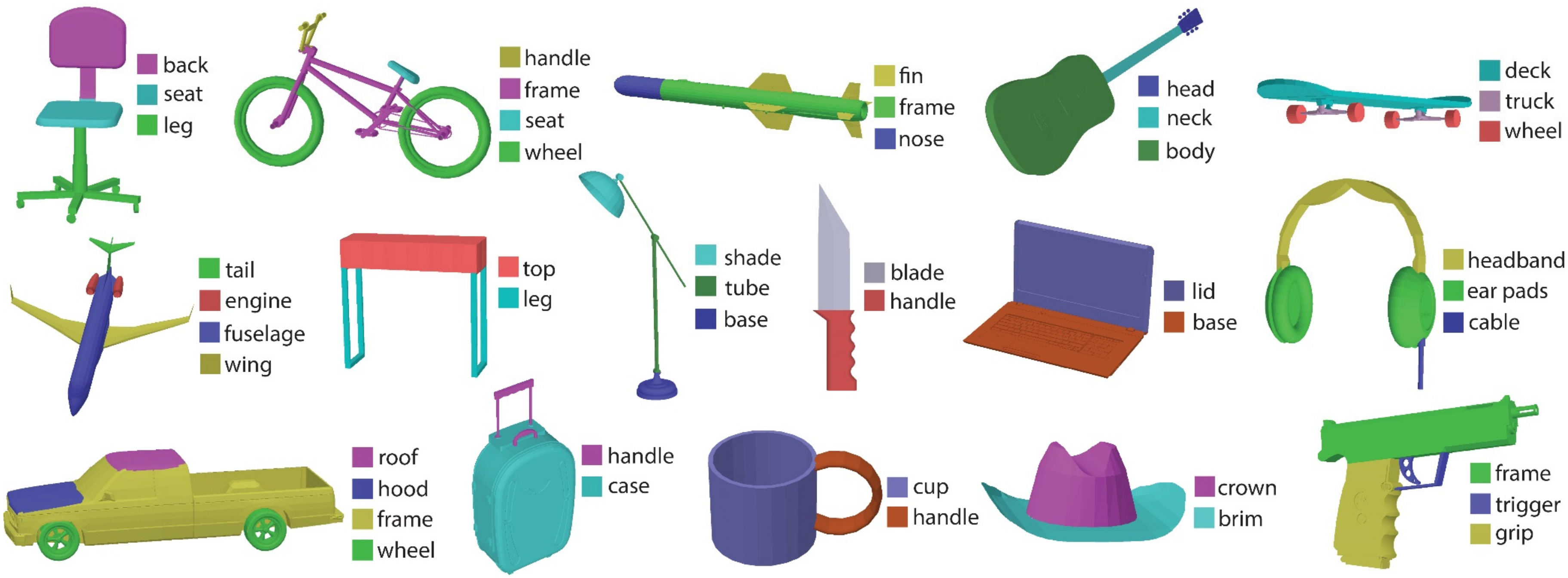
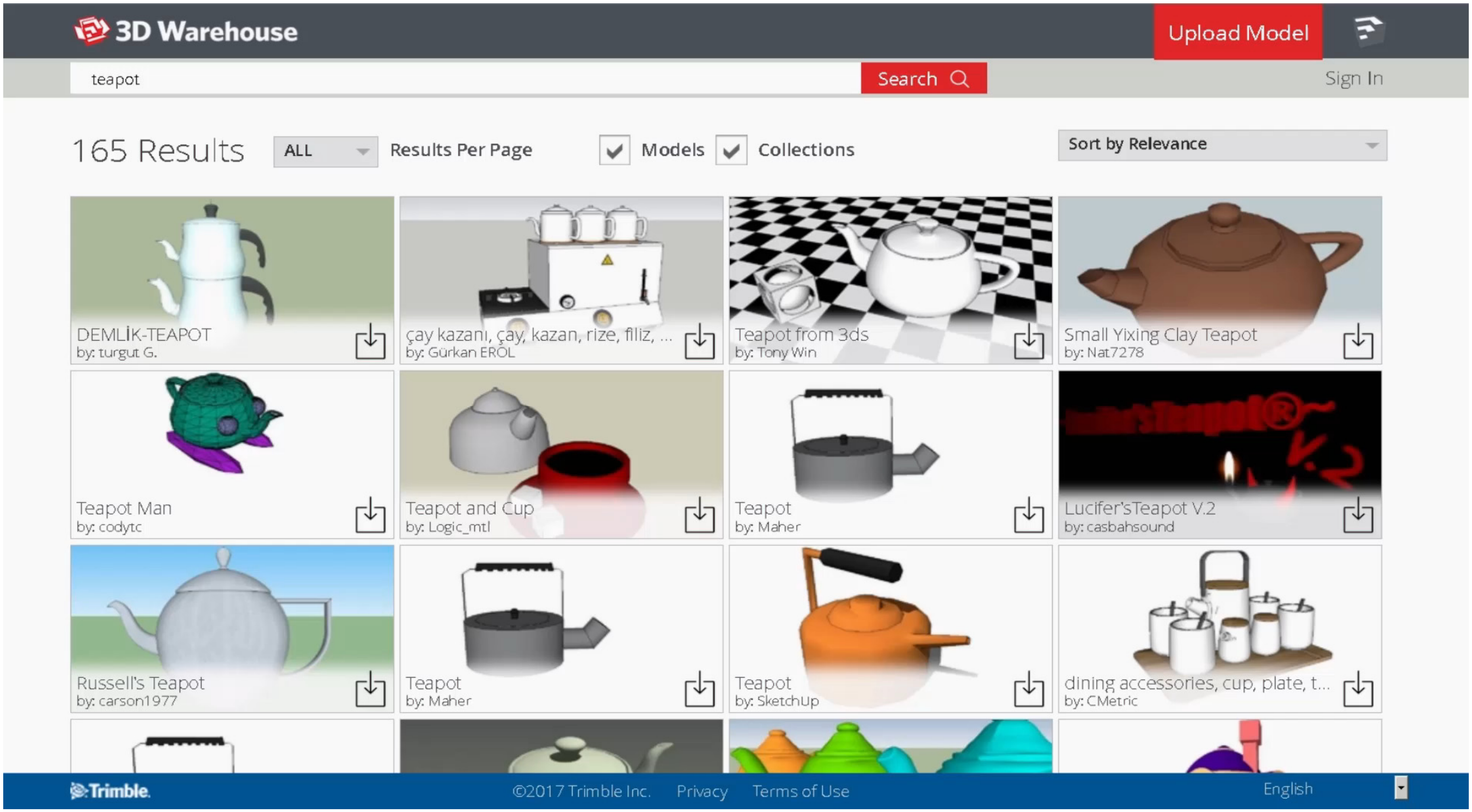
3D Shape Analysis with Multi-view Convolutional Networks



Evangelos Kalogerakis

UMASS AMHERST

3D model repositories

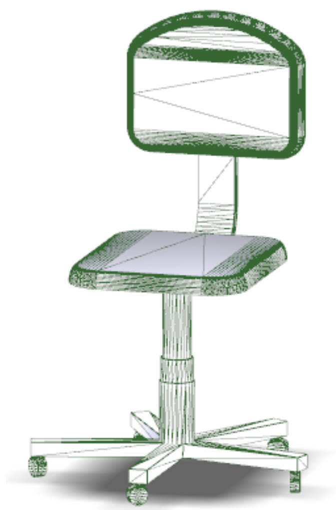


[3D Warehouse - video]

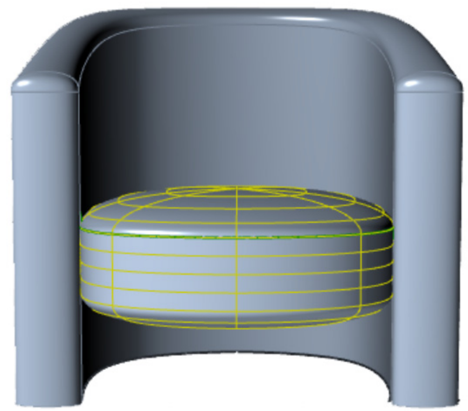
3D geometry acquisition

[KinectFusion - video]

3D shapes come in various "flavors"



Polygon meshes

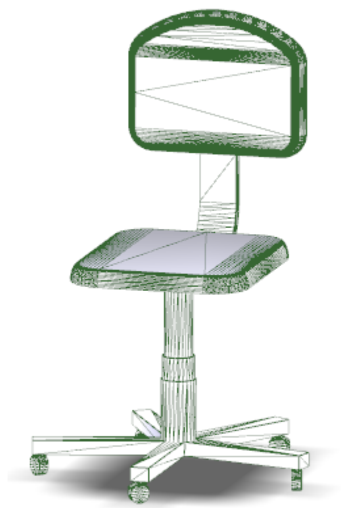


Analytic surfaces

Point Clouds

May have different resolution, non-manifold geometry, arbitrary or no texture and interior, disjoint parts, noise…

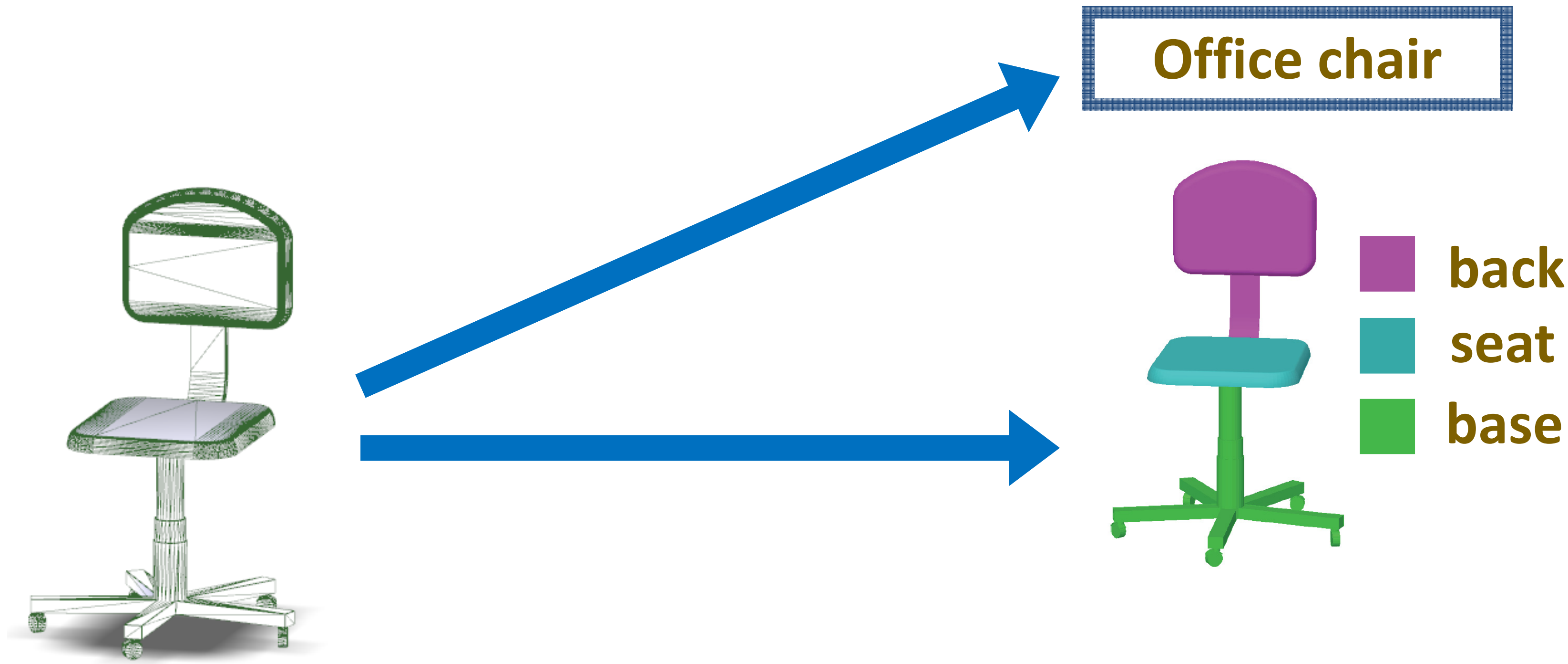
We need algorithms that “understand” shapes



