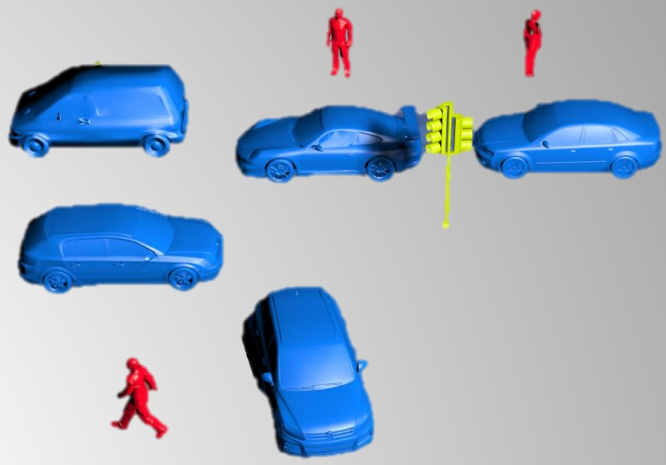


3D-RCNN: Instance-level 3D Object Reconstruction via Render-and-Compare

Abhijit Kundu
Google Research

Yin Li
CMU

James M. Rehg

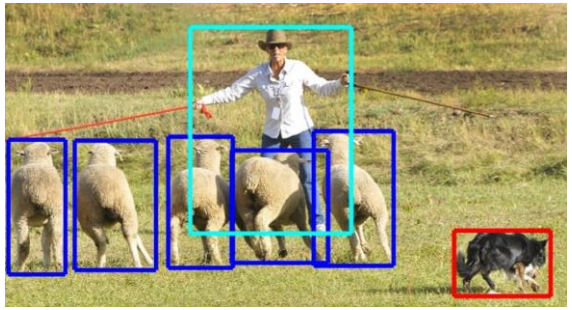
Georgia Institute of Technology

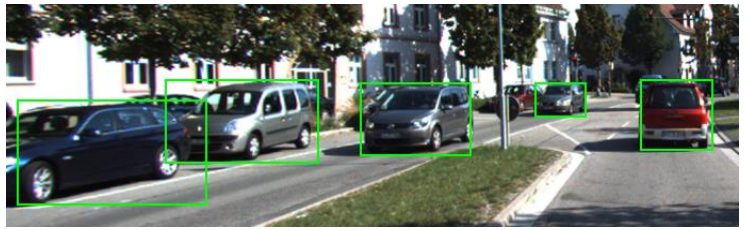
“Perception is our best guess as to **what is in the world** given our current sensory input and our prior experience”

- Helmholtz 1866

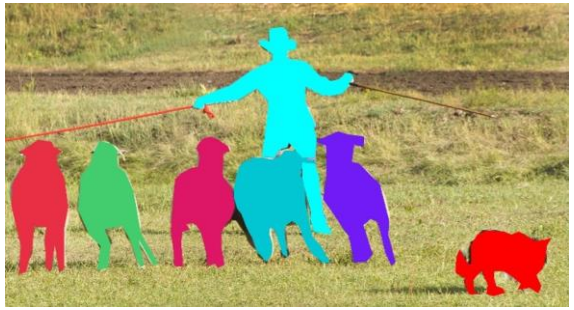
Most scene representations are 2D

Object detection boxes

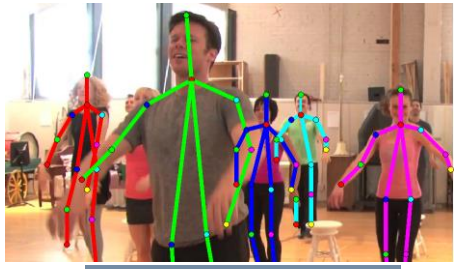




Instance segmentation masks

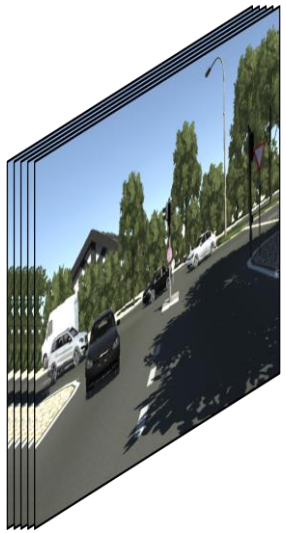


Person body keypoints & parts

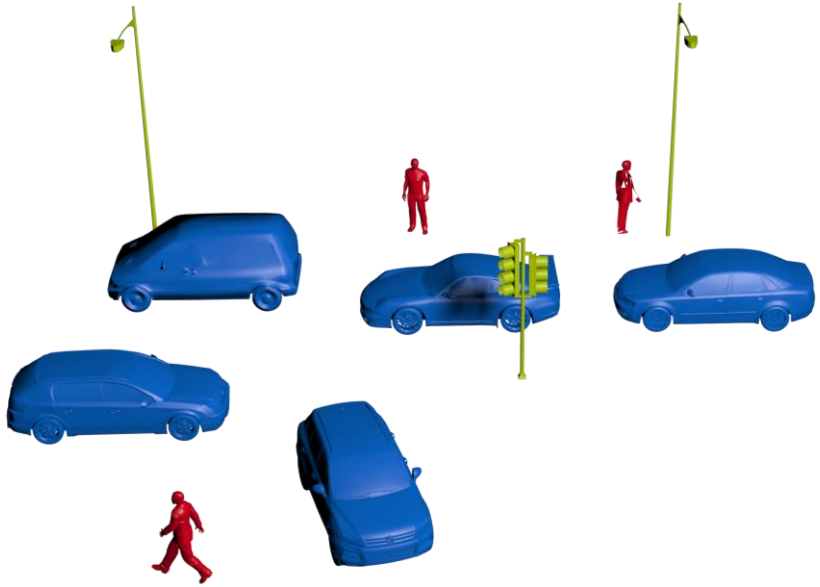


Our goal: 3D representation for object

Problem Statement: Given an image, estimate 3D **shape** and 3D **pose** of all object instances



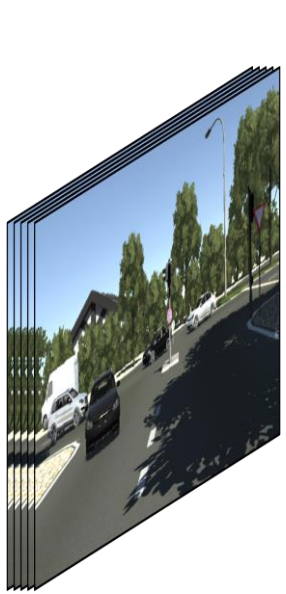
Image



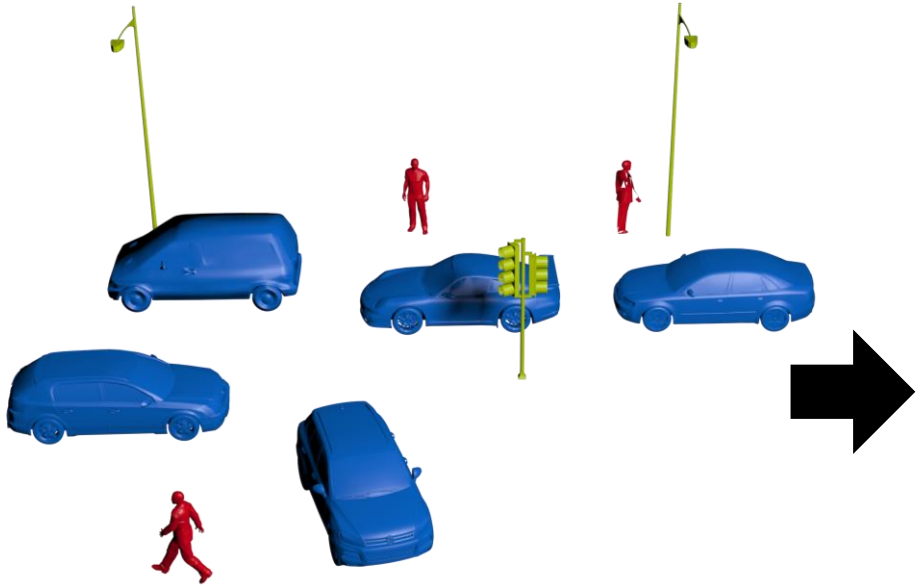
3D object instance reconstructions

Goal: 3D representation for objects

Problem Statement: Given an image, estimate 3D **shape** and 3D **pose** of all object instances

