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

Abhijit Kundu*, Yin Li†, and James M. Rehg‡

This work was done at Georgia Tech. Authors currently at {*Google, †CMU, ‡ Amazon}.

Introduction

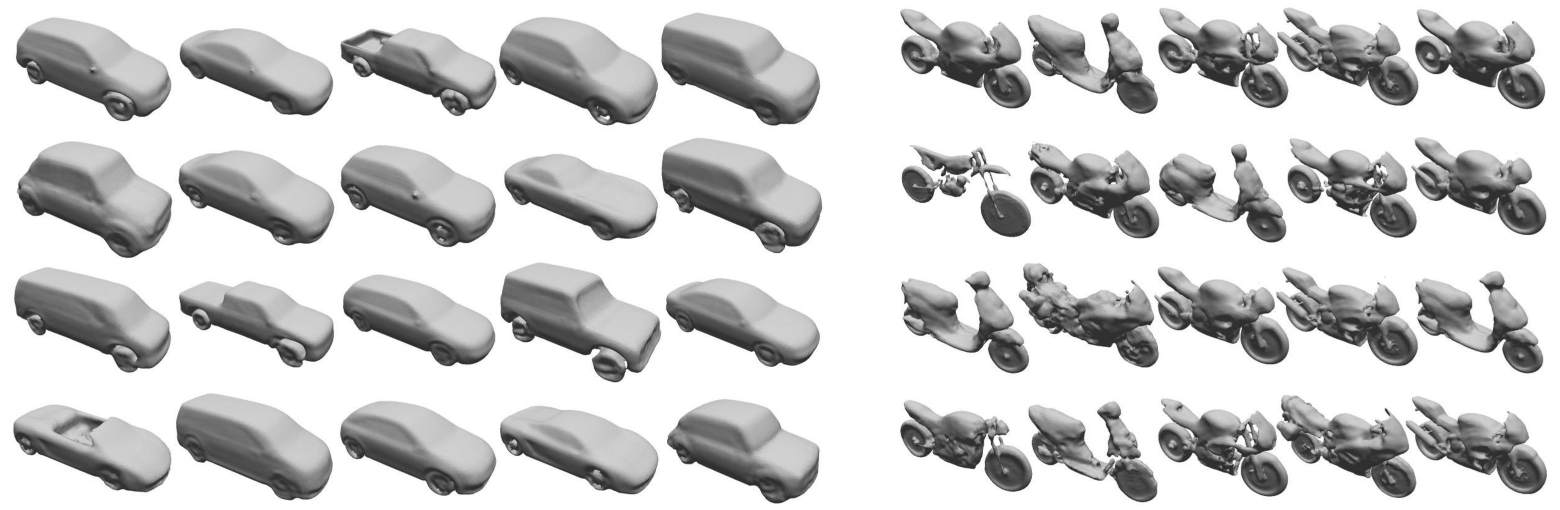
Problem Statement: Given an image, estimate **3D shape** and **3D pose** of all object instances.

Key Points:

- Fast inverse-graphics network
- Exploits 3D CAD datasets for instance-level, class-specific shape prior
- Novel parametrization of 3D shape and pose
- Differentiable Render-and-Compare (allows 2D supervision)
- Many 2D outputs (e.g. instance segmentation, depth-map) comes free

Shape Representation

- Learn class specific low dimensional parametric shape space
- For rigid objects: PCA on volumetric representation (TSDF)
- For articulated objects: SMPL (Lopper et al.)