Geometric representation

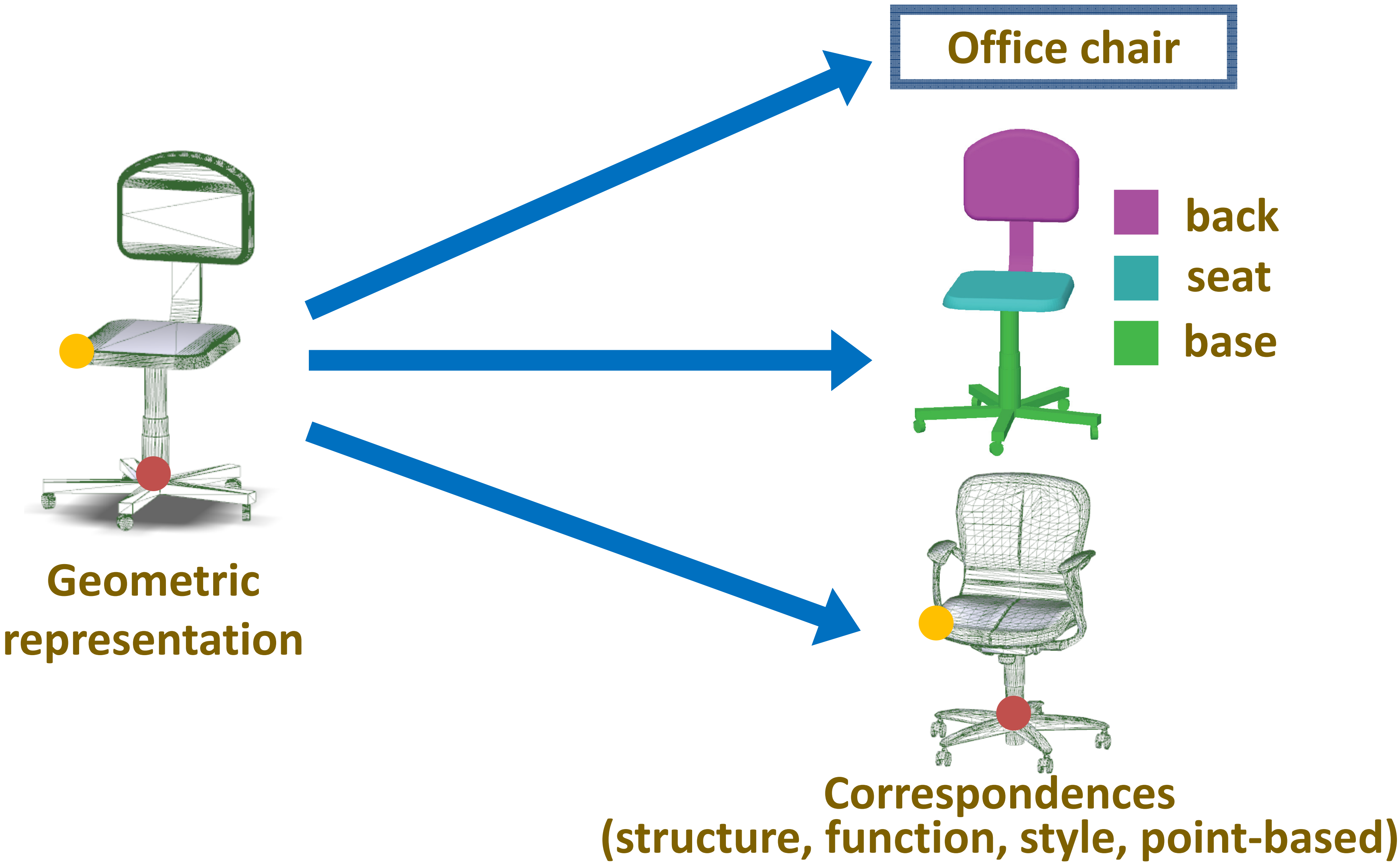
Office chair

We need algorithms that “understand” shapes

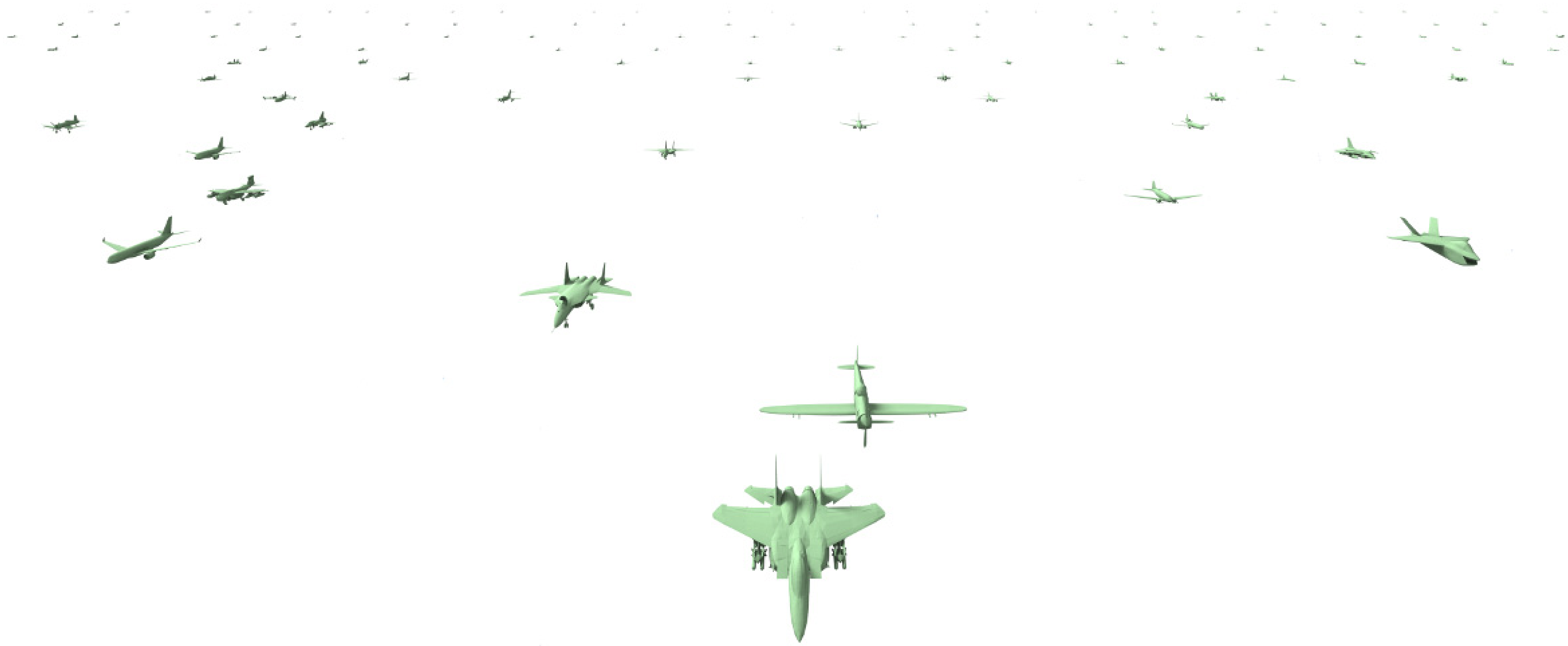


Geometric representation

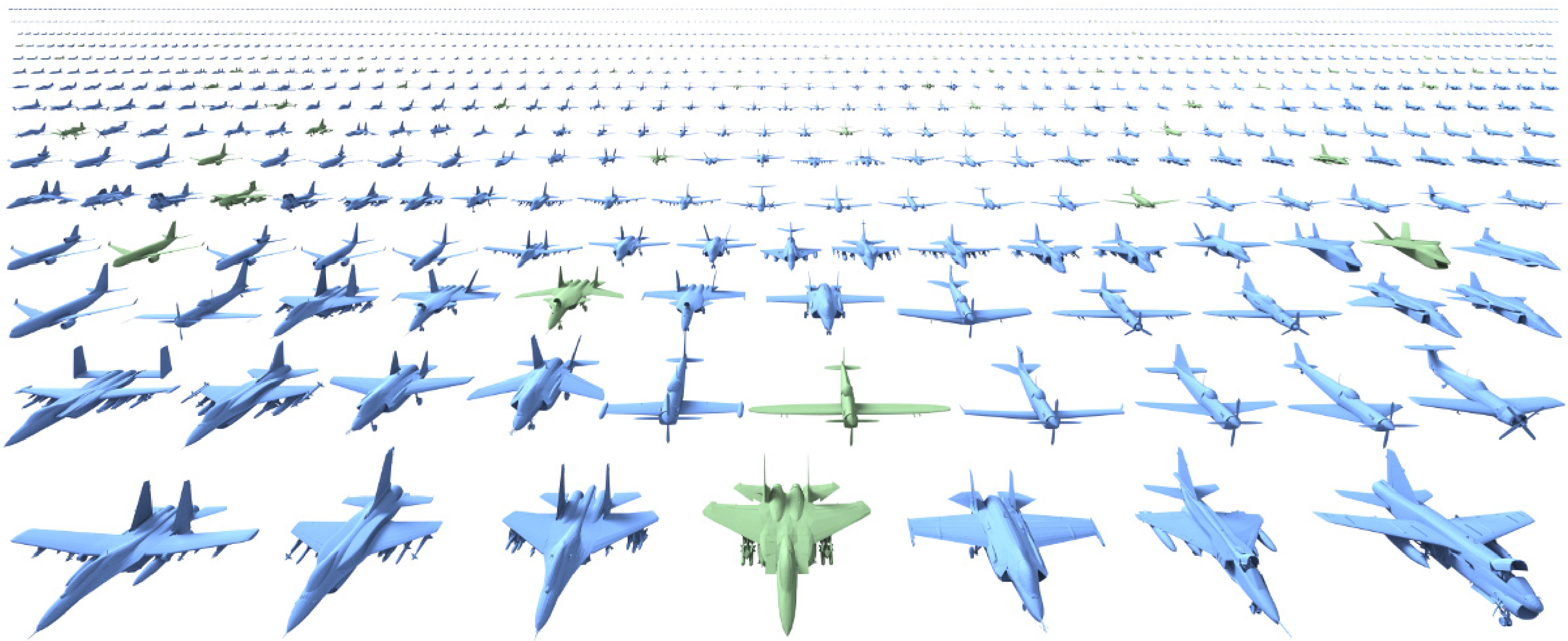
We need algorithms that “understand” shapes



Why shape understanding?
Generative models of shapes

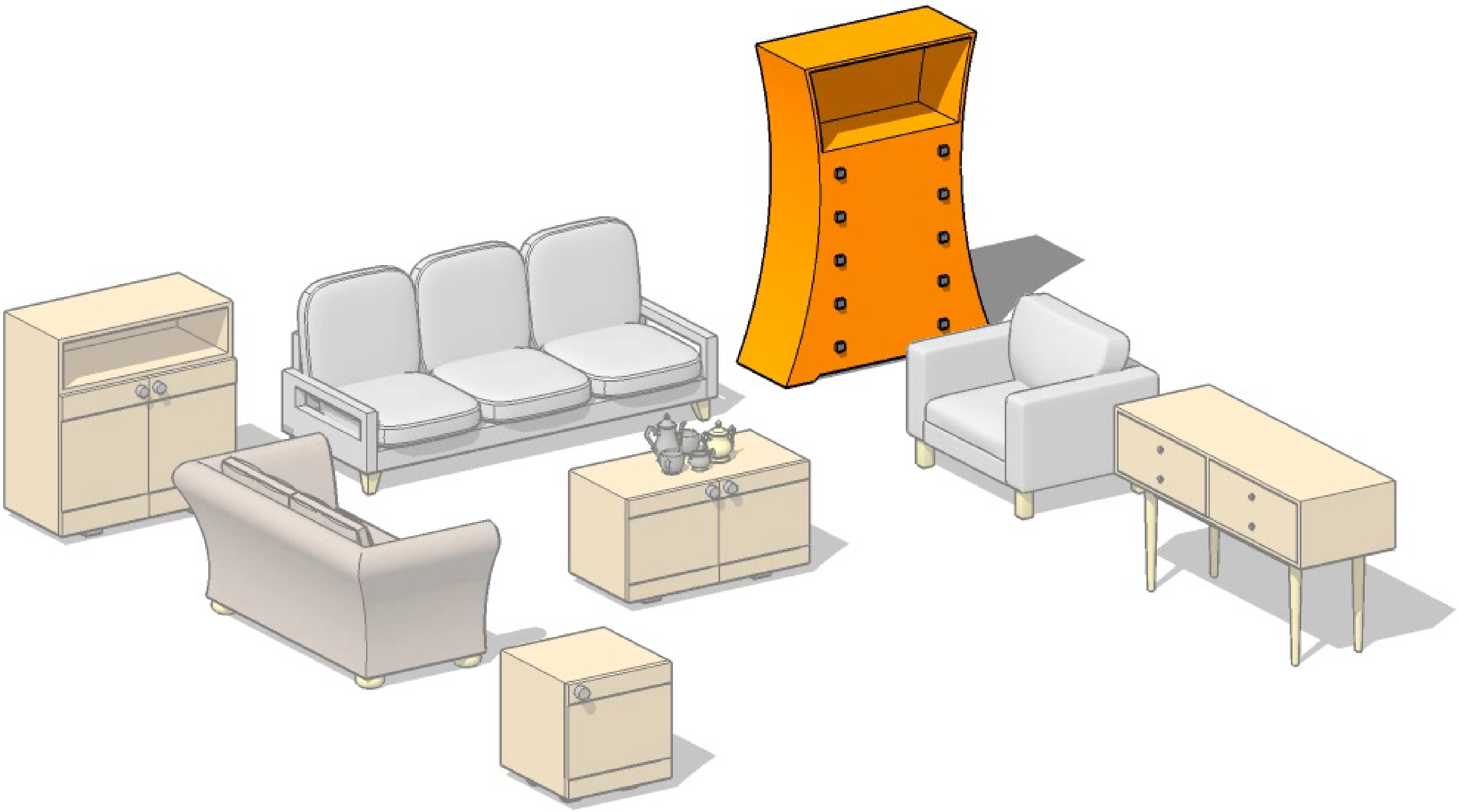


Why shape understanding?
Generative models of shapes



Kalogerakis, Chaudhuri, Koller, Koltun, SIGGRAPH 2012

Why shape understanding? Scene design

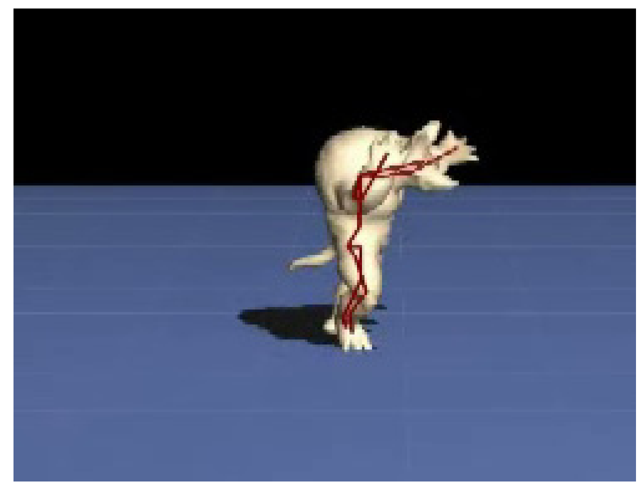


Lun, Kalogerakis, Wang, Sheffer, SIGGRAPH ASIA 2016

Why shape understanding?
Texturing

Kalogerakis, Hertzmann, Singh, SIGGRAPH 2010

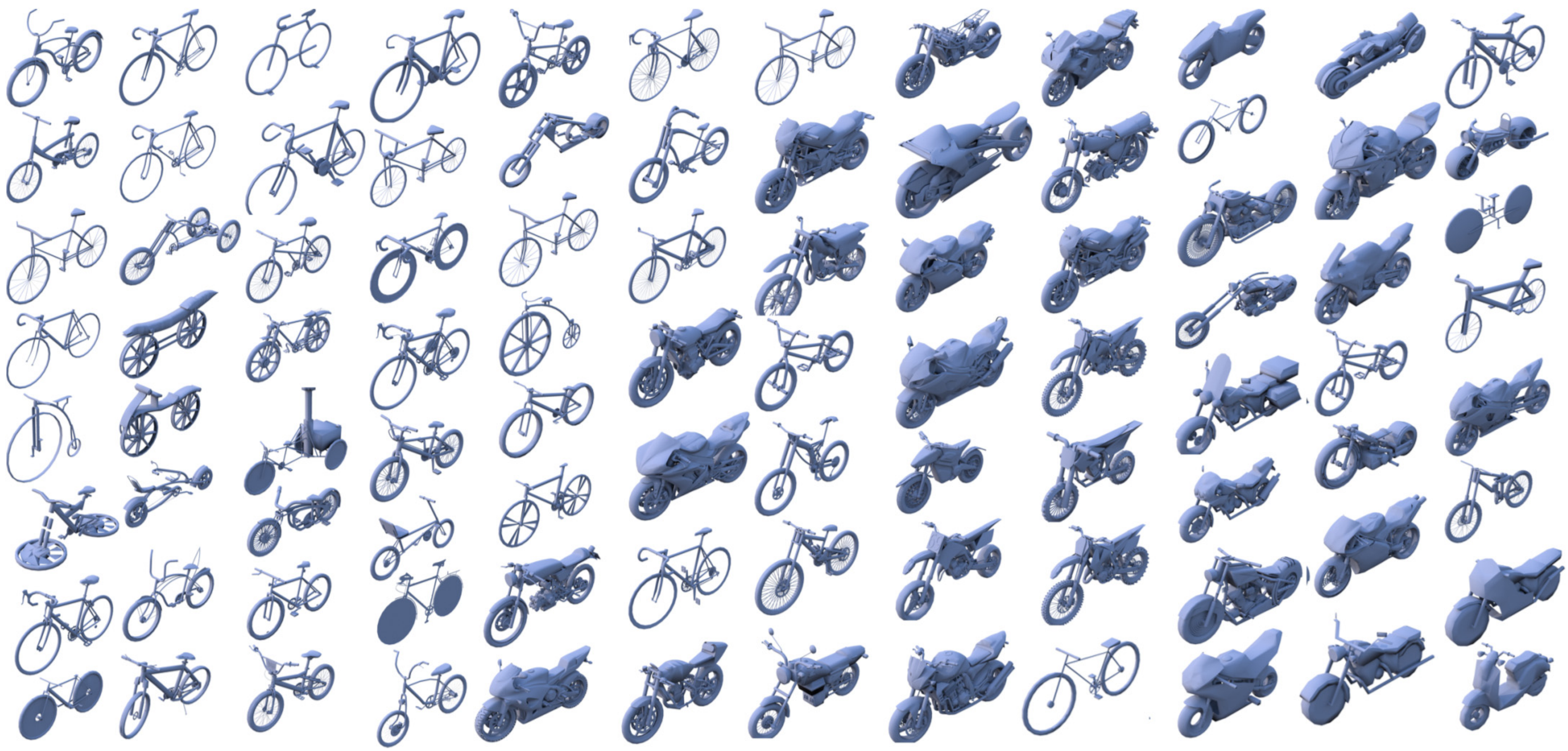
Why shape understanding?
Character Animation



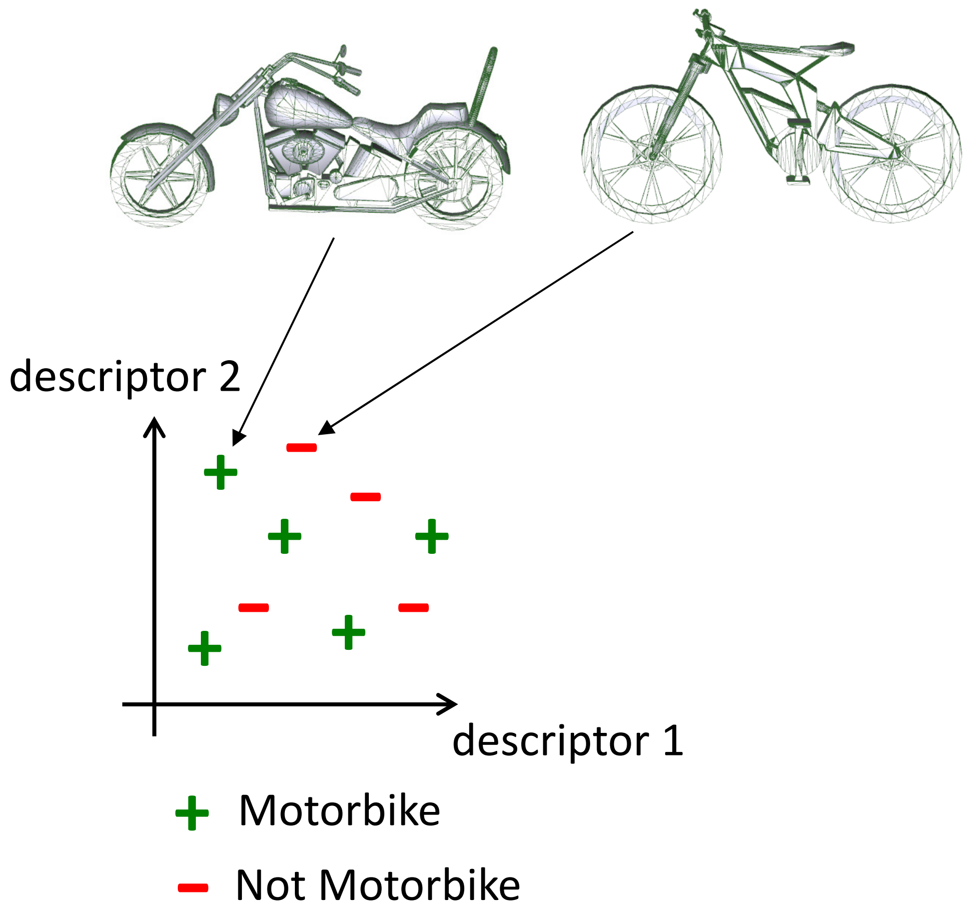
Simari, Nowrouzezahrai, Kalogerakis, Singh, SGP 2009

How can we perform shape understanding?

