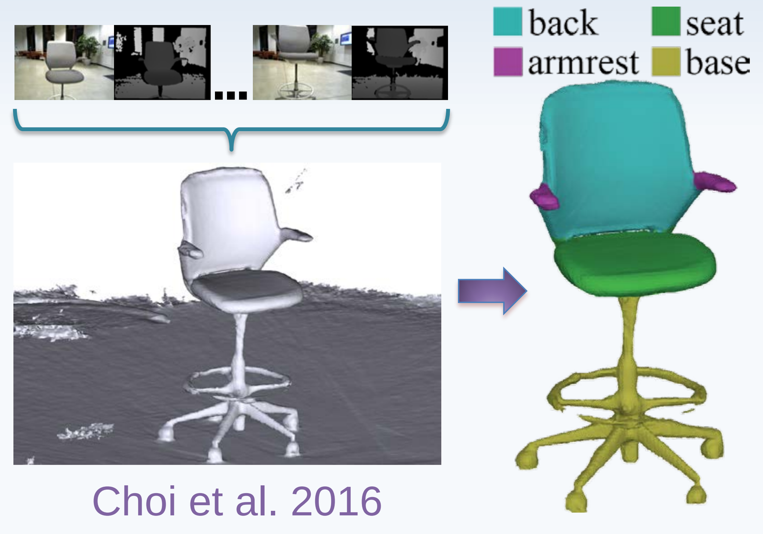
3D Shape Segmentation with Projective Convolutional Networks

Evangelos Kalogerakis — UMass Amherst
Melinos Averkiou — University of Cyprus
Subhransu Maji — UMass Amherst
Siddhartha Chaudhuri — IIT Bombay

Overview

Motivation: recognizing parts in 3D shapes is fundamental to several applications in 3D computer vision, computer graphics, and robotics

3D Modeling and Animation



Parsing RGBD data

Challenges: subtlety in 3D geometric cues, arbitrary orientation, noise, varying resolution, arbitrary or no interior, missing texture, non-manifold geometry, shape part variability, need to parse local and global context

Earlier work: "hand-engineered" geometric descriptors, heuristic processing stages, low resolution, lack of generality & robustness

Our approach: combine fully convolutional net (**FCN**) operating on rendered shape views with surface-based graphical model (**CRF**)

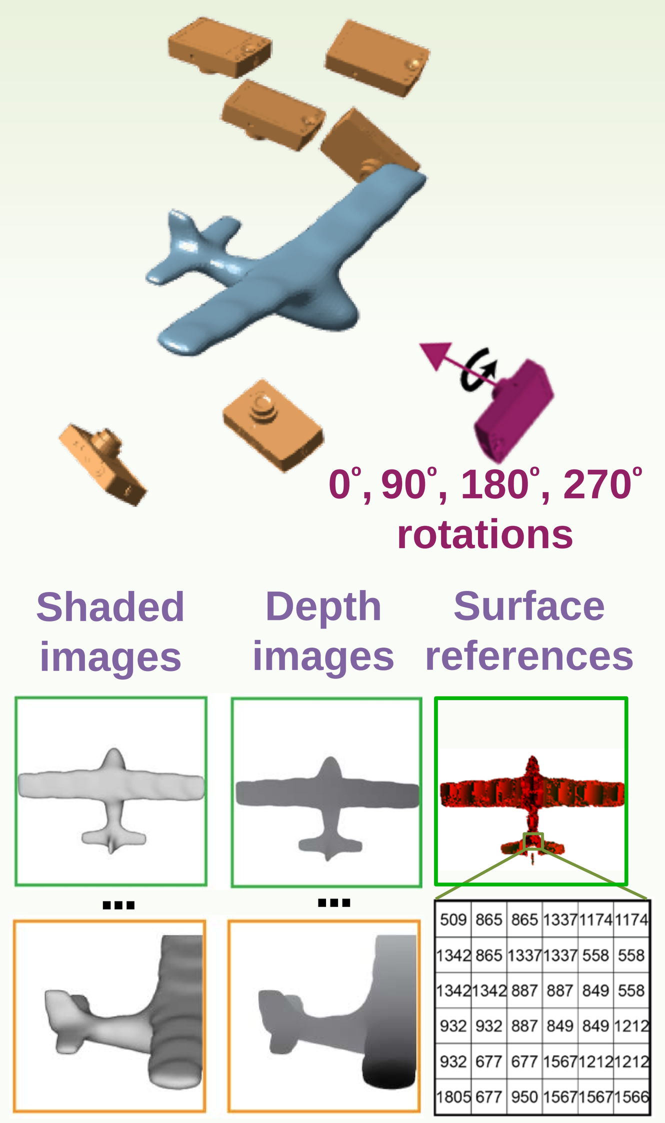
Key ideas:

