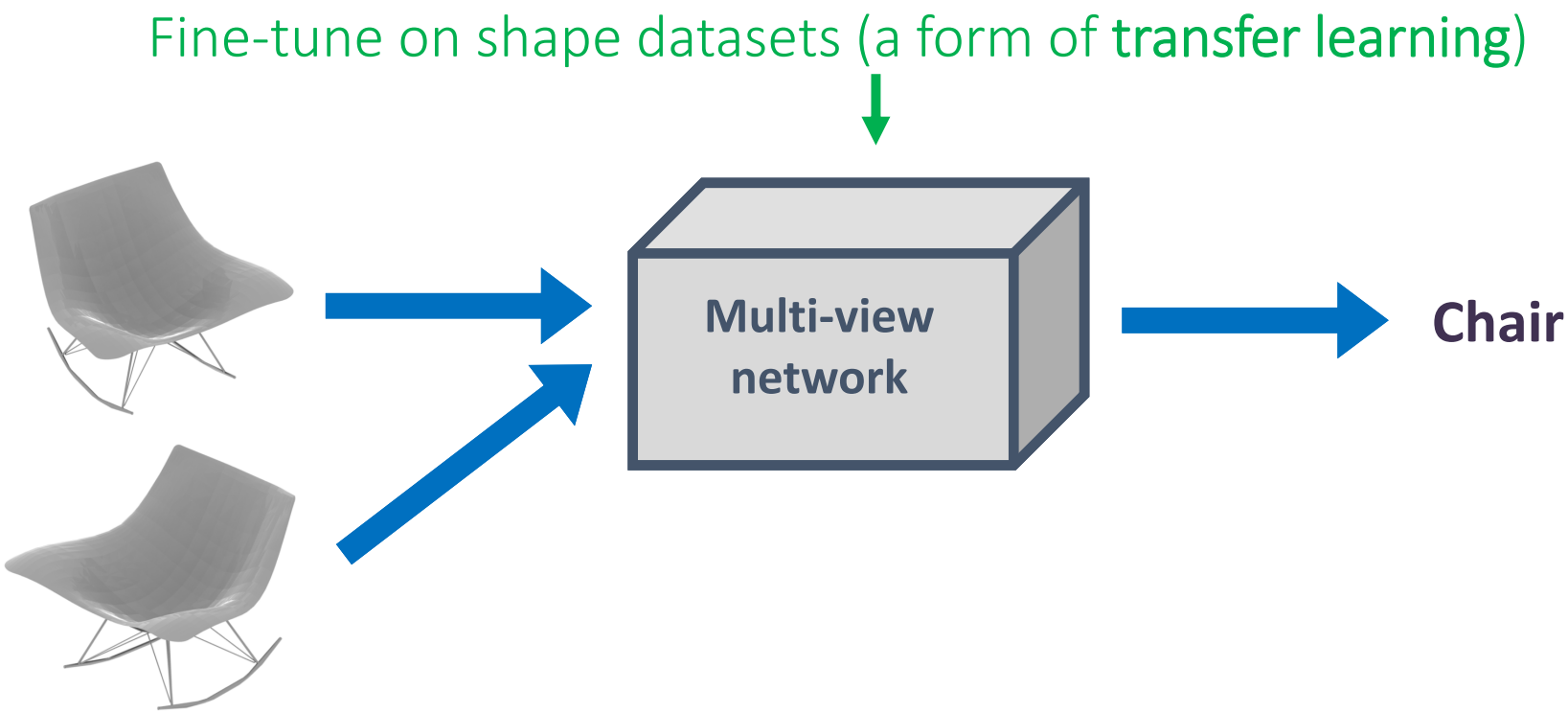
The multi-view approach

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

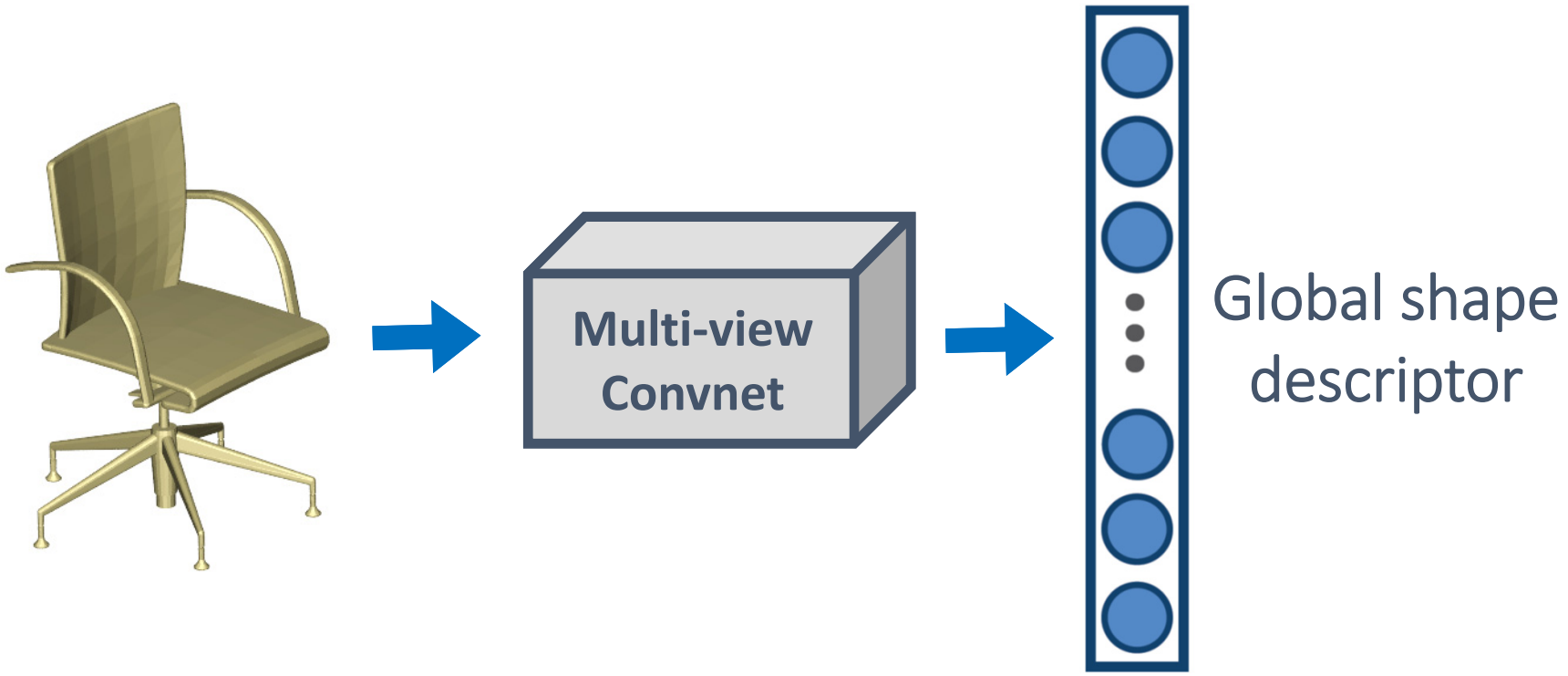


The multi-view approach

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

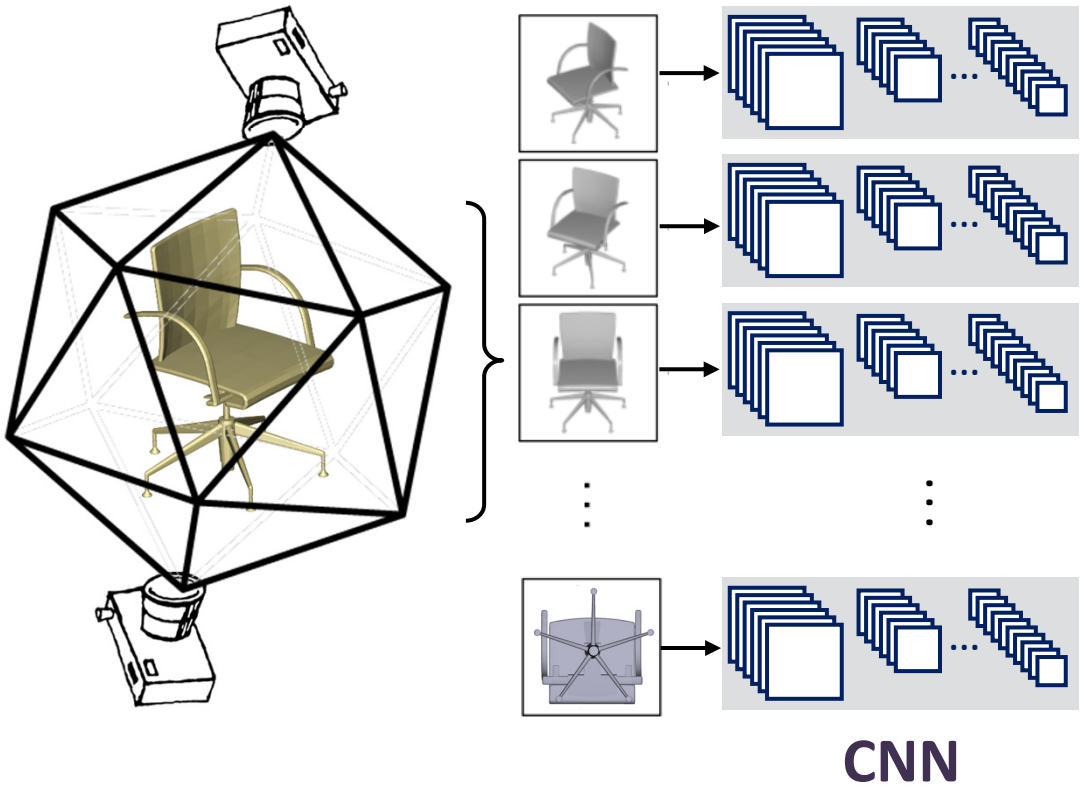


Goal

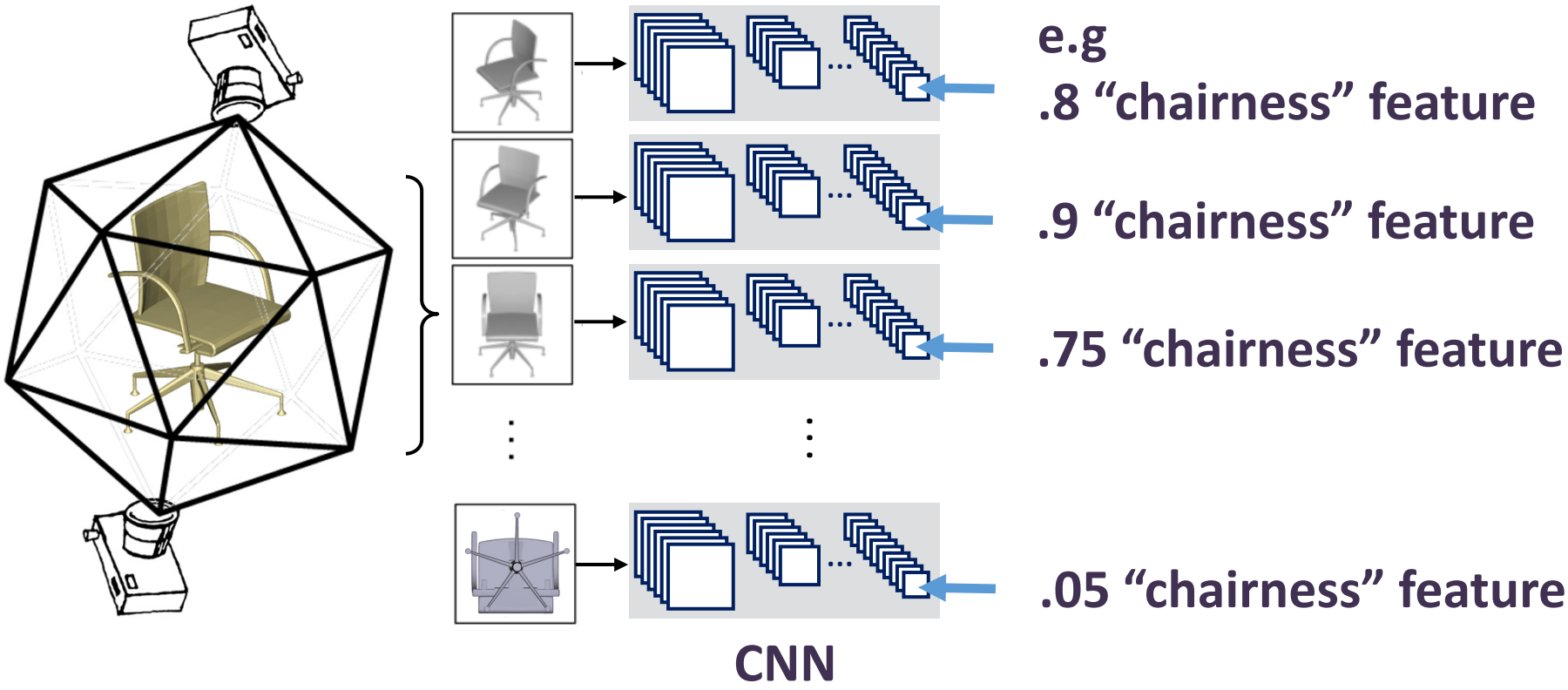


Su, Maji, Kalogerakis, Learned-Miller, ICCV 2015

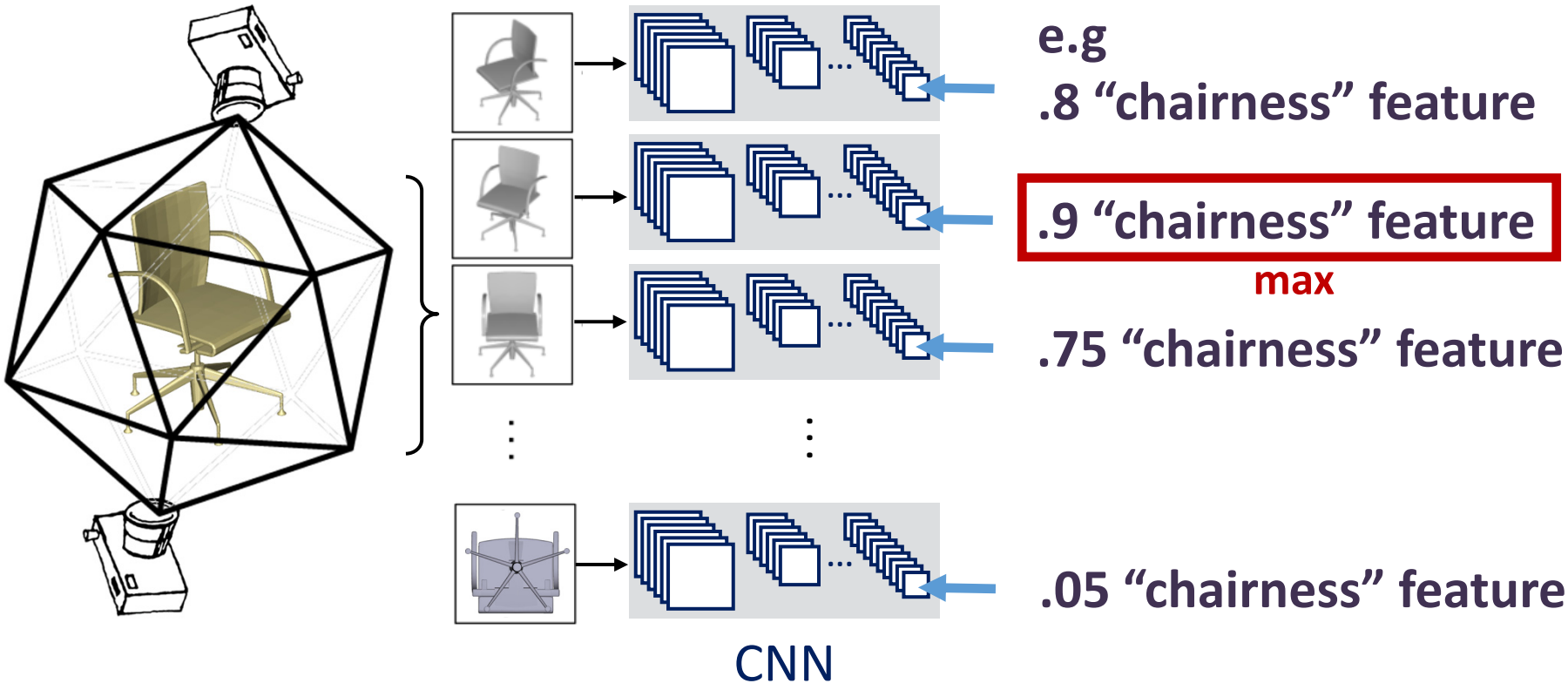
Shape recognition with multi-view CNNs



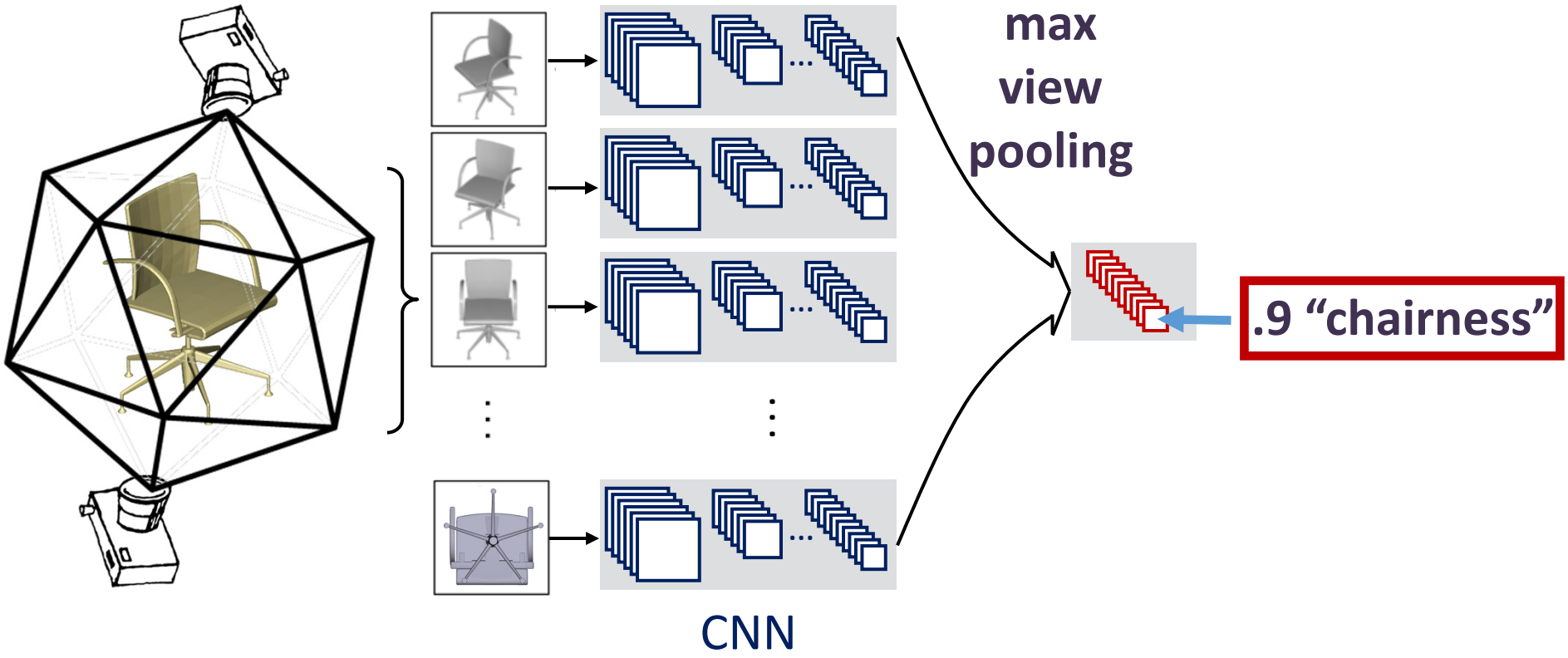
View Pooling



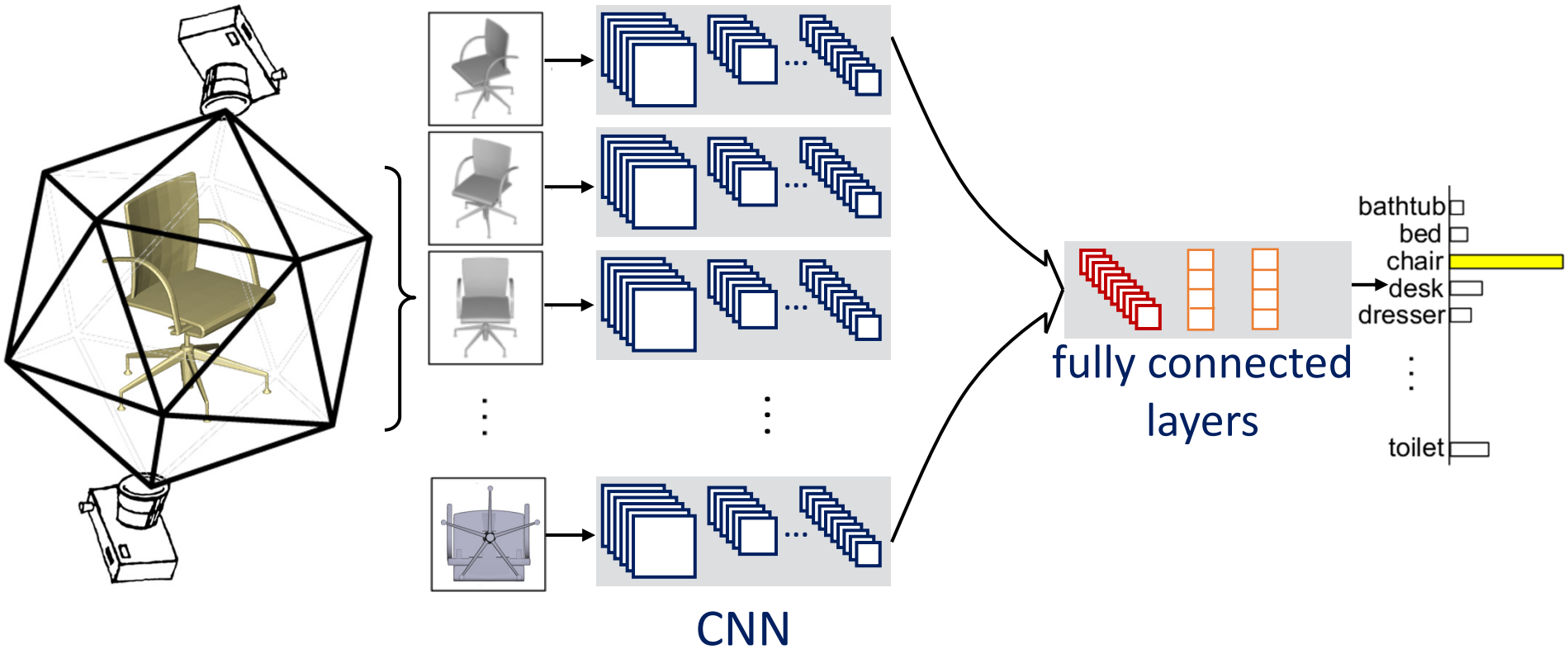
View Pooling



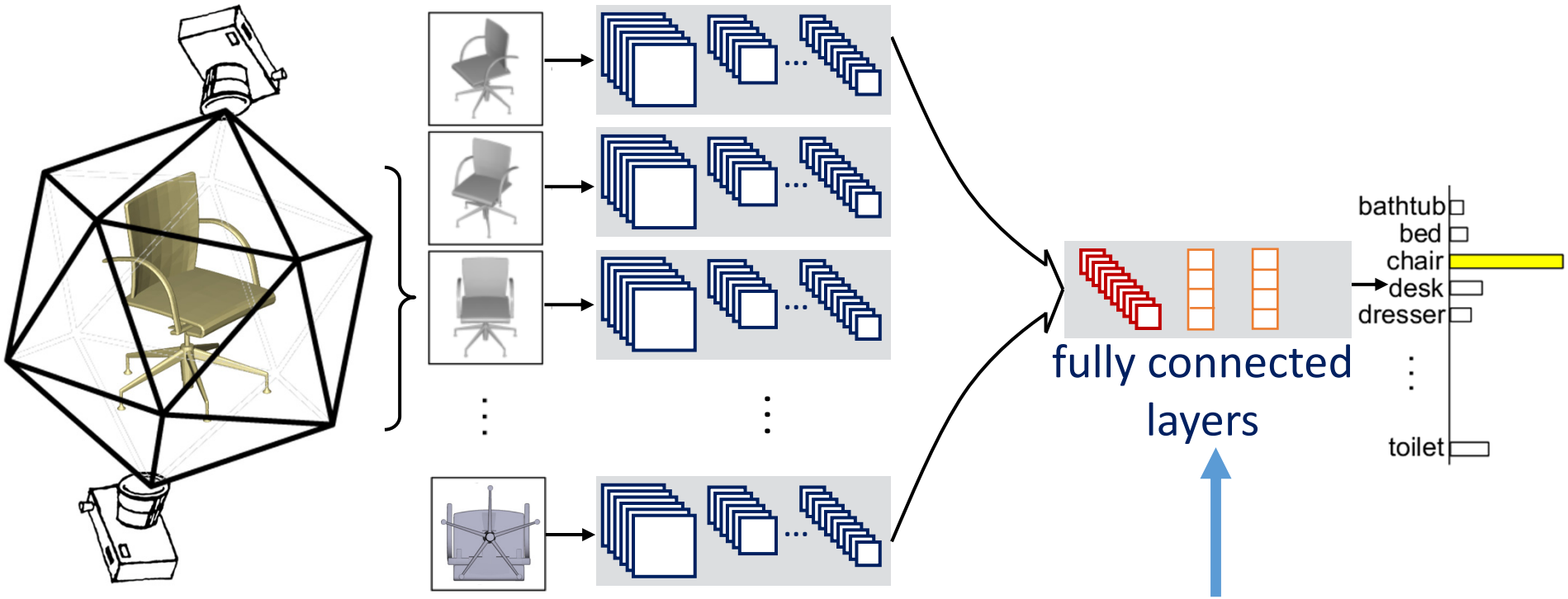
View Pooling



Classification



Training



Softmax loss: $$L(\mathbf{w}) = -\sum_{\text{shape } i} \log P(y_i = y_i^{(gt)})$$