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

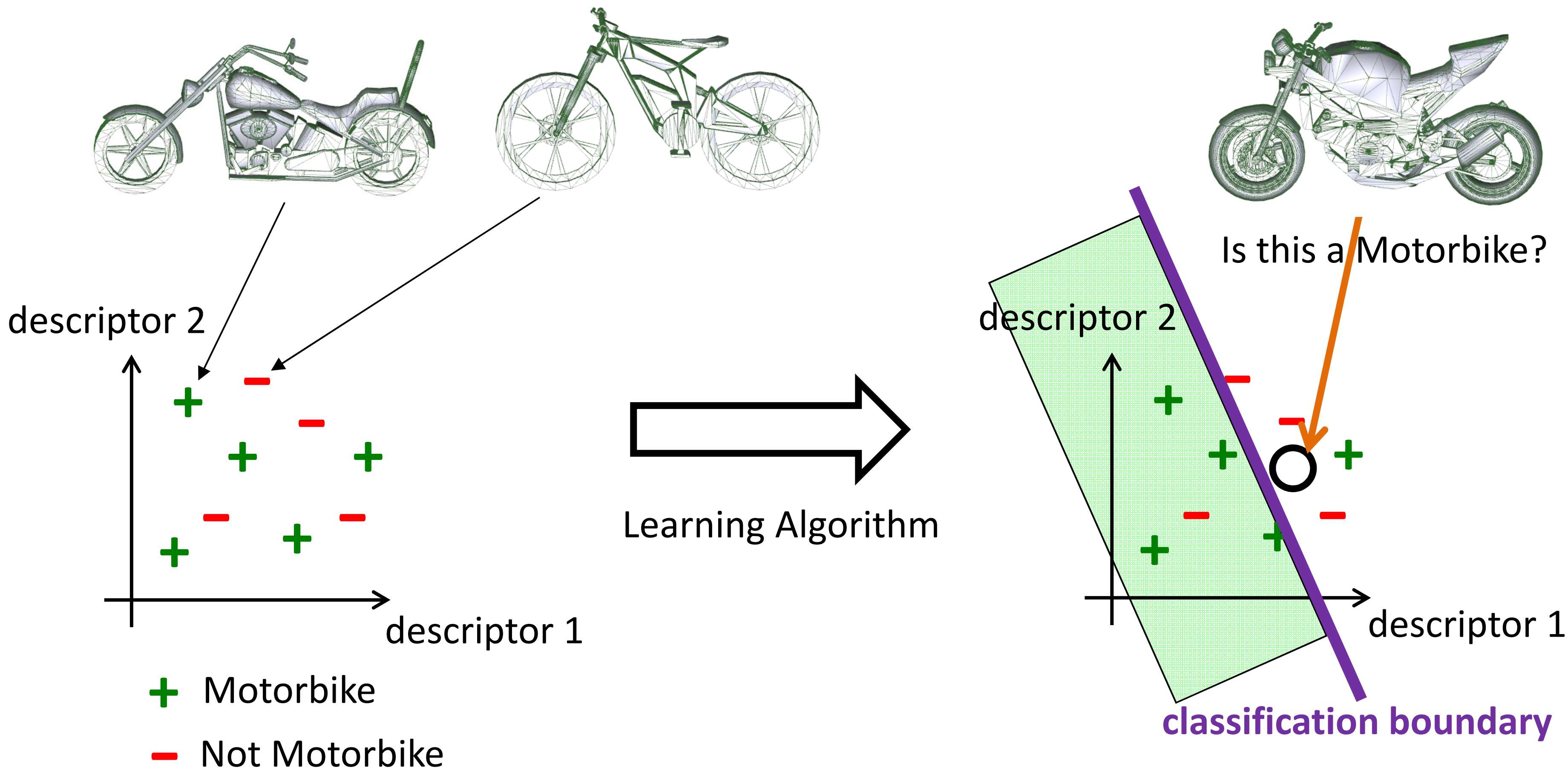


The importance of good shape descriptors



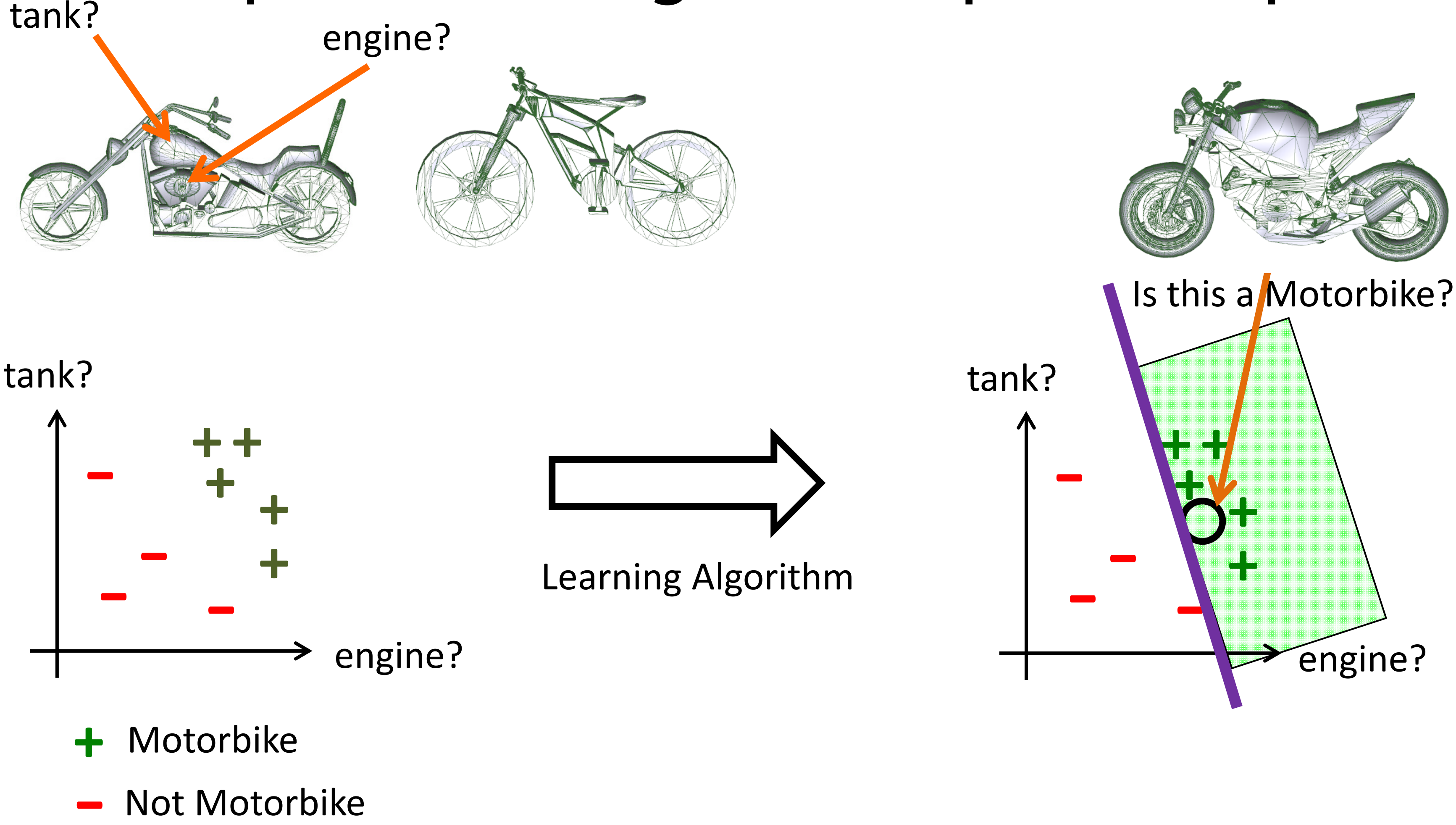
"Old-style" descriptors: surface curvature, spin images, PCA…

The importance of good shape descriptors



“Old-style” descriptors: surface curvature, spin images, PCA…

The importance of good shape descriptors

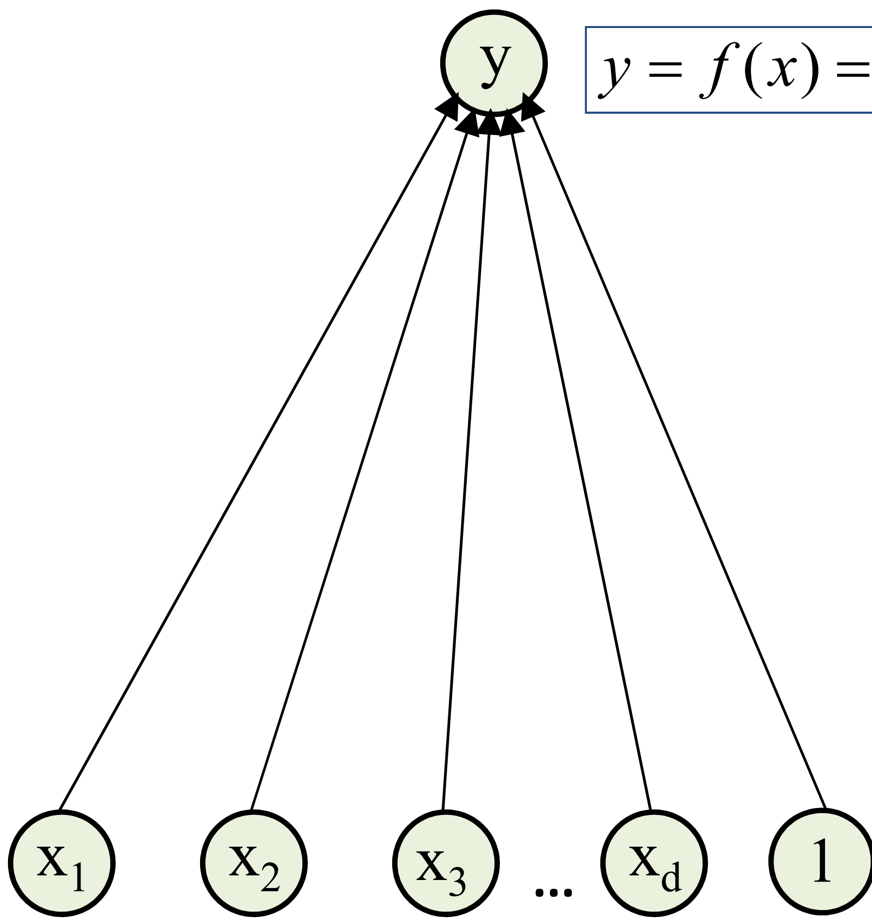


Need descriptors that capture semantics, function…

From "shallow" mappings…

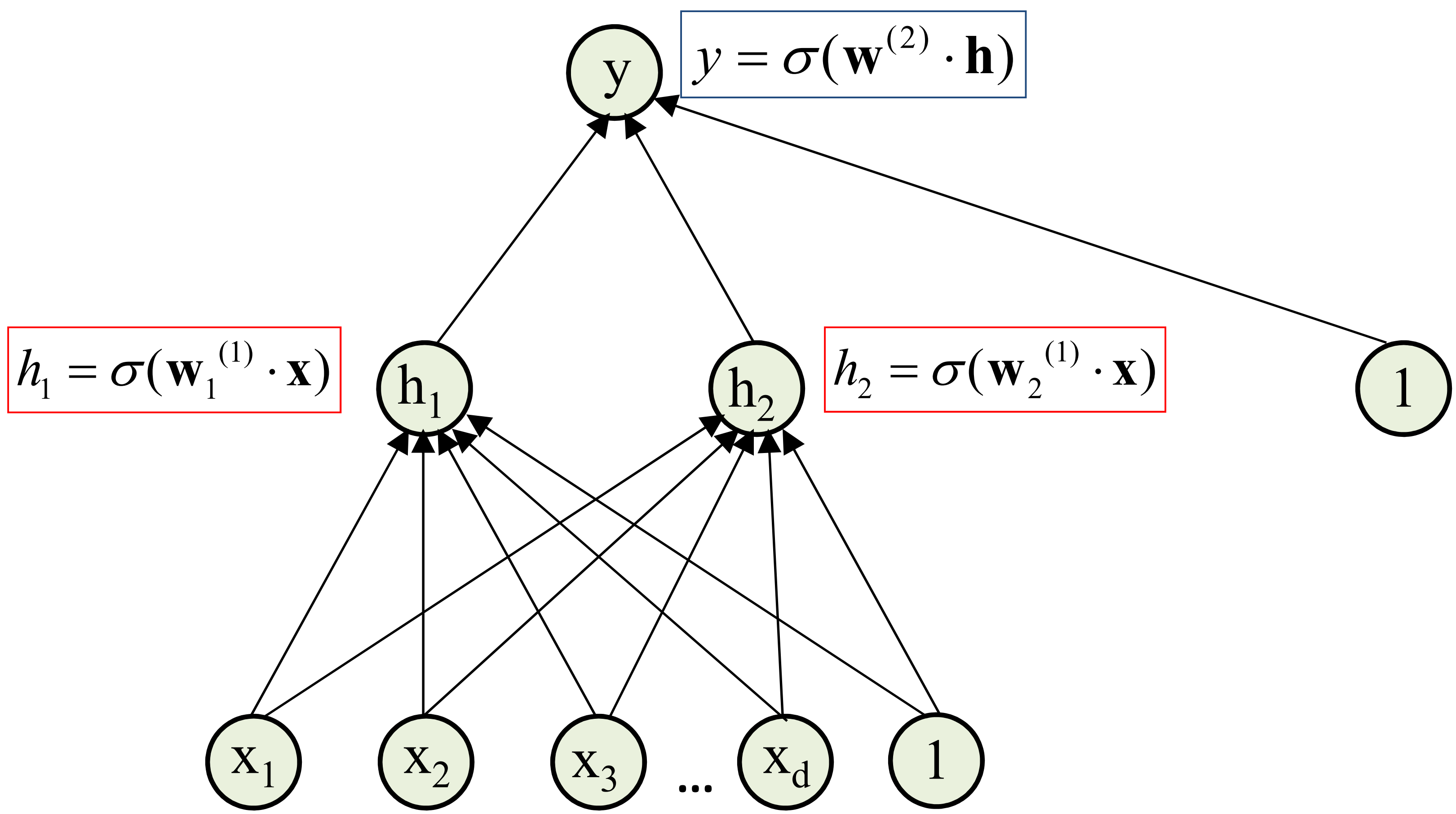
Old-style approach: output is a **direct function** of hand-engineered shape descriptors