- **Adaptive view selection** per shape to maximally cover its surface
- **Multi-scale representation** of the surface information
- Initialize network from **pre-trained** image-based architectures
- **End-to-end training** of the whole network (FCN & CRF)
- **Projective layer** for mapping view representations to surfaces

Key advantages:

- **High-resolution** shape analysis
- **Robustness** to geometric representation artifacts (noise, irregular tessellation, arbitrary interior, non-manifold geometry)
- Transfer learning from **massive image datasets**
- **Rotational invariance**
- CNN representation power is focused on the **shape surface**

Method

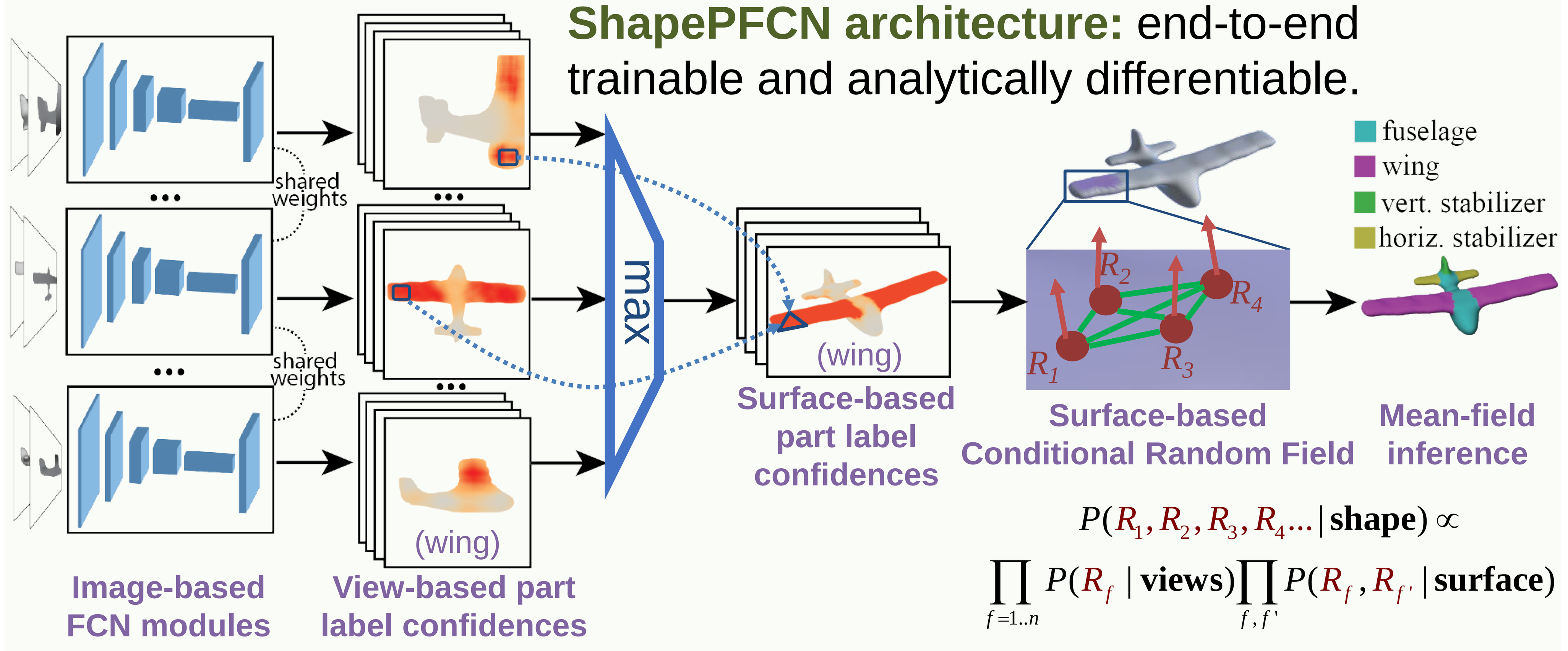


Rendering stage: infer set of viewpoints that **maximally covers the surface** of the input shape across multiple scales.
To favor rotational invariance, perform in-plane camera rotations.
Views are not ordered, number of viewpoints differ per shape, and no view correspondences across shapes are assumed.