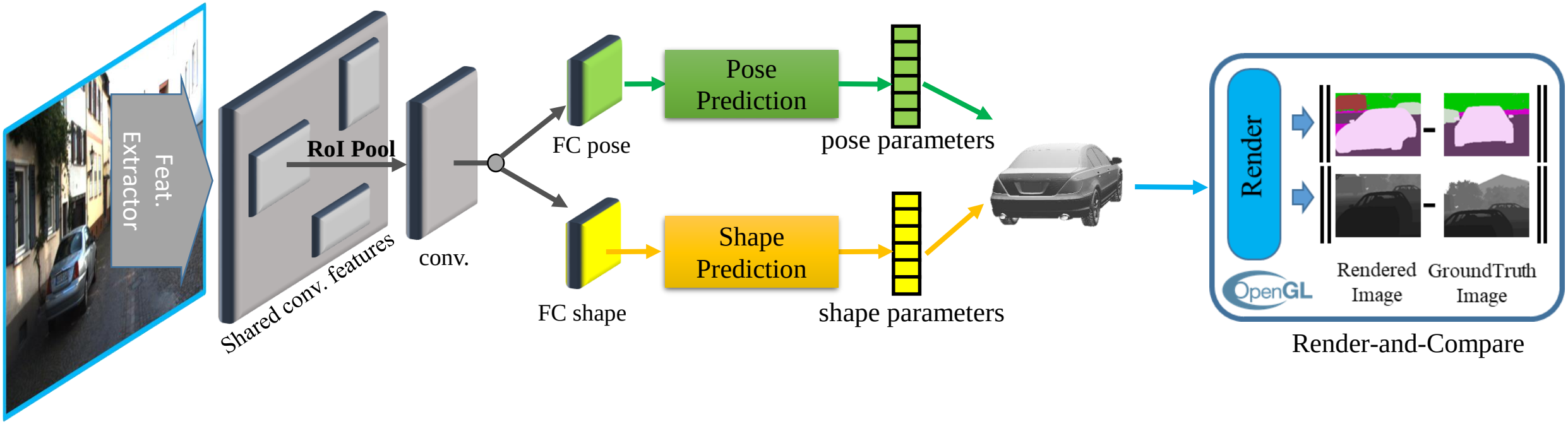


Image



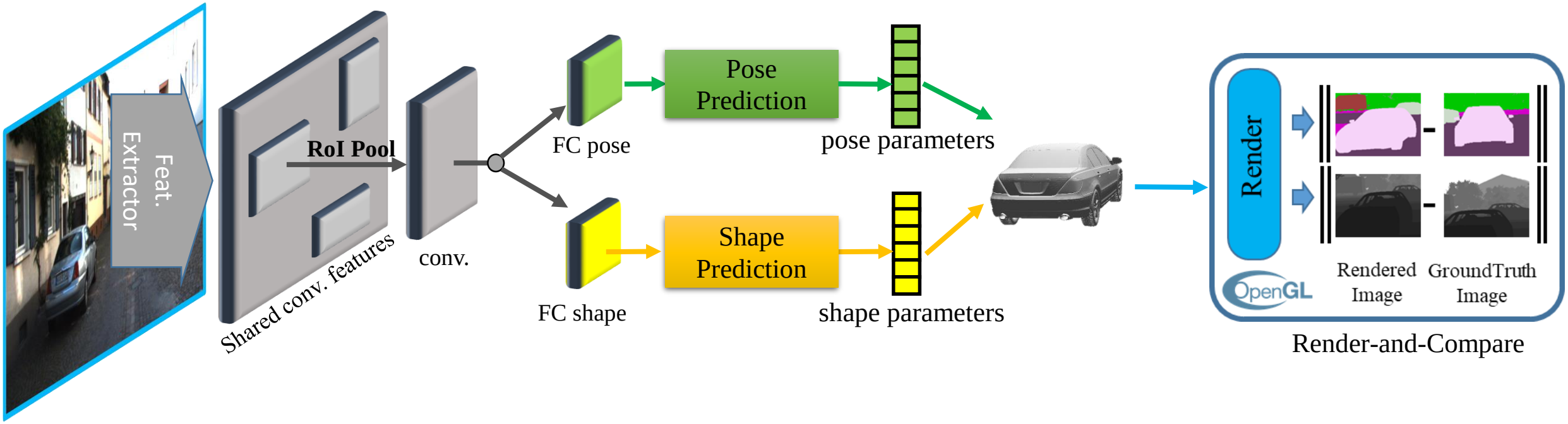
3D object instance reconstructions

3D-RCNN: Fast Inverse-Graphics Framework



- Faster-RCNN meta-architecture
- Shape and pose predictions are made on top of RoI-Pooled features

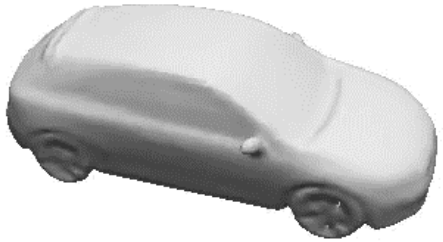
3D-RCNN: Key Questions

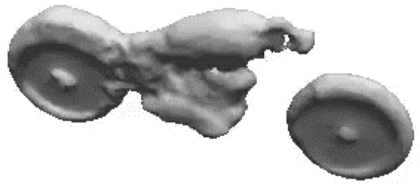


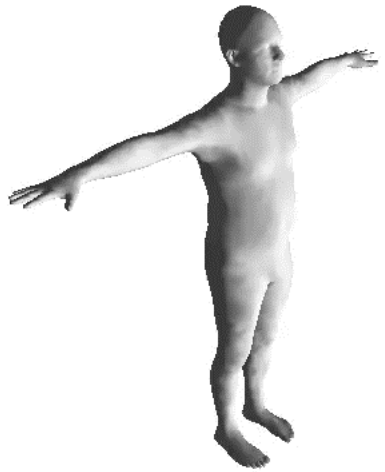
- How do we parameterize shape and pose?
- How to achieve 3D equivariance?
- How to train without 3D supervision?

Shape Representation

- Exploit CAD datasets
  - Learn low dimensional parametric **shape space** for each class
  - For **rigid** objects: PCA on volumetric representation (TSDF)
  - For *articulated* objects: **SMPL**