$$y = f(x) = \sigma(\mathbf{w} \cdot \mathbf{x})$$



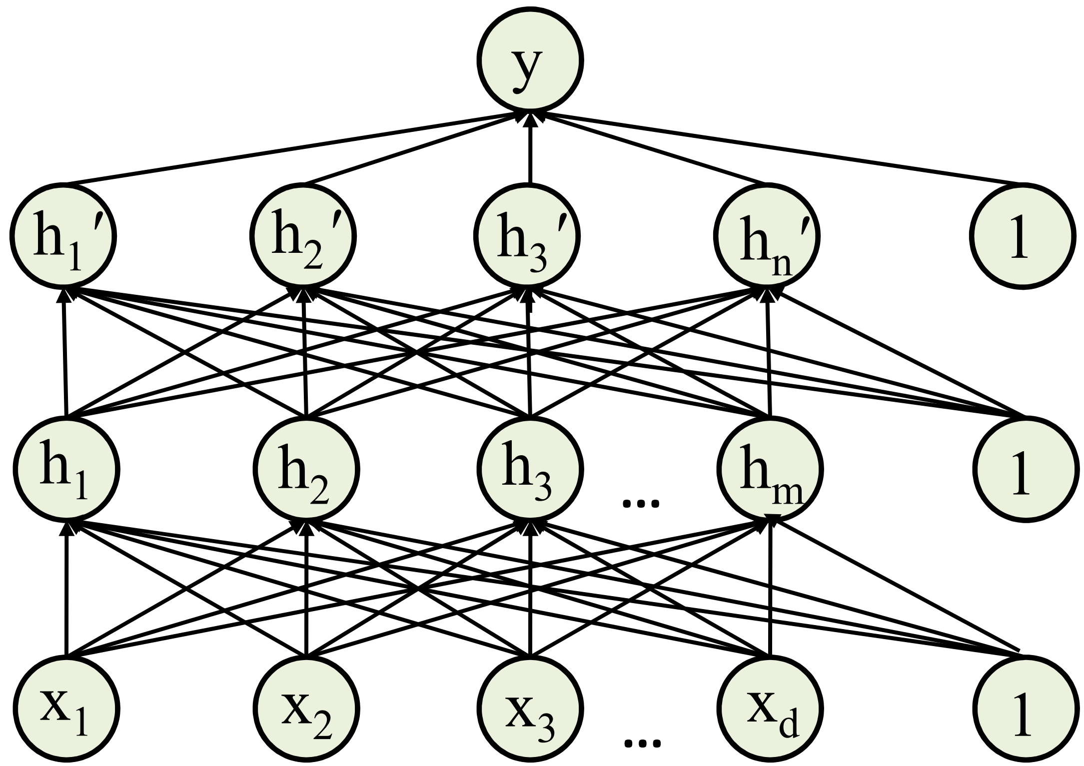
… to neural nets

Introduce **intermediate learned functions** that yield optimized descriptors.

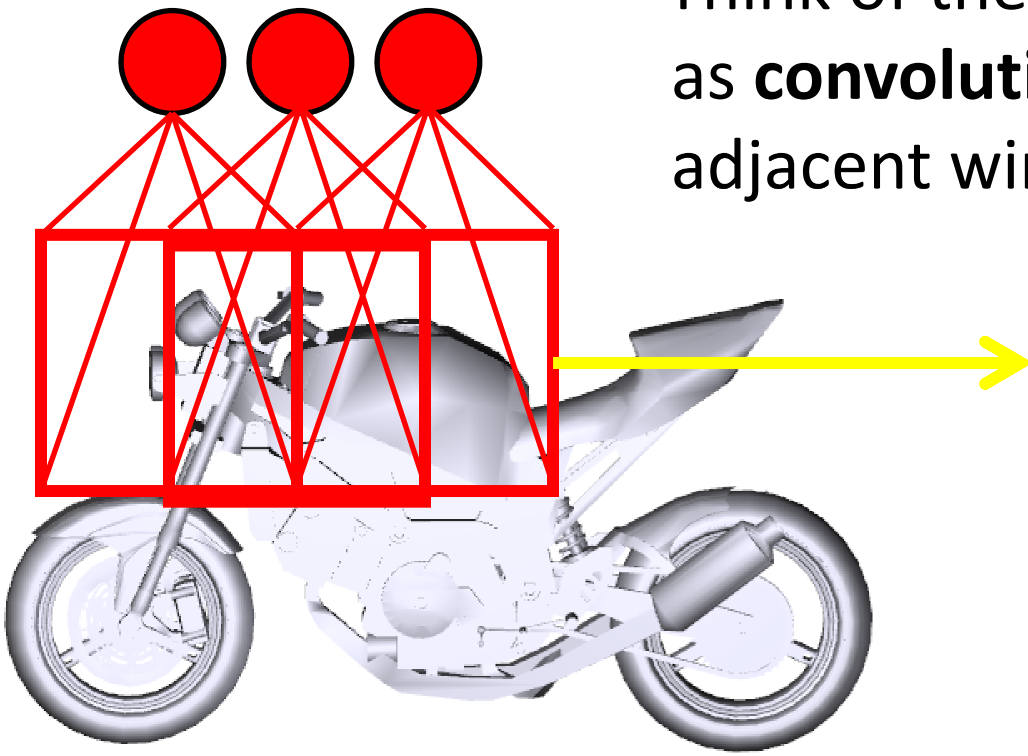


… to deep neural nets

Stack several layers…



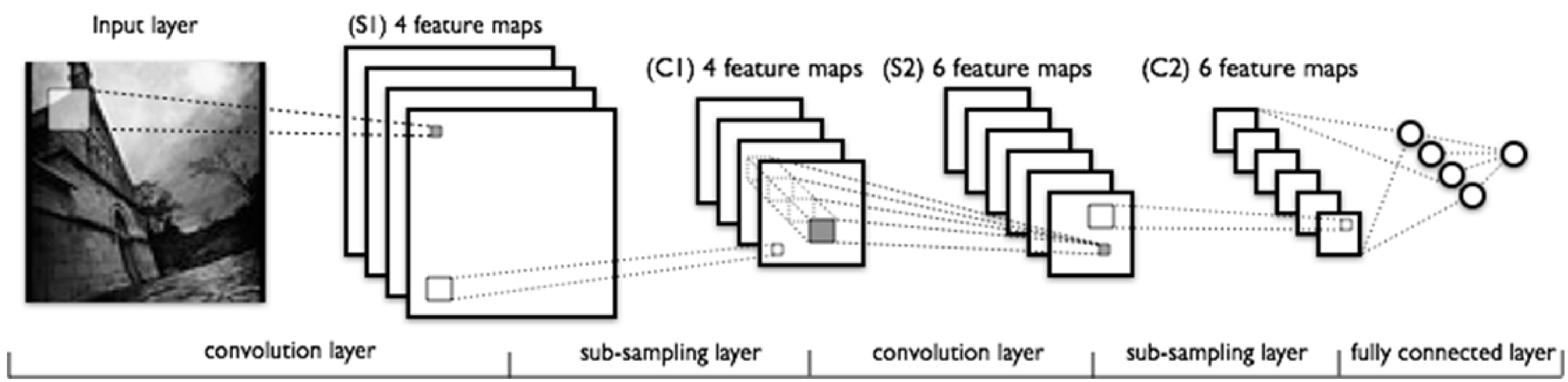
Convolutional neural networks



Think of these intermediate functions as **convolutional filters** acting on small adjacent windows

Convolutional neural networks

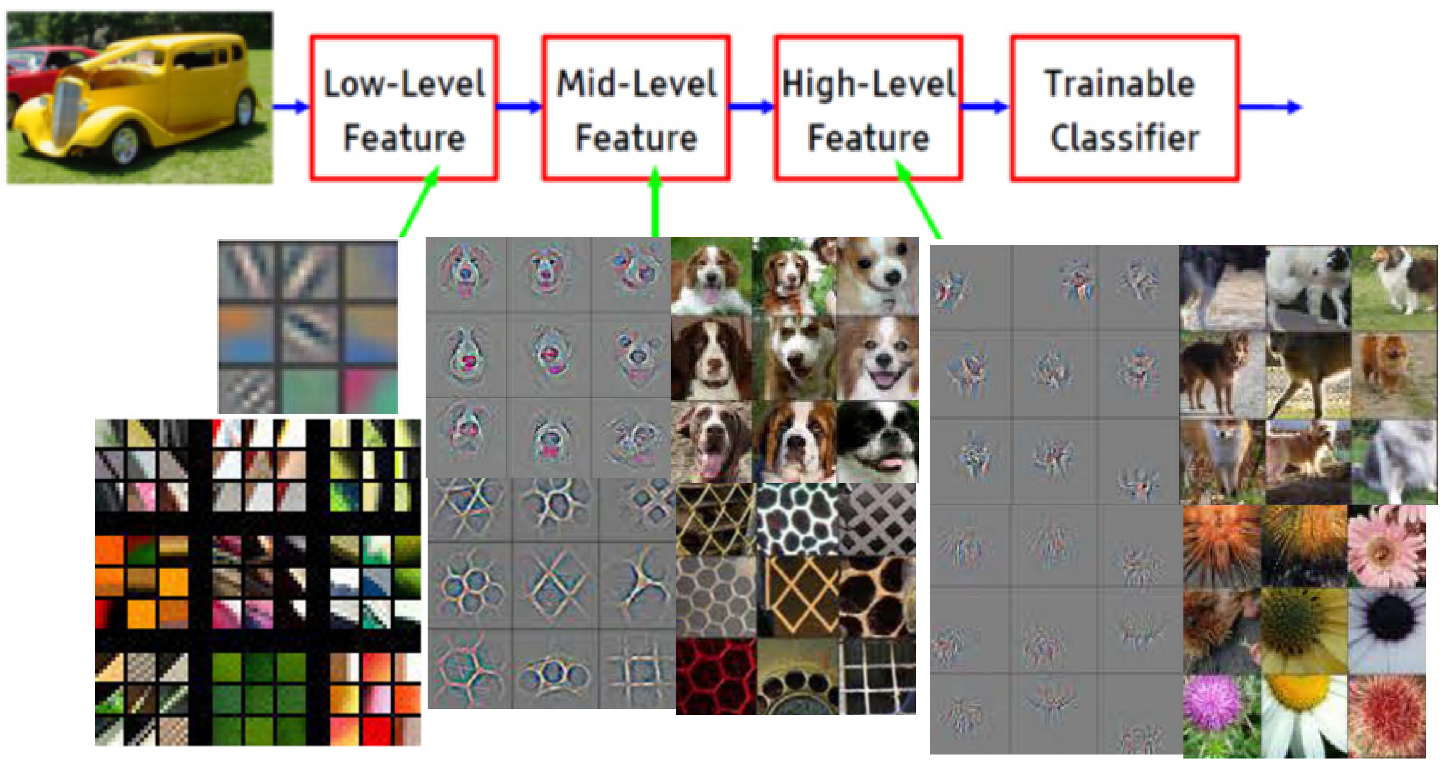
Basic idea: interchange several **convolutional** and **pooling** (subsampling) **layers**.



Source: http://deeplearning.net/tutorial/lenet.html

The image processing “success story”

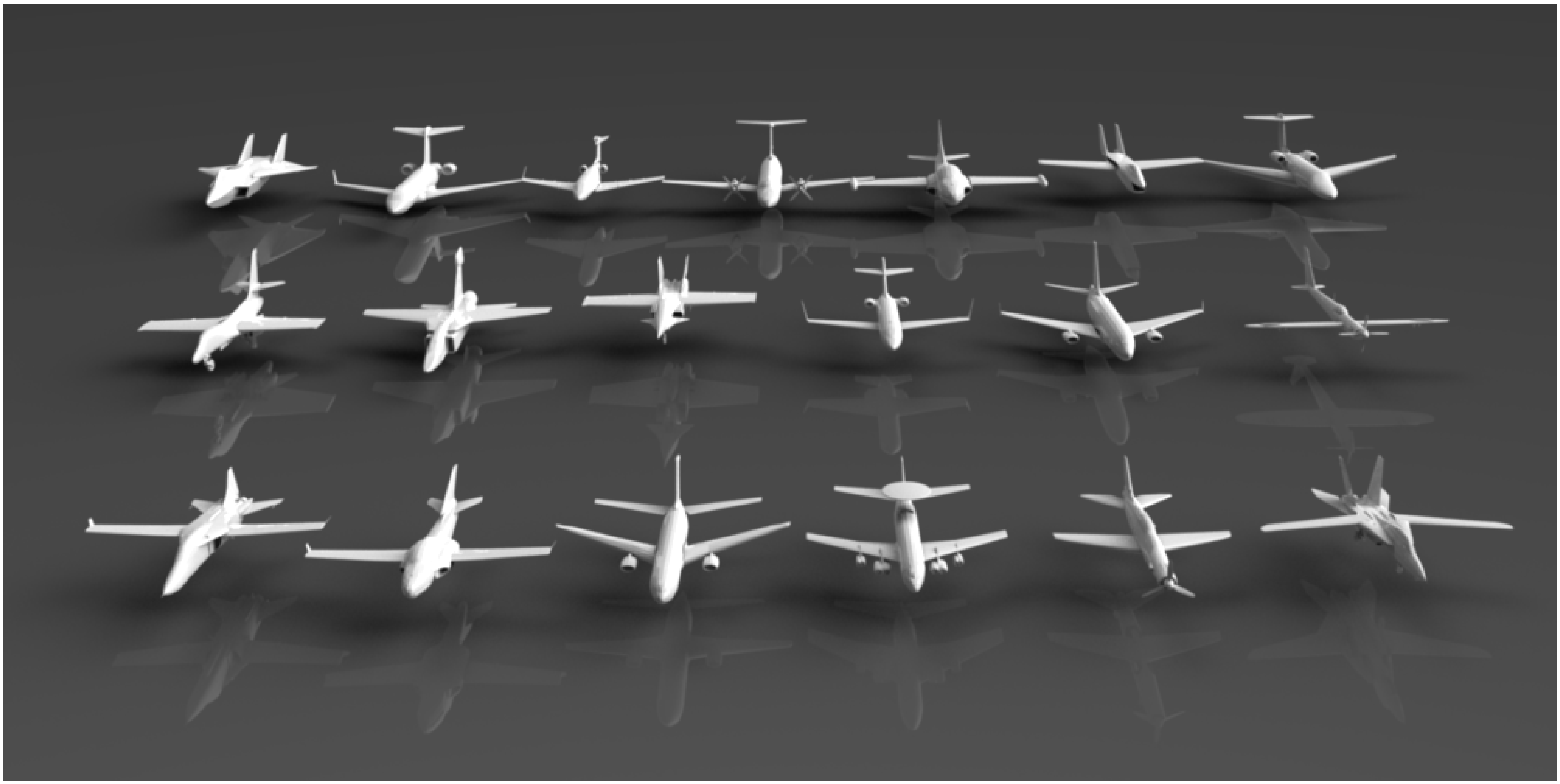
The convolution filters capture **various hierarchical patterns** (edges, sub-parts, parts…). Convnets have achieved high accuracy in several image-processing tasks.