ModelNet40: Classification & Retrieval

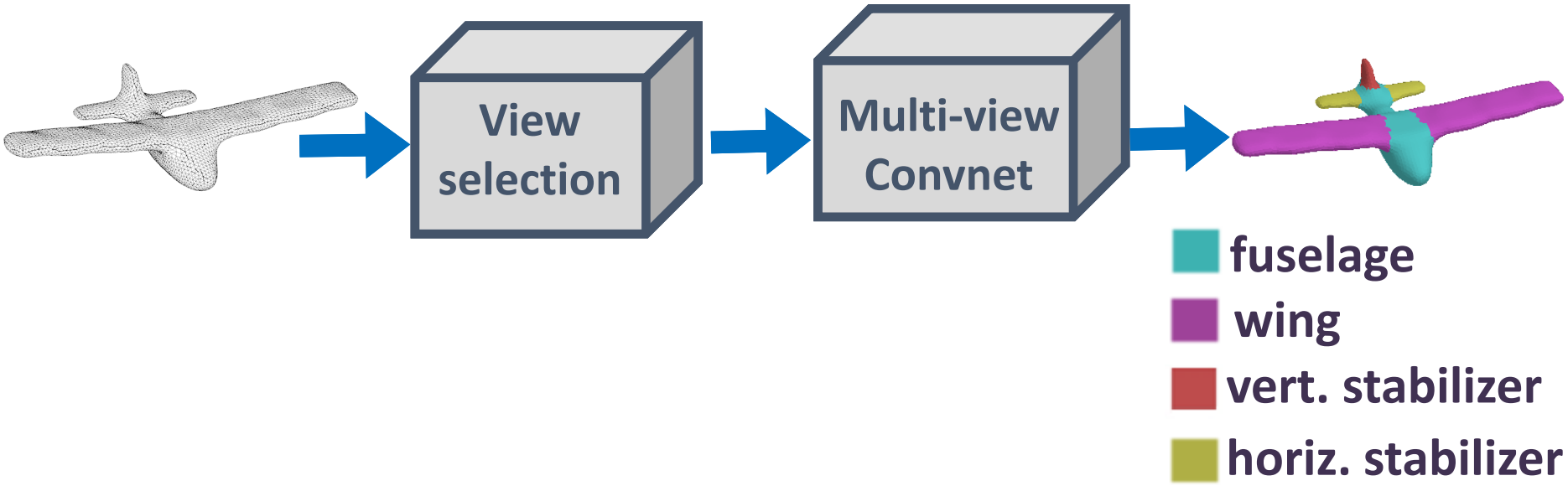
Method	Classification (Accuracy)
Spherical Harmonics [Kazhdan et al.]	68.2%
LightField [Chen et al.]	75.5%
Volumetric Net [Wu et al.]	77.3%
ImageNet-trained CNN (VGG-M, 1 view)	83.0%
Multi-view convnet (MVCNN)	90.1%

See also another recent multi-view kind of network:
RotationNet: Joint Object Categorization and Pose Estimation Using Multiviews from Unsupervised Viewpoints (97% performance in ModelNet40)
https://arxiv.org/abs/1603.06208

3D Deep Learning approaches

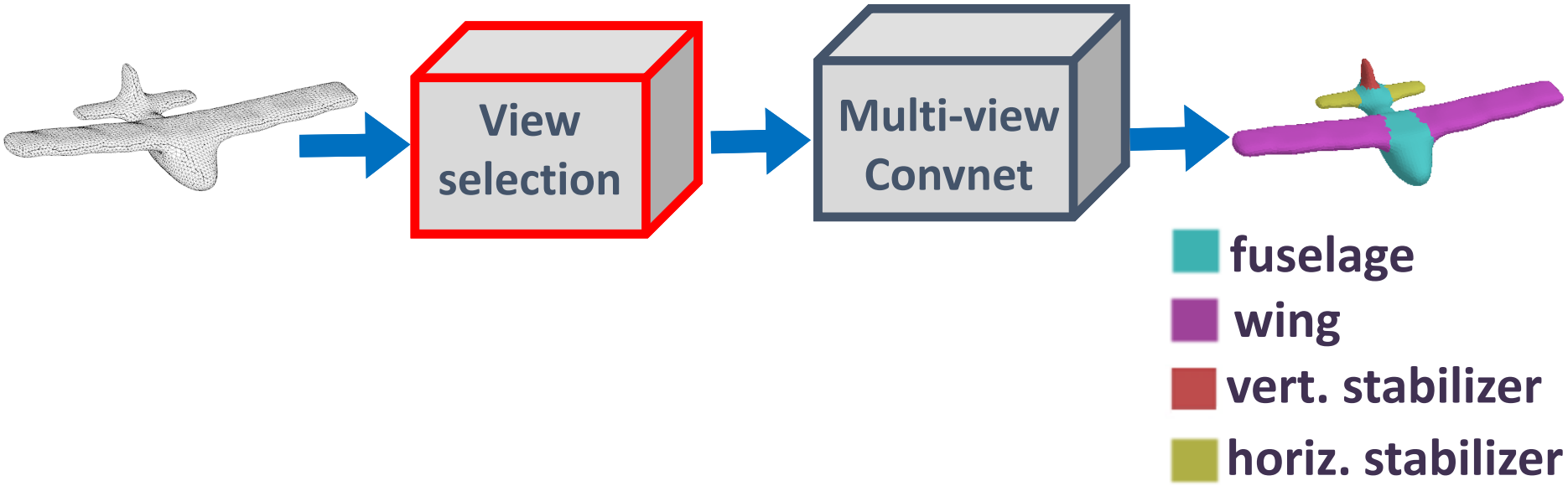
- **The Multi-View approach**
  - Recognition
  - **Segmentation**
  - Correspondences
- The Voxel approach
- The Point approach
- The Graph approach

View-based convnets for 3D shapes - Segmentation Pipeline



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

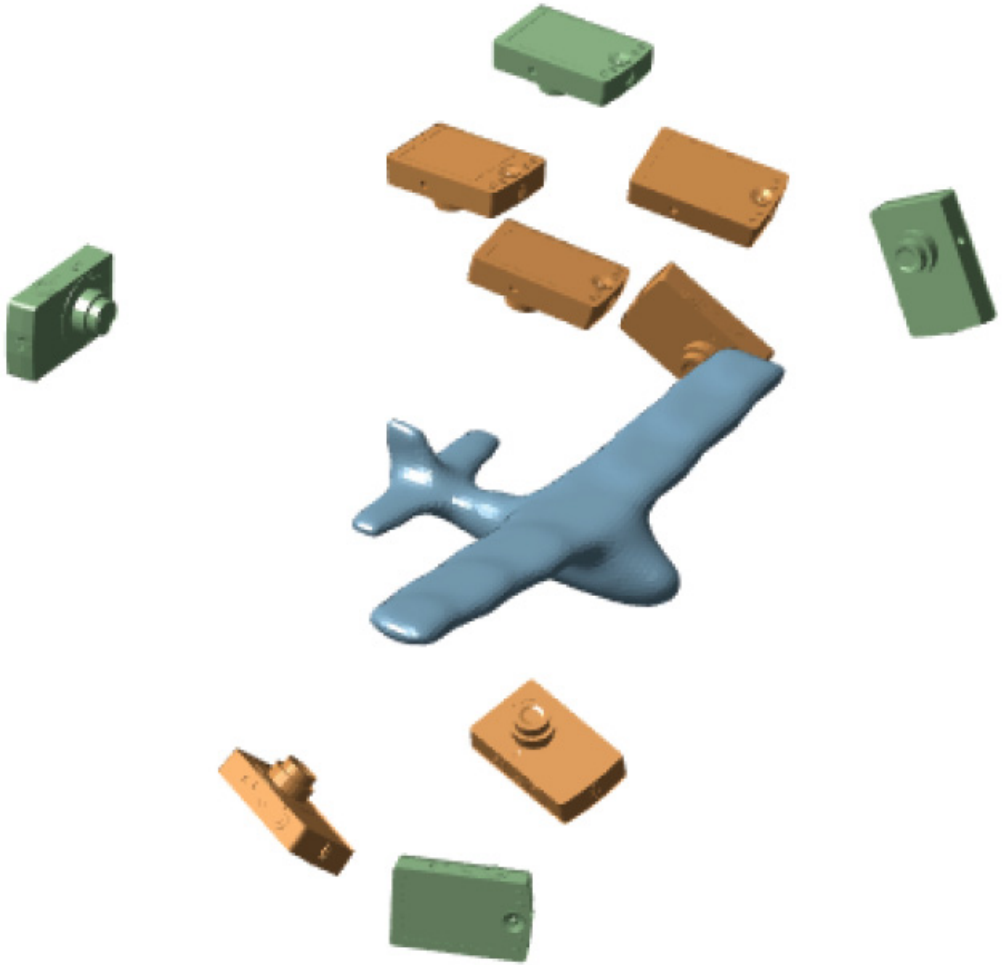
View-based convnets for 3D shapes - Segmentation Pipeline



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

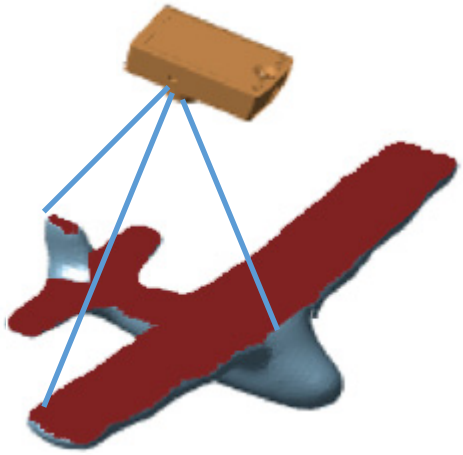
Input: shape as a collection of rendered views

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



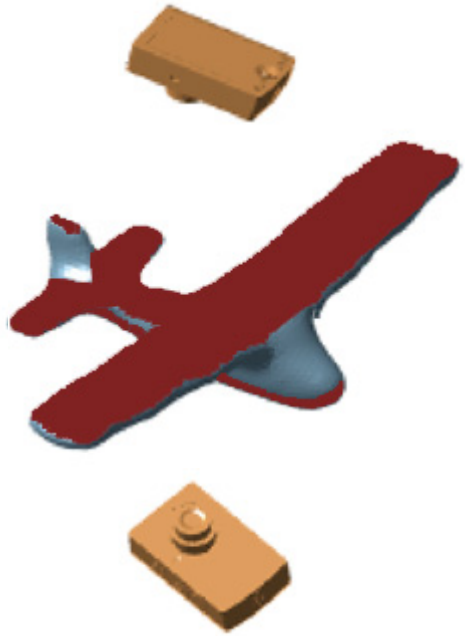
Input: shape as a collection of rendered views

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**.

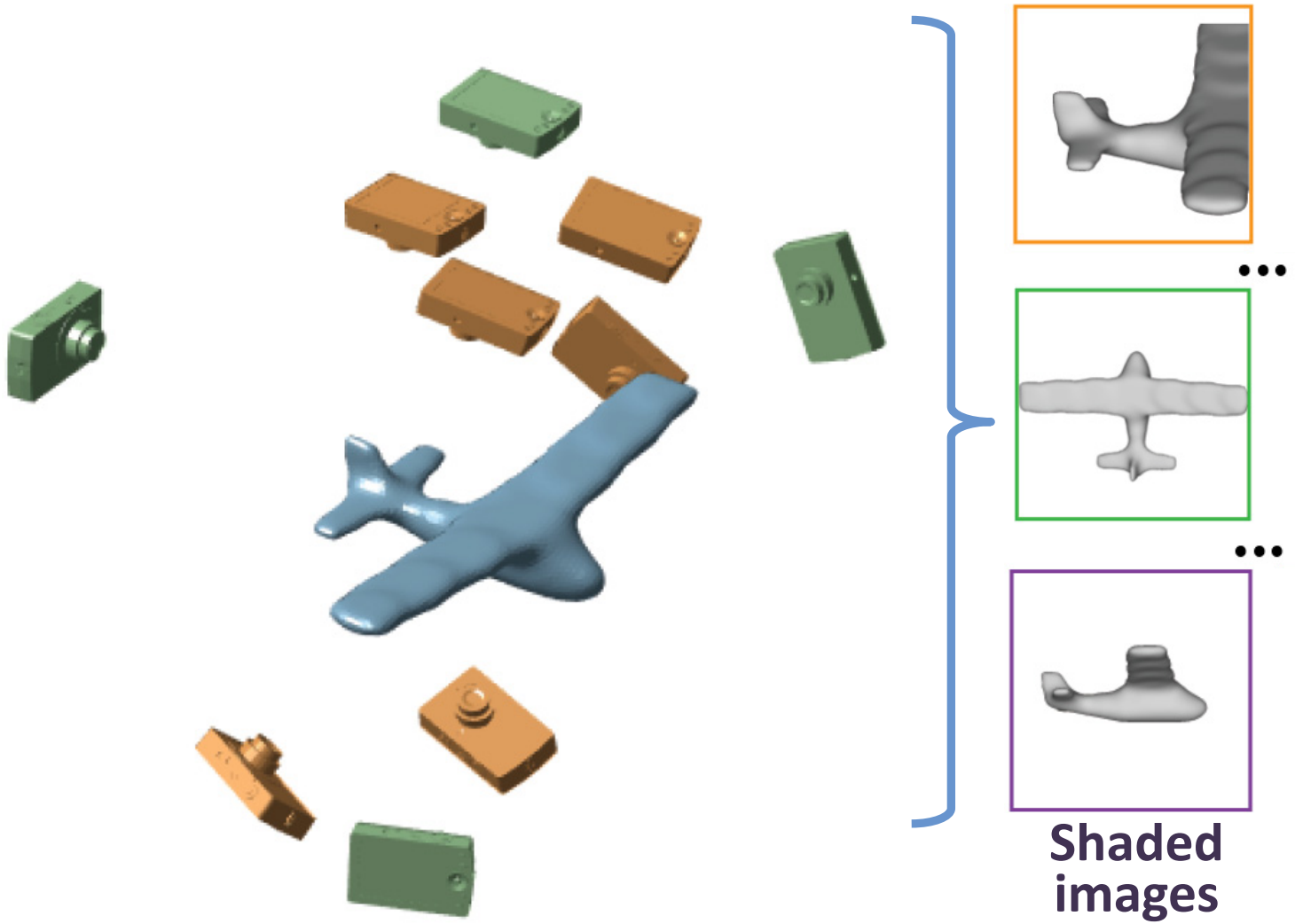


Input: shape as a collection of rendered views

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

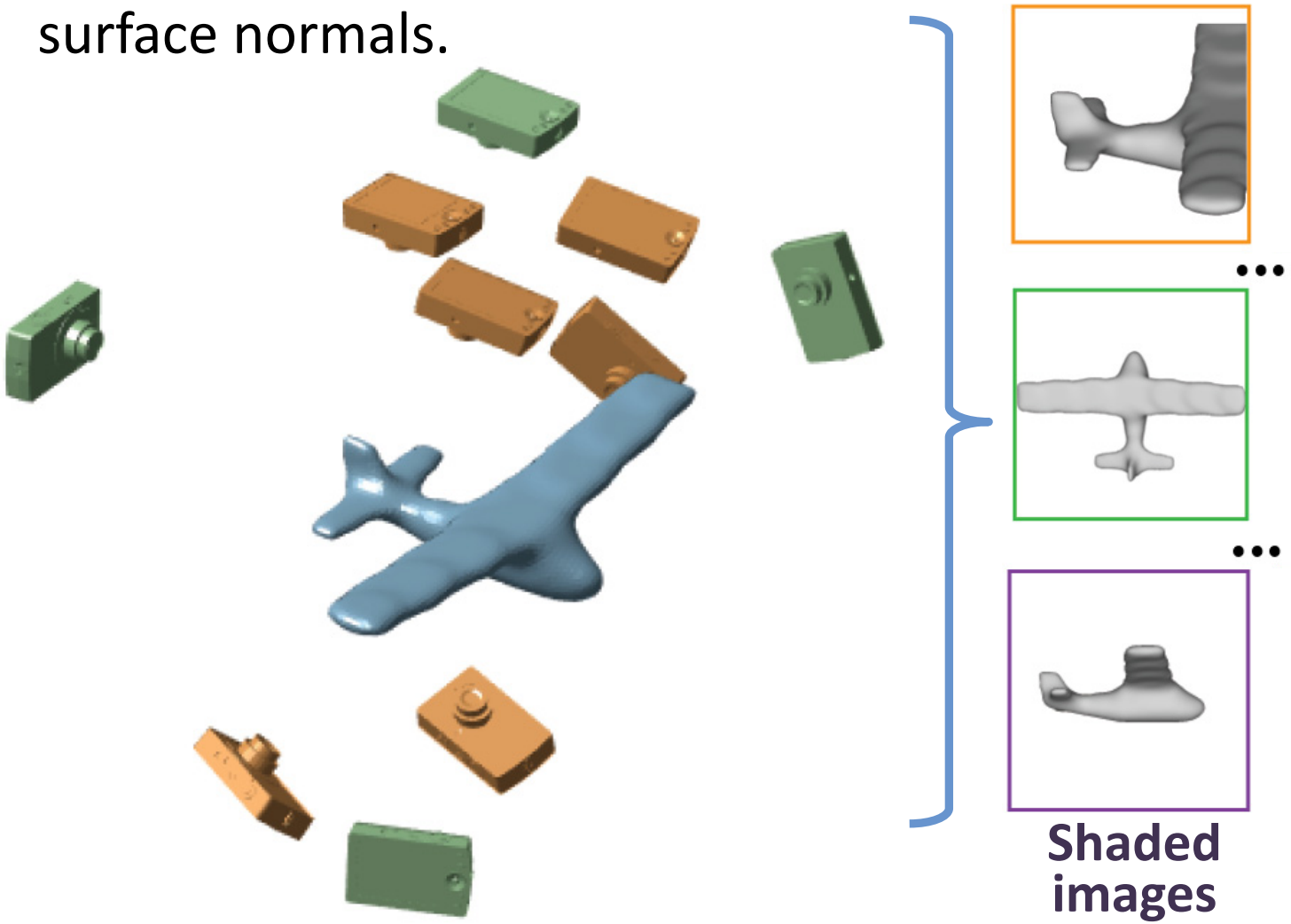
Input: shape as a collection of rendered views

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



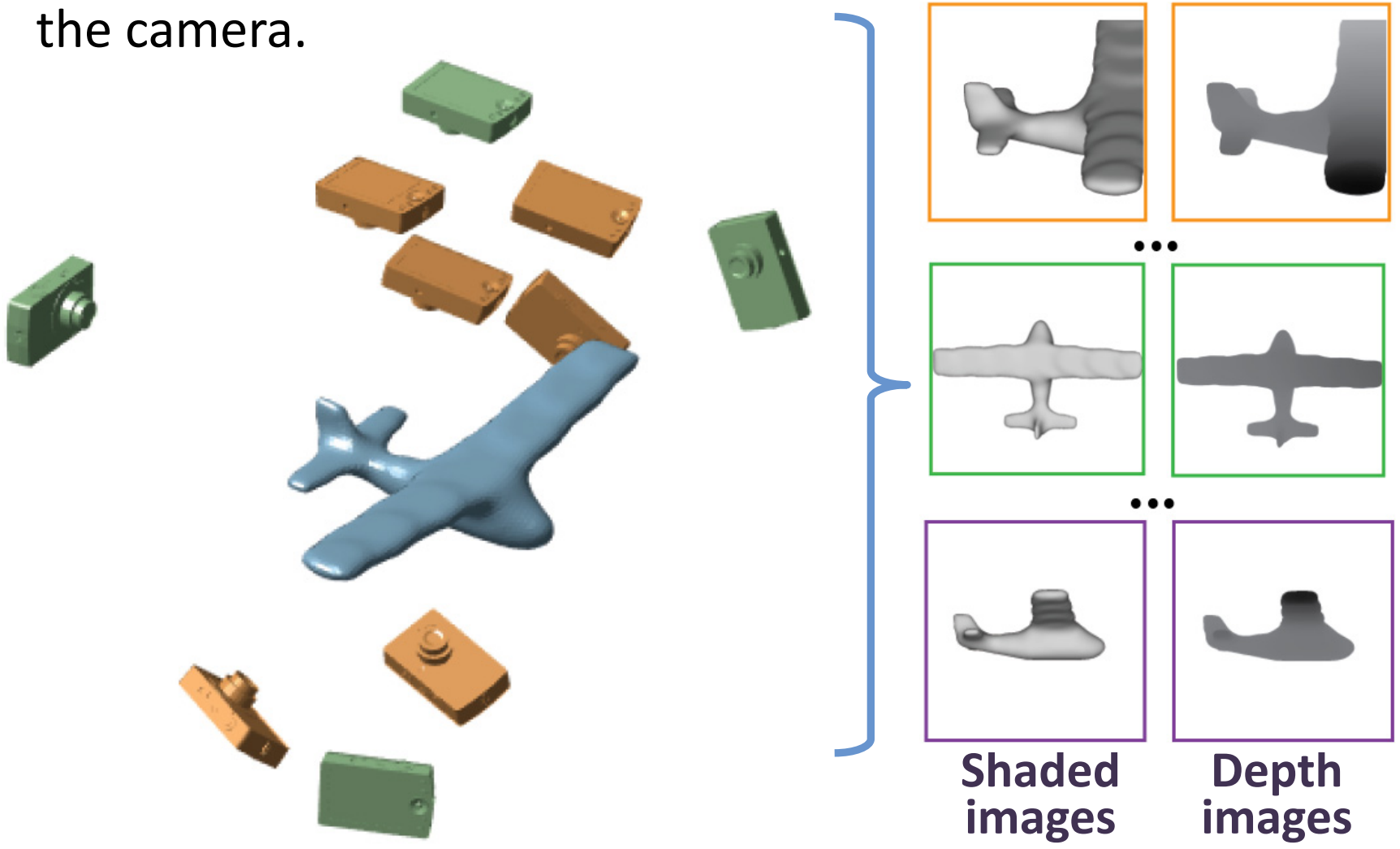
Input: shape as a collection of rendered views