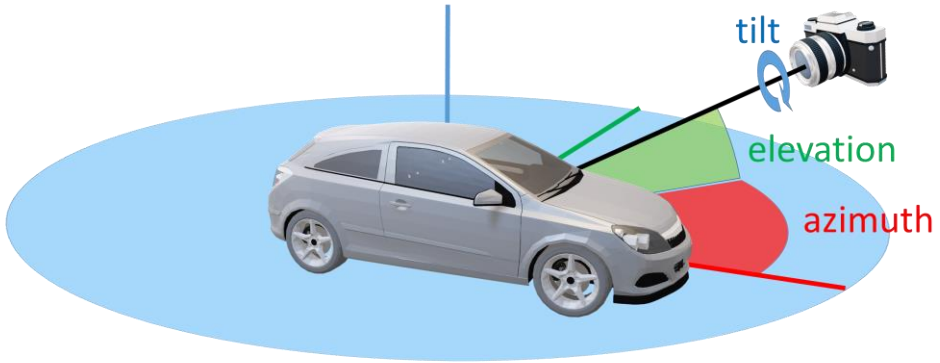




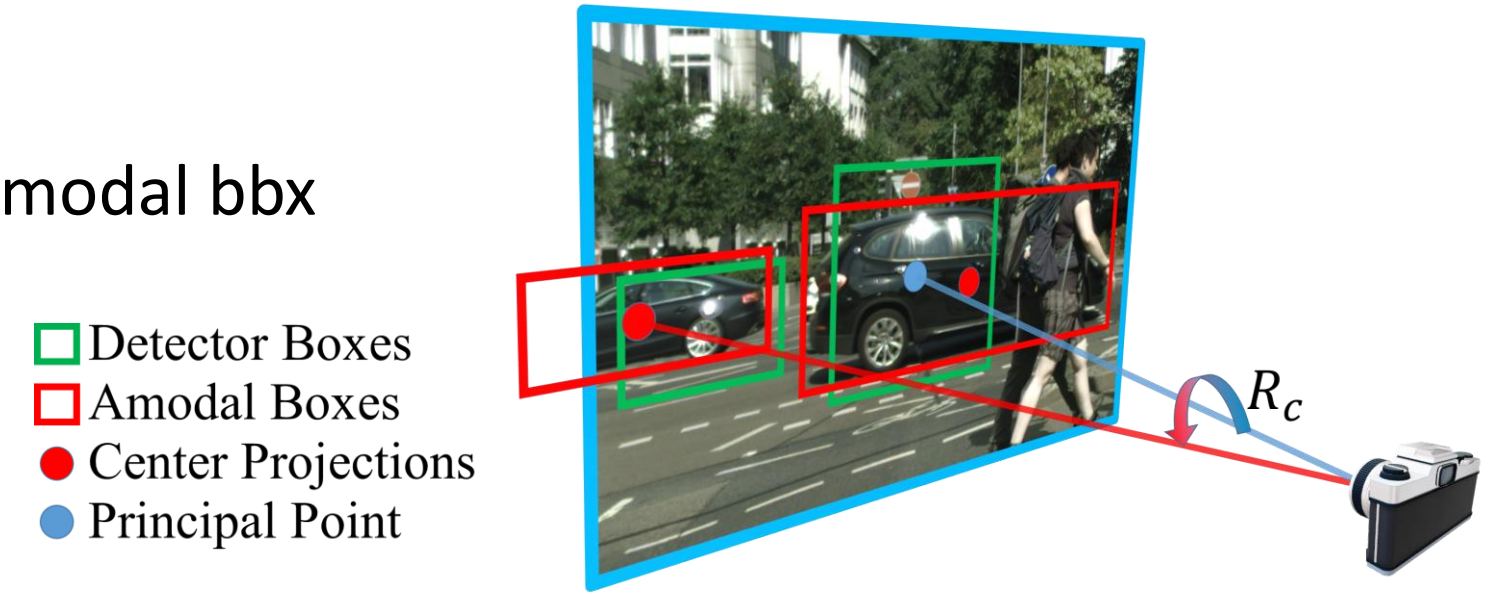
SMPL [Loper et al. **SIGGHRAPH 2015**]

Pose Representation

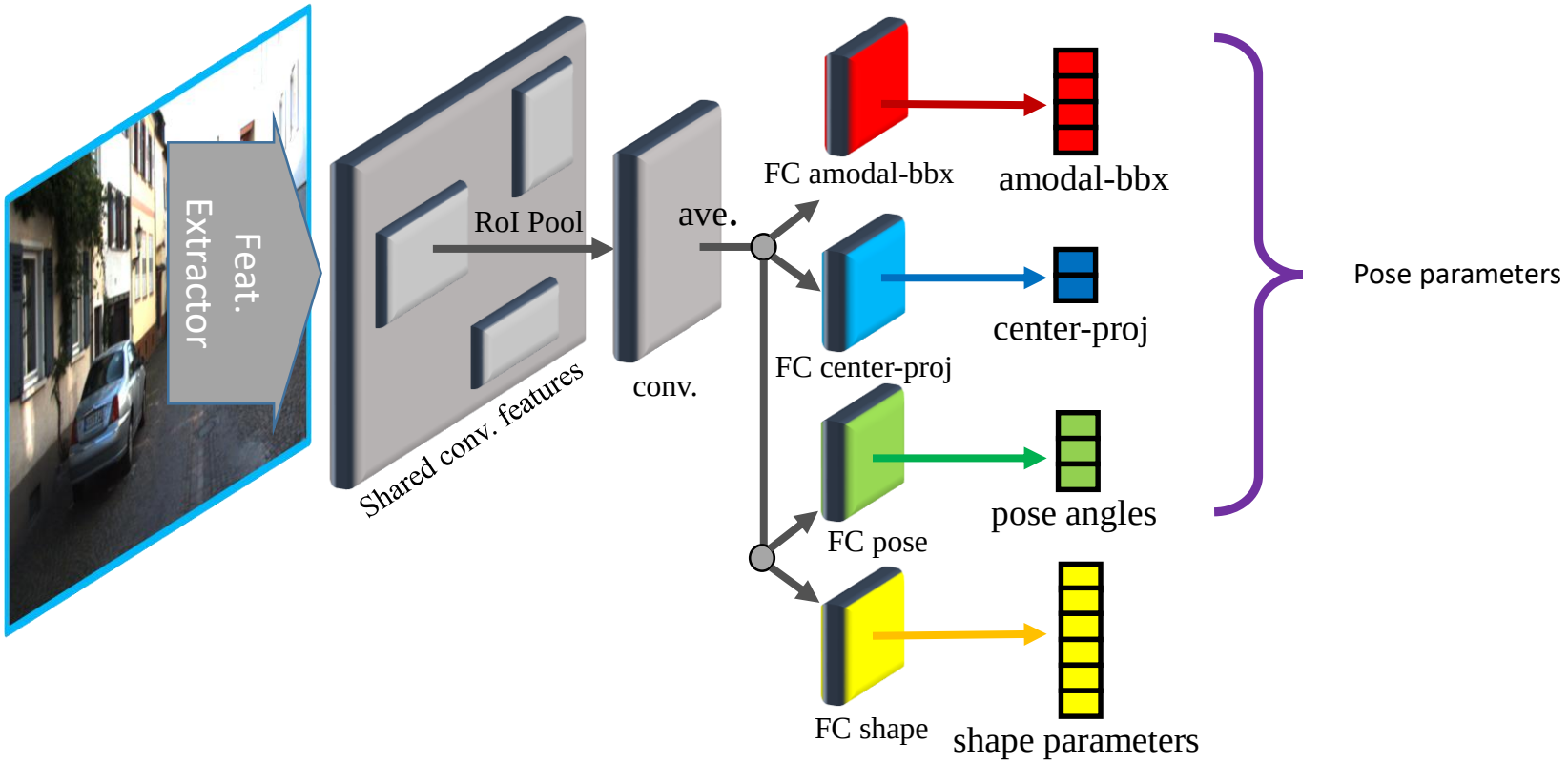
- Pose-angles (**allocentric** viewpoint)
  - Allocentric vs Egocentric