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

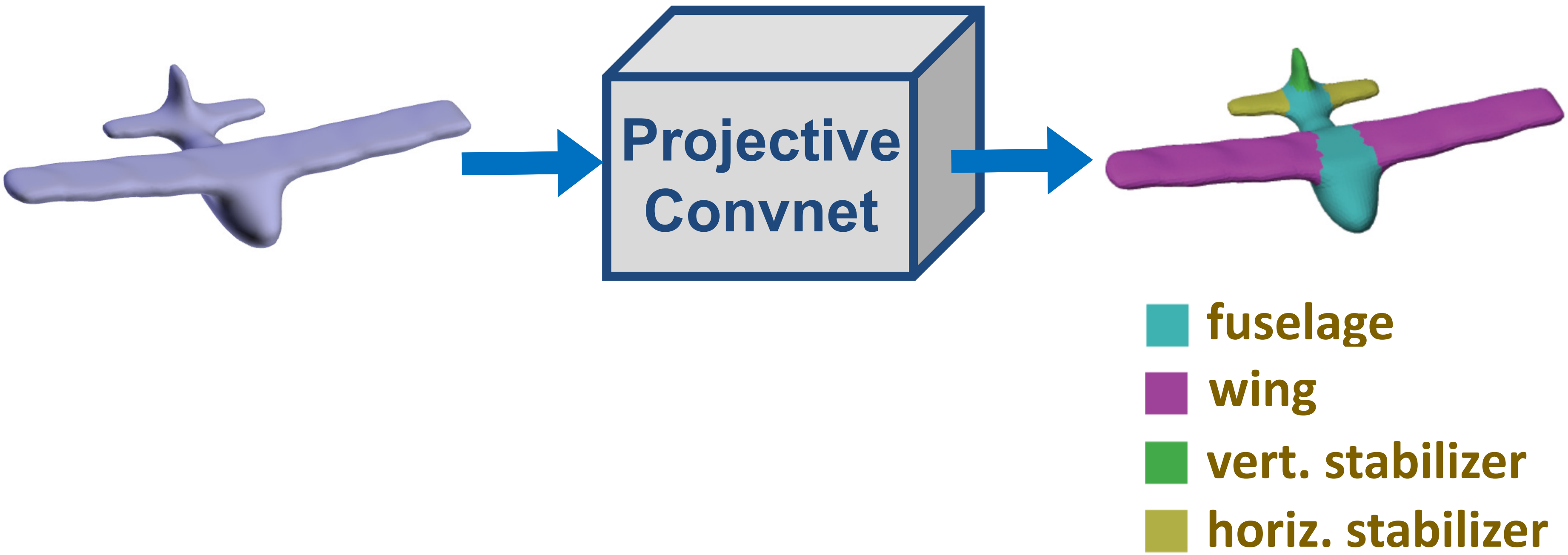
How can we apply convnets for 3D shapes?

Motivated by the **success of image-based architectures** and the fact that 3D shapes are often **designed for viewing…**



View-based convnets for 3D shapes

… we introduced **view-based** convnets for 3D shape analysis!



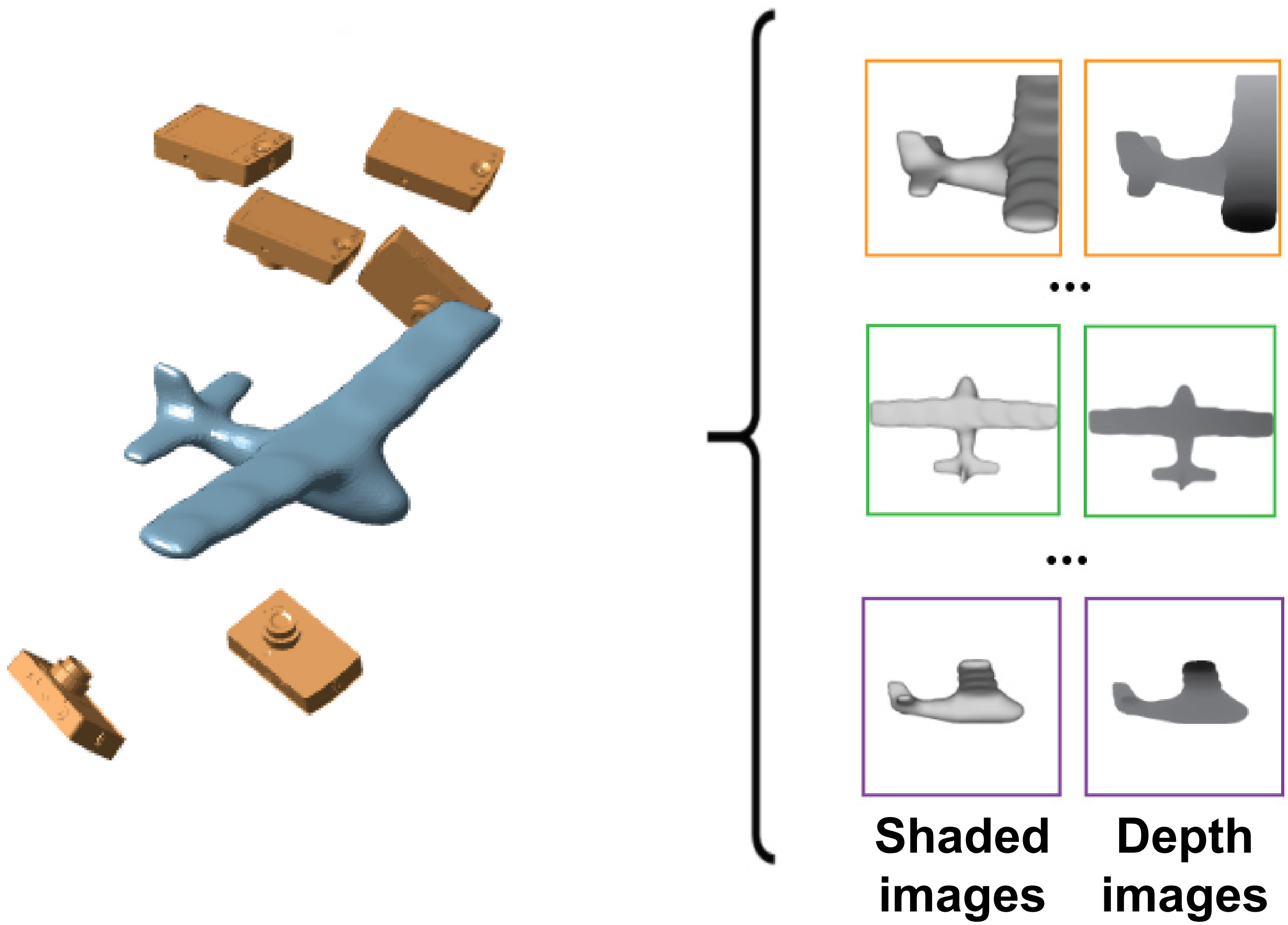
E. Kalogerakis, M. Averkiou, S, Maji, S. Chaudhuri, CVPR 2017 (oral)

Input: shape as a collection of rendered views

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

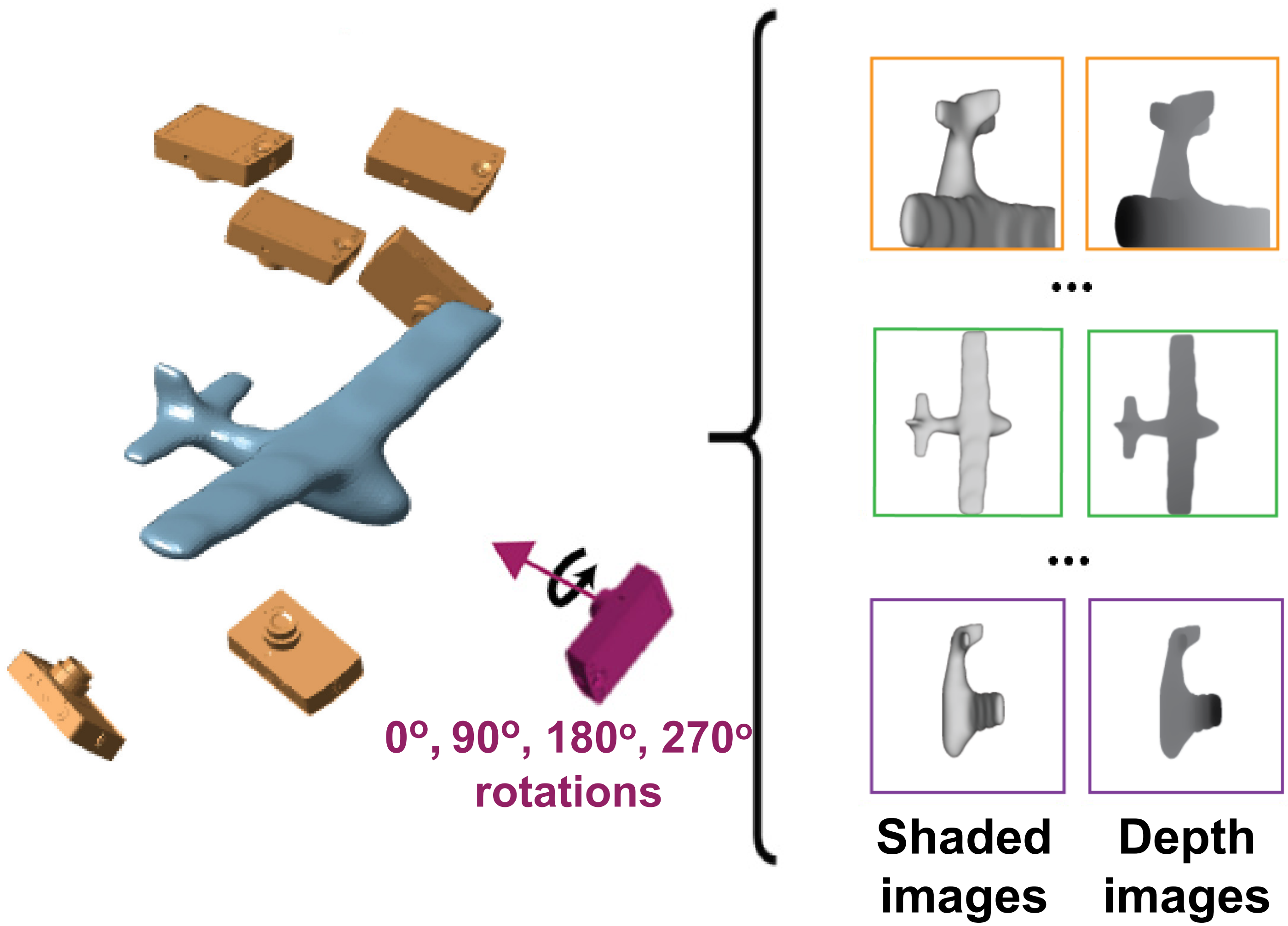
Input: shape as a collection of rendered views

Render **depth & shaded images** (normal dot view vector)



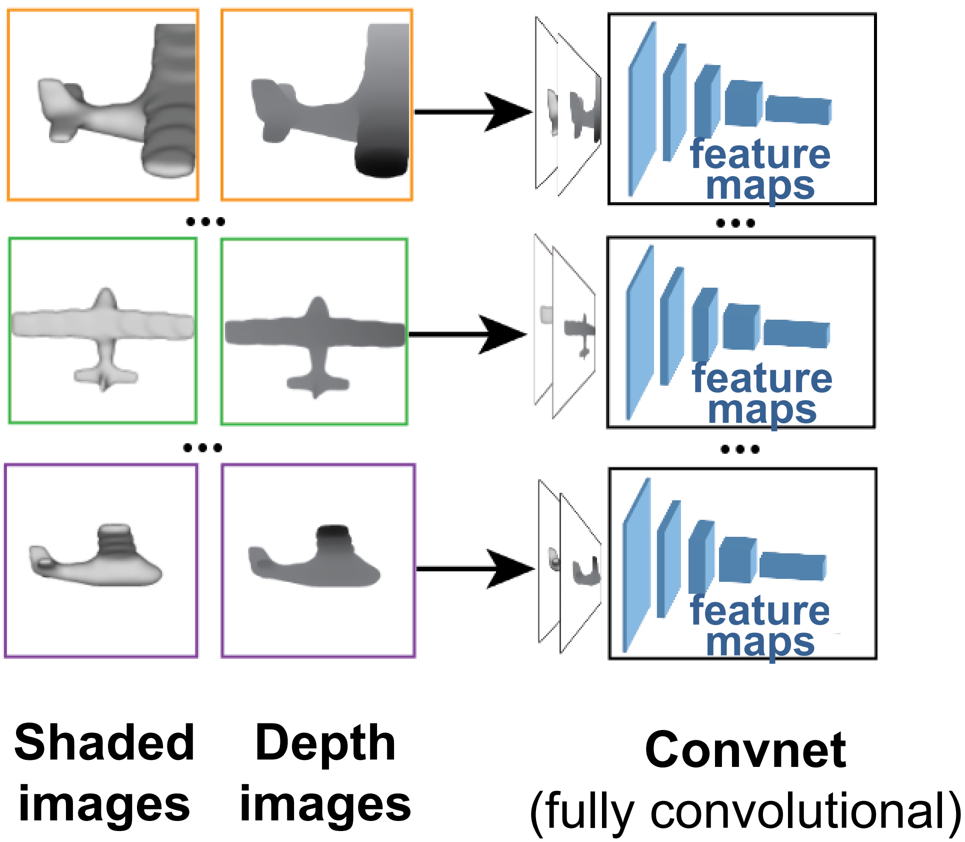
Input: shape as a collection of rendered views

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



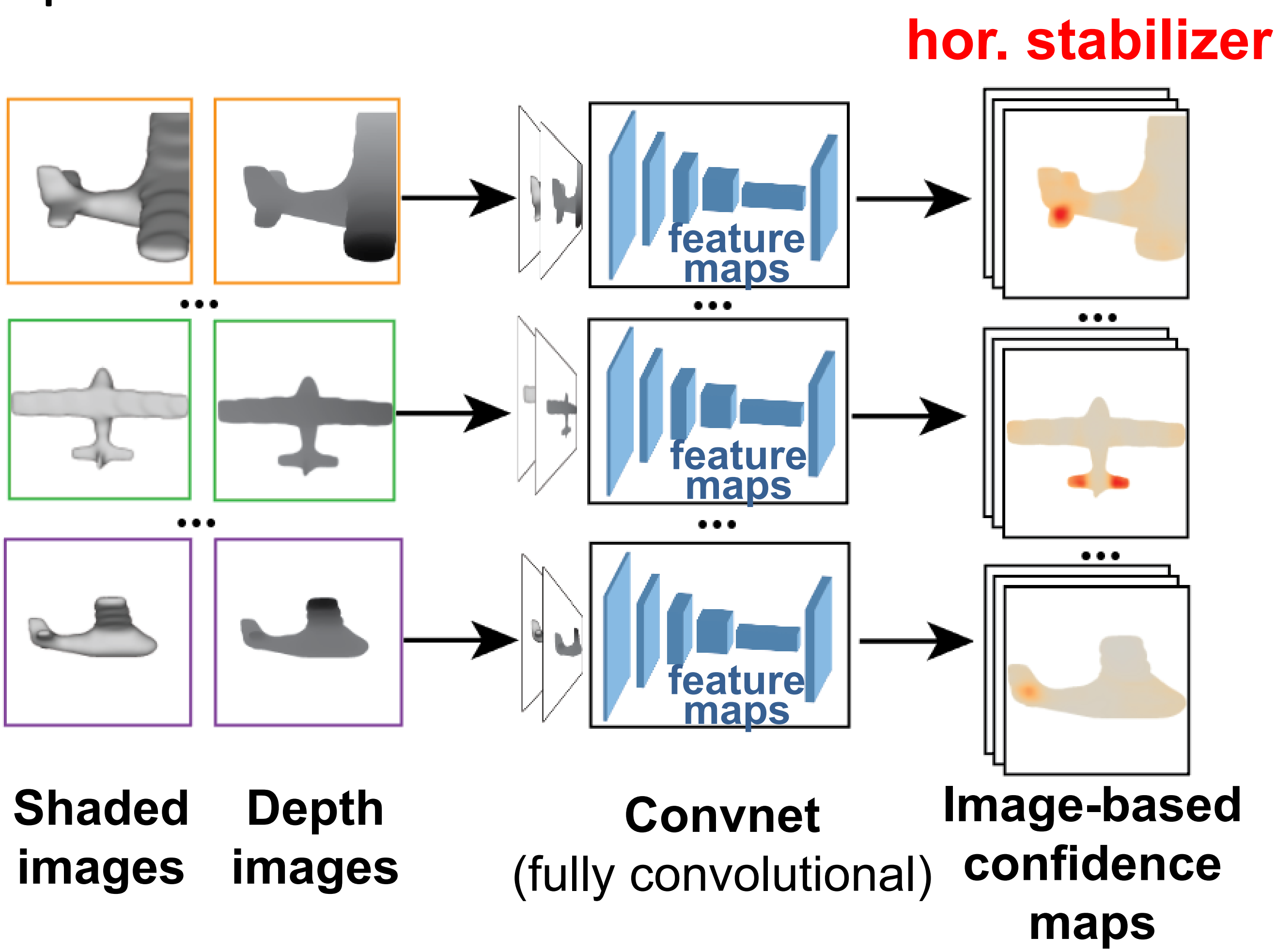
Projective convnet architecture

Each **pair of depth & shaded images** is processed by a convnet. Views are **not ordered** (**no view correspondence** across shapes). Convnets have **shared parameters**.