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



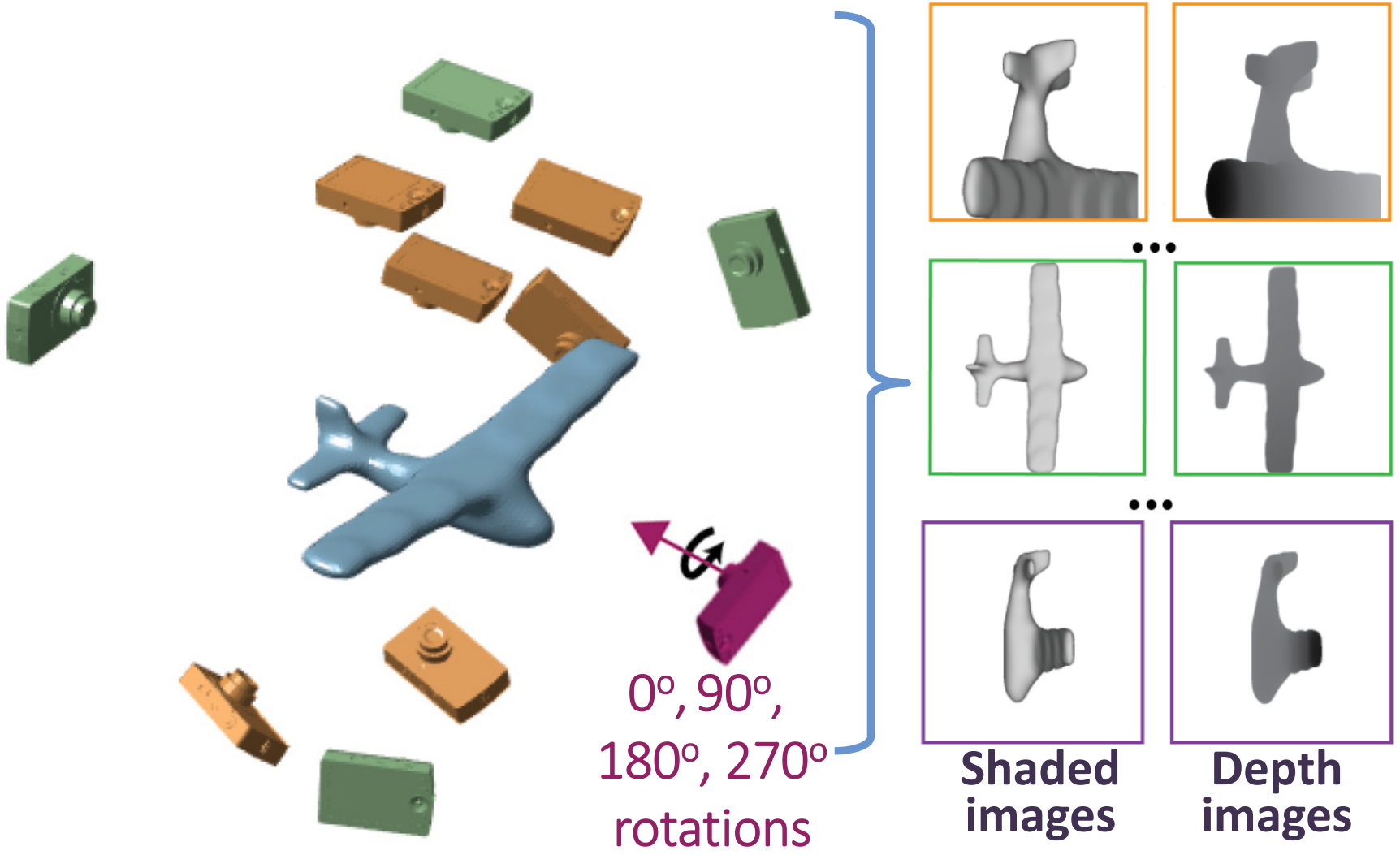
Input: shape as a collection of rendered views

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



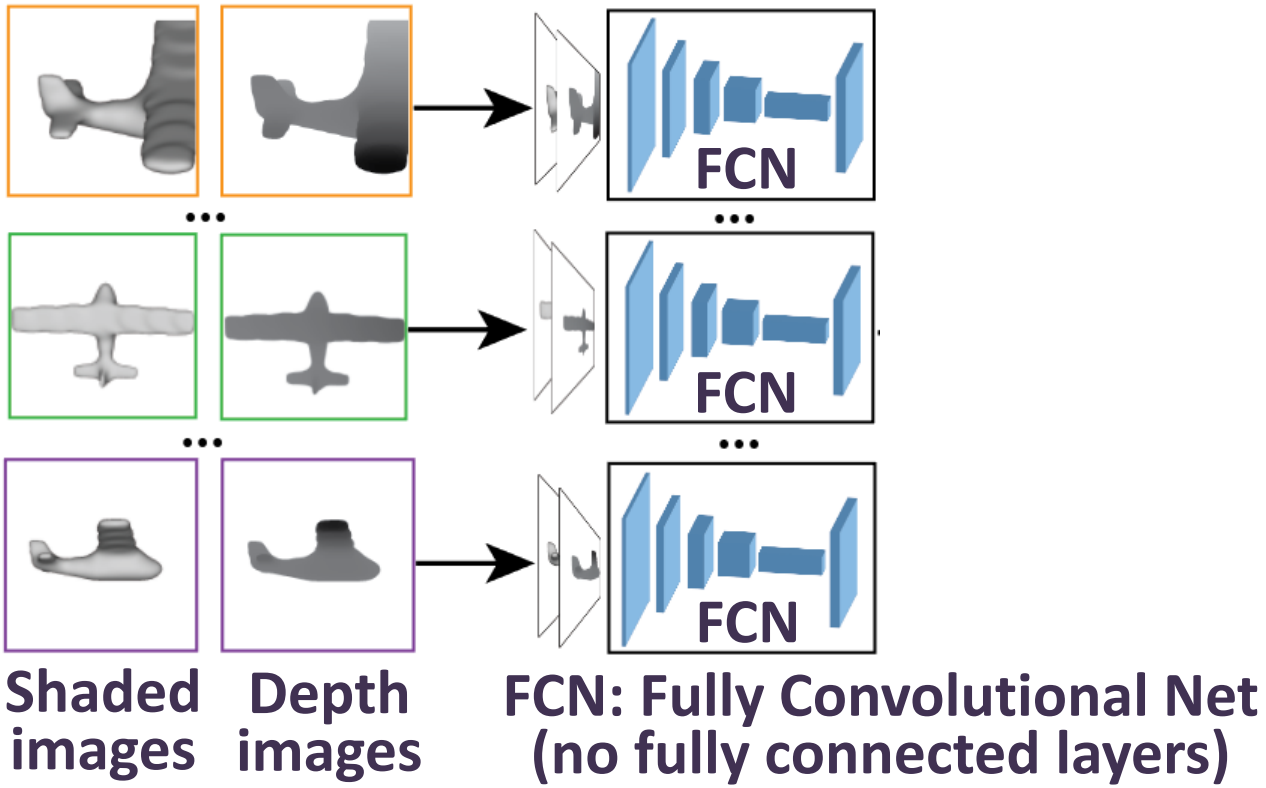
Input: shape as a collection of rendered views

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



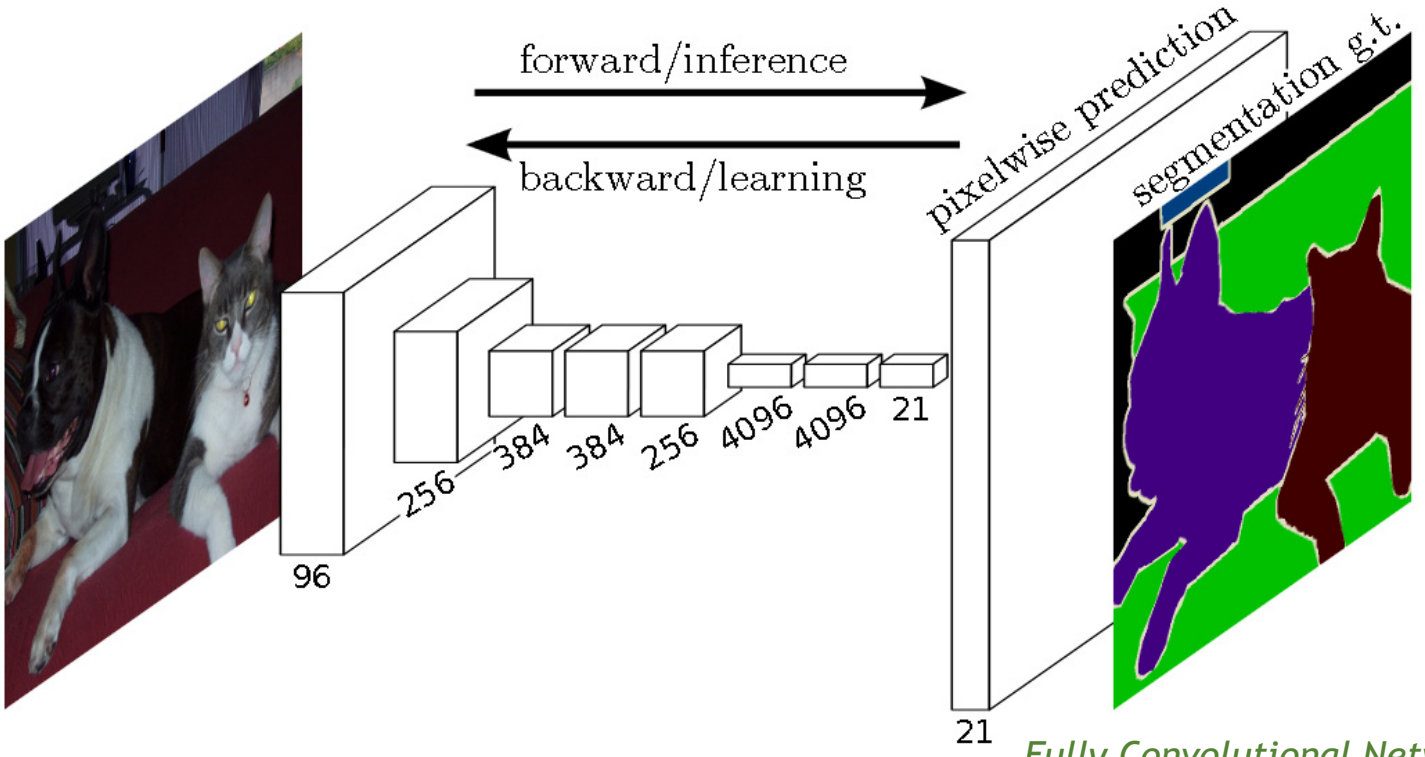
Projective convnet architecture

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



Fully Convolutional Nets

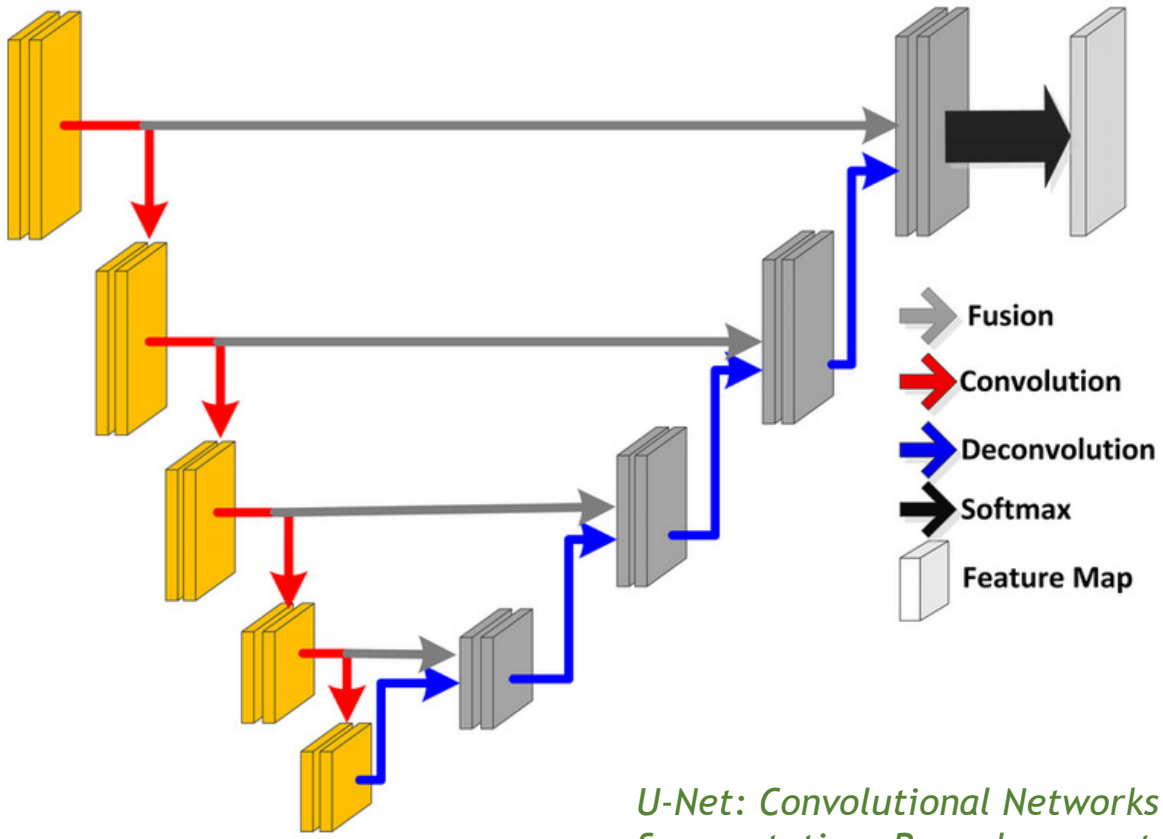
Instead of predicting a property for the whole image, output **dense predictions** e.g., probabilities of part labels per pixel **without fully connected layers**



Fully Convolutional Networks for Semantic Segmentation, Long et al.

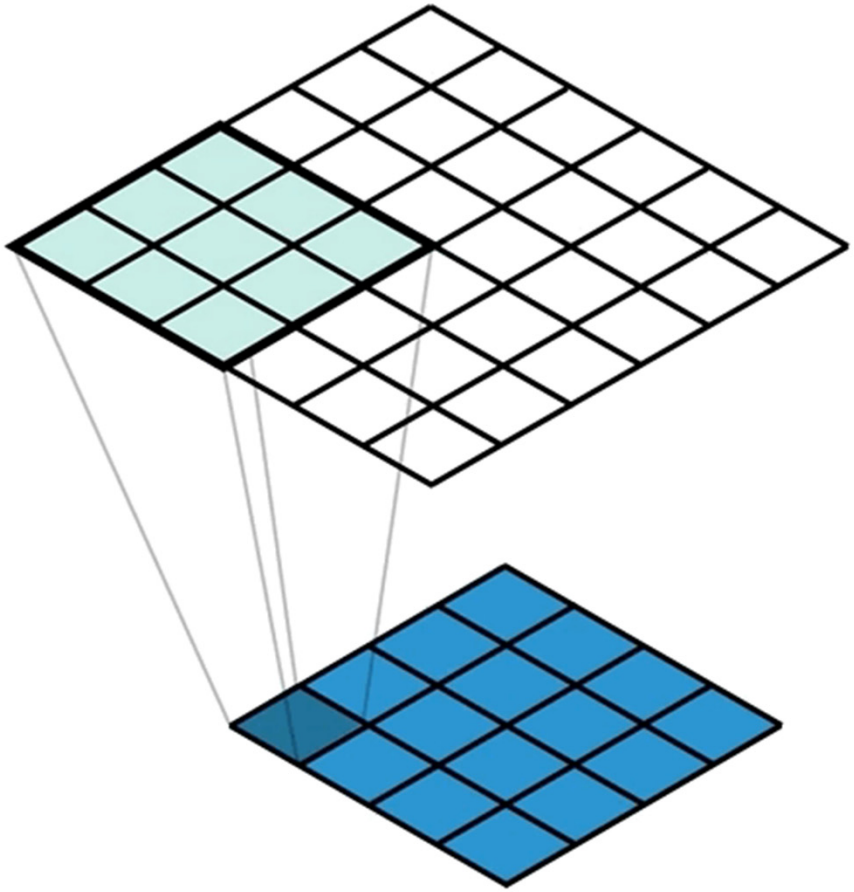
“U-Net” architecture

If the decoder relies exclusively on the last feature map of the encoder, then it can easily fail to produce fine-grained details. Concatenate feature maps of the encoder in the decoder (“U-Net”)



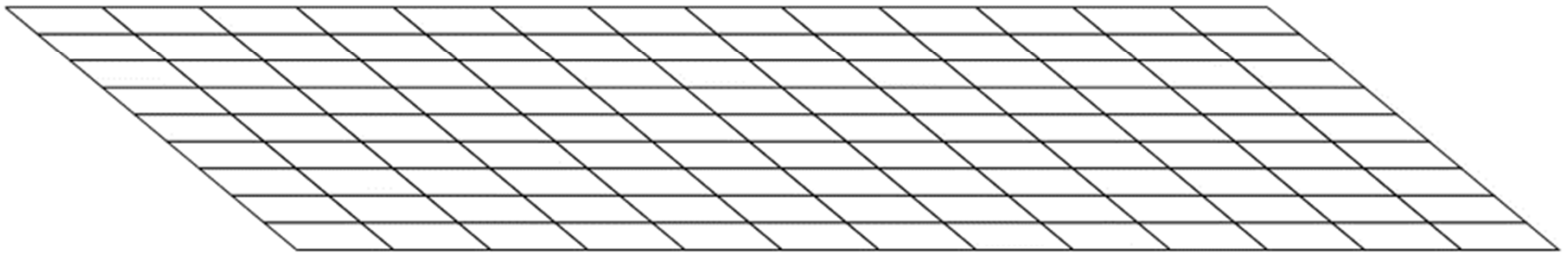
U-Net: Convolutional Networks for Biomedical Image Segmentation, Ronneberger et al.

Transpose Convolution



https://medium.com/apache-mxnet/transposed-convolutions-explained-with-ms-excel-52d13030c7e8

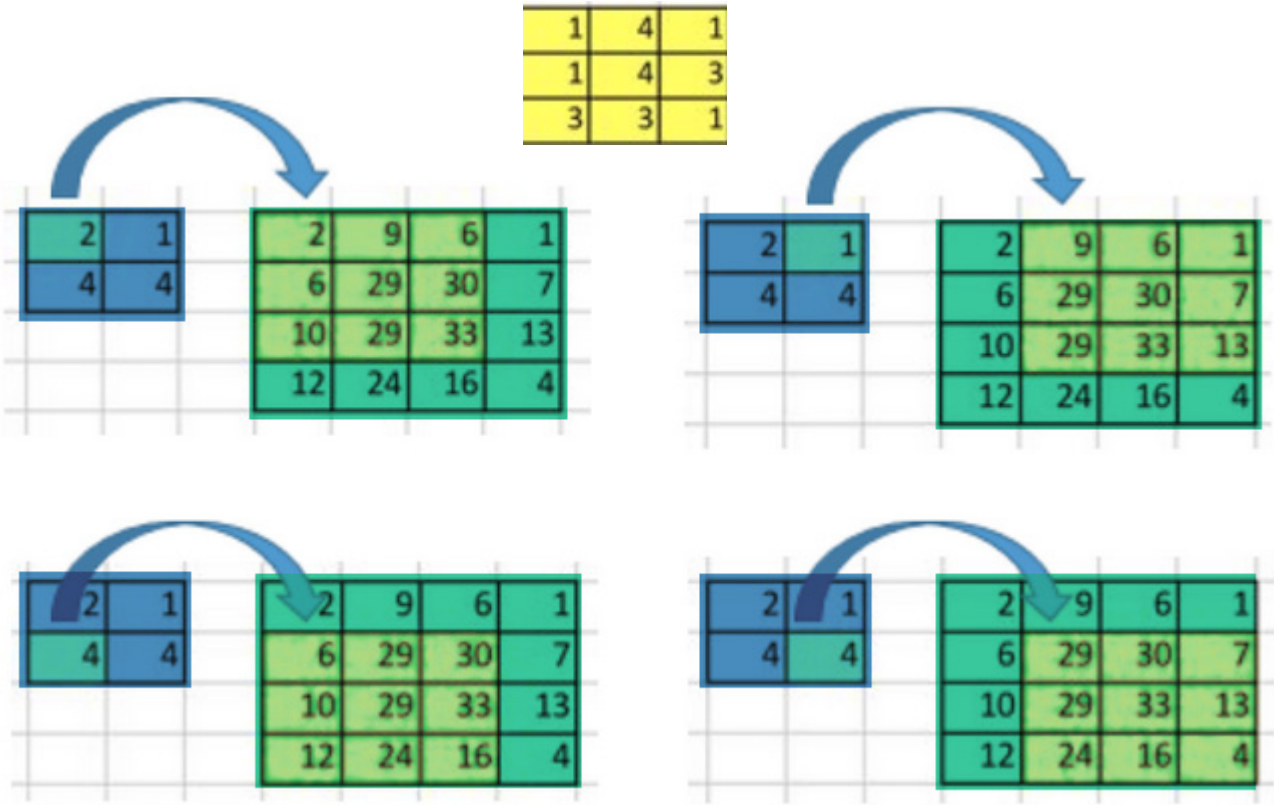
Transpose Convolution



See also: https://distill.pub/2016/deconv-checkerboard/

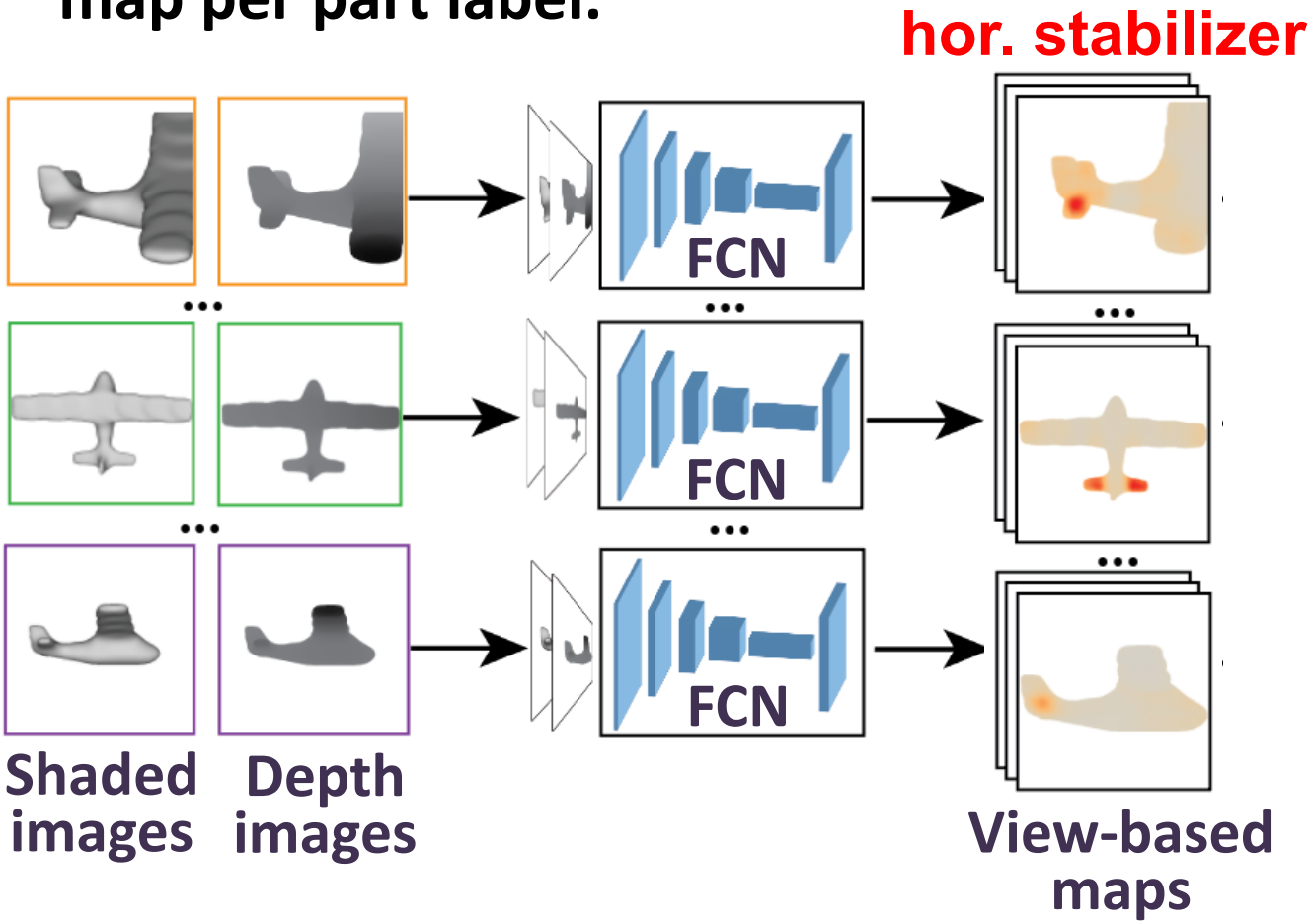
Transpose Convolution

Like convolution, but “reversed”. **Blue-ish** is the input image (2x2), filter weights are yellow (3x3), output is green-ish (4x4)