Encode surface position & normals: render **shaded images** (normal dot view vector) and **depth images** relative to the cameras.

Render **surface reference images**: each pixel stores a pointer to a surface element.

ShapePFCN architecture: end-to-end trainable and analytically differentiable.



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

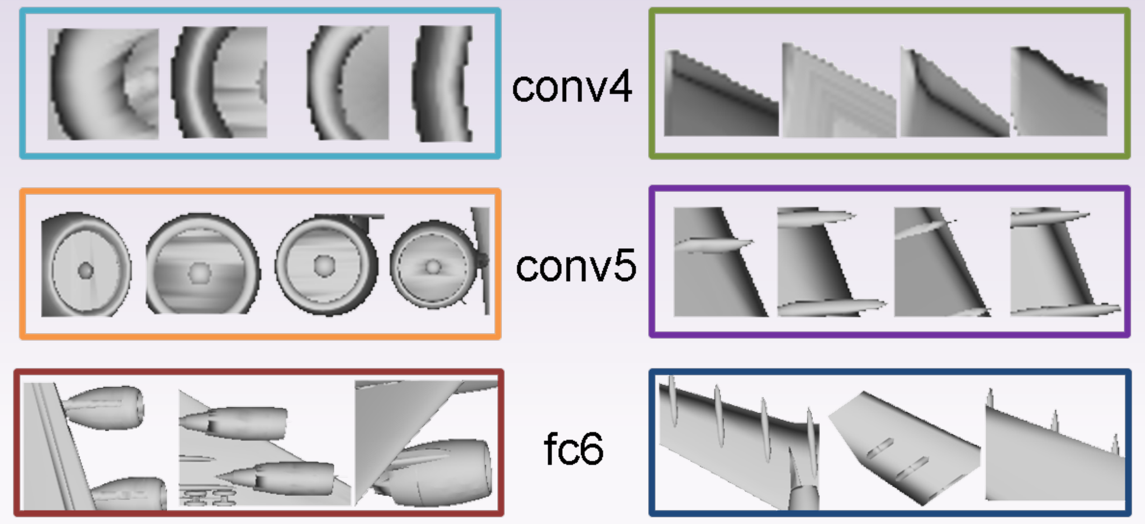
The pairs of shaded and depth images are passed into **FCN branches with shared filters.** Their outputs are image-based confidences per label.

The image-based label confidences are **aggregated on the surface** via the surface references & a projection layer.

Our **surface CRF** uses the surface confidences as unary terms. Pairwise terms use geodesic distances & normals for **coherent labeling**.

Results

Top filter activations: after training, filters are sensitive to different local surface patterns (triangular, circular patches etc). In upper layers, different filters are sensitive to various shape sub-parts and parts.