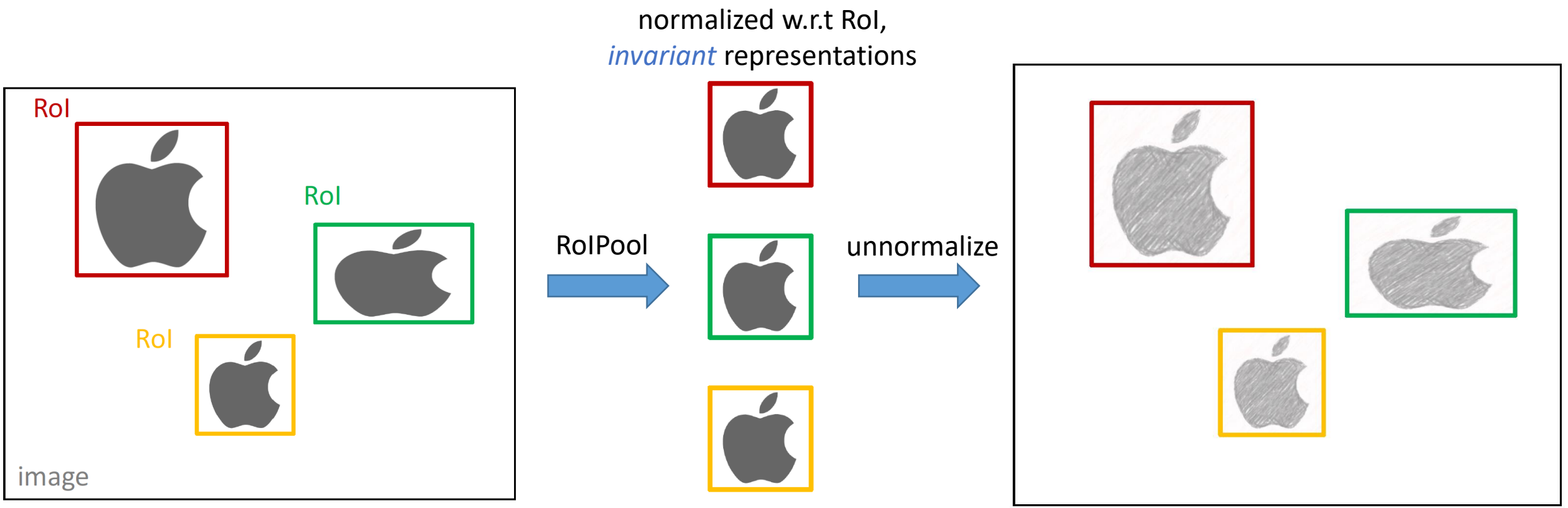
- 2D Center-proj and 2D Amodal bbx



Pose Representation



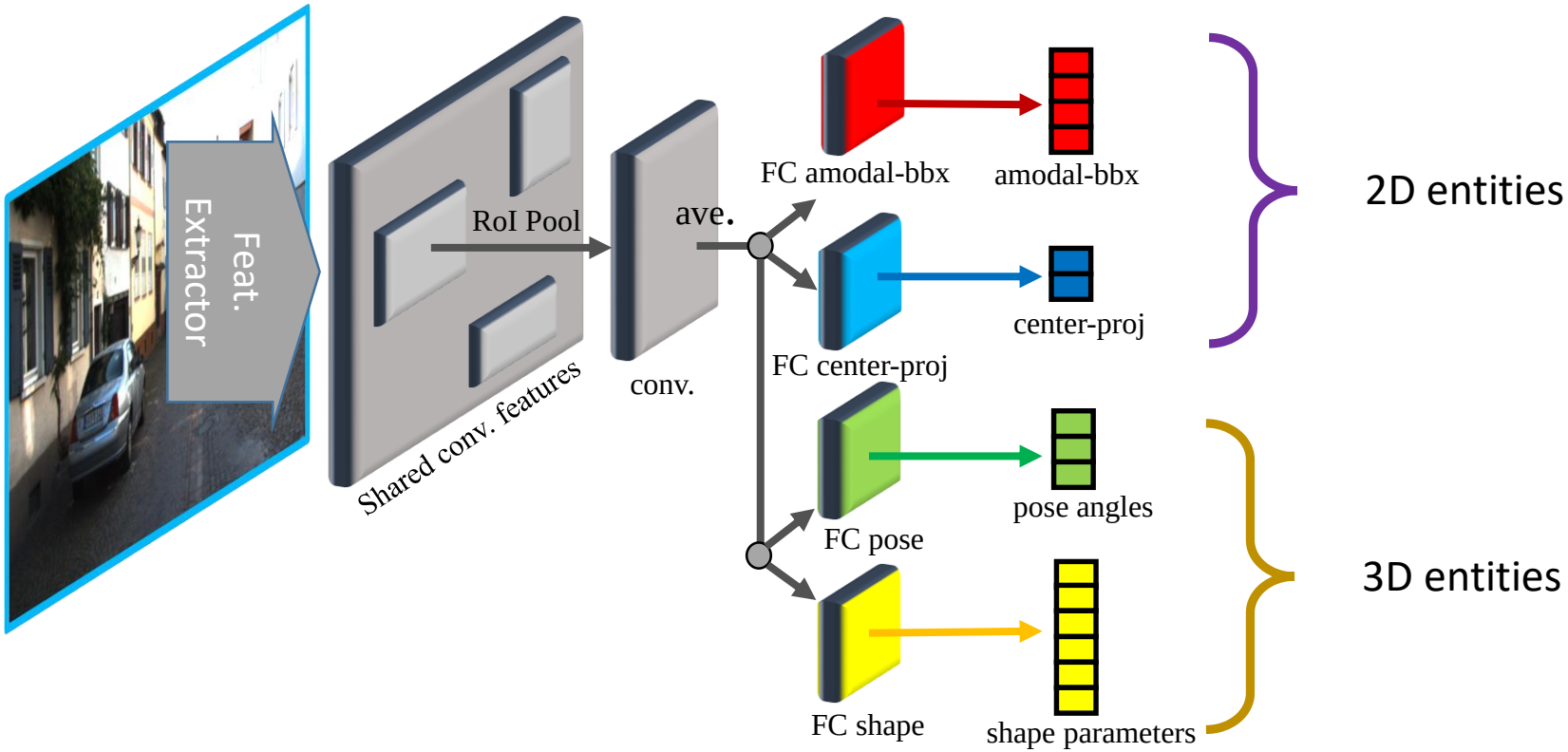
On Equivariance



RoIPool creates *scale* and *aspect-ratio* ***invariant*** representations
RoIPool + "output pasted back" (unnormalize) provides *scale* and *aspect-ratio* ***equivariance***

slide courtesy **Kaiming He**. Mask R-CNN: A Perspective on Equivariance

3D Equivariance: The problem



$$\text{amodal-box } \hat{\boldsymbol{a}} = \left[\frac{x_a - x_r}{w_r}, \frac{y_a - y_r}{h_r}, \log\frac{w_a}{w_r}, \log\frac{h_a}{h_r}\right]$$

$$\text{center-proj } \hat{\boldsymbol{c}} = \left[\frac{x_c - x_r}{w_r}, \frac{y_c - y_r}{h_r}\right]$$

We can normalize/de-normalize the 2D targets w.r.t RoI, but we cannot do this for 3D targets

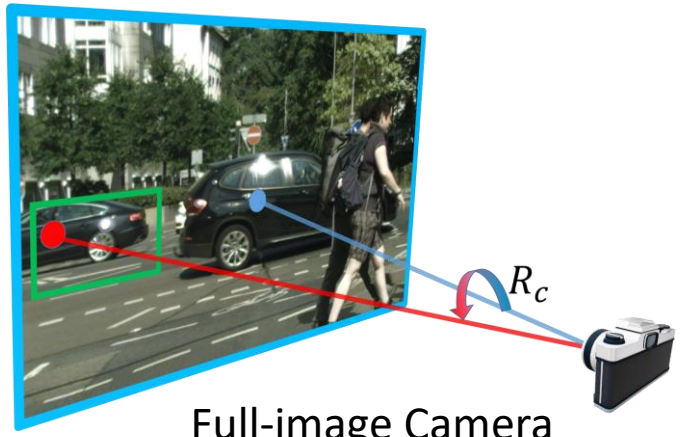
3D Equivariance: The problem

Full image: All three persons have the exact same shape parameters

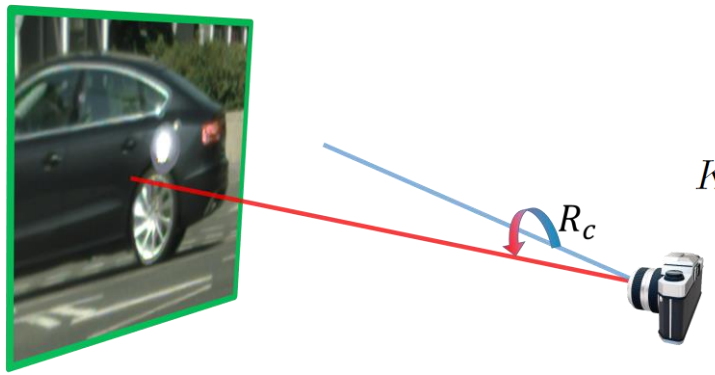
RoI Images

3D Equivariance: RoI Camera

- We interpret the RoI transformation as image formed by a virtual camera (RoI Camera) which is rotated and has different intrinsics than full-image camera