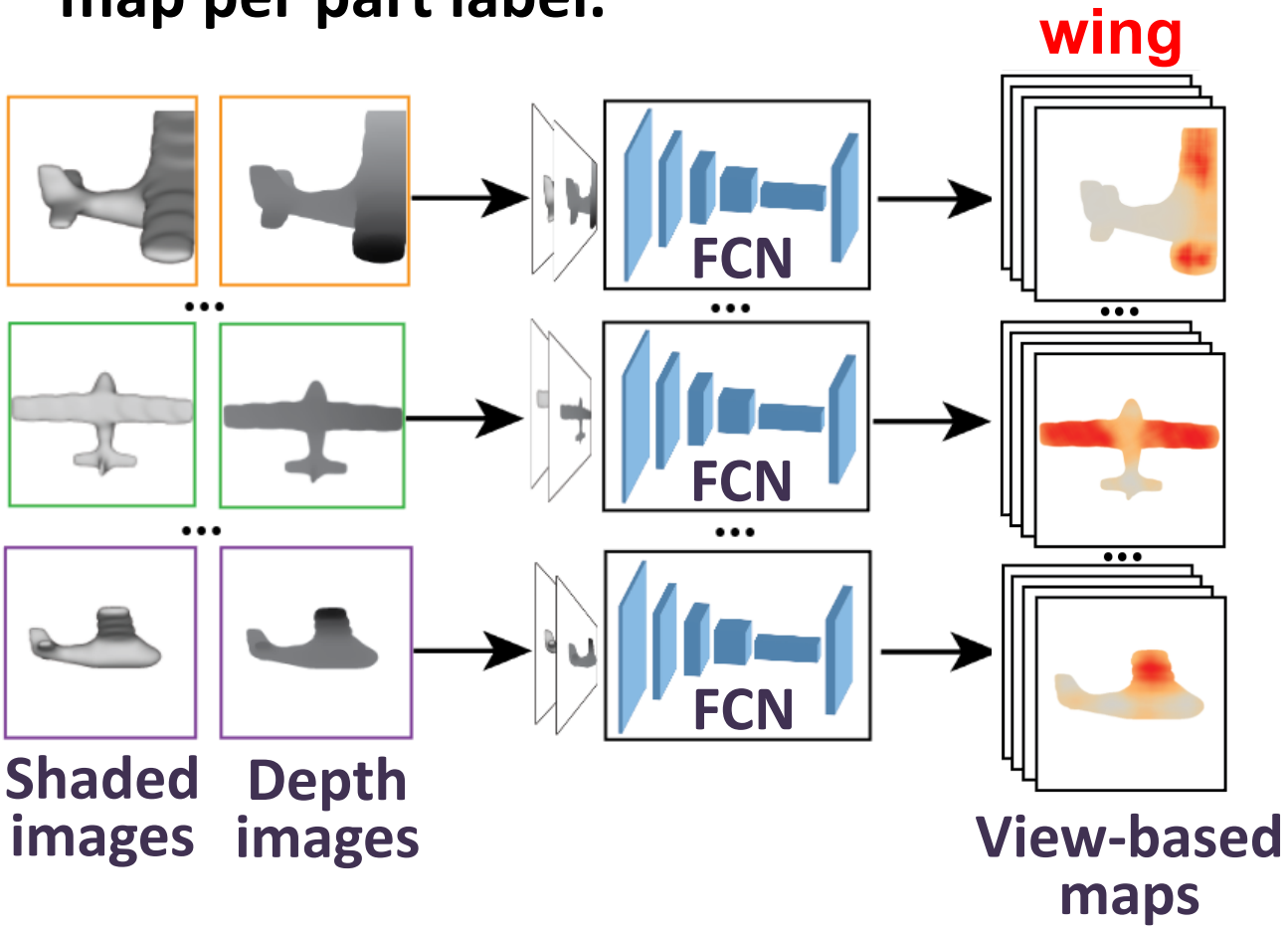
Projective convnet architecture

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



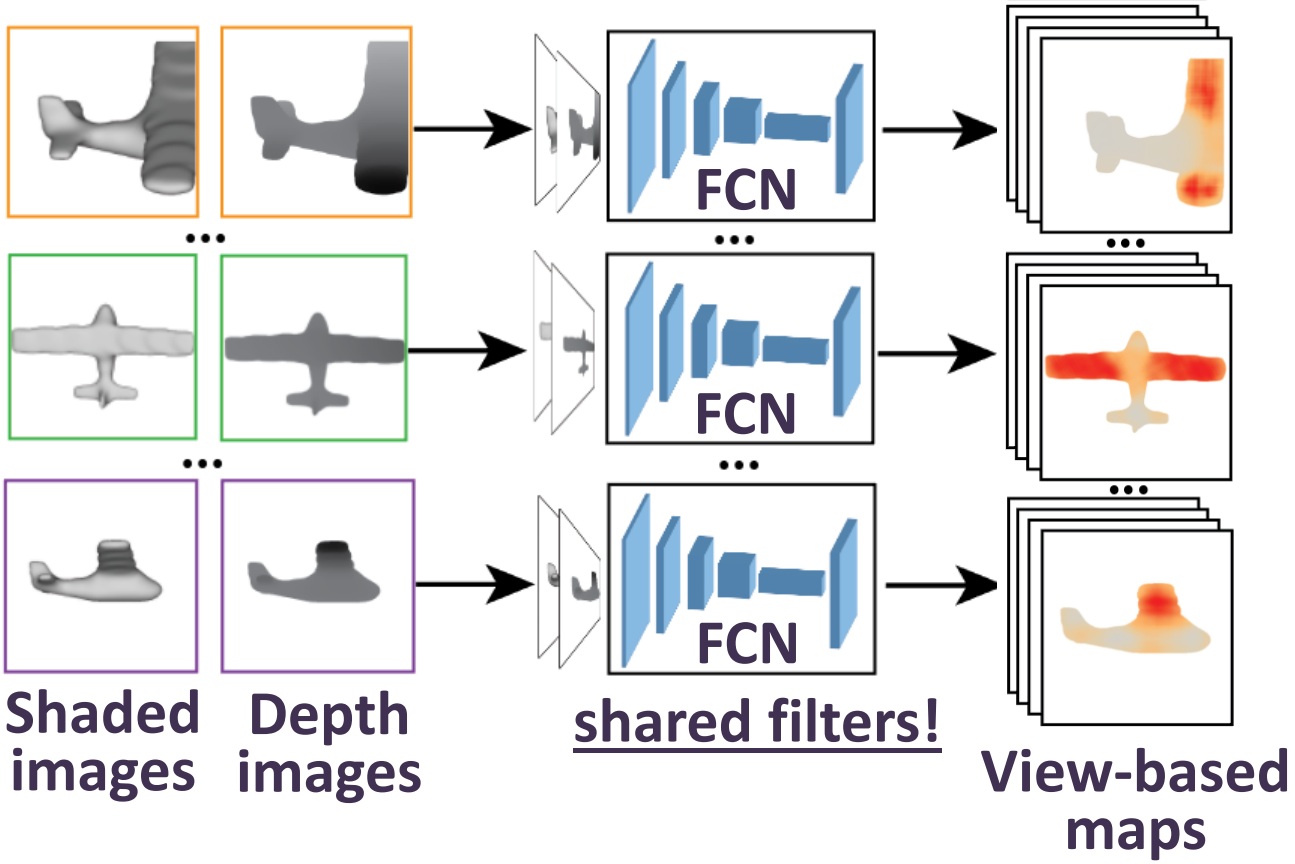
Projective convnet architecture

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



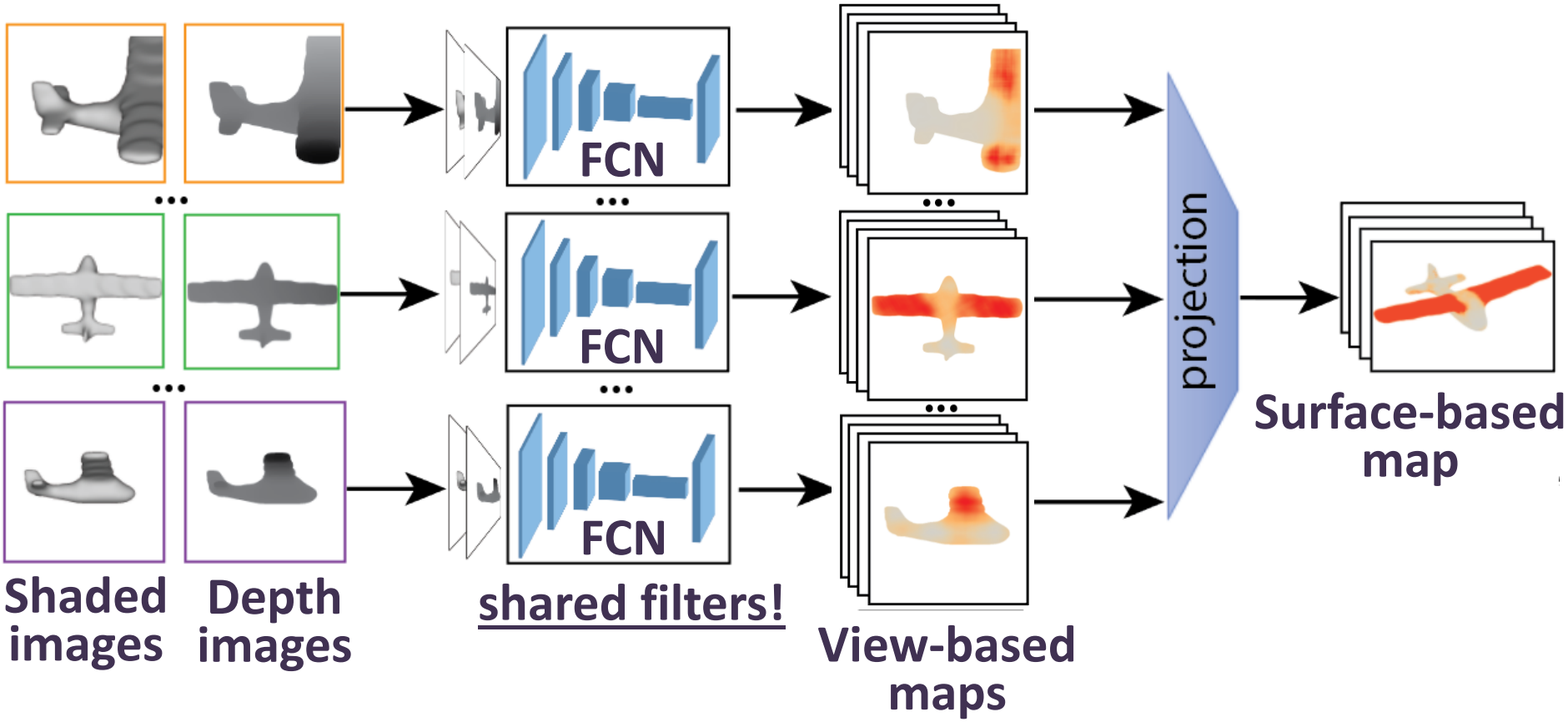
Projective convnet architecture

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



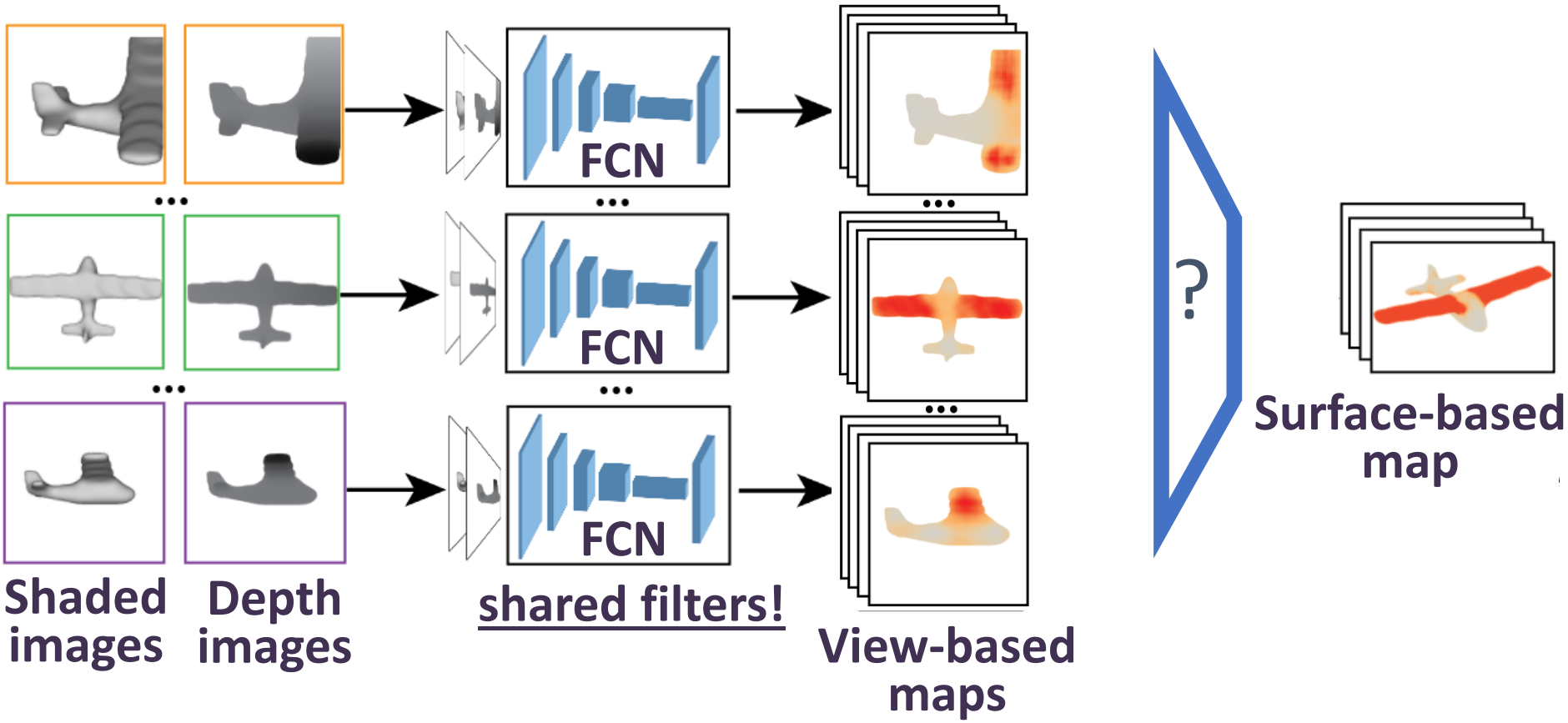
Projective convnet architecture

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



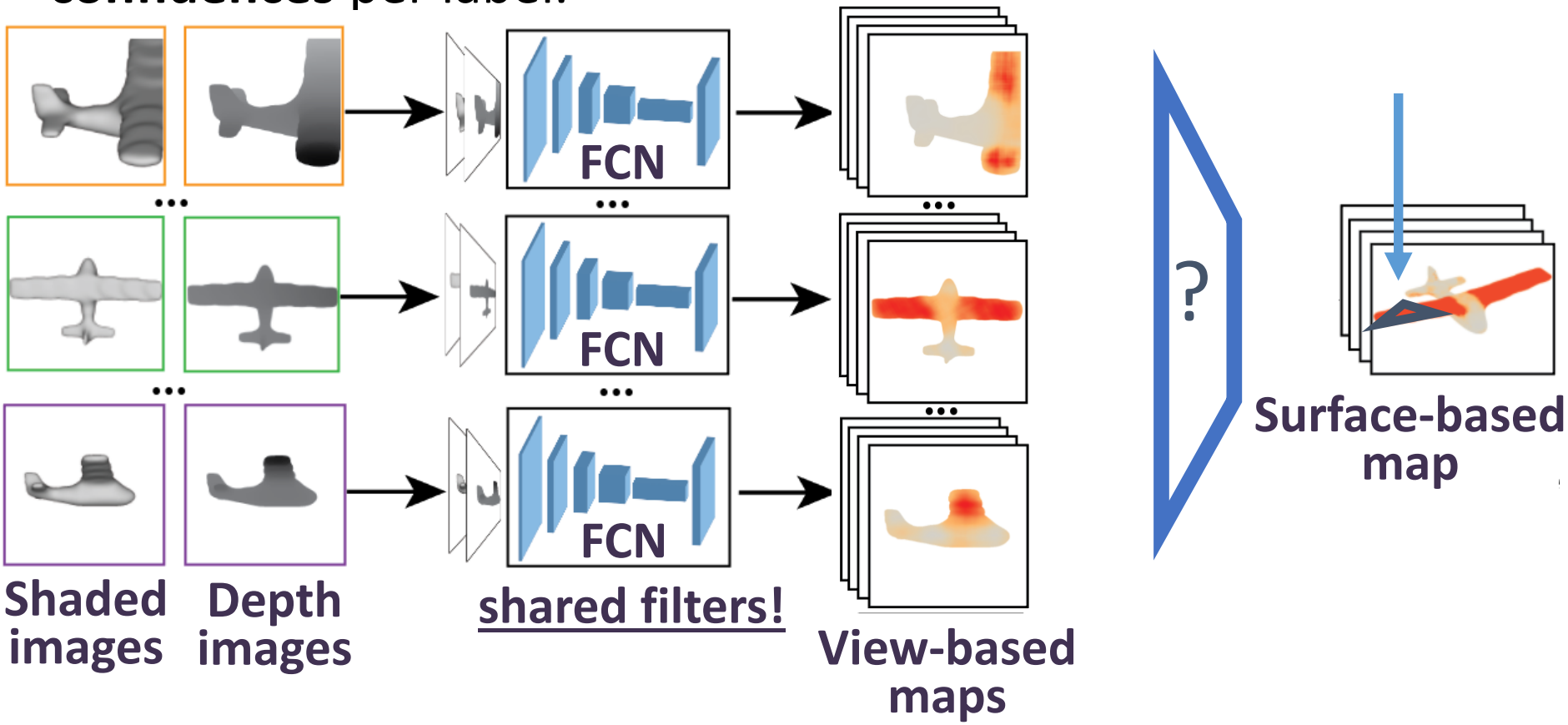
Projective convnet architecture

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



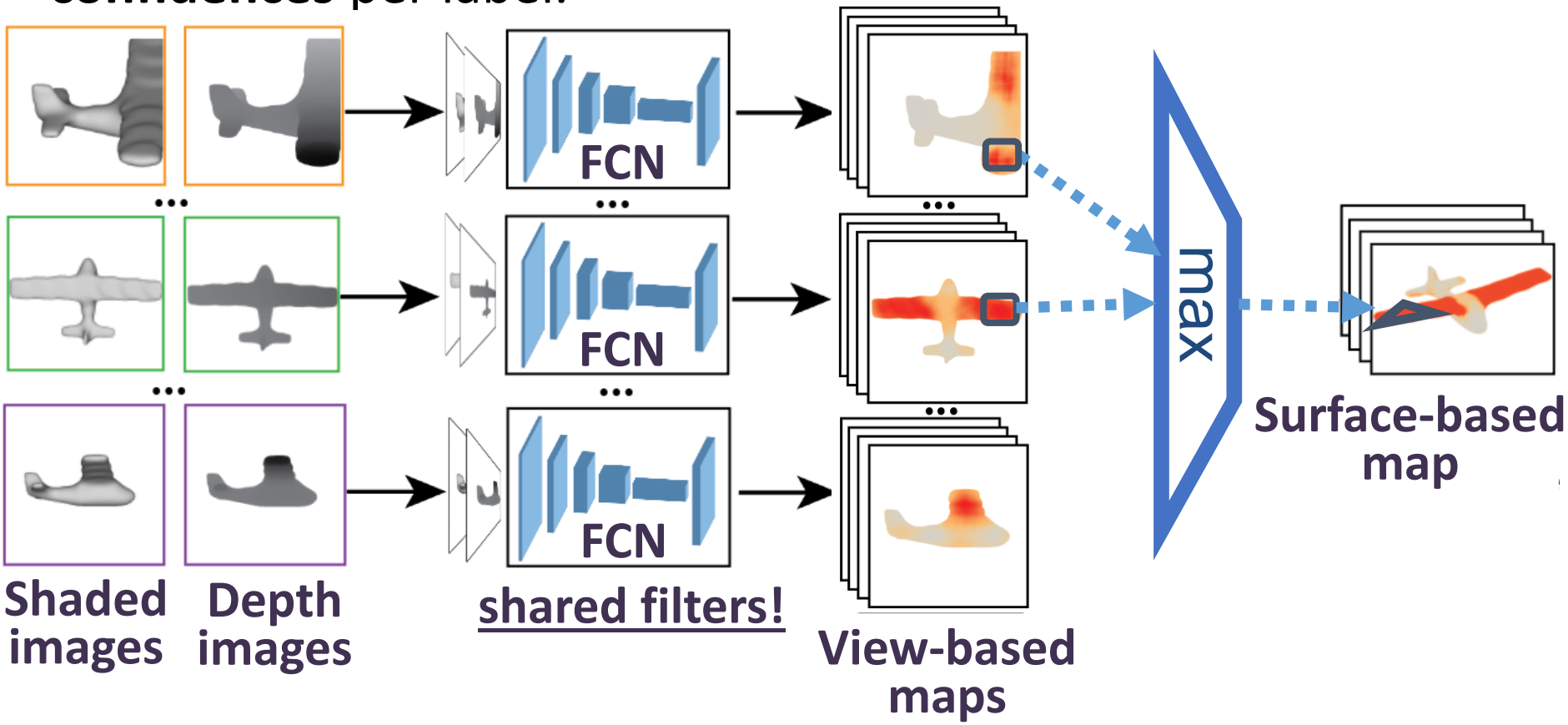
Projective convnet architecture

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



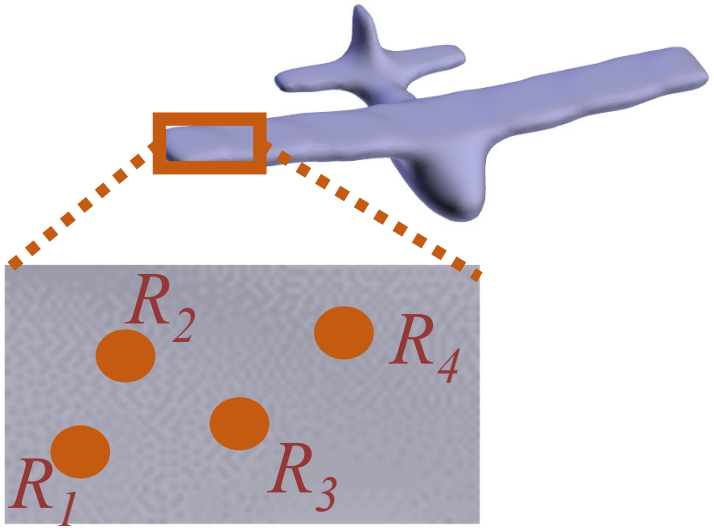
Projective convnet architecture

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.**

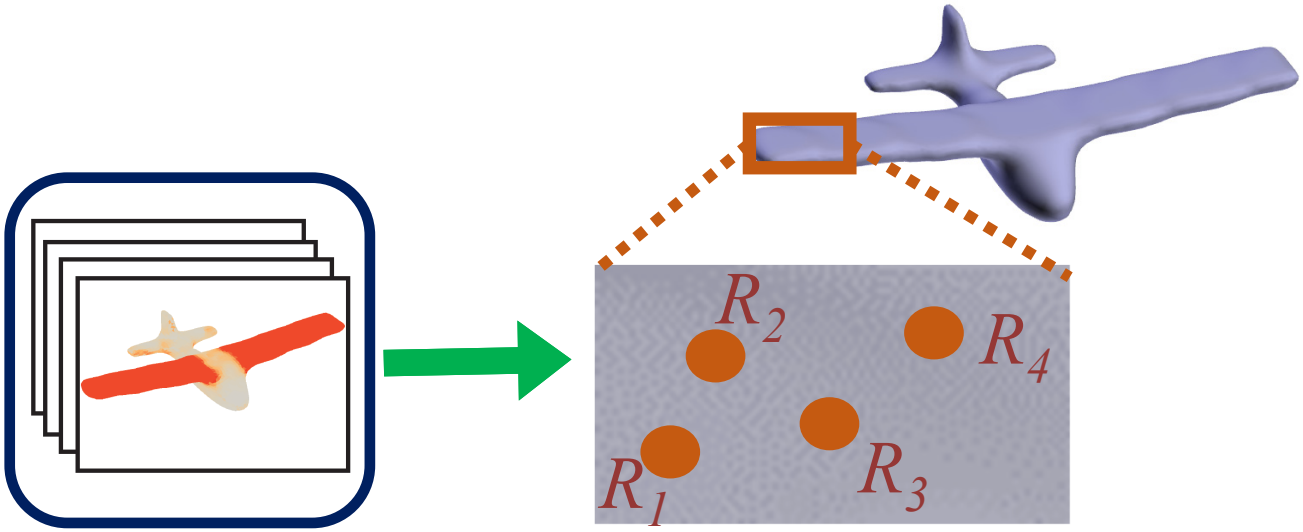


$R_1, R_2, R_3, R_4 \ldots$ random variables taking values:

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

Surface model for spatially coherent labeling

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



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

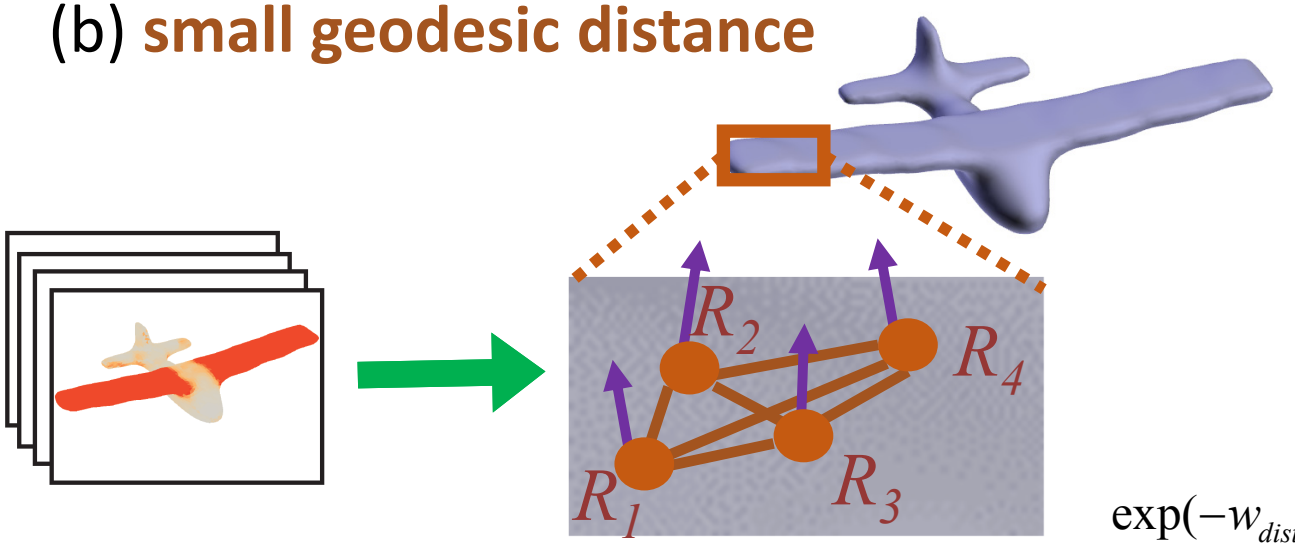
Unary factors (FCN confidences)

Surface model for spatially coherent labeling

Pairwise terms **favor same label** for triangles with:

(a) **similar surface normals**

(b) **small geodesic distance**



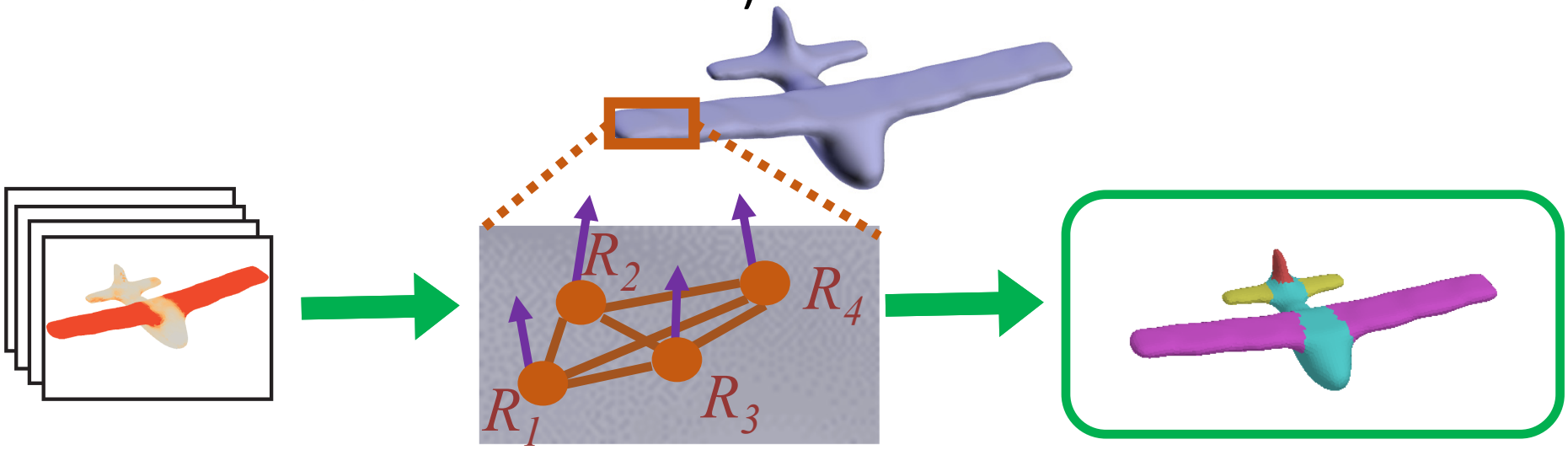
$$\exp(-w_{dist} \cdot \mathbf{d_{f,f'}}^2) \qquad \exp(-w_{angle} \cdot \boldsymbol{\omega}_{\mathbf{f,f'}}^2)$$