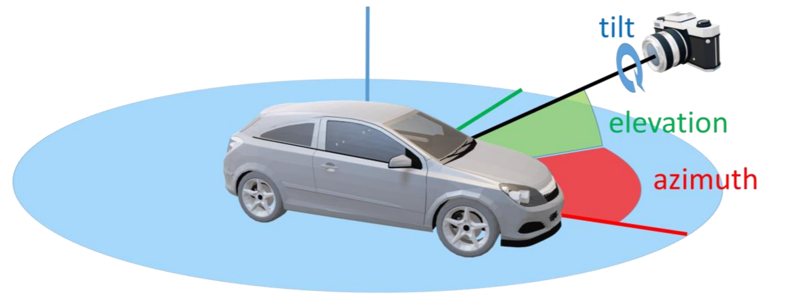


Pose Representation

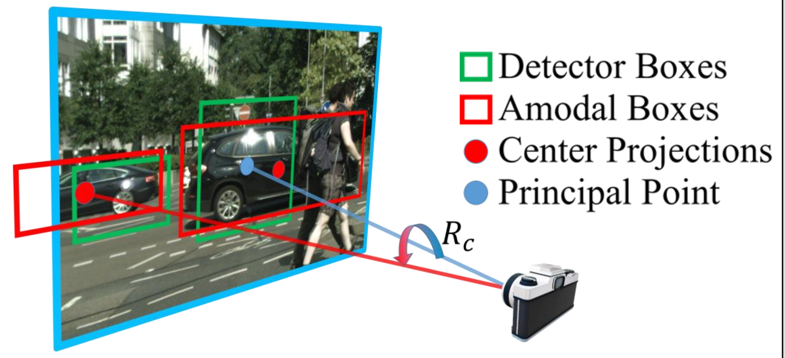
Which representation is better learnable target? **Egocentric** vs **Allocentric**

objects with same egocentric orientation

objects with same allocentric orientation



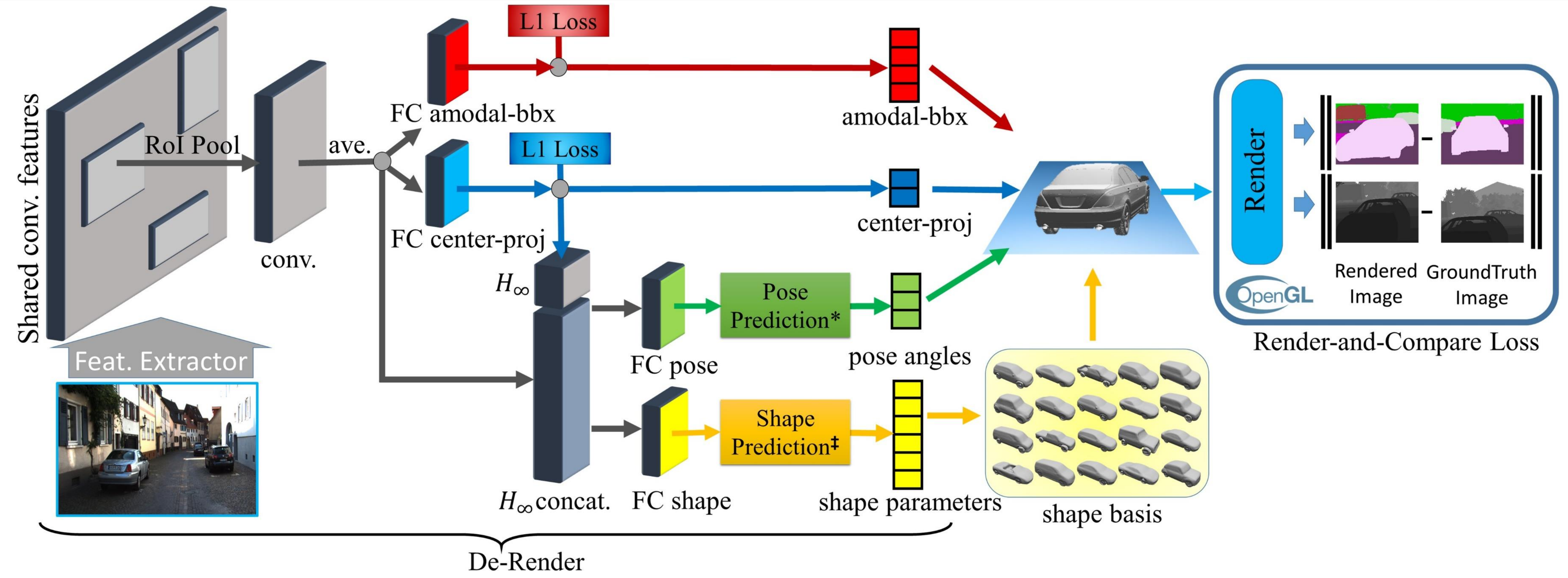
pose-angles = allocentric viewpoint + joint angles (articulated objects)



center-proj and amodal-bbx

pose-angles, center-proj and amodal-bbx completely describe the 3D pose.