Full-image Camera



RoI Camera

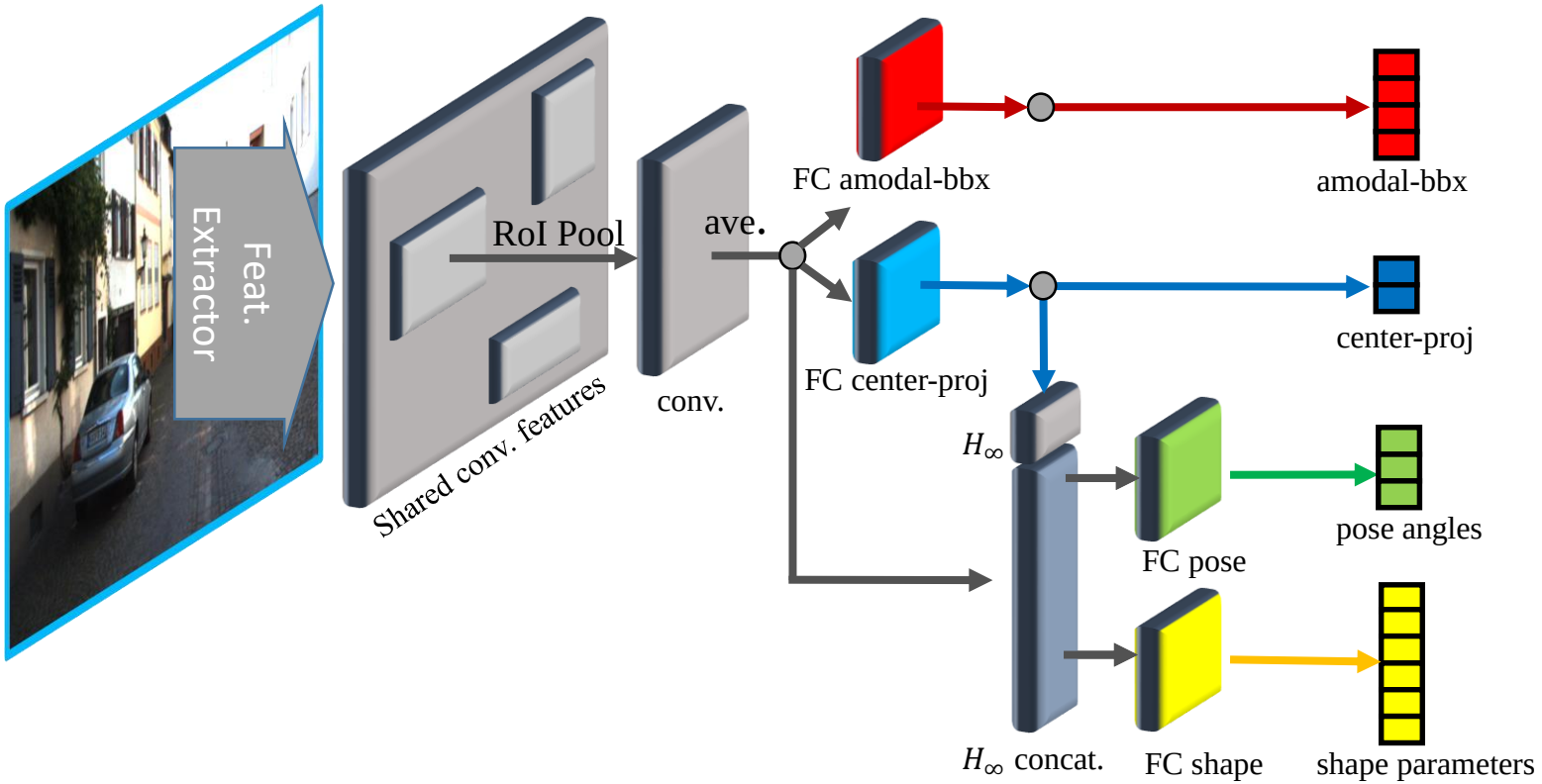
Detector Boxes
Center Projections
Principal Point

$$K_c = \begin{bmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix}, K_r = \begin{bmatrix} f_x f_w / r_w & 0 & f_w/2 \\ 0 & f_y f_h / r_h & f_h/2 \\ 0 & 0 & 1 \end{bmatrix}$$

- Two cameras under pure rotation are related by the ***infinite-homography*** matrix