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

Pairwise factors (geodesic+normal distance)

Inference

Infer **most likely joint assignment** to all surface random variables of the probabilistic model (**Conditional Random Field – we will discuss later in class**)

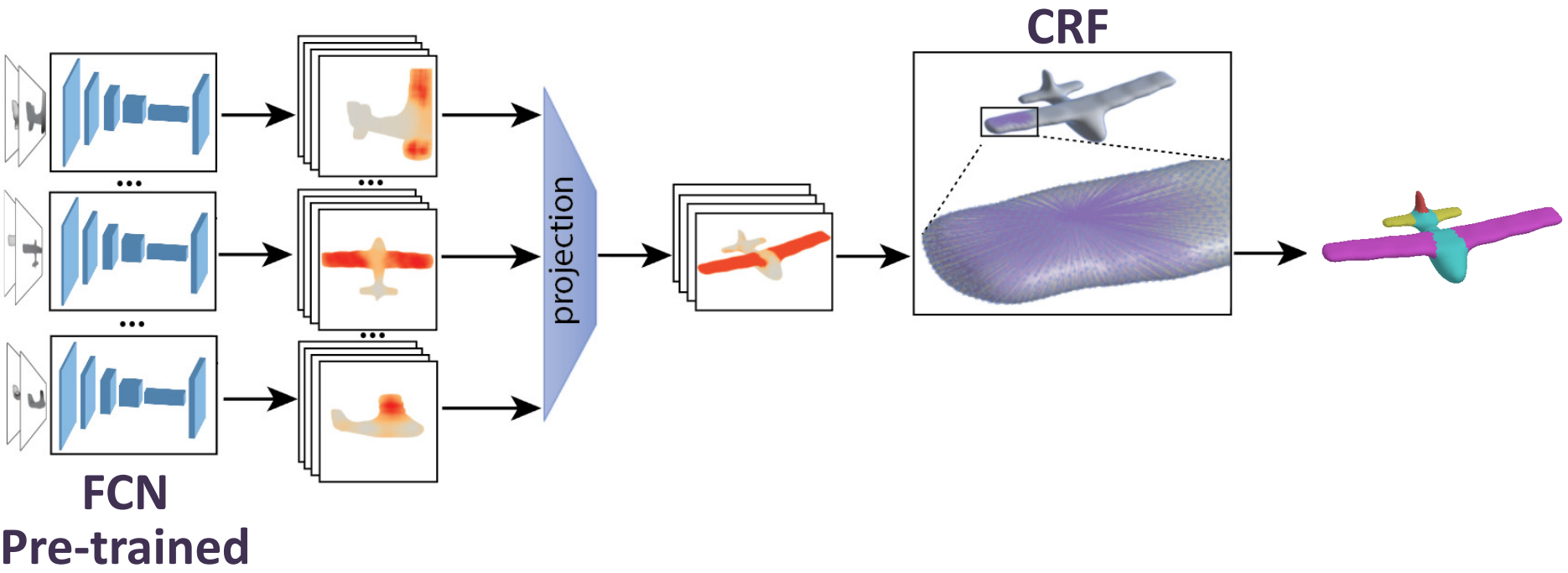


$$\max P(R_1, R_2, R_3, R_4 \ldots \mid \textbf{shape}) = \frac{1}{Z} \prod_{f=1..n} P(R_f \mid \textbf{views}) \prod_{f,f'} P(R_f, R_{f'} \mid \textbf{surface})$$

MAP assignment (mean–field inference… we will discuss later in the course)

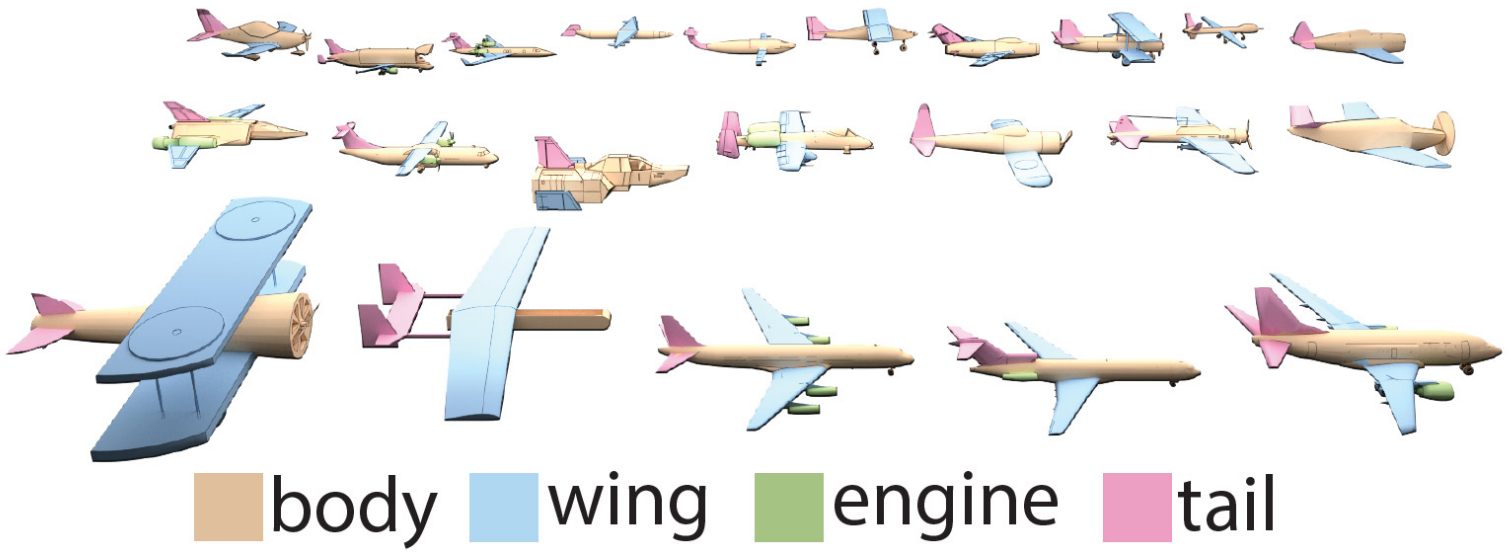
Training

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



Training

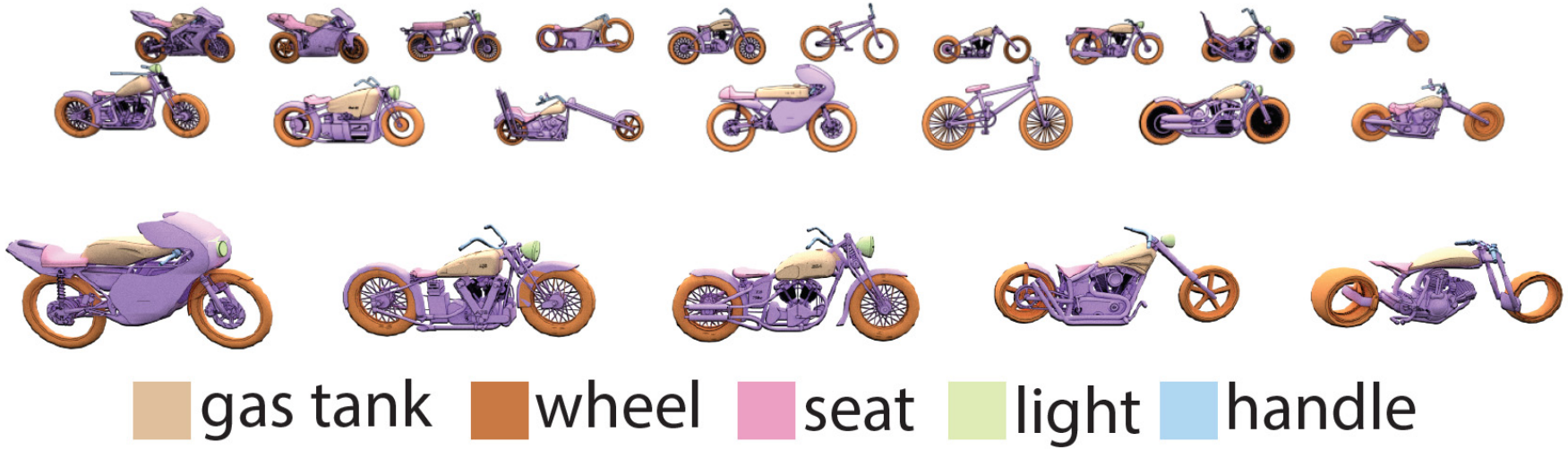
The architecture is trained **end-to-end** with analytic gradients. Training starts from a **pretrained image-based net** (VGG16), then **fine-tune on segmented shape datasets.**



[Yi et al. 2016]

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)

	ShapeBoost	Guo et al.	ShapePFCN
	81.2	80.6	**87.5**
Ignore easy classes (2 or 3 part labels) →	76.8	76.8	**84.7**

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

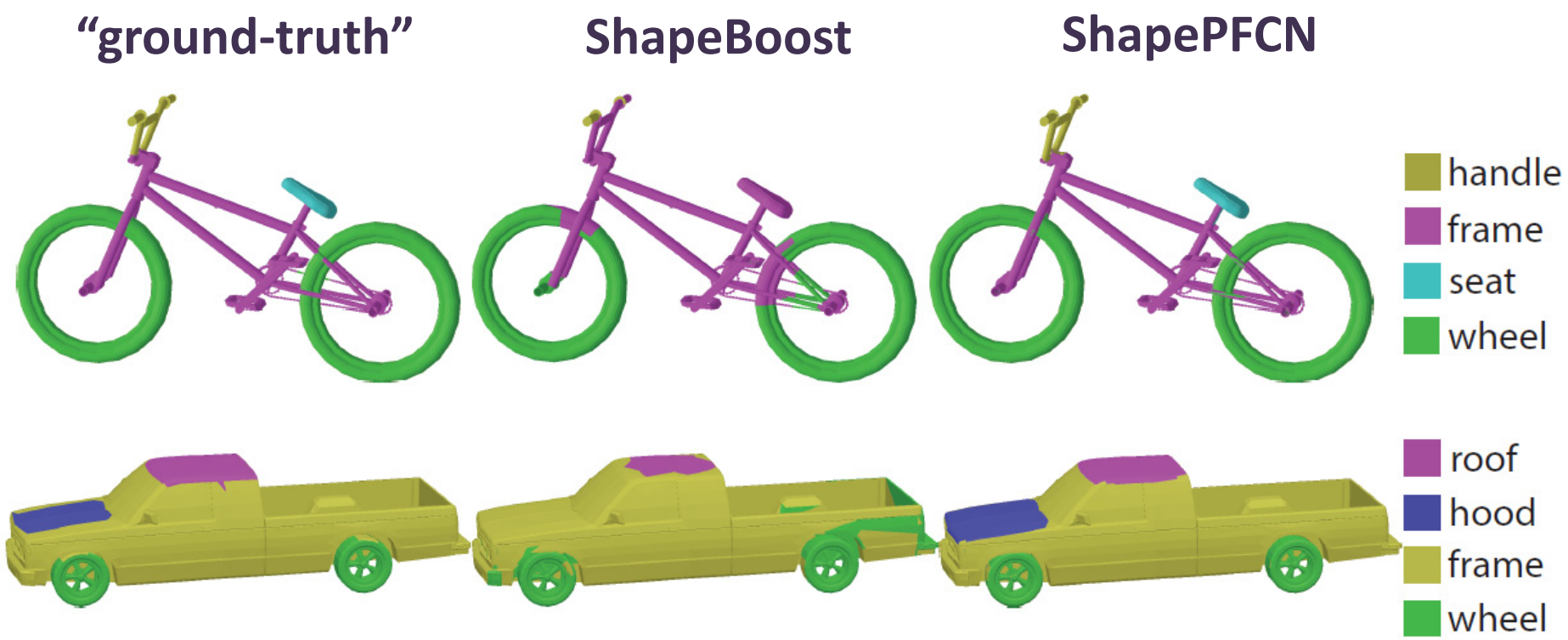
Results

Labeling accuracy on **ShapeNet** test dataset:

(assume consistent upright orientation + render y-coords)

	ShapeBoost	Guo et al.	ShapePFCN
	81.2	80.6	**89.4**
Ignore easy classes (2 or 3 part labels) →	76.8	76.8	**86.6**

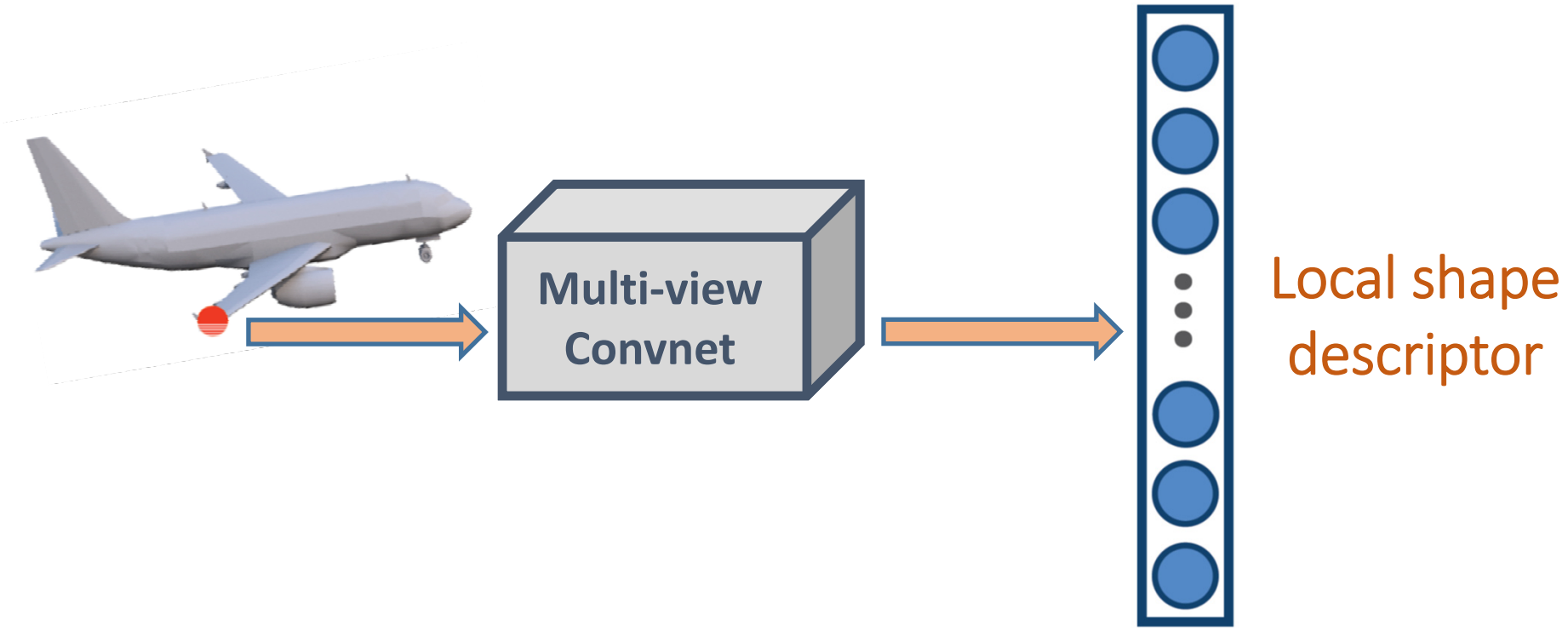
~10% improvement in labeling accuracy for complex categories (vehicles, furniture)