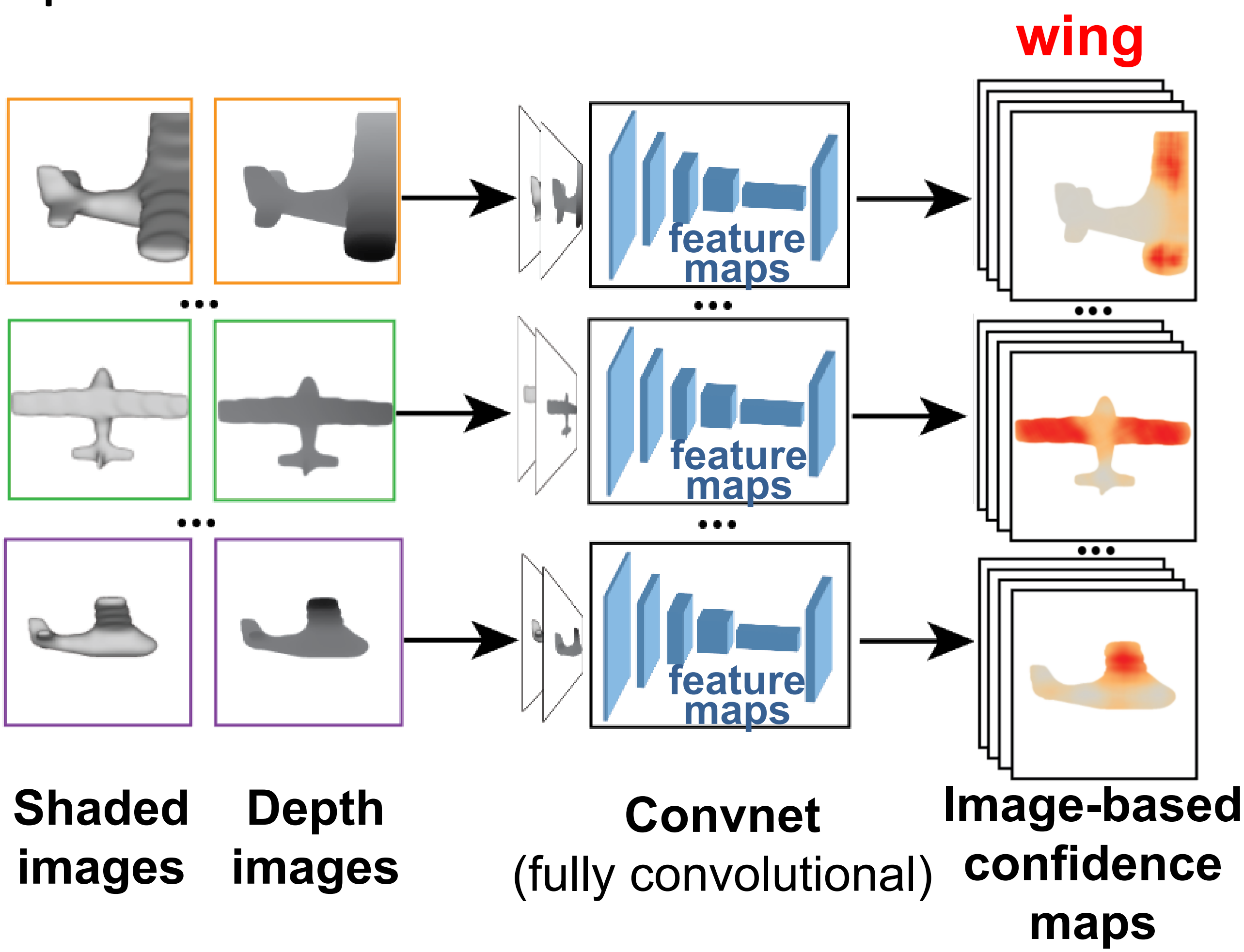
Projective convnet architecture

The output of each convnet branch is a **confidence map** per part label.



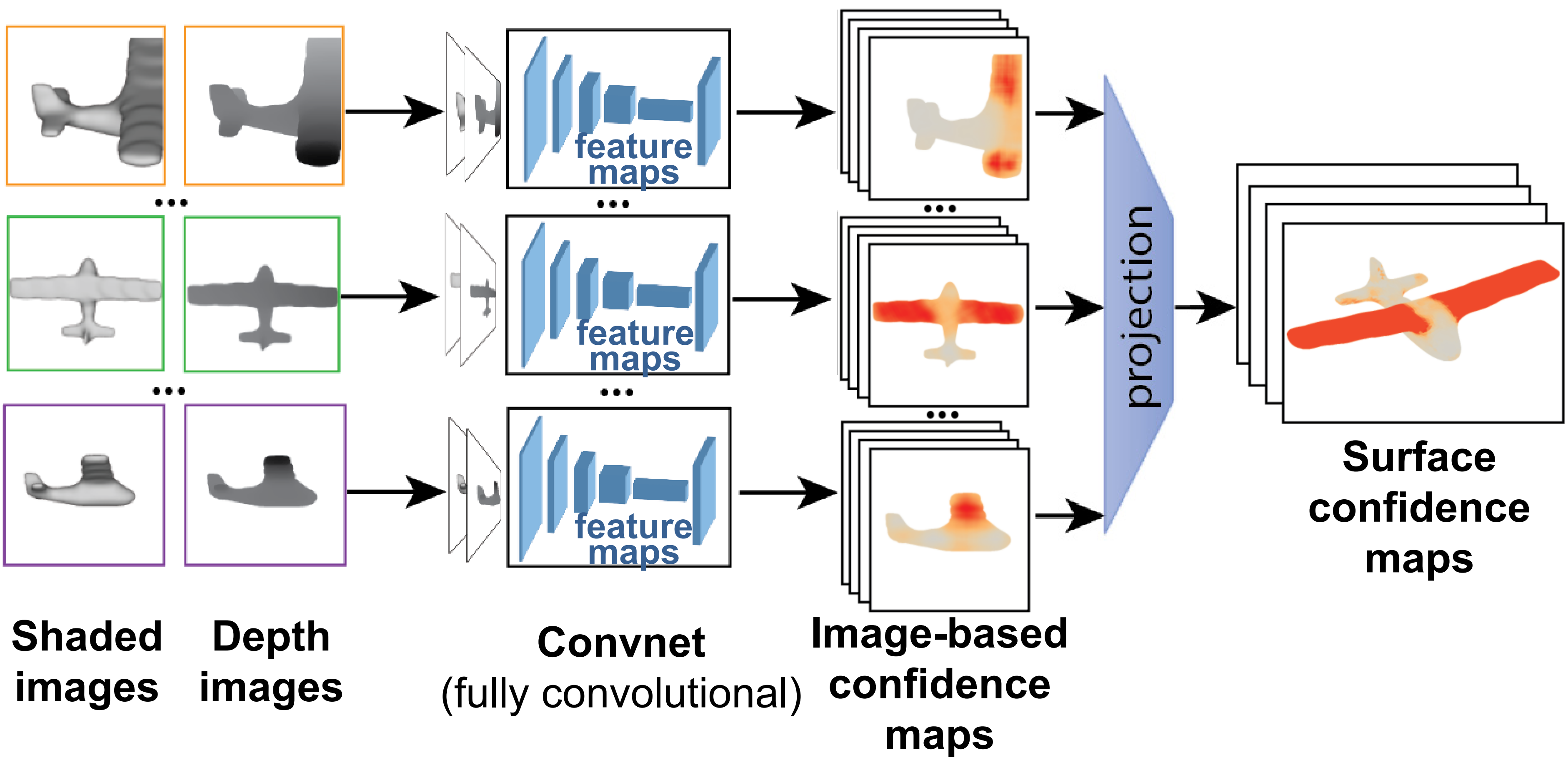
Projective convnet architecture

The output of each convnet branch is a **confidence map** per part label.



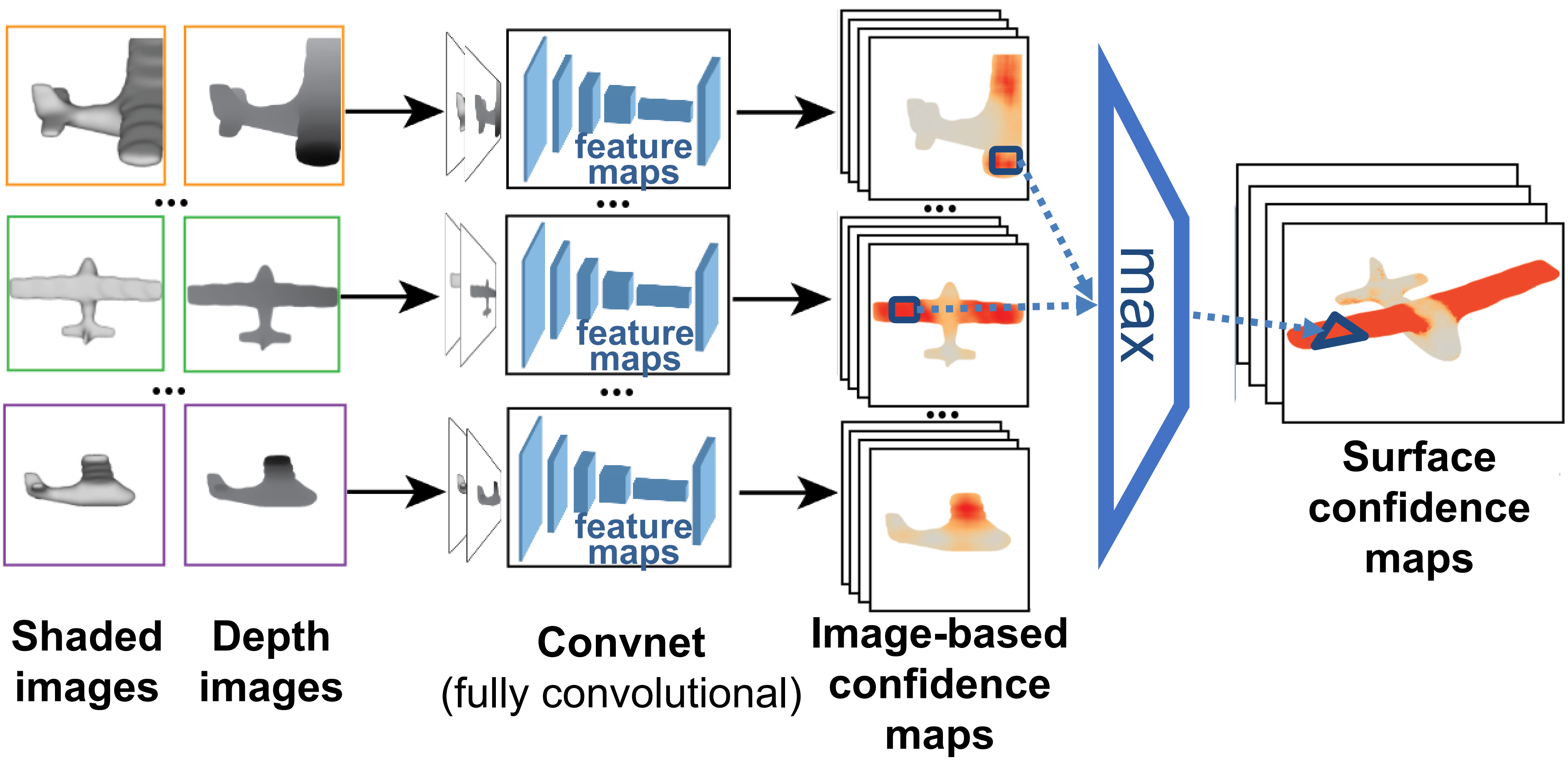
Projective convnet architecture

Since we want our output on the surface, we **aggragate the image confidences across all views onto the surface**.



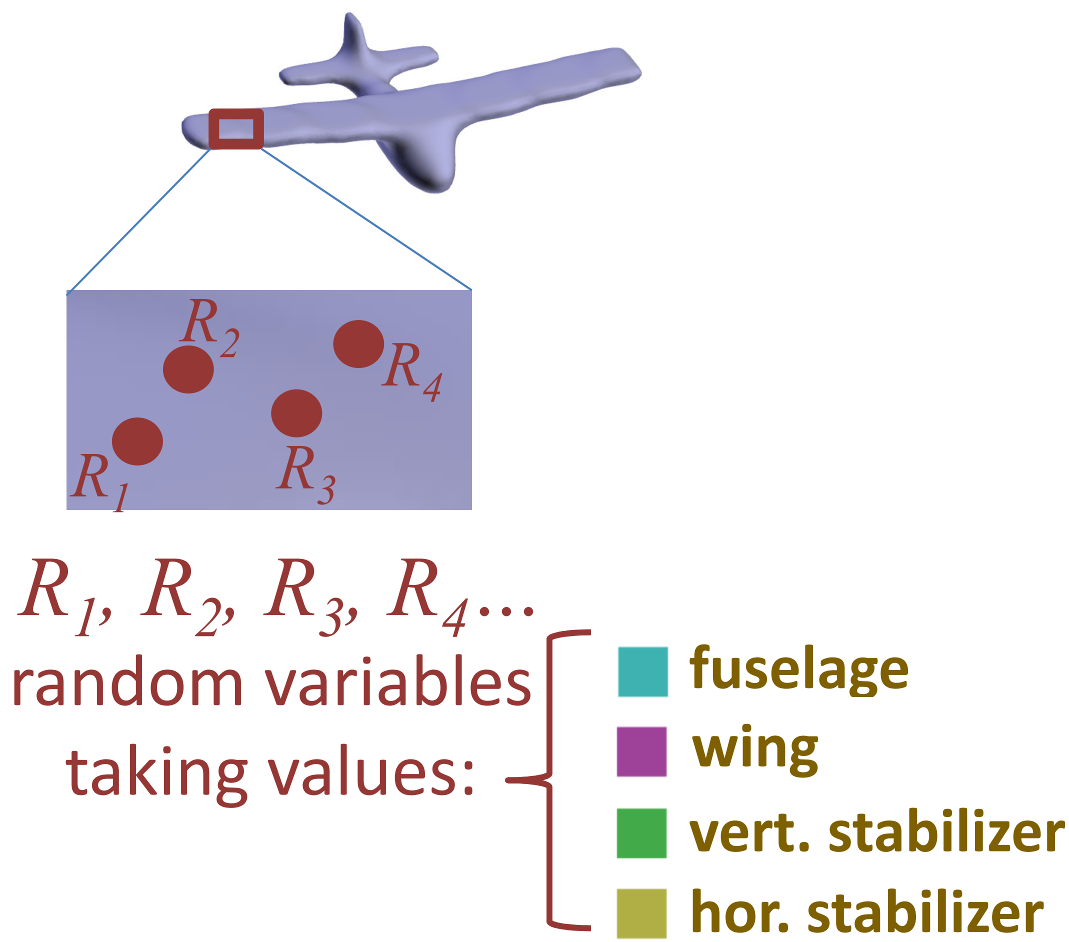
Projective convnet architecture

For each face / surface point, find all pixels that include it across all views, and use the **max** of confidence per label.



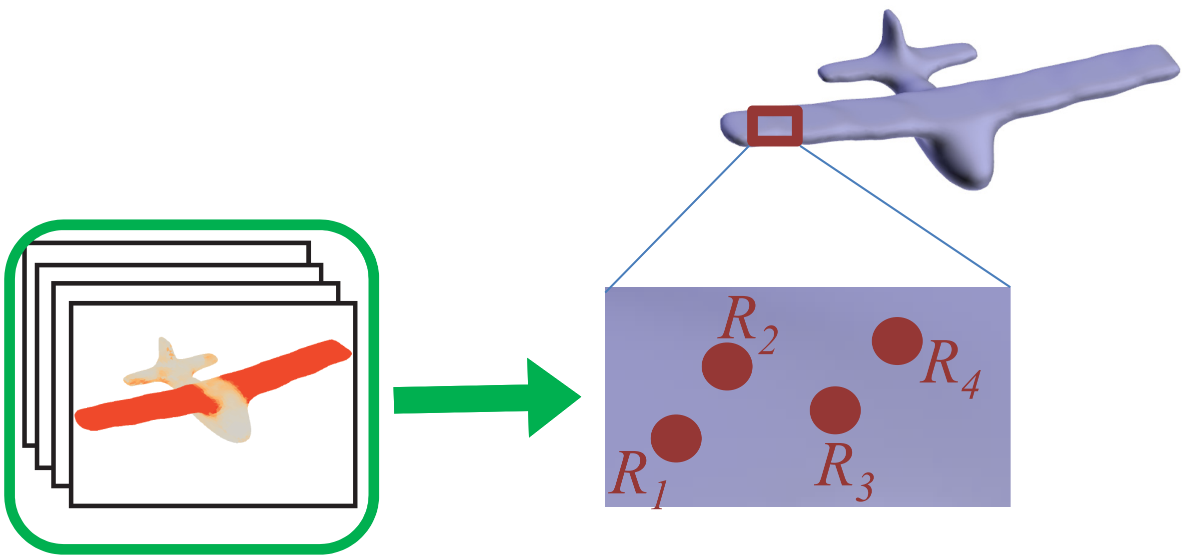
Projective convnet architecture: CRF layer

The last layer performs **inference in a probabilistic model defined on the surface** to promote coherent labeling.