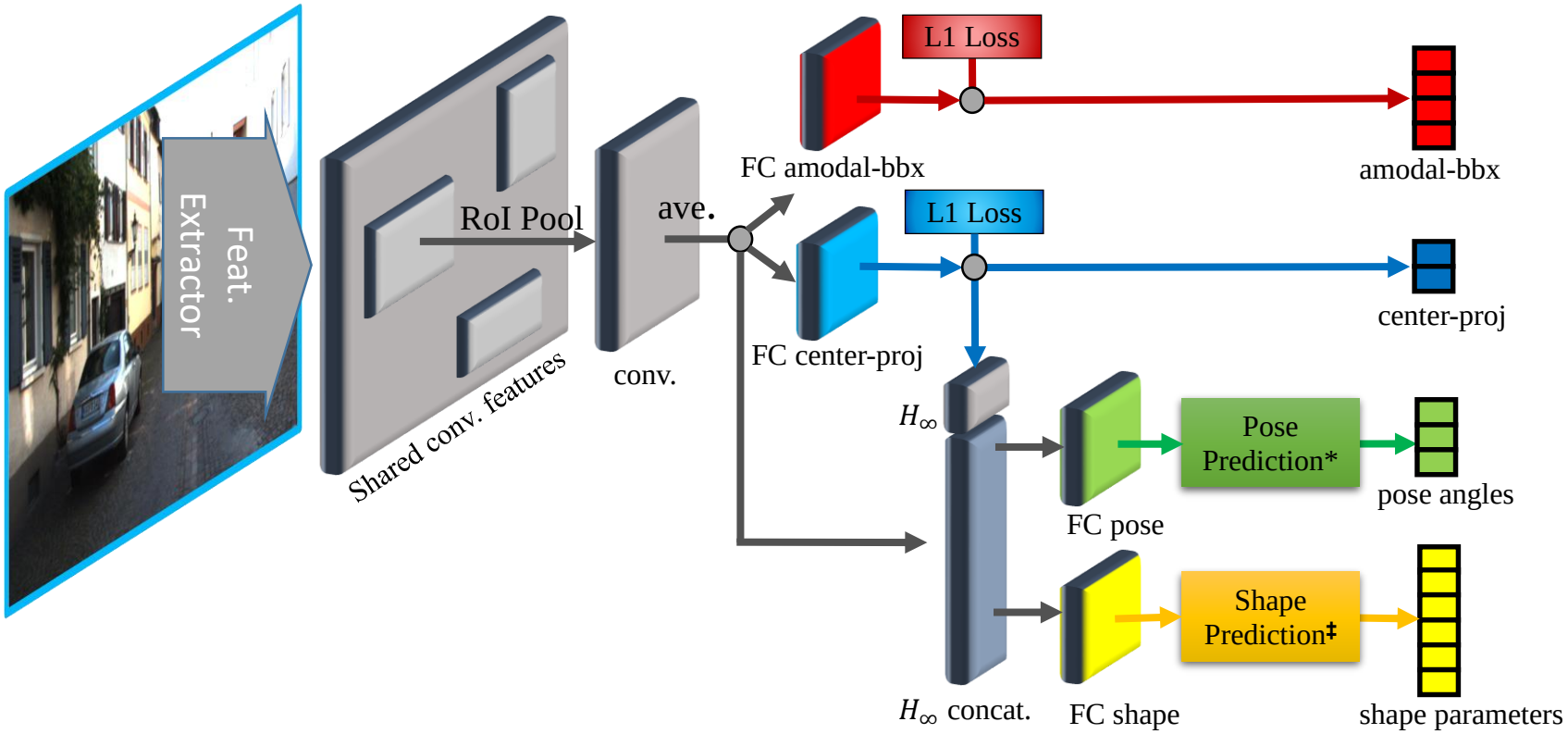
$$H_\infty = K_r R_c^{-1} K_c^{-1}$$

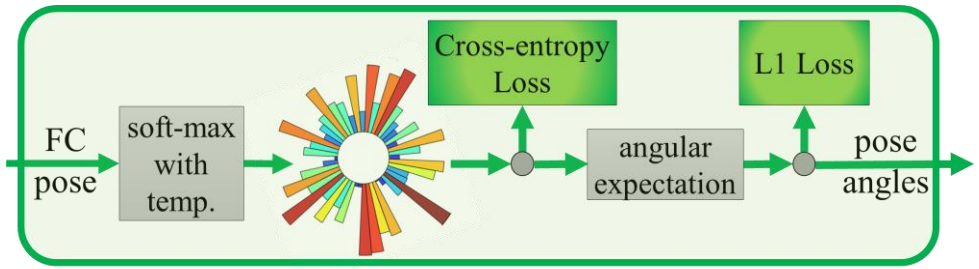
3D Equivariance: Solution



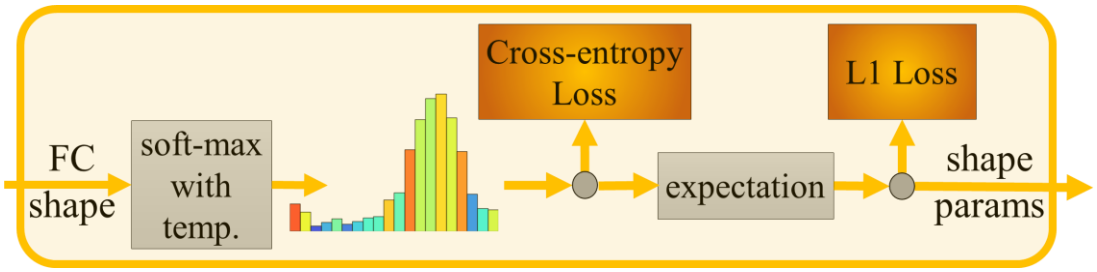
We need to feed the RoI transformation information captured in the ***infinite-homography*** matrix for the 3D shape and 3D pose predictions

3D-RCNN: Training with Direct 3D Supervision



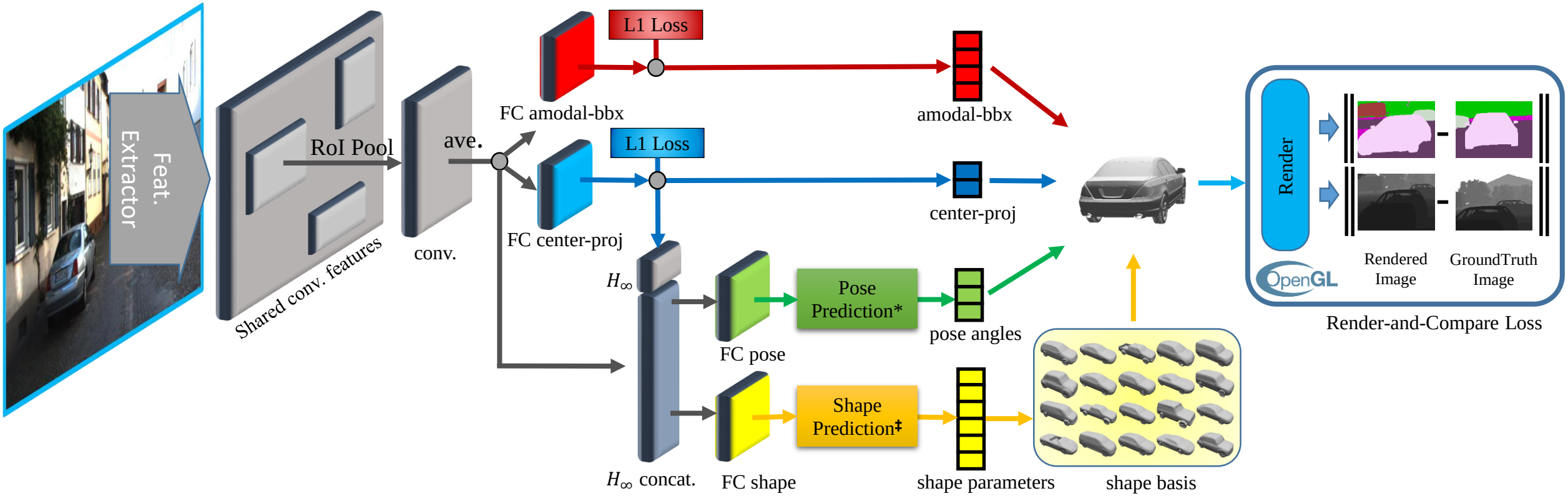


Pose Prediction*



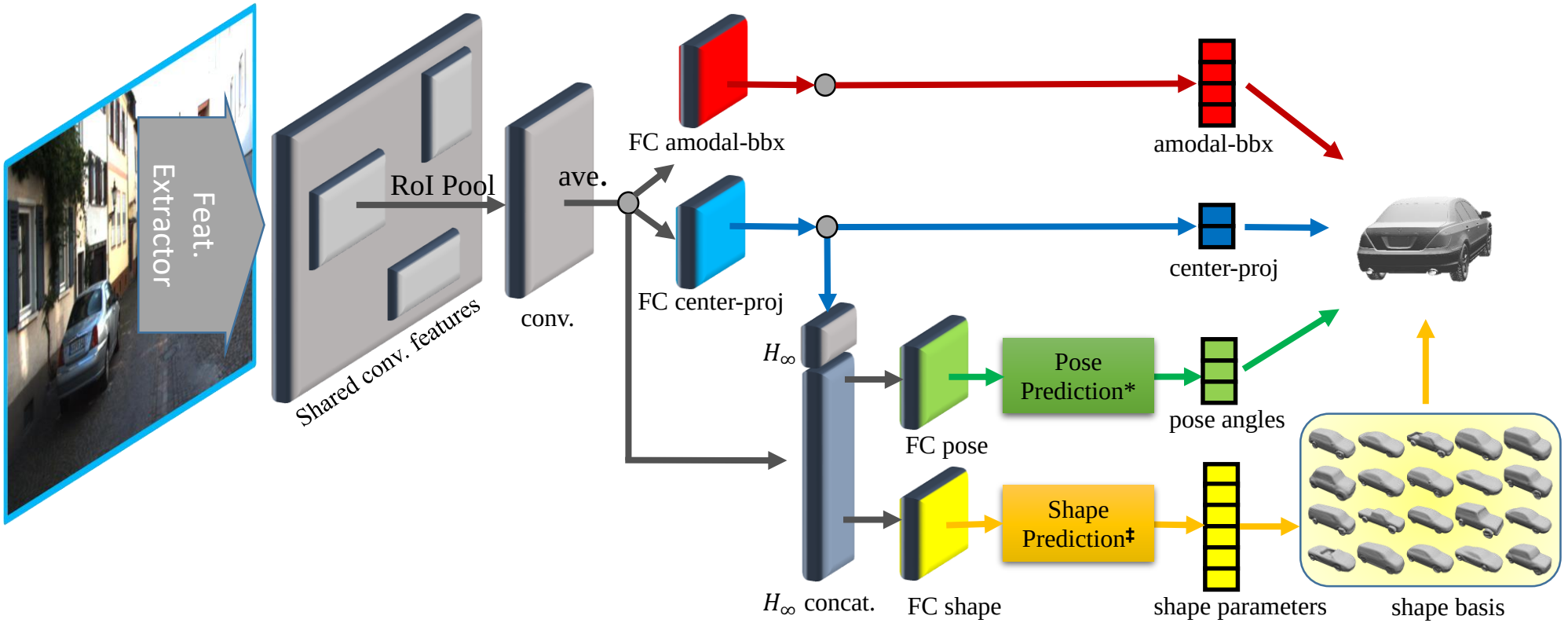
Shape Prediction ‡

3D-RCNN: Training with Render-and-Compare



- We use finite difference for computing derivatives. This is possible because:
  - small number of parameters per instance: 19 (4+2+3+10) for rigid objects, 88 (19 + 69) for person
  - Non-photorealistic rendering is fast
  - The entire shape-decoding, render, and compare happens in GPU (CUDA-GL interop driver)