3D-RCNN



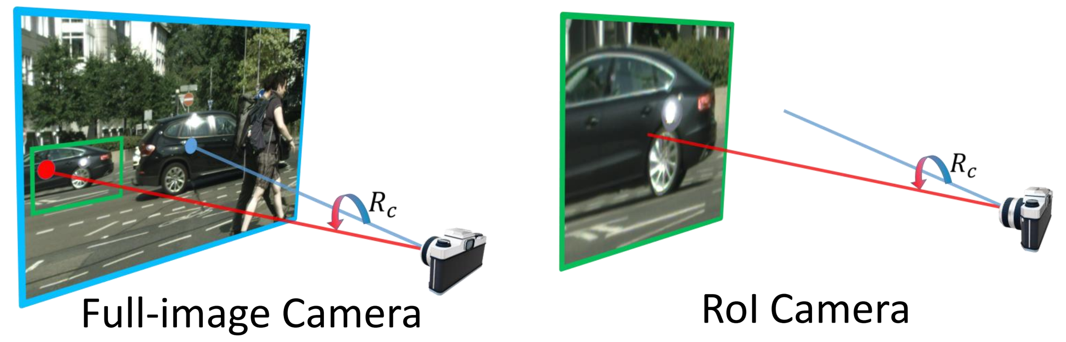
3D Equivariance

- RoIPool creates *scale* and *aspect-ratio* **invariant** representations
- How can we obtain 3D **equivariance**?
- 2D un-normalization (e.g. in box, mask) not possible for 3D targets

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

ROI images

We interpret the RoI transformation as image formed by a virtual camera RoI-Camera (rotated and with different intrinsics)