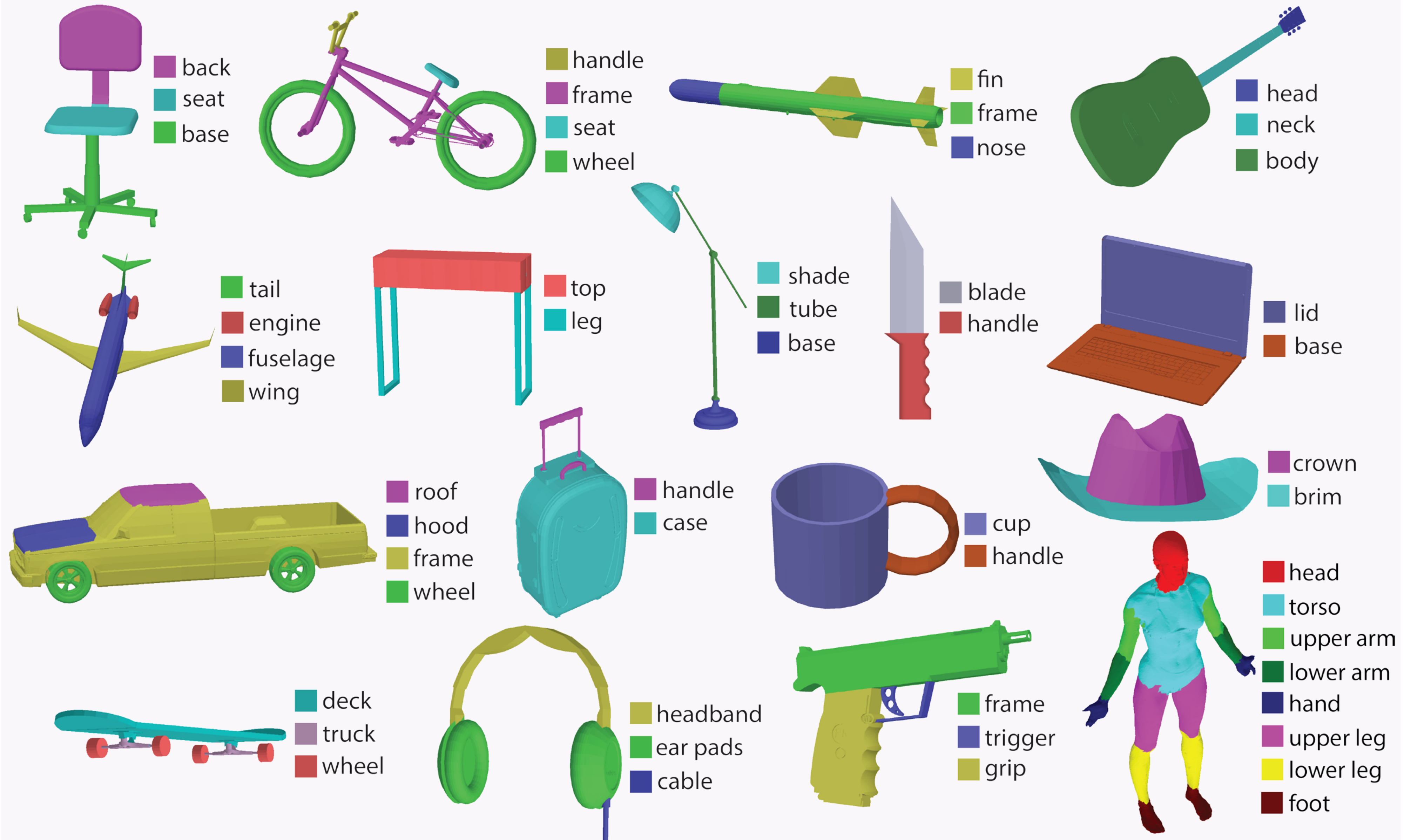


Experiments: 3D ShapeNet (16 classes), L-PSB & COSEG (30 classes)

	ShapeBoost	Guo et al.	ShapePFCN
Category Avg.	83.0	82.2	**88.4**
Category Avg. (>3 labels)	76.9	77.2	**85.0**
Dataset Avg.	81.2	80.6	**87.5**
Dataset Avg. (>3 labels)	76.8	76.8	**84.7**

Average labeling accuracy on segmented ShapeNetCore

note: per category training, 50% training / 50% testing, max 500 shapes per class, no assumption on shape orientation



Project page with datasets, results and source code:
http://people.cs.umass.edu/~kalo/papers/shapepfcn/index.html