3D-RCNN: Inference



Fast Inference: Approximately 200ms per image with ResNet50

3D-RCNN: Experiments on Pascal3D+

- Joint detection and 3D Pose estimation

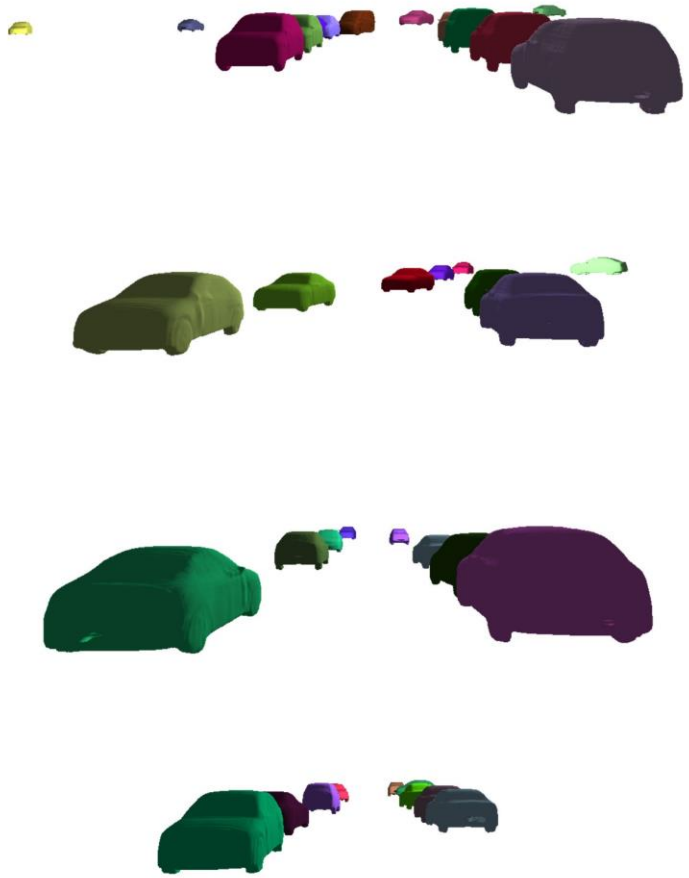
Pascal3D+	Bicycle				Motorcycle				Car			
	AVP_4	AVP_8	AVP_{16}	AVP_{24}	AVP_4	AVP_8	AVP_{16}	AVP_{24}	AVP_4	AVP_8	AVP_{16}	AVP_{24}
Pepik et al.	43.9	40.3	22.9	16.7	31.8	32.0	16.7	10.5	36.9	36.6	29.6	24.6
Tulsiani & Malik	59.4	54.8	42.0	33.4	61.1	59.5	38.8	34.3	55.2	51.5	42.8	40.0
RenderForCNN	50.5	41.1	25.8	22.0	50.8	39.9	31.4	24.4	41.8	36.6	29.7	25.5
Poirson et al.	62.1	56.4	39.6	29.4	62.7	58.6	40.4	30.3	51.4	45.2	35.4	35.7
Massa et al.	67.0	62.5	43.0	39.4	71.5	64.0	49.4	37.5	58.3	55.7	46.3	44.2
Xiang et al.	60.4	36.3	23.7	16.4	60.7	37.0	23.4	19.9	48.7	37.2	31.4	24.6
Our Method	**74.3**	**67.2**	**51.0**	**42.1**	**74.4**	**72.3**	**52.2**	**47.1**	**71.8**	**65.5**	**55.6**	**52.1**