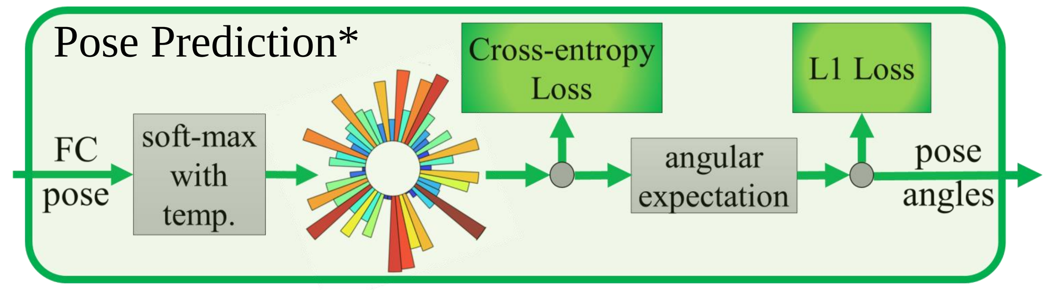


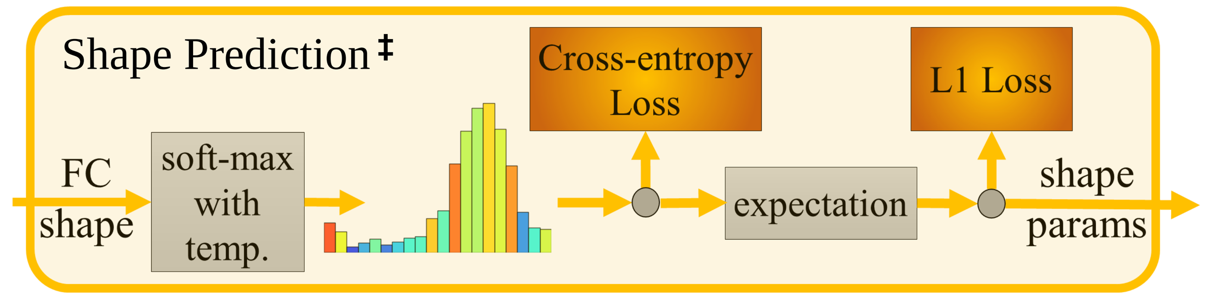
Full-image Camera   RoI Camera

Two cameras under pure rotation are related by the ***infinite-homography*** matrix $H_\infty = K_r R_c^{-1} K_c^{-1}$

Direct 3D Supervision

Whenever 3D ground-truth Is available, we use direct supervision. For shape and pose direct supervision, we use a combination of both *regression* and *classification* loss (can be interpreted as **soft-arg-max**).