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

3D Deep Learning approaches

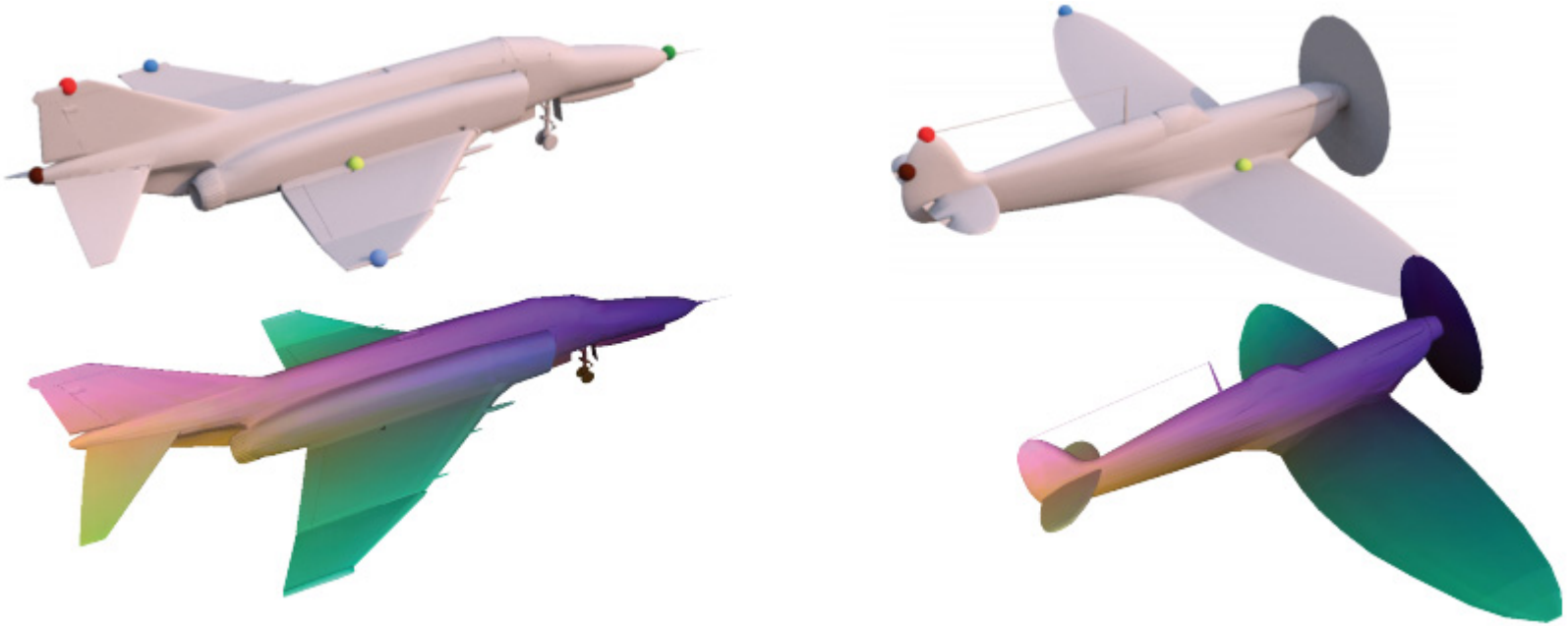
- **The Multi-View approach**
  - Recognition
  - Segmentation
  - **Correspondences**
- The Voxel approach
- The Point approach
- The Graph approach

Goal



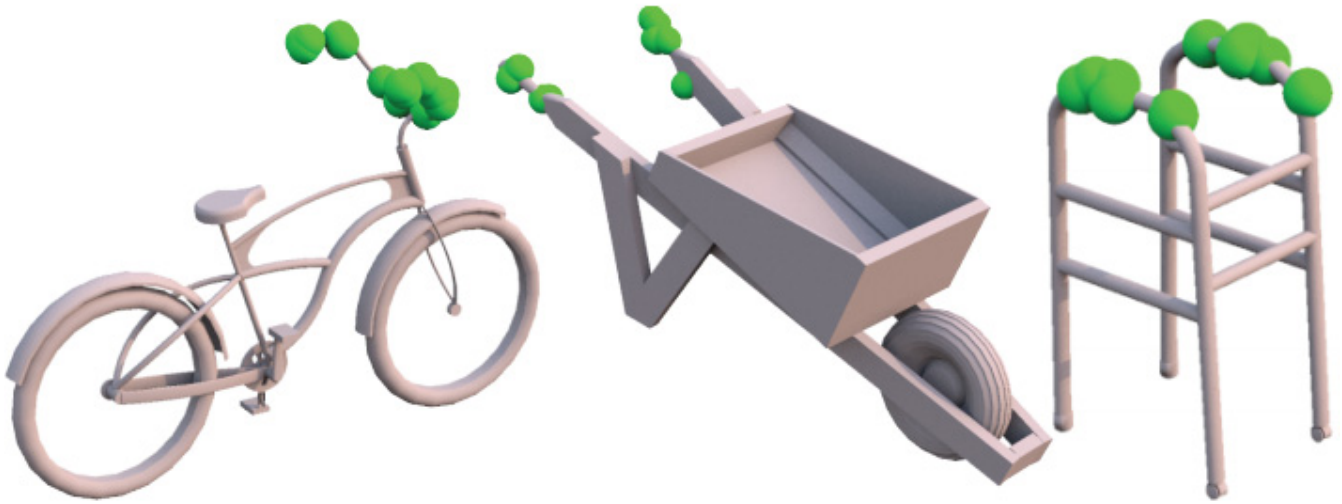
Huang, Kalogerakis, Chaudhuri, Ceylan, Kim, Yumer (TOG 2018)

Applications of local descriptors: keypoint prediction & correspondences



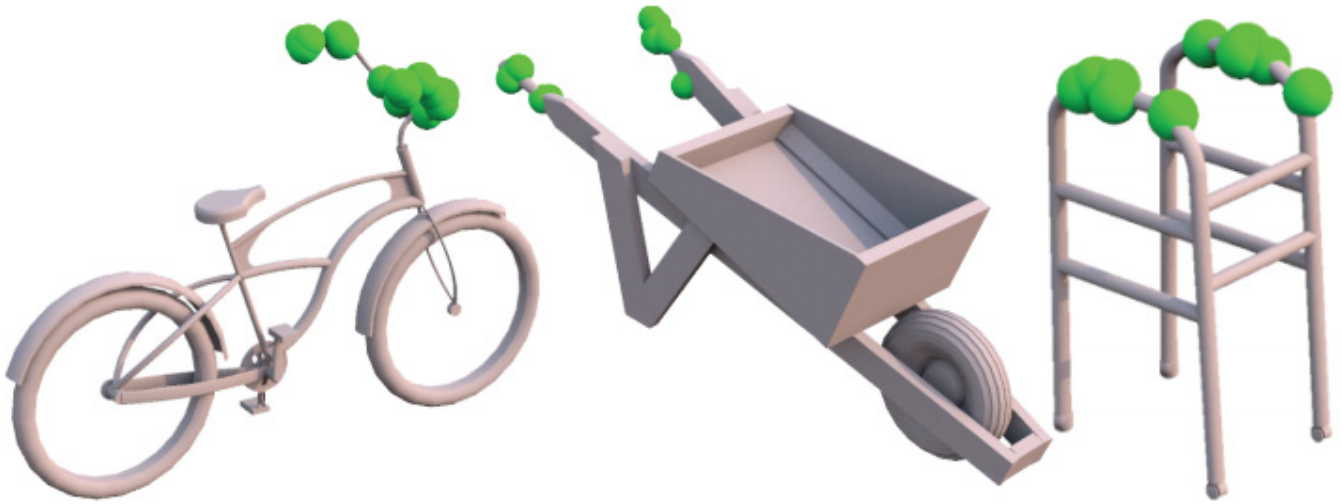
(similar colors correspond to points with similar descriptors)

Applications of local descriptors: affordance prediction

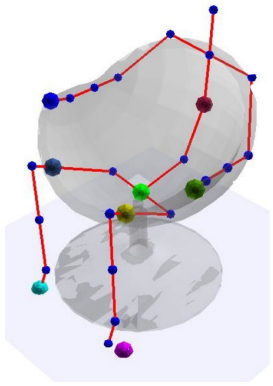


Where humans tend to place their palms when they interact with these objects?

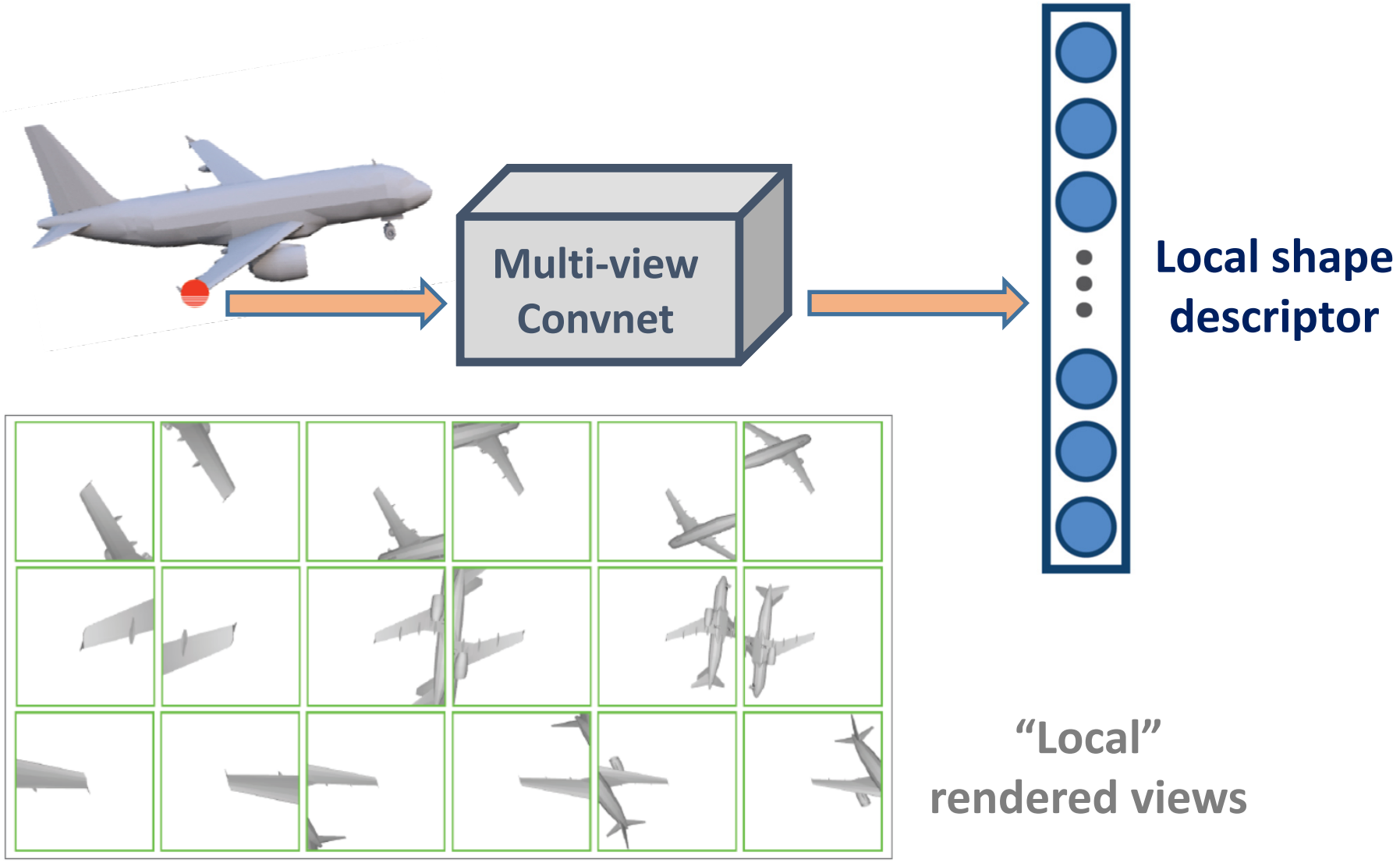
Applications of local descriptors: affordance prediction



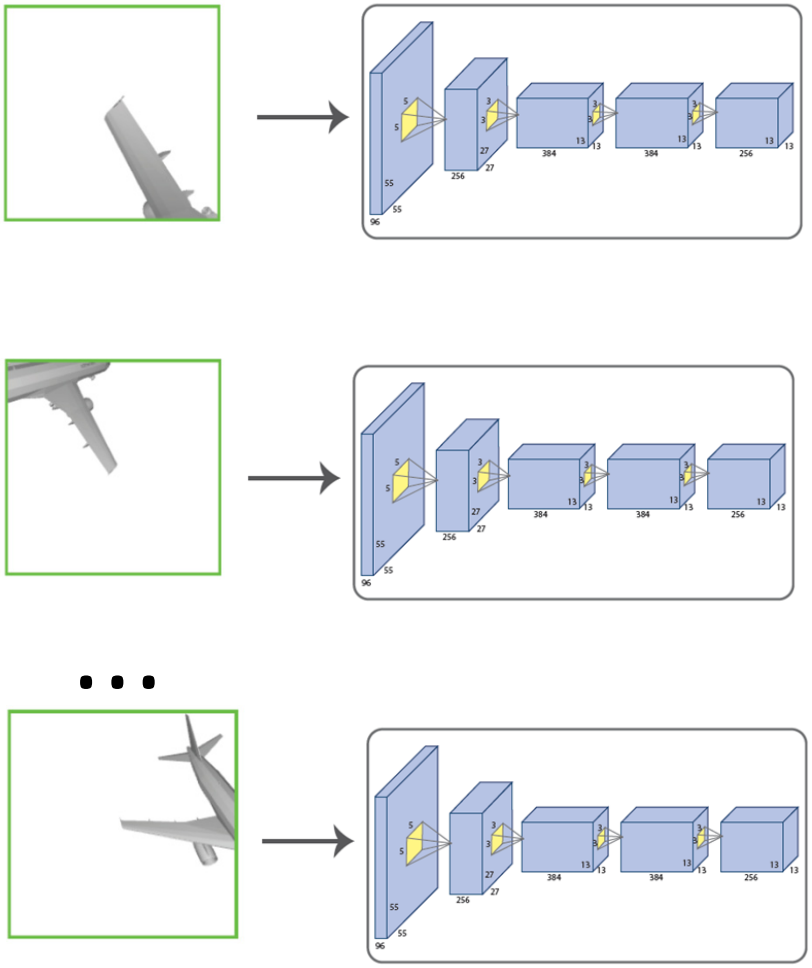
How would you place a human body relative to this object?



Goal

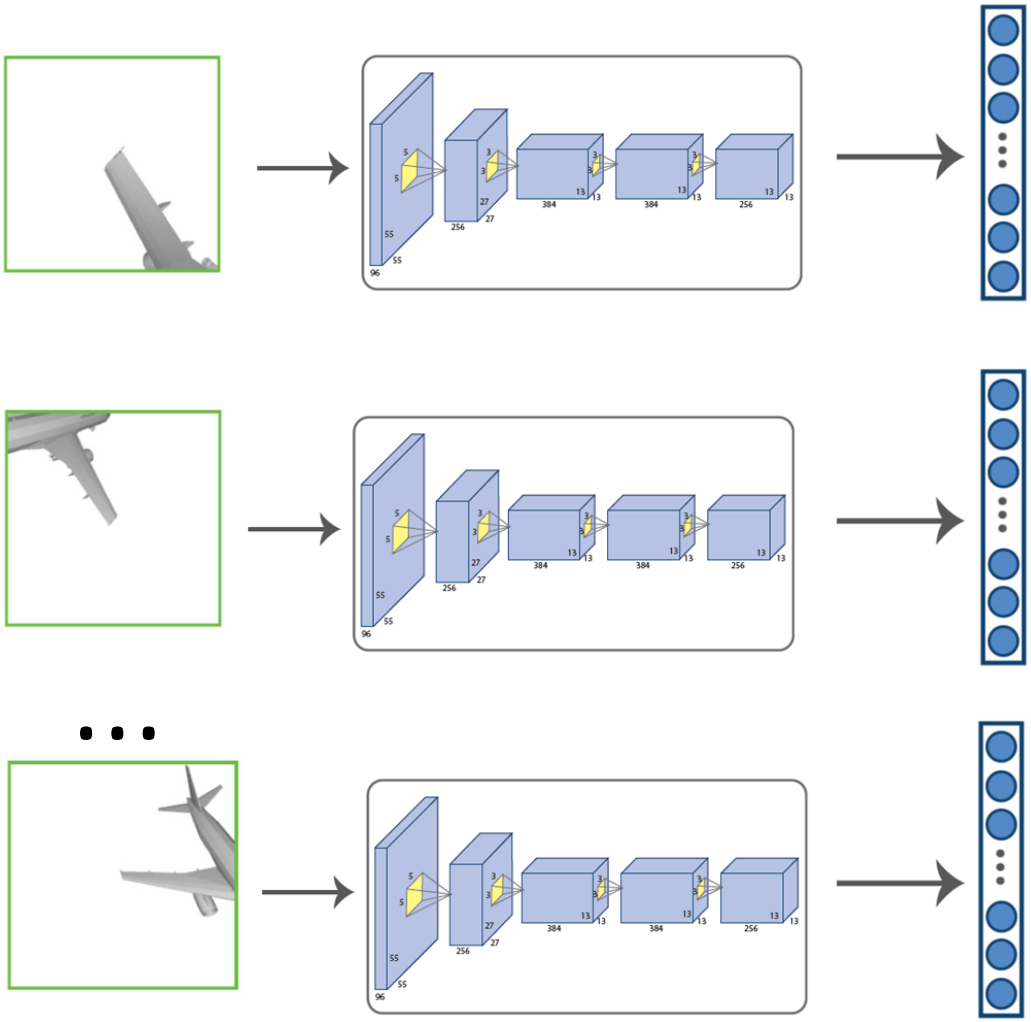

Multi-view
Convnet
Local shape
descriptor
"Local"
rendered views

Local MVCNN



CNN branches
(shared parameters)

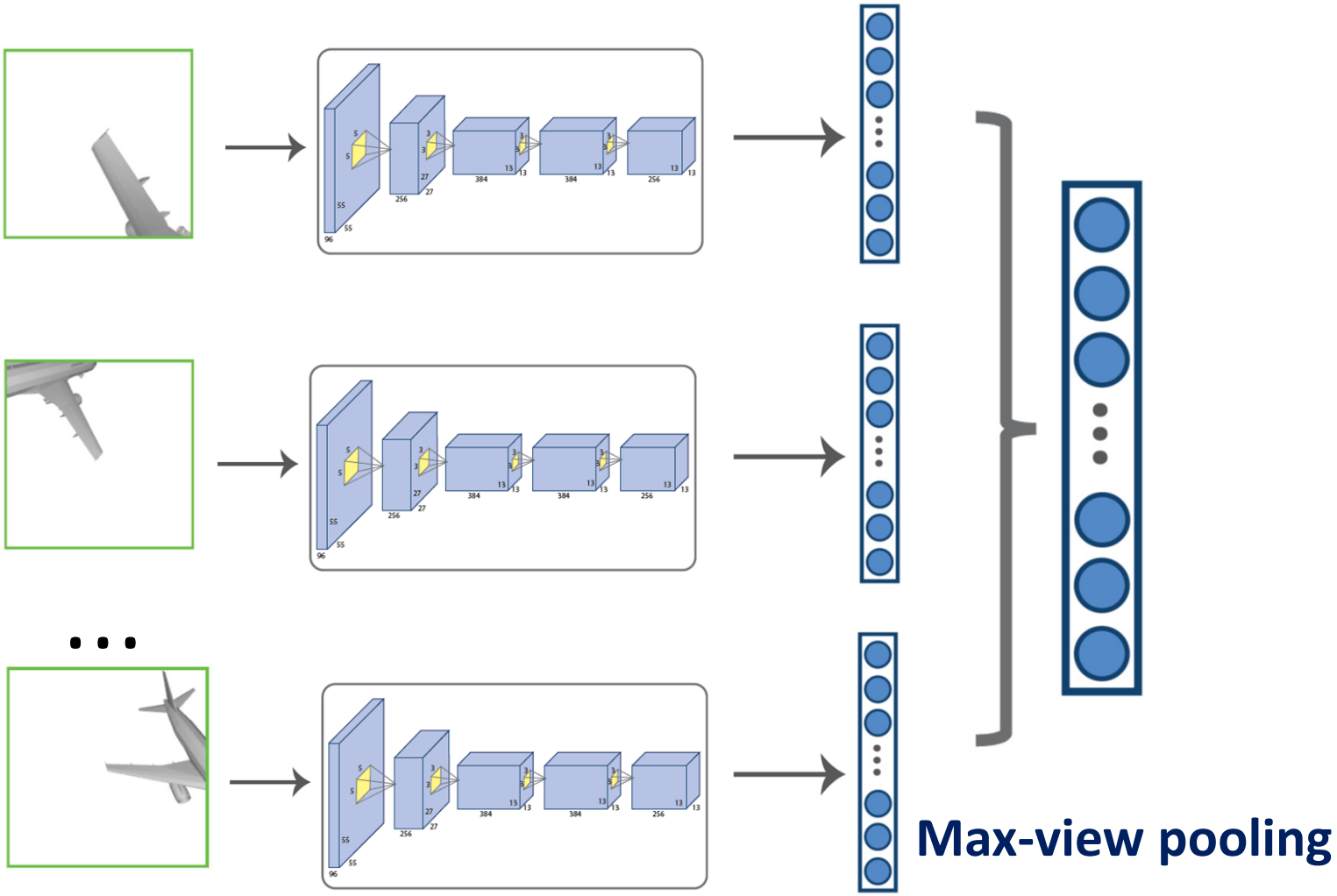
Local MVCNN



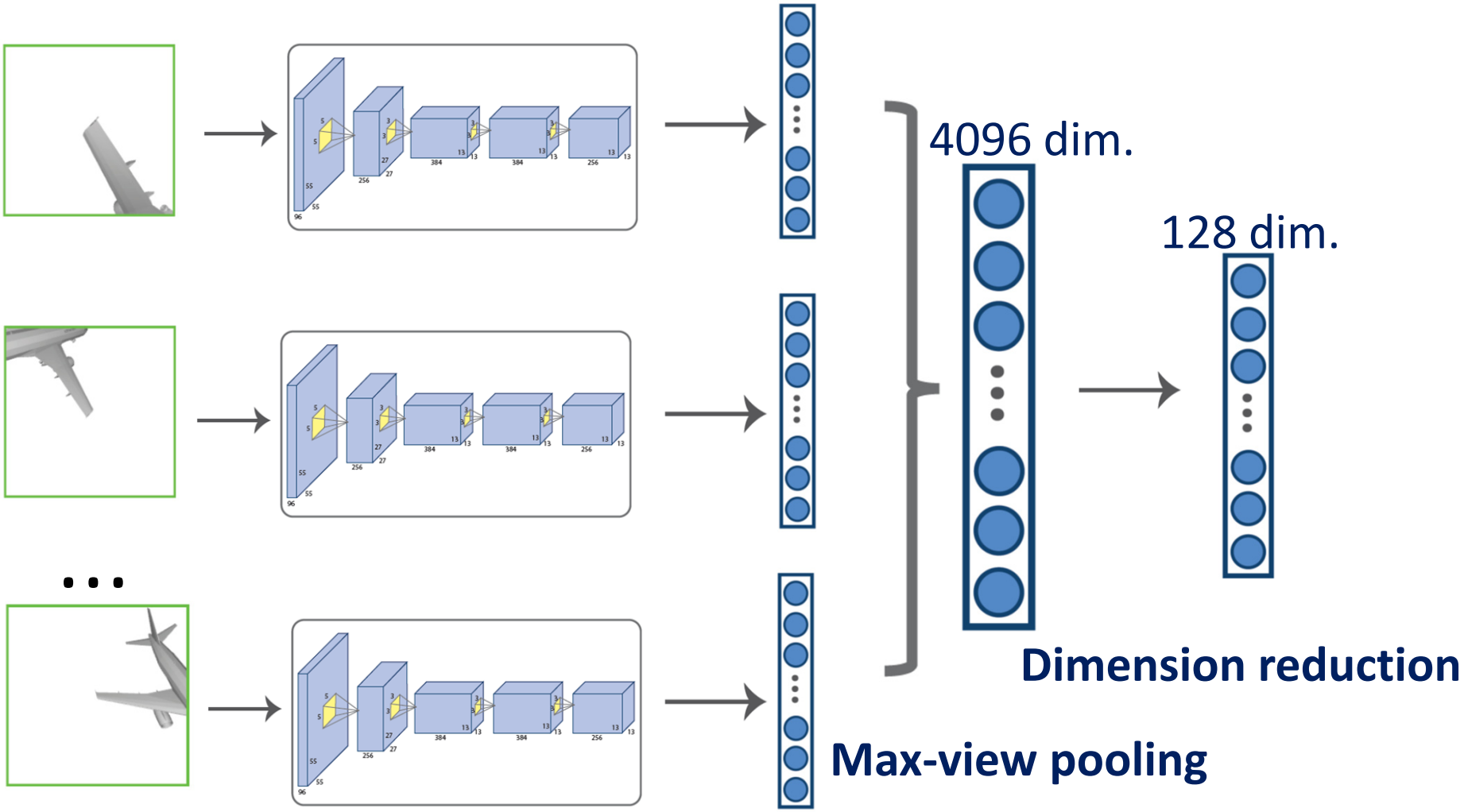
CNN branches (shared parameters)