AVP: Average Viewpoint Precision

3D-RCNN: Experiments on KITTI

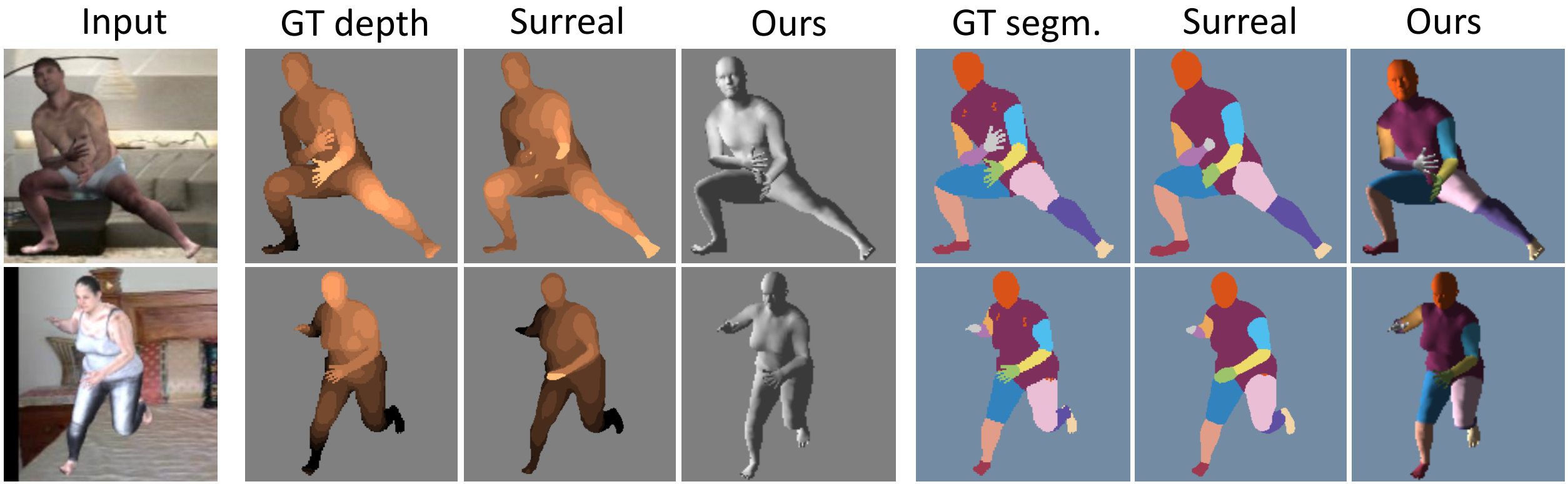
Input Image

3D object instances



3D object instances overlaid

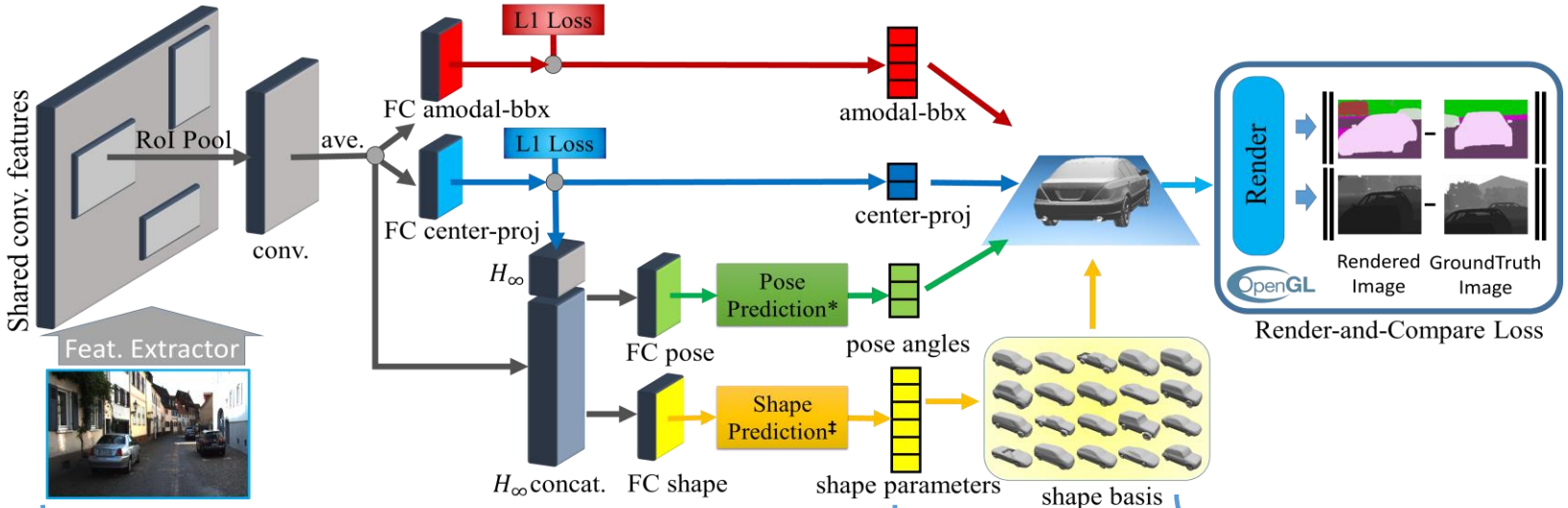
3D-RCNN: Experiments on **Surreal**



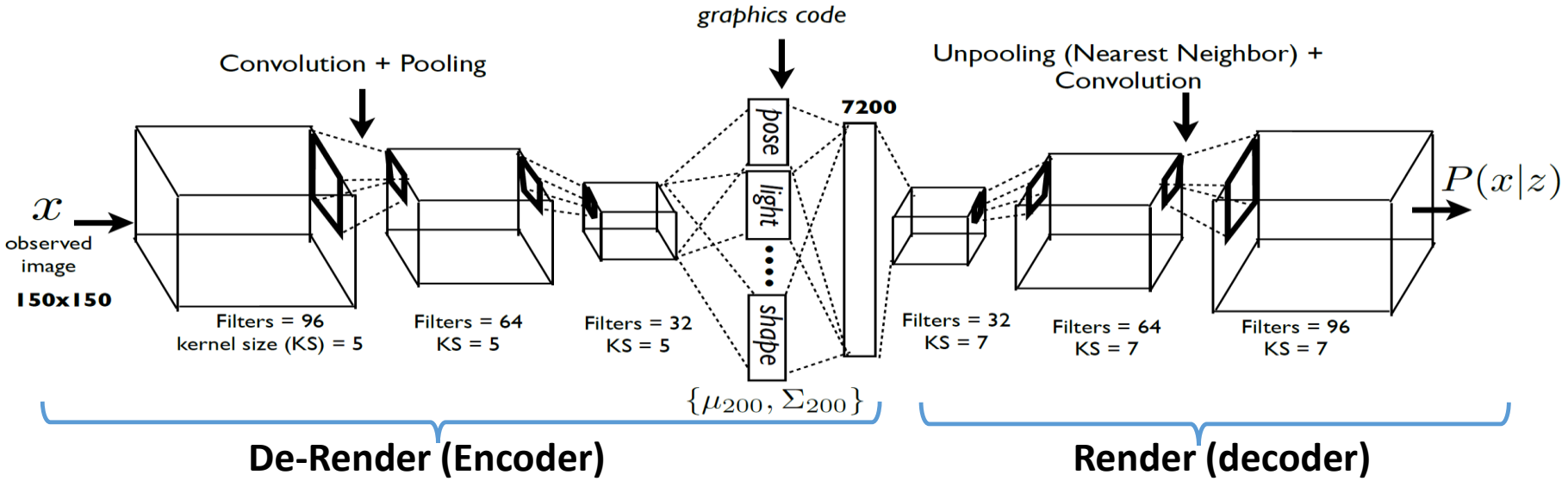
Surreal: Varol et al. Learning from Synthetic Humans. CVPR 2017

Modern Inverse-Graphics architectures

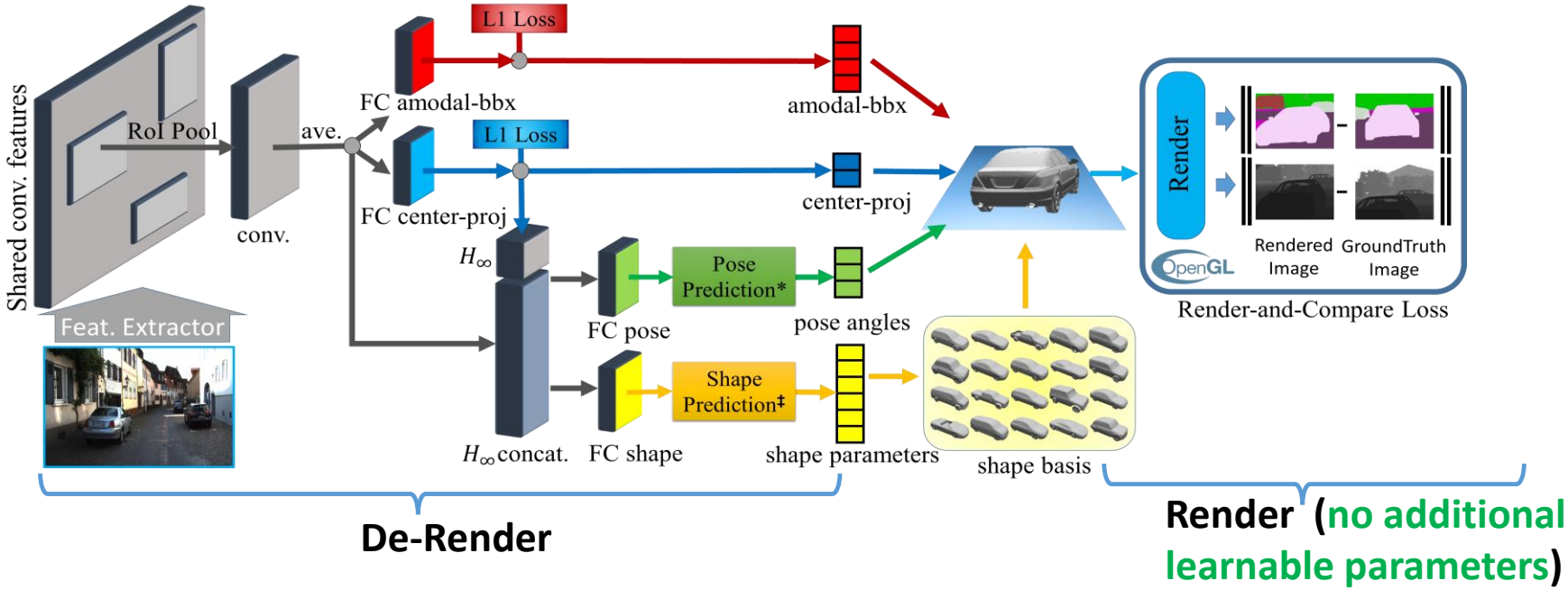
3D-RCNN



Kulkarni et. al **Deep Convolutional Inverse Graphics Network.** NIPS 2015



3D-RCNN: Summary



- Fast inverse-graphics network
- Exploits 3D CAD datasets for instance-level, class-specific shape prior
- Novel parametrization of shape and pose
- Differentiable Render-and-Compare

Questions and Feedback

Thank you for attending my talk