Projective convnet architecture: CRF layer

It has the form of a **Conditional Random Field** whose unary term represents the surface-based label confidences

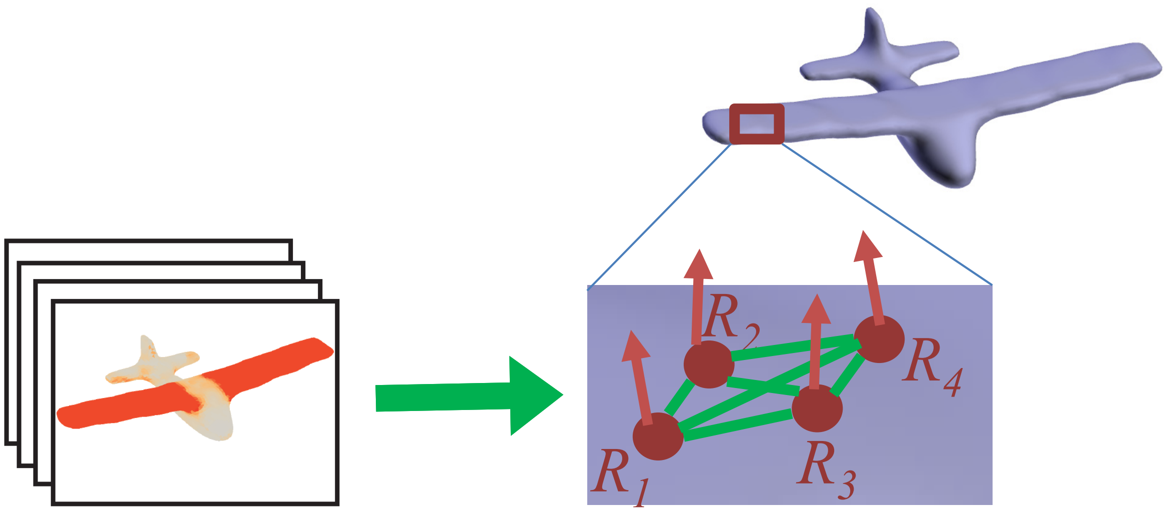


$$P(R_1, R_2, R_3, R_4 \ldots \mid \textbf{shape}) = \frac{1}{Z} \prod_{f=1..n} P(R_f \mid \textbf{views}) \prod_{i,j} P(R_f, R_{f'} \mid \textbf{surface})$$

Unary factor (convnet)

Projective convnet architecture: CRF layer

Pairwise terms **favor same label** for triangles or points with **similar surface normals** and **small geodesic distance**

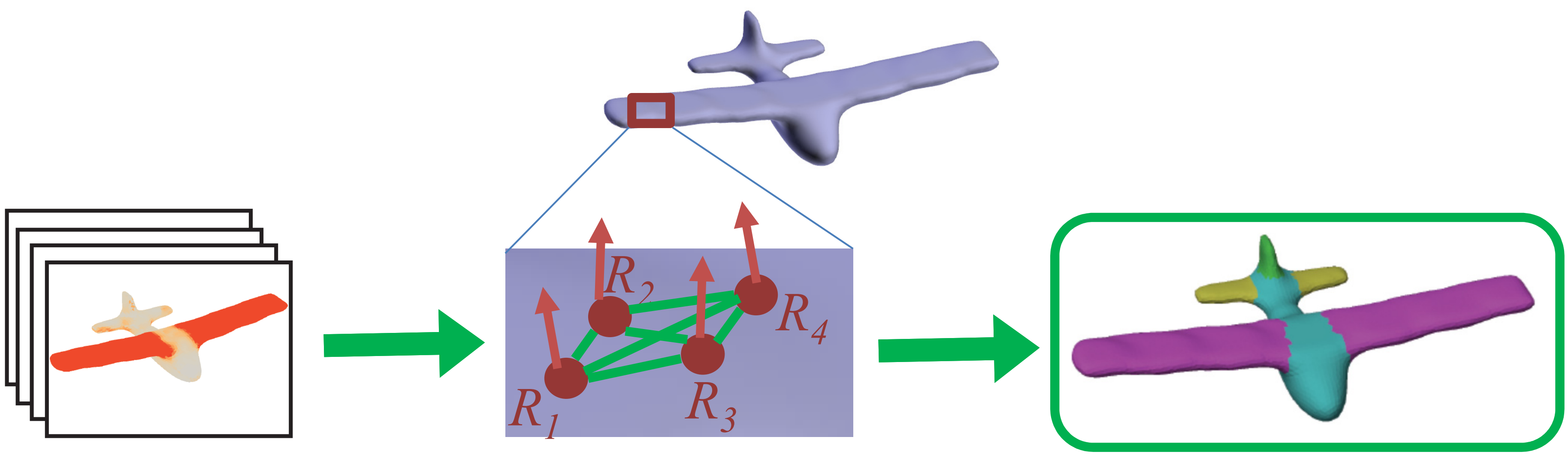


$$P(R_1, R_2, R_3, R_4 \ldots \mid \textbf{shape}) = \frac{1}{Z} \prod_{f=1..n} P(R_f \mid \textbf{views}) \prod_{i,j} P(R_f, R_{f'} \mid \textbf{surface})$$

Pairwise factor (geodesic+normal dist.)

Projective convnet architecture: CRF layer

Inference aims to find the **most likely joint assignment** to all surface random variables (optimization problem)



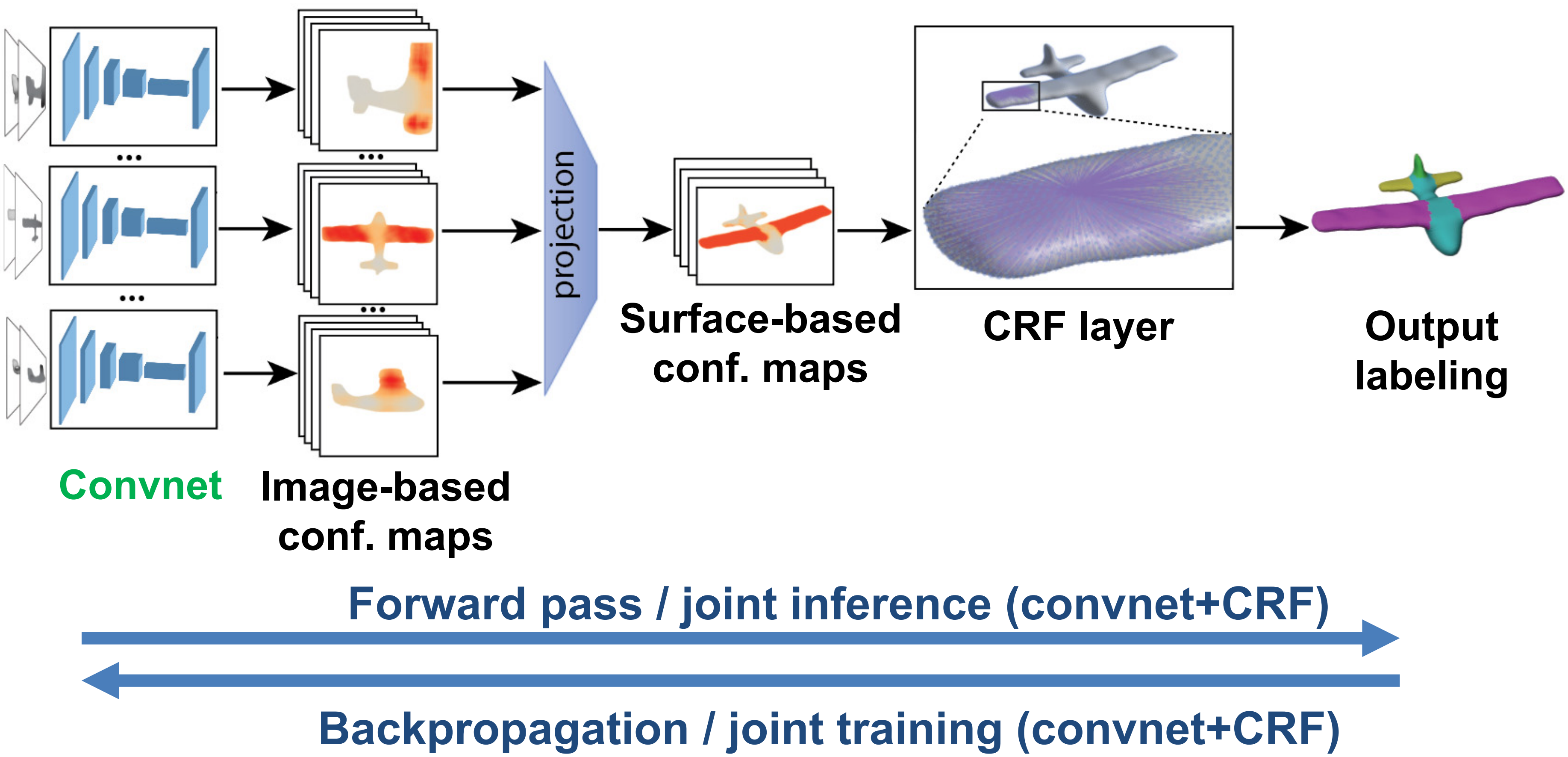
$$\max P(R_1, R_2, R_3, R_4 \ldots \mid \textbf{shape}) = \frac{1}{Z} \prod_{f=1..n} P(R_f \mid \textbf{views}) \prod_{i,j} P(R_f, R_{f'} \mid \textbf{surface})$$

MAP assignment (mean–field inference)

Training

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

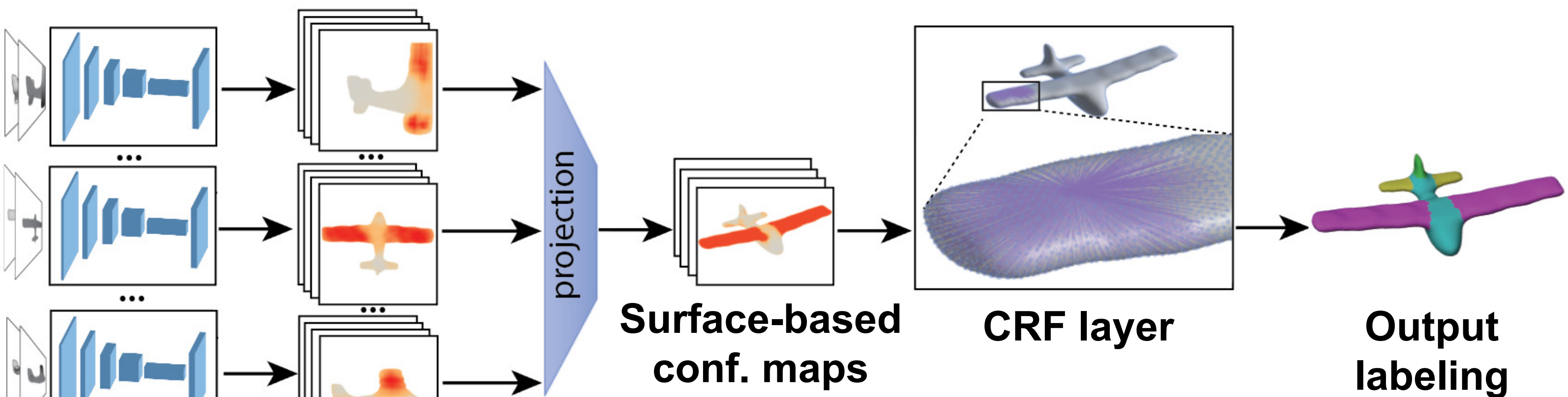
Training starts from a **pretrained image-based net** (VGG16)



Training

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

Training starts from a **pretrained image-based net** (VGG16)



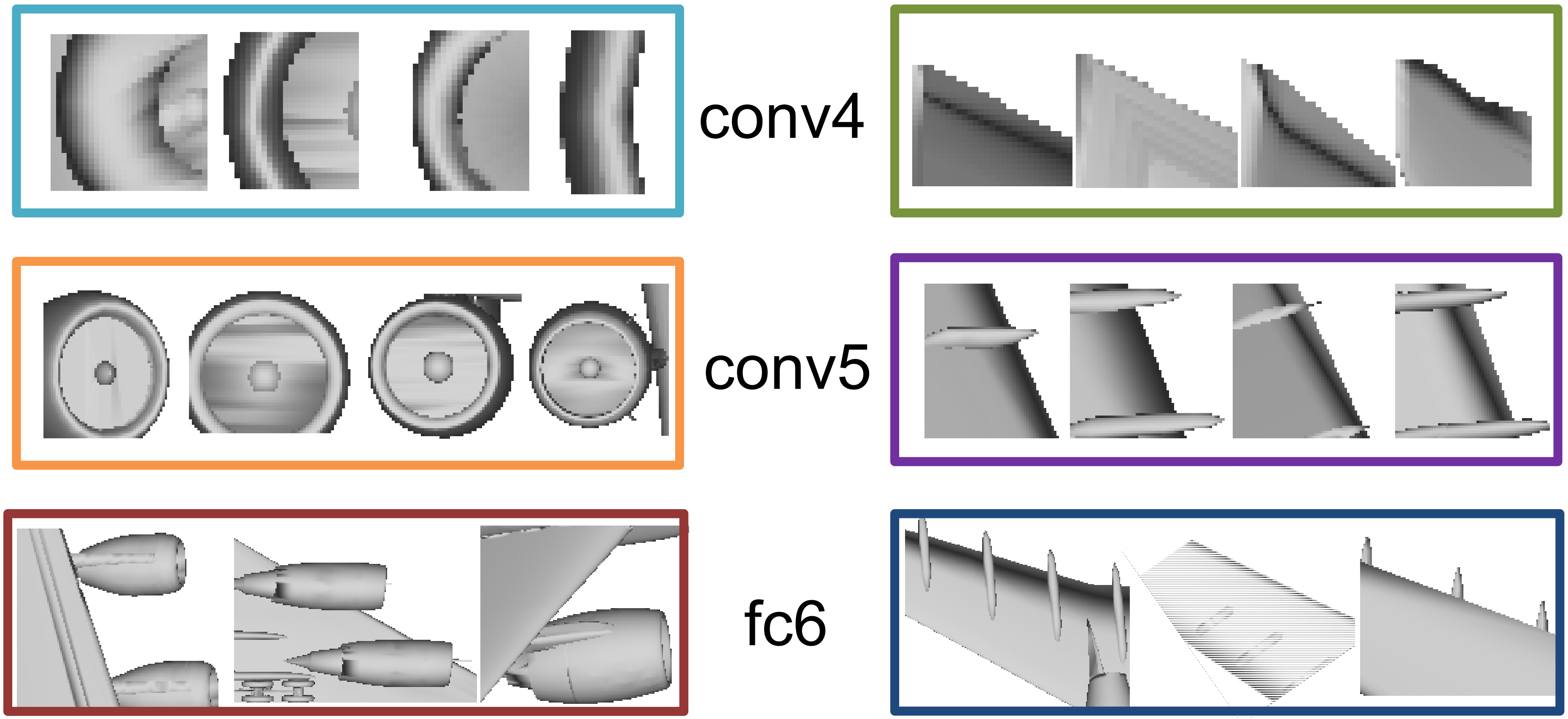
$$L = \frac{1}{|S|} \sum_{s \in S} \log P(\mathbf{R}_s = \mathbf{T}_s) + \lambda ||\boldsymbol{\theta}||^2$$

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

Backpropagation / joint training (convnet+CRF)

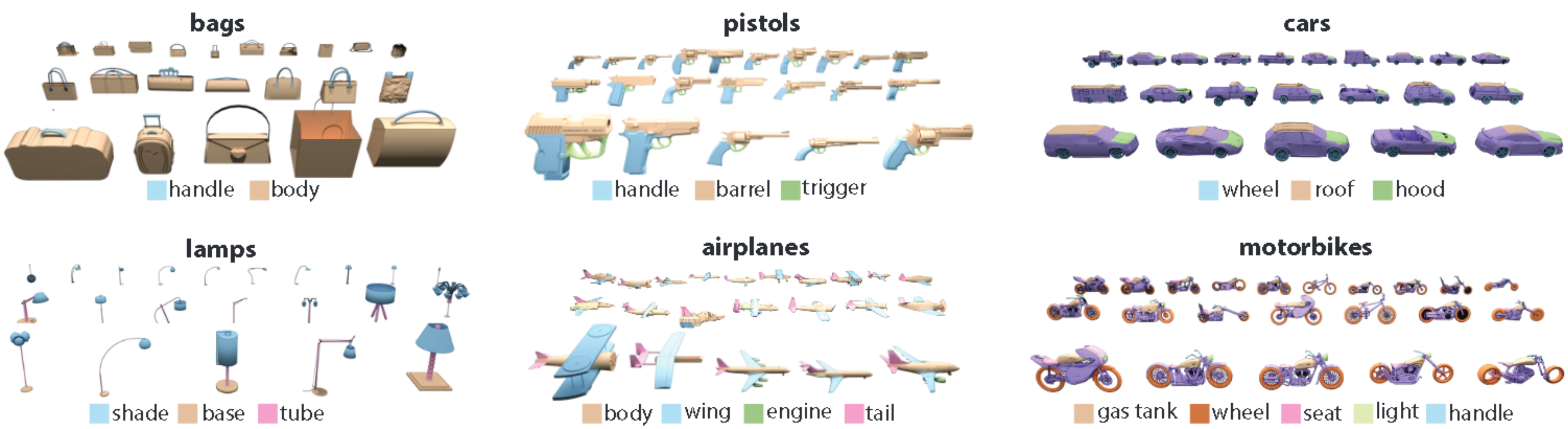
What are the learned filters doing?