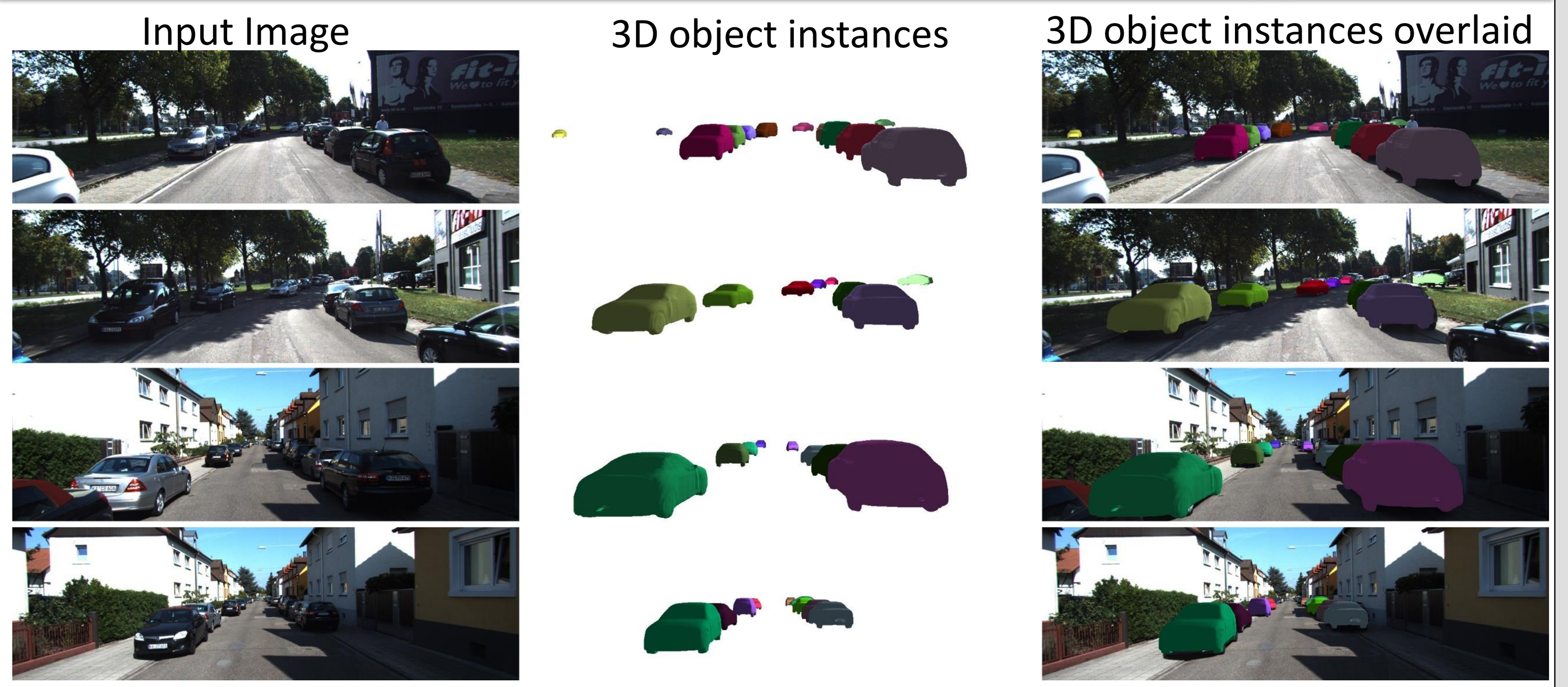


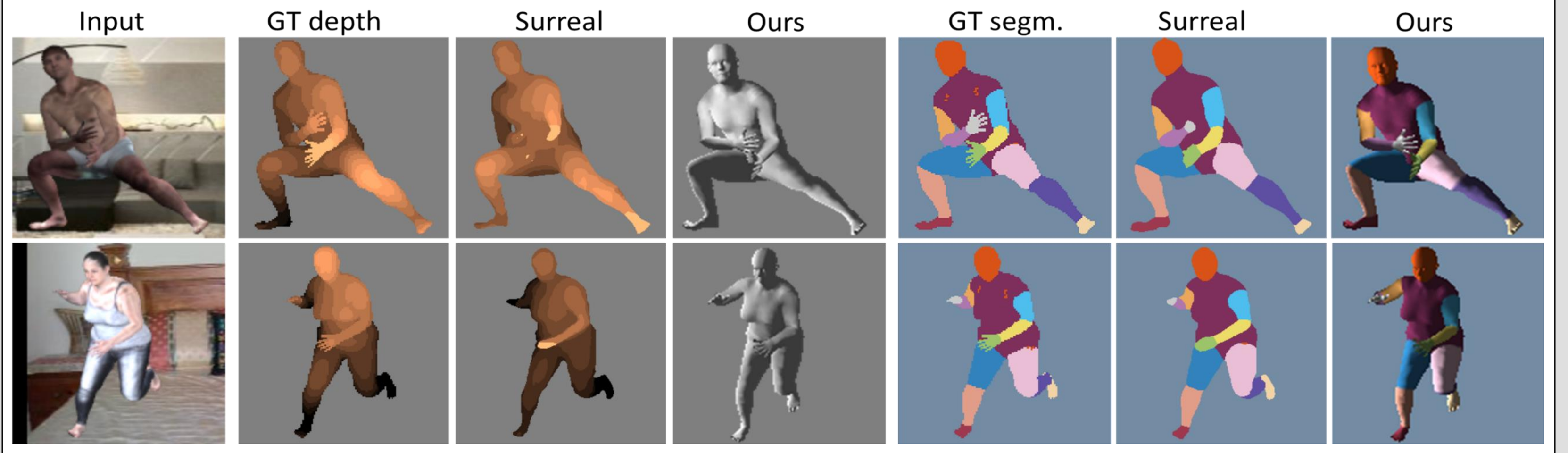


Render-and-Compare

- Render-and-Compare allows us to train from 2D annotation. We use **finite difference** for computing derivatives. This is possible because:
  - We have 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)

Experiments





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

KITTI Validation set	Easy			Moderate			Hard		
	AP↑	AOS↑	AAE↓	AP↑	AOS↑	AAE↓	AP↑	AOS↑	AAE↓
SubCNN 2016	90.53%	85.90%	12.24°	85.71%	84.21%	15.20°	72.71%	70.61%	17.14°
Our Method (original box)	90.53%	90.50%	1.99°	85.71%	85.57%	**4.51°**	72.71%	71.98%	**6.50°**
Our Method (rendered box)	**90.76%**	**90.73%**	**1.98°**	**89.31%**	**89.15%**	4.90°	**79.89%**	**79.51%**	7.94°

AP: Average Precision **AAE**: arccos(2*(AOS/AP)-1) **AOS**: Average Orientation Similarity

Future Work

- Extend to Video (shape constancy, smooth motion, tracking)
- Exploit rich self-supervised/predictive learning signal due to 3D representation