Local view-based descriptors

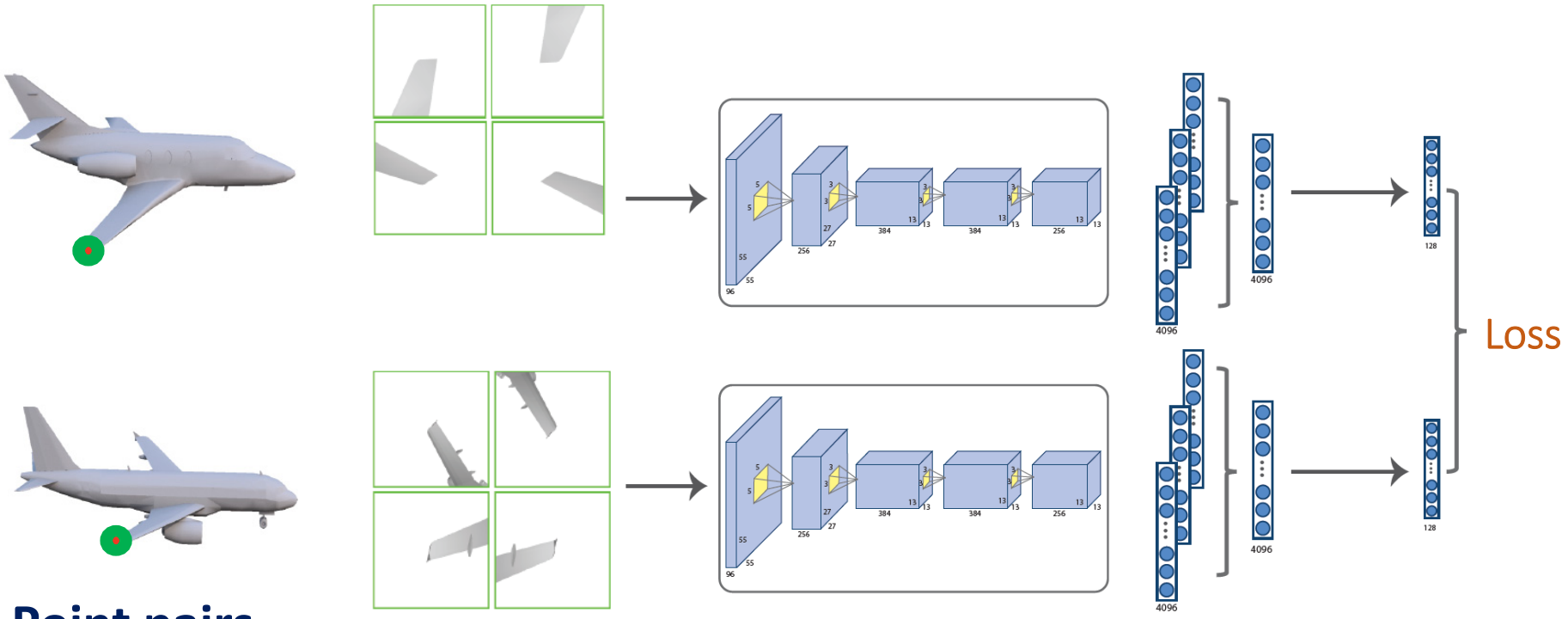
Local MVCNN


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



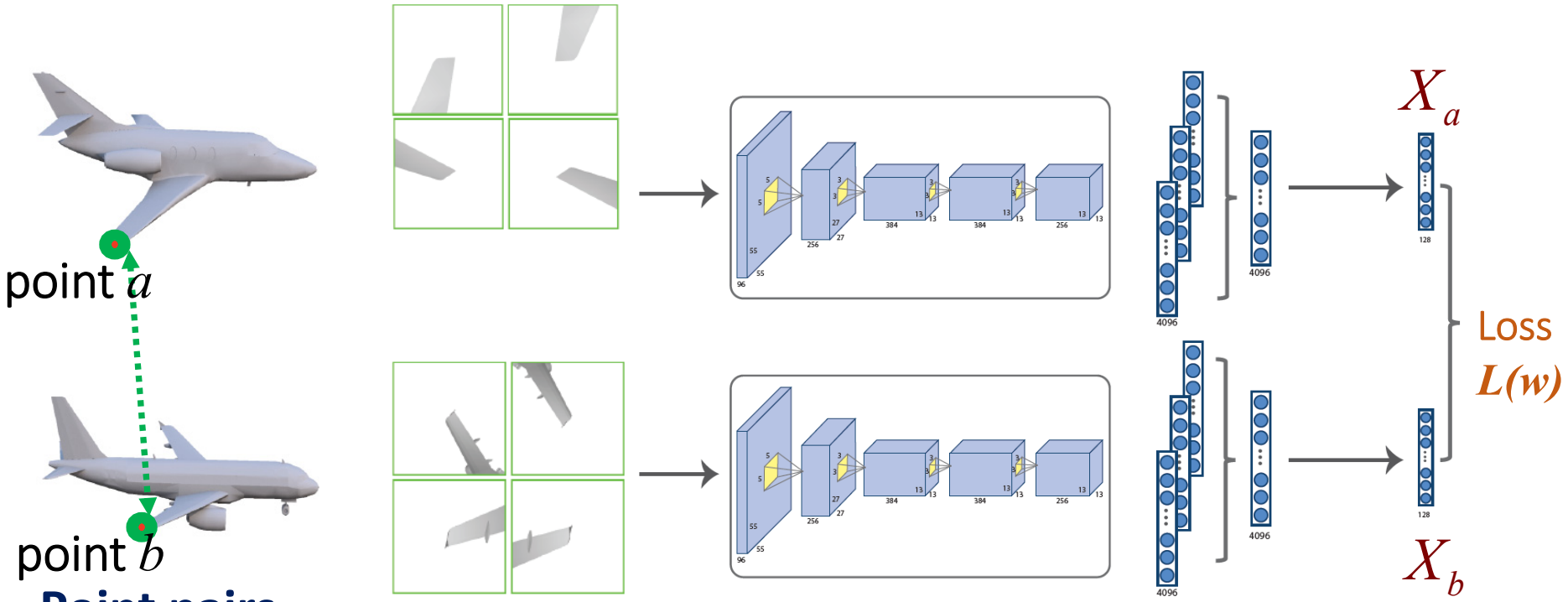
Point pairs from two shapes

Local Rendered views

CNN branches

View Pooling & dim. reduction

Training



Point pairs from two shapes

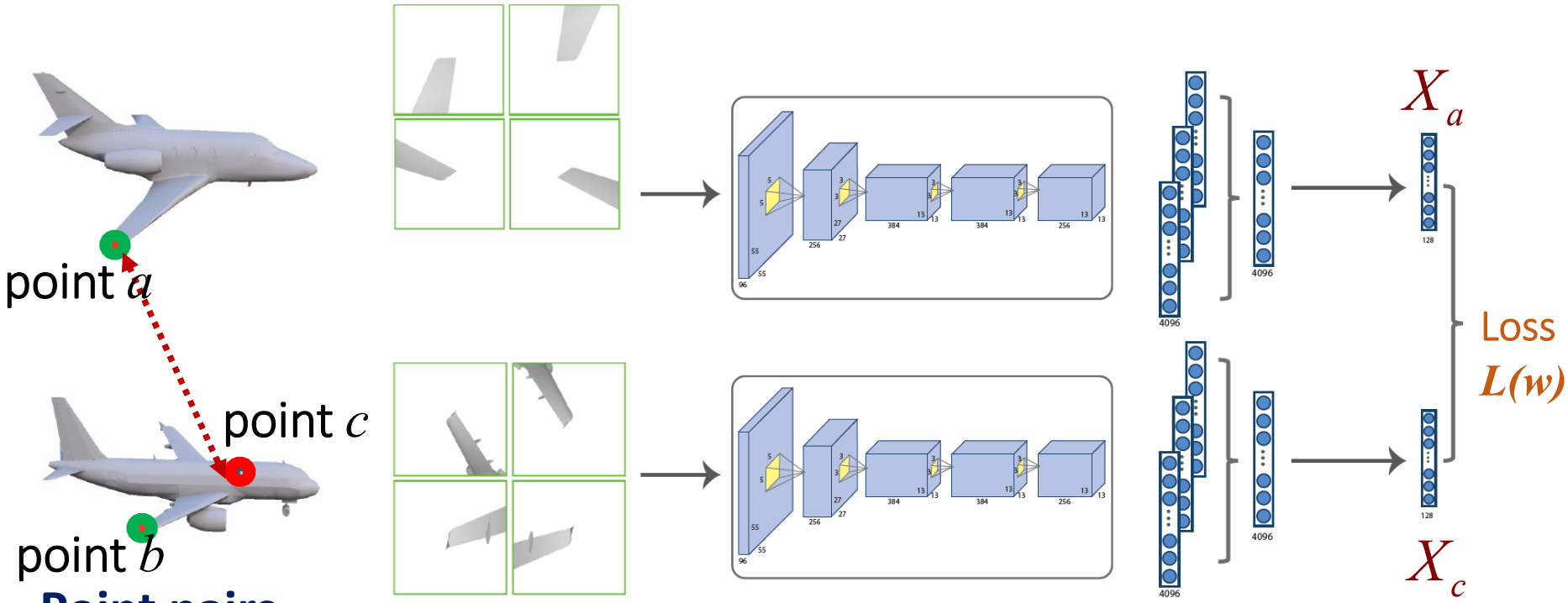
Local Rendered views

CNN branches

View Pooling & dim. reduction

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

Training: Siamese Architecture



Point pairs from two shapes

Local Rendered views

CNN branches

View Pooling & dim. reduction

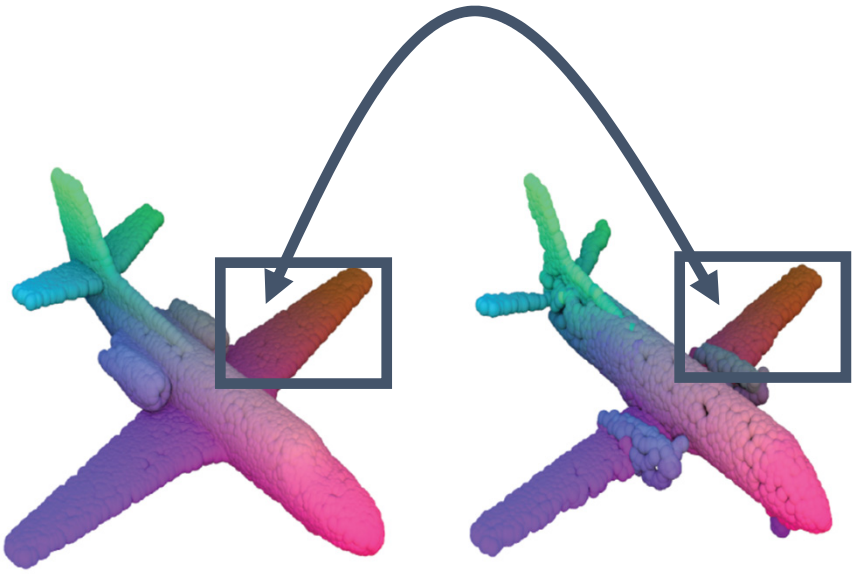
Known as Contrastive loss:

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

See also original "siamese Architecture" paper Hadsell et al. "Dimensionality Reduction by Learning an Invariant Mapping" + triplet loss Scroff et al. "FaceNet: A Unified Embedding for Face Recognition and Clustering"

Training dataset generation

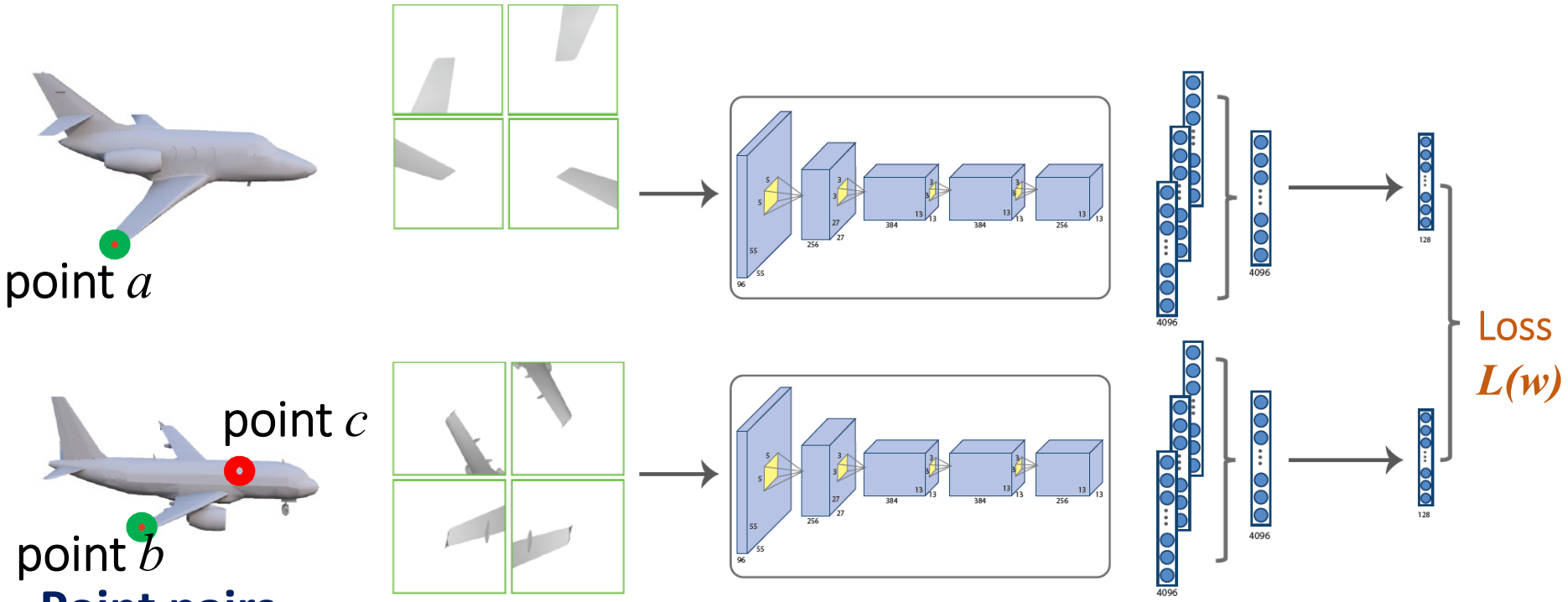
Non-rigid alignment **per part**
from segmented ShapeNetCore



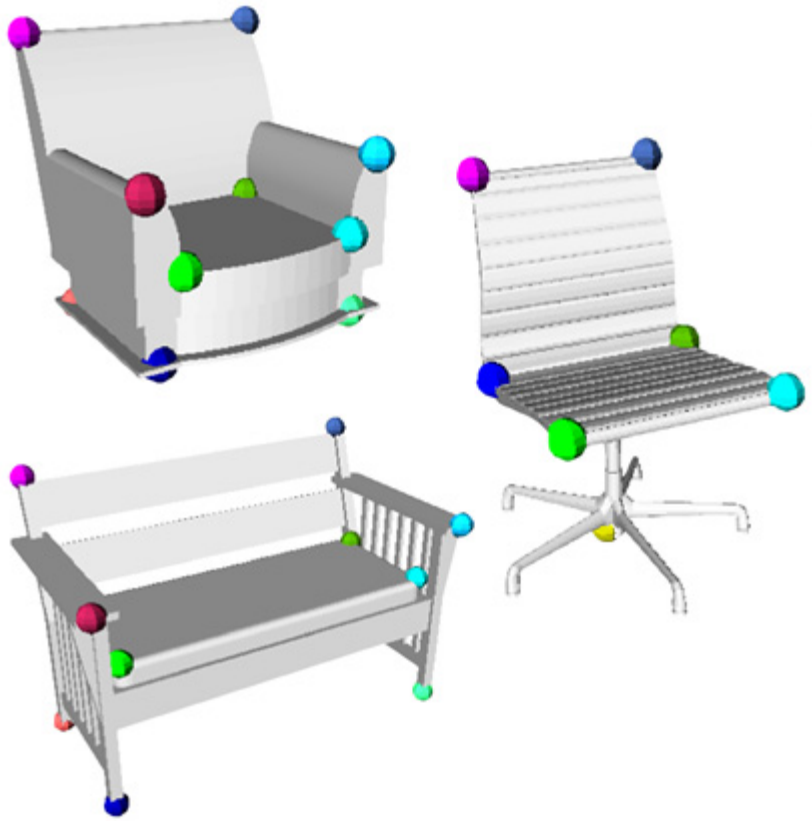
(corresponding points have same color)

Note: we will discuss non-rigid alignment methods later in the course

Training

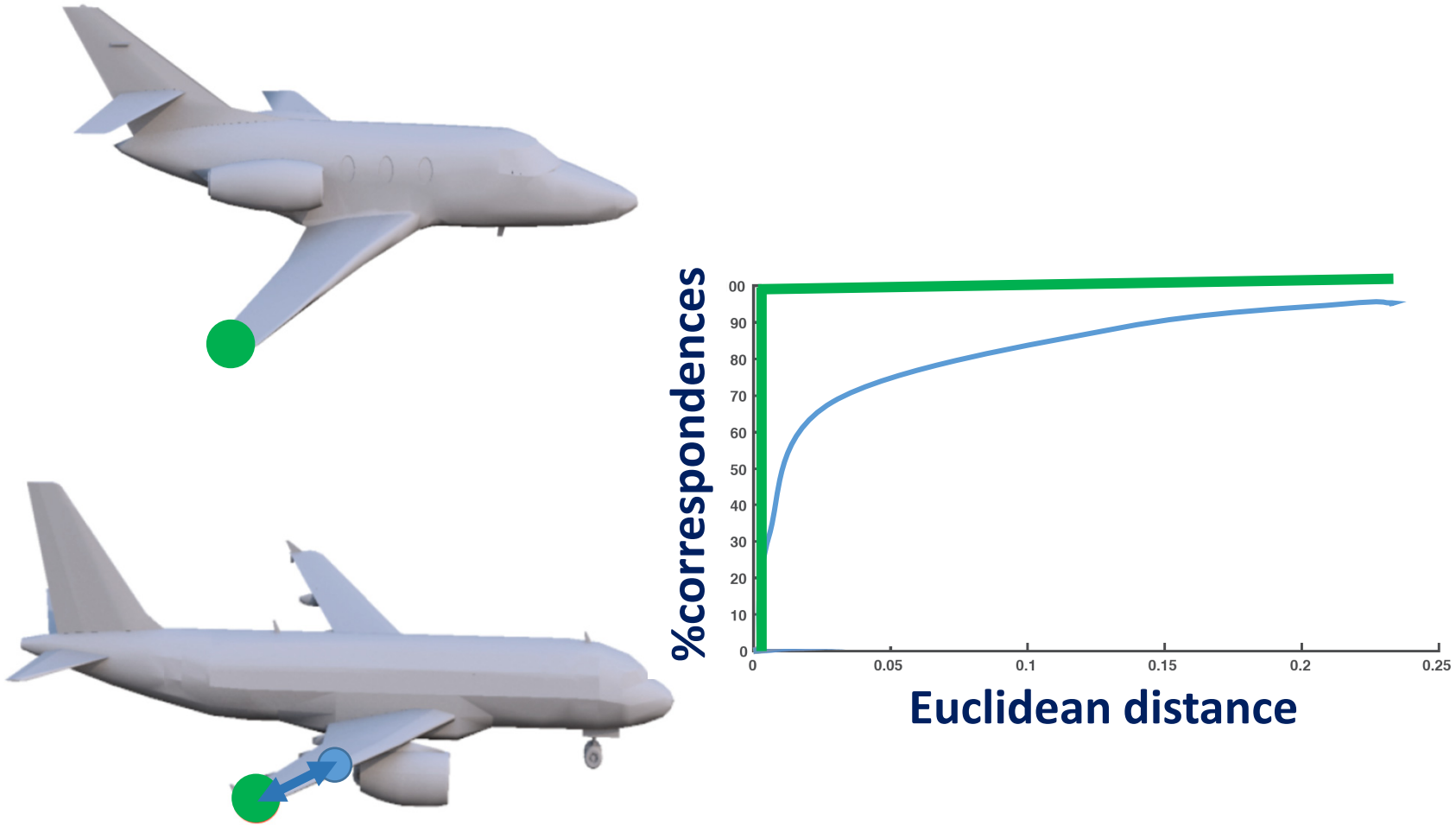

point a
point c
point b
Loss
$L(w)$
Point pairs from two shapes
Local Rendered views
CNN branches (initialized from AlexNet)
View Pooling & dim. reduction
Backpropagation