Activated in the presence of certain surface patterns / patches



Dataset used in experiments

Evaluation on ShapeNetCore (human labeled shapes).
50% used for training / **50%** used for test split **per category**.



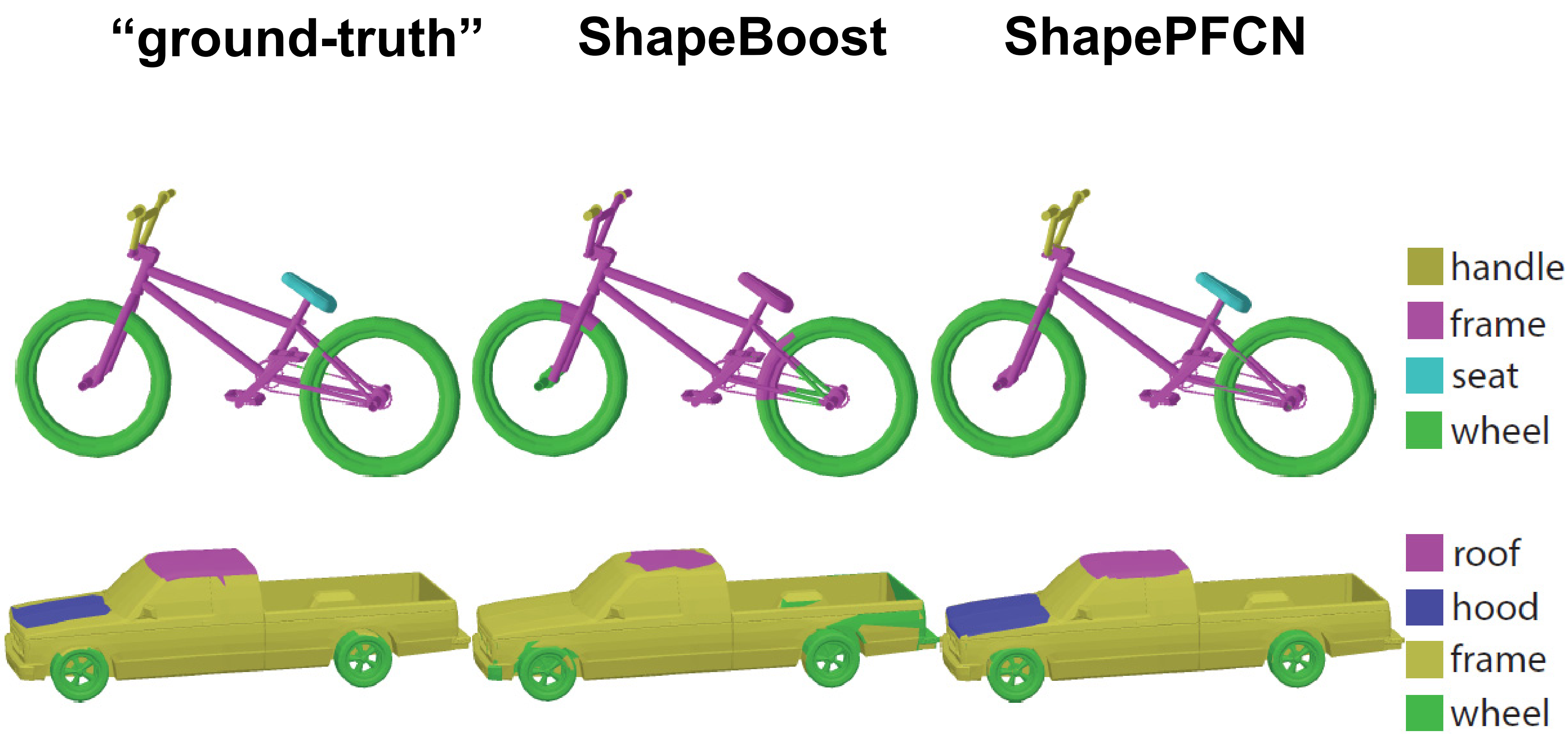
[Yi et al. 2016]

ShapeNetCore: **8% improvement in labeling accuracy** for complex categories (vehicles, furniture etc)

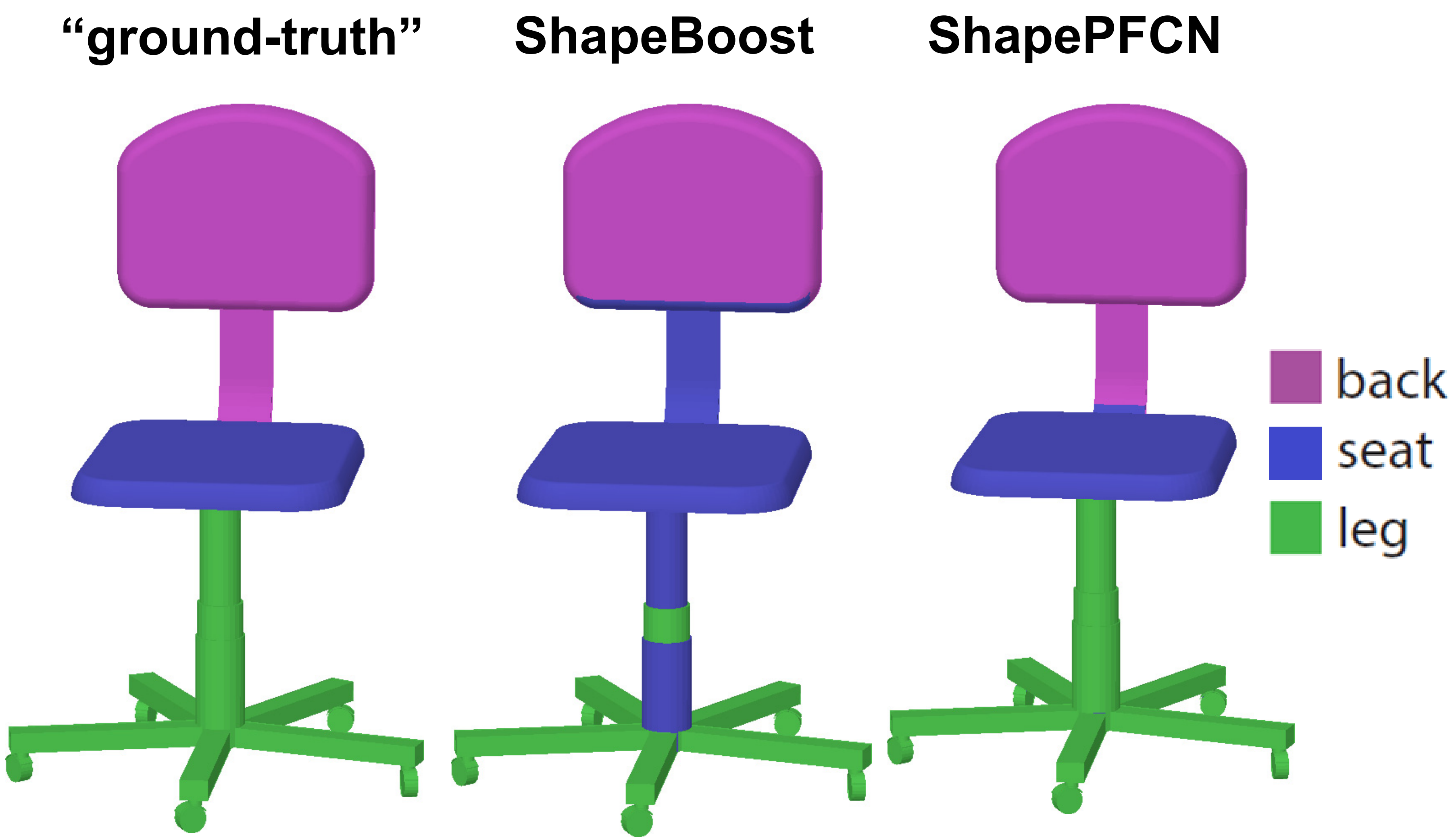
	#train/test shapes	#part labels	ShapeBoost	Guo et al.	ShapePFCN
Airplane	250 / 250	4	85.8	87.4	**90.3**
Bag	38 / 38	2	93.1	91.0	**94.6**
Cap	27 / 28	2	85.9	85.7	**94.5**
Car	250 / 250	4	79.5	80.1	**86.7**
Chair	250 / 250	4	70.1	66.8	**82.9**
Earphone	34 / 35	3	81.4	79.8	**84.9**
Guitar	250 / 250	3	89.0	89.9	**91.8**
Knife	196 / 196	2	81.2	77.1	**82.8**
Lamp	250 / 250	4	71.7	71.6	**78.0**
Laptop	222 / 223	2	86.1	82.7	**95.3**
Motorbike	101 / 101	6	77.2	80.1	**87.0**
Mug	92 / 92	2	94.9	95.1	**96.0**
Pistol	137 / 138	3	88.2	84.1	**91.5**
Rocket	33 / 33	3	79.2	76.9	**81.6**
Skateboard	76 / 76	3	91.0	89.6	**91.9**
Table	250 / 250	3	74.5	77.8	**84.8**

ShapeNetCore: **8% improvement in labeling accuracy** for complex categories (vehicles, furniture etc)

	#train/test shapes	#part labels	ShapeBoost	Guo et al.	ShapePFCN
Airplane	250 / 250	4	85.8	87.4	**90.3**
Bag	38 / 38	2	93.1	91.0	**94.6**
Cap	27 / 28	2	85.9	85.7	**94.5**
Car	250 / 250	4	79.5	80.1	**86.7**
Chair	250 / 250	4	70.1	66.8	**82.9**
Earphone	34 / 35	3	81.4	79.8	**84.9**
Guitar	250 / 250	3	89.0	89.9	**91.8**
Knife	196 / 196	2	81.2	77.1	**82.8**
Lamp	250 / 250	4	71.7	71.6	**78.0**
Laptop	222 / 223	2	86.1	82.7	**95.3**
Motorbike	101 / 101	6	77.2	80.1	**87.0**
Mug	92 / 92	2	94.9	95.1	**96.0**
Pistol	137 / 138	3	88.2	84.1	**91.5**
Rocket	33 / 33	3	79.2	76.9	**81.6**
Skateboard	76 / 76	3	91.0	89.6	**91.9**
Table	250 / 250	3	74.5	77.8	**84.8**



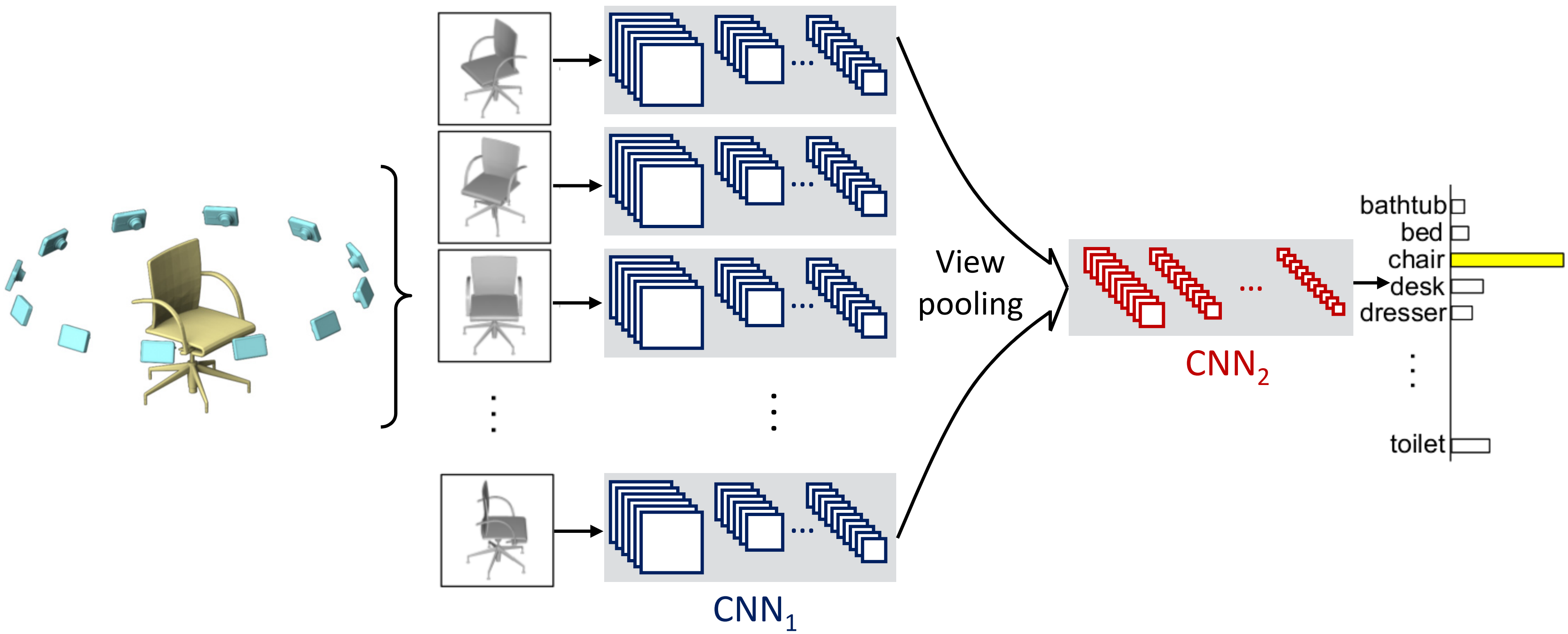
“ground-truth”
ShapeBoost
ShapePFCN
handle
frame
seat
wheel
roof
hood
frame
wheel



“ground-truth”
ShapeBoost
ShapePFCN
back
seat
leg

Shape recognition with multi-view CNNs

An earlier version of a view-based CNN for shape recognition



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

Summary

- Inspired 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**
- **Fast** processing at **high-resolutions**
- **Robust** to input geometric representation artifacts (e.g., irregular tessellation, polygon soups, etc)
- Initialized from image-based architectures **pretrained on massive image datasets** (filters capture shape+texture)

Thank you!

Acknowledgements: NSF (CHS-1422441, CHS-1617333, IIS- 1617917), NVidia, Adobe, Facebook, Qualcomm.

Experiments were performed in the **UMass GPU cluster (400 GPUs!)** obtained under a grant by the MassTech Collaborative.

Our project web page:

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