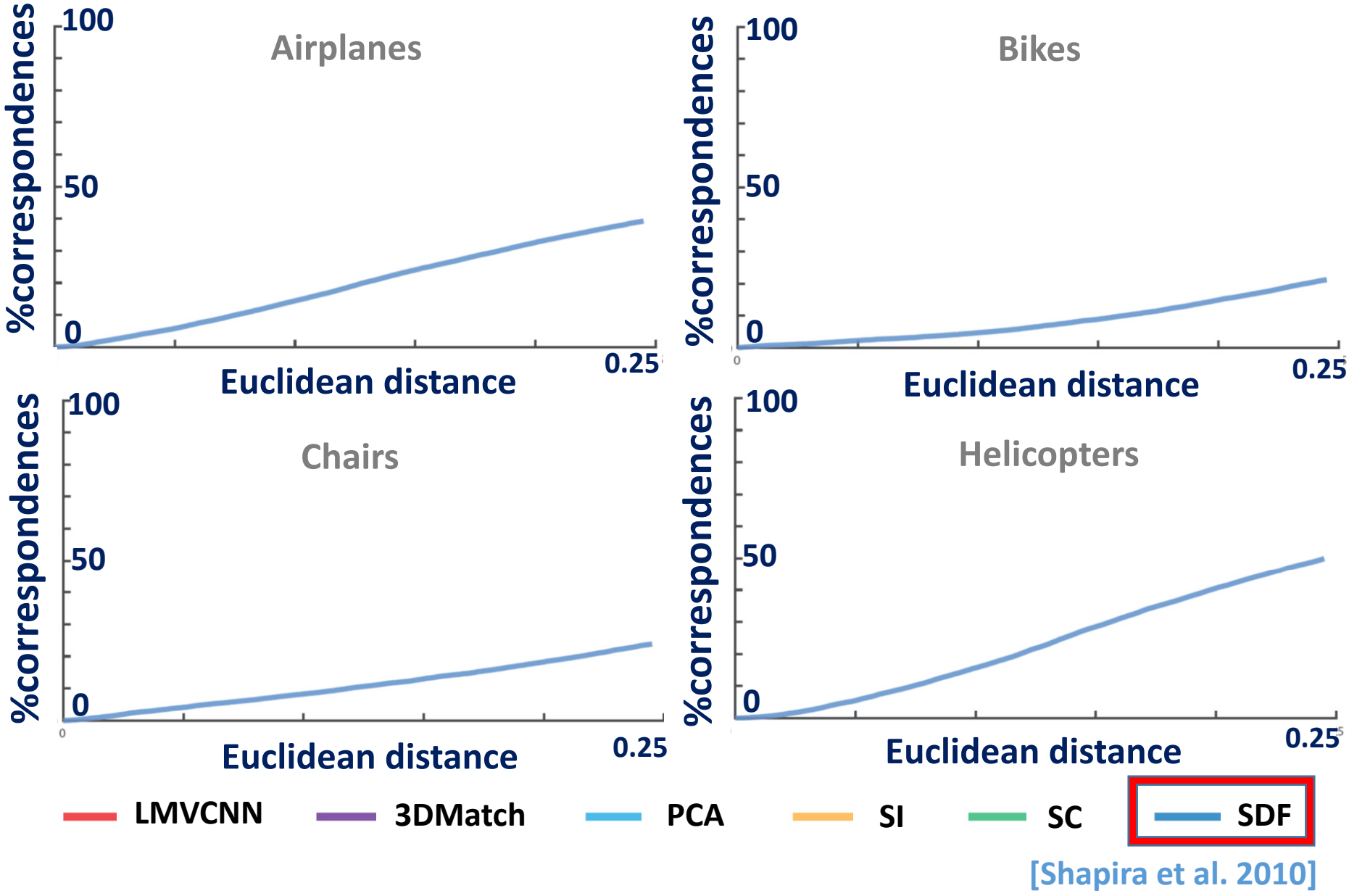
Evaluation: correspondence accuracy



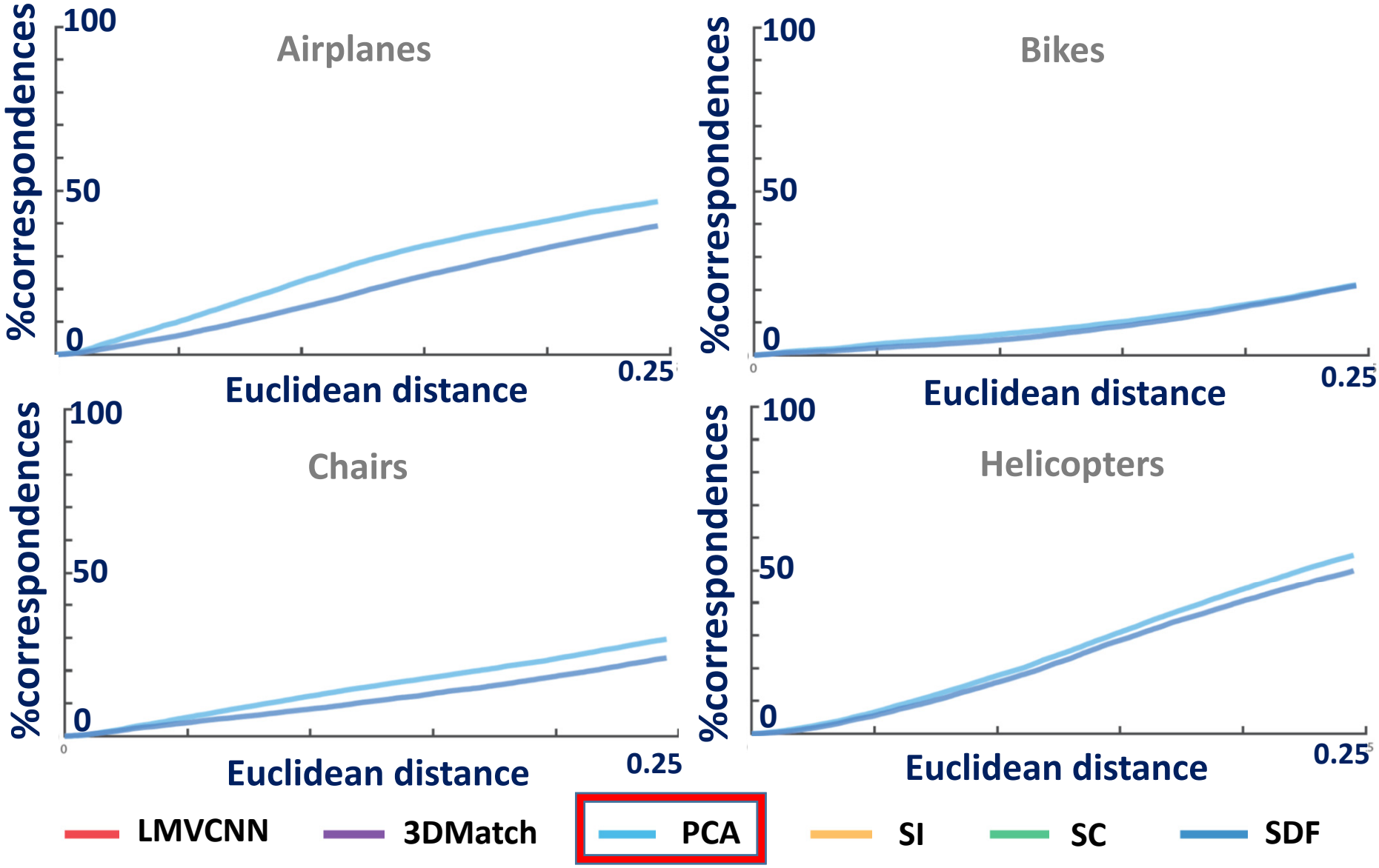
[Kim et al. 2013]

Evaluation: correspondence accuracy

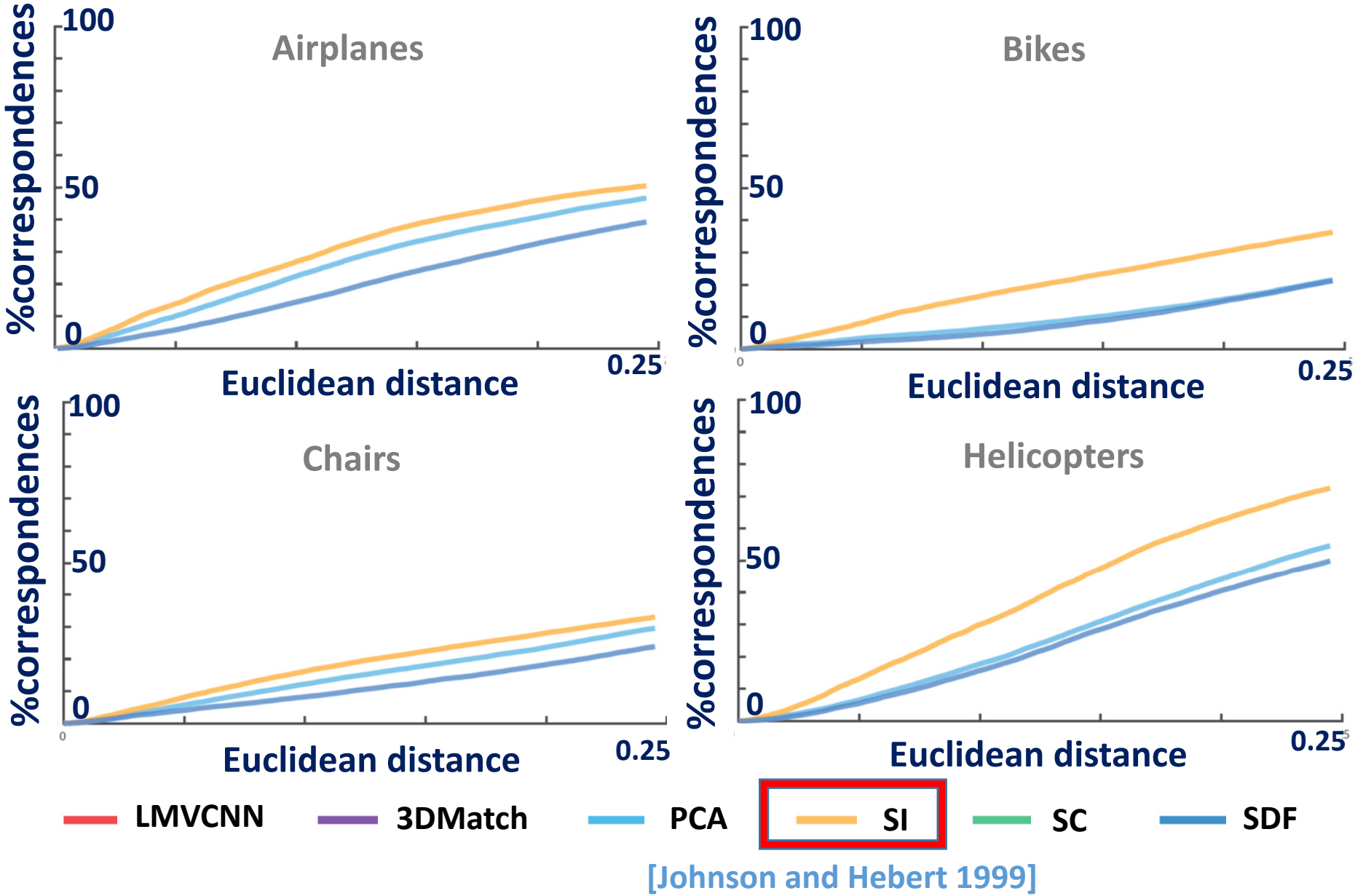




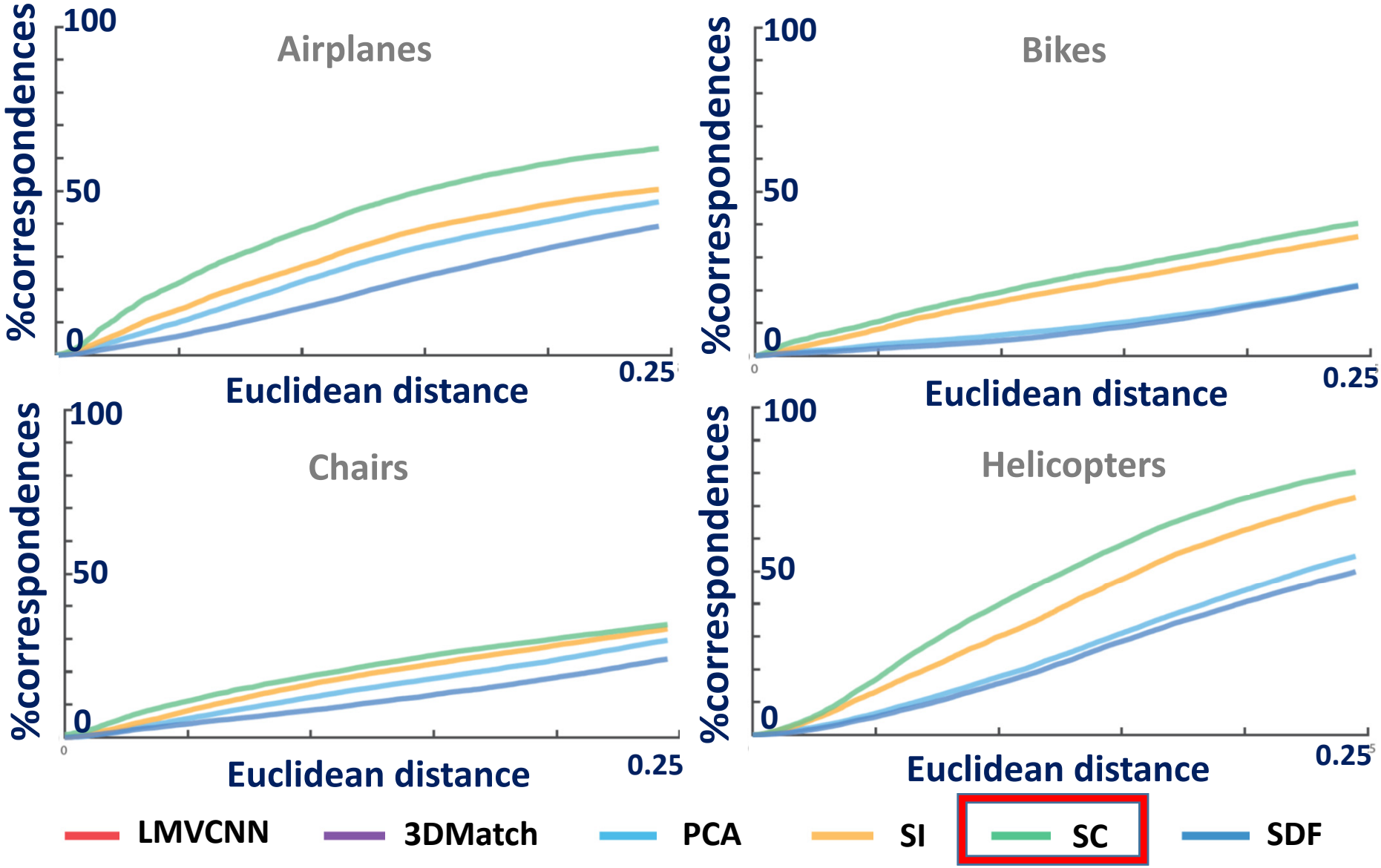
[Shapira et al. 2010]



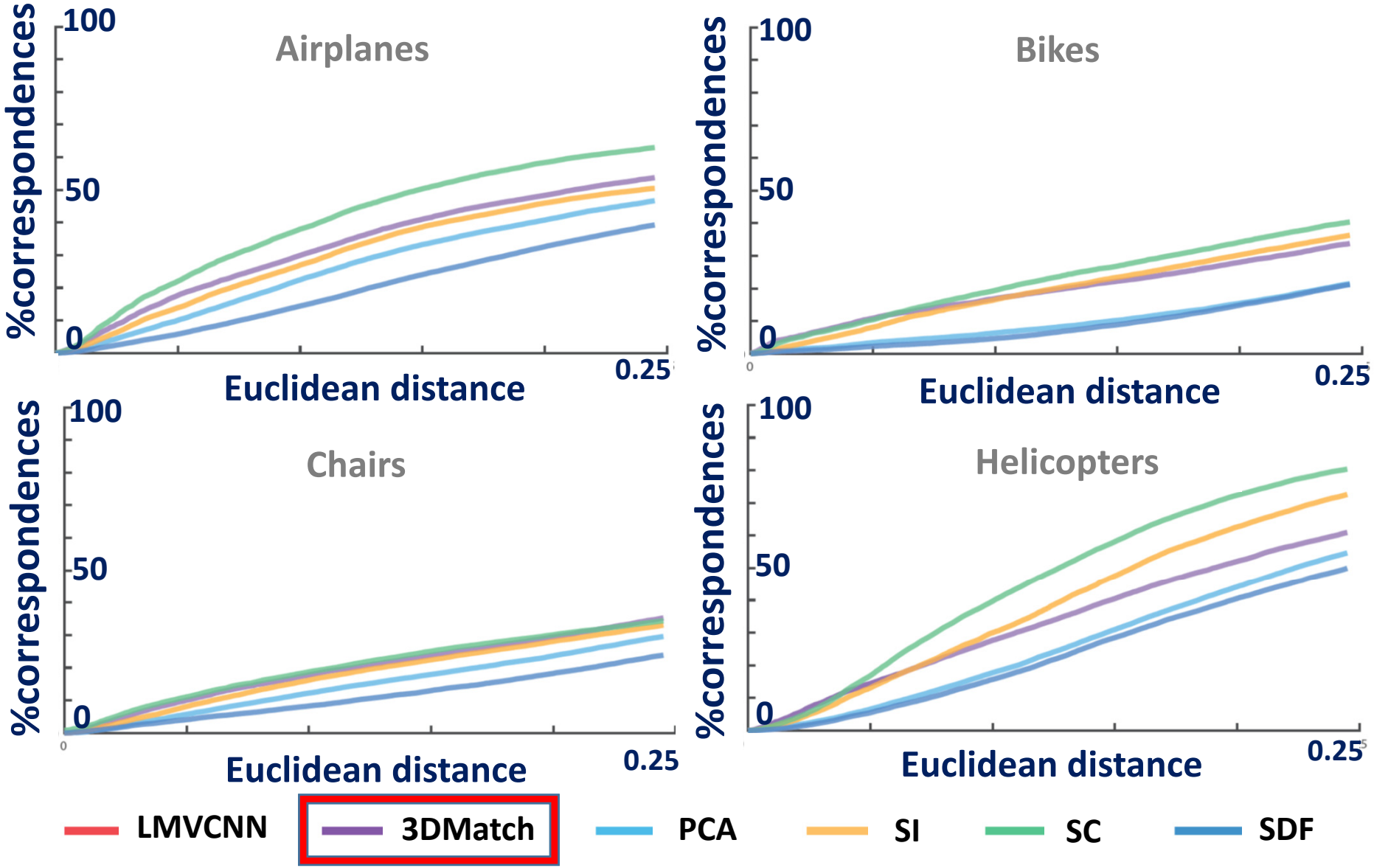
[Kalogerakis et al. 2010]



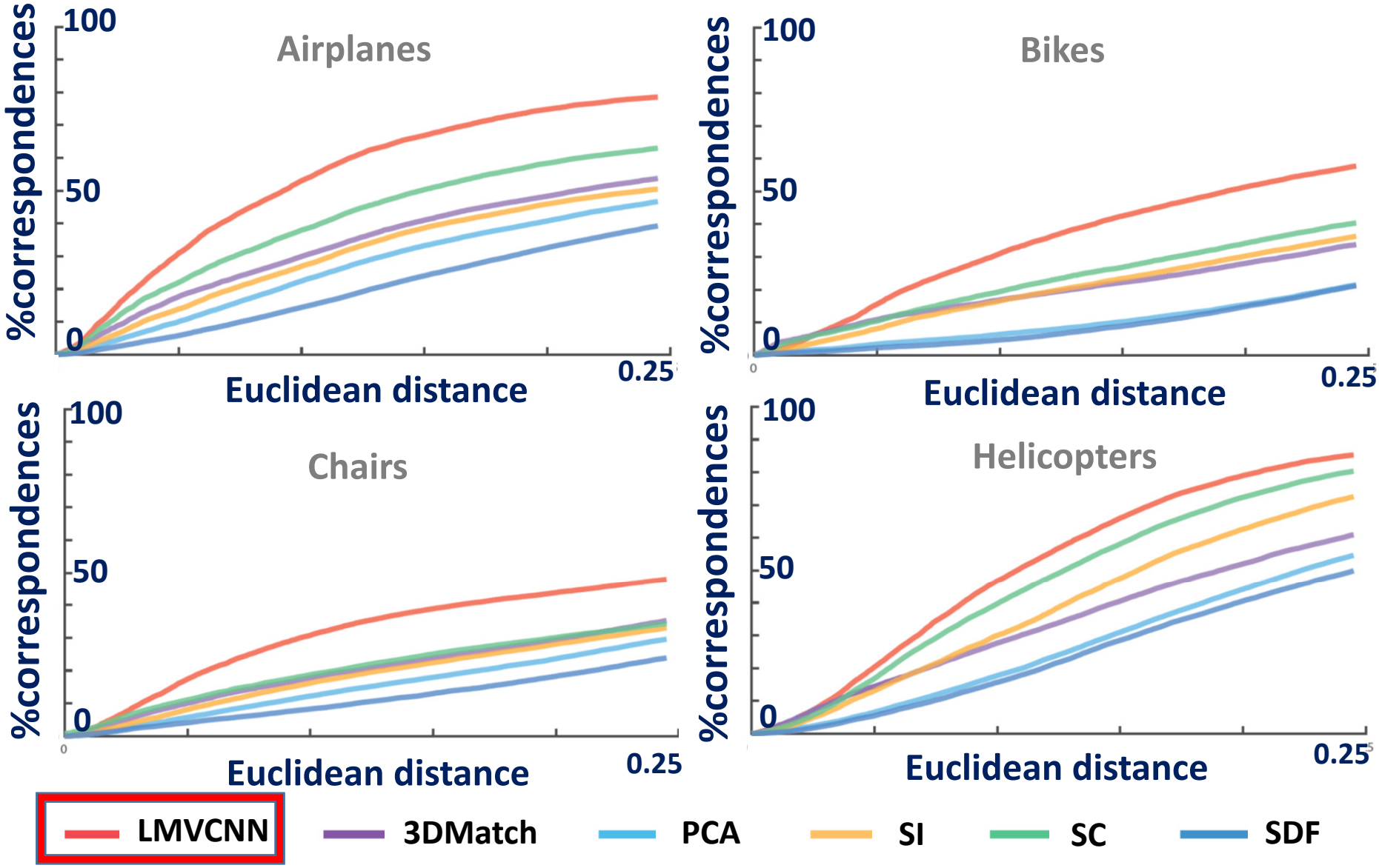
[Johnson and Hebert 1999]



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



[Zeng et al. 2017 - volumetric net]



Our method – Local Multi-view CNN (LMVCNN)

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