3D Object Detection in Point Clouds

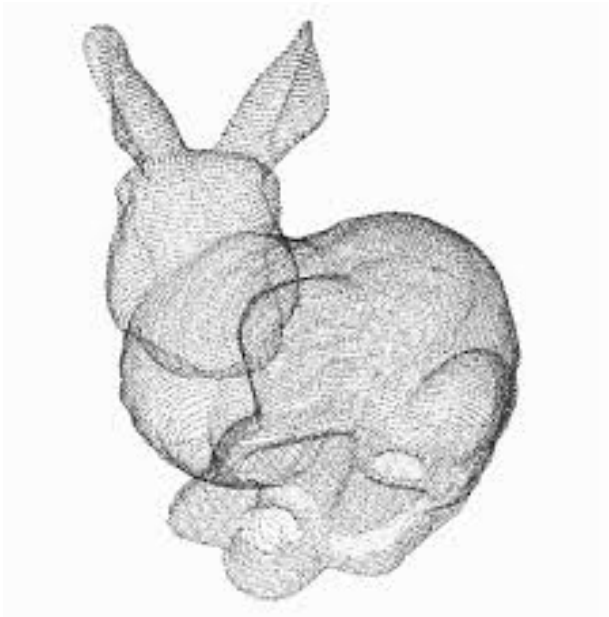
Chaoda Zheng

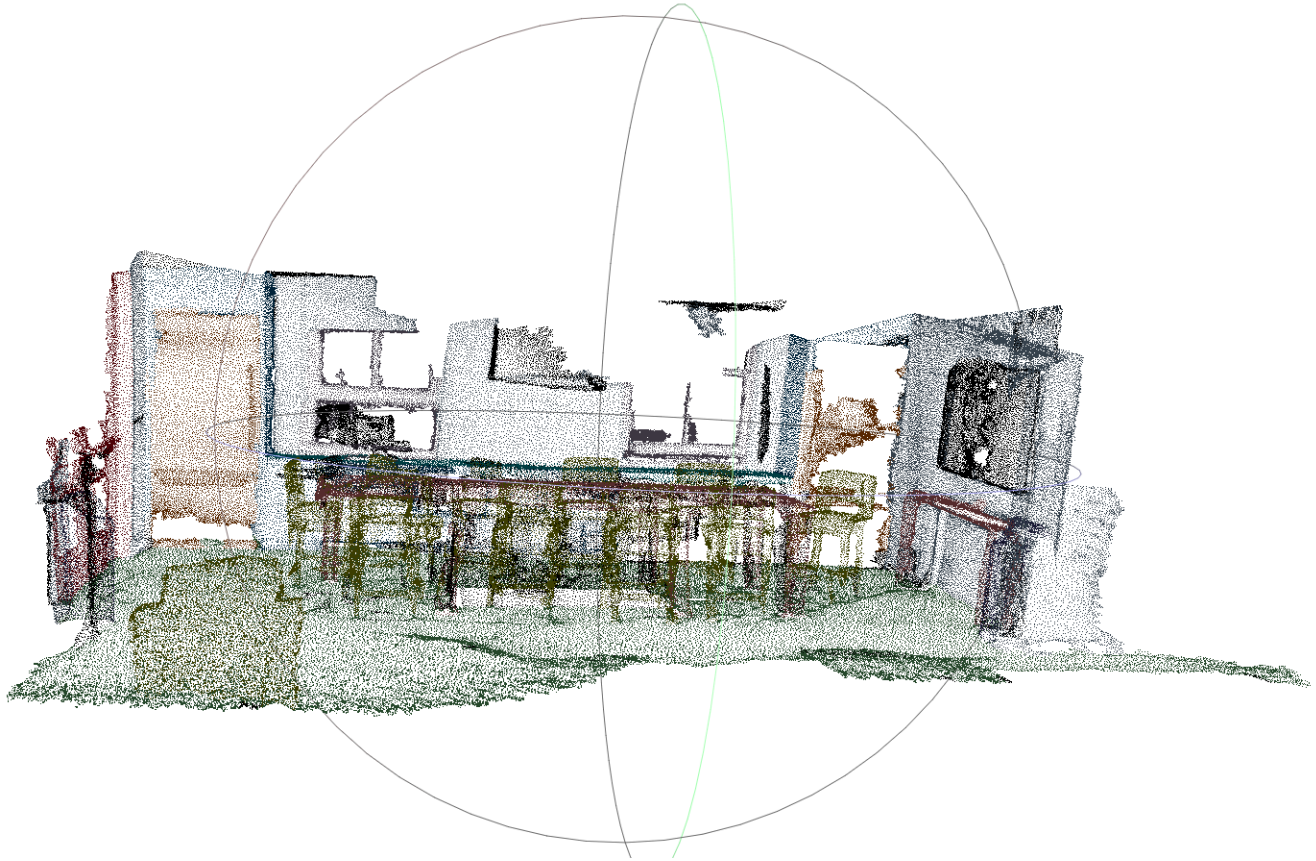
2020.4.2

3 works by Charles R. Qi

- Frustum PointNets for 3D Object Detection from RGB-D Data (CVPR 2018)
- Deep Hough Voting for 3D Object Detection in Point Clouds (ICCV 2019)
- ImVoteNet: Boosting 3D Object Detection in Point Clouds with Image Votes (CVPR 2020)

Point Cloud





A set of surface points

RGB-D

+

RGB

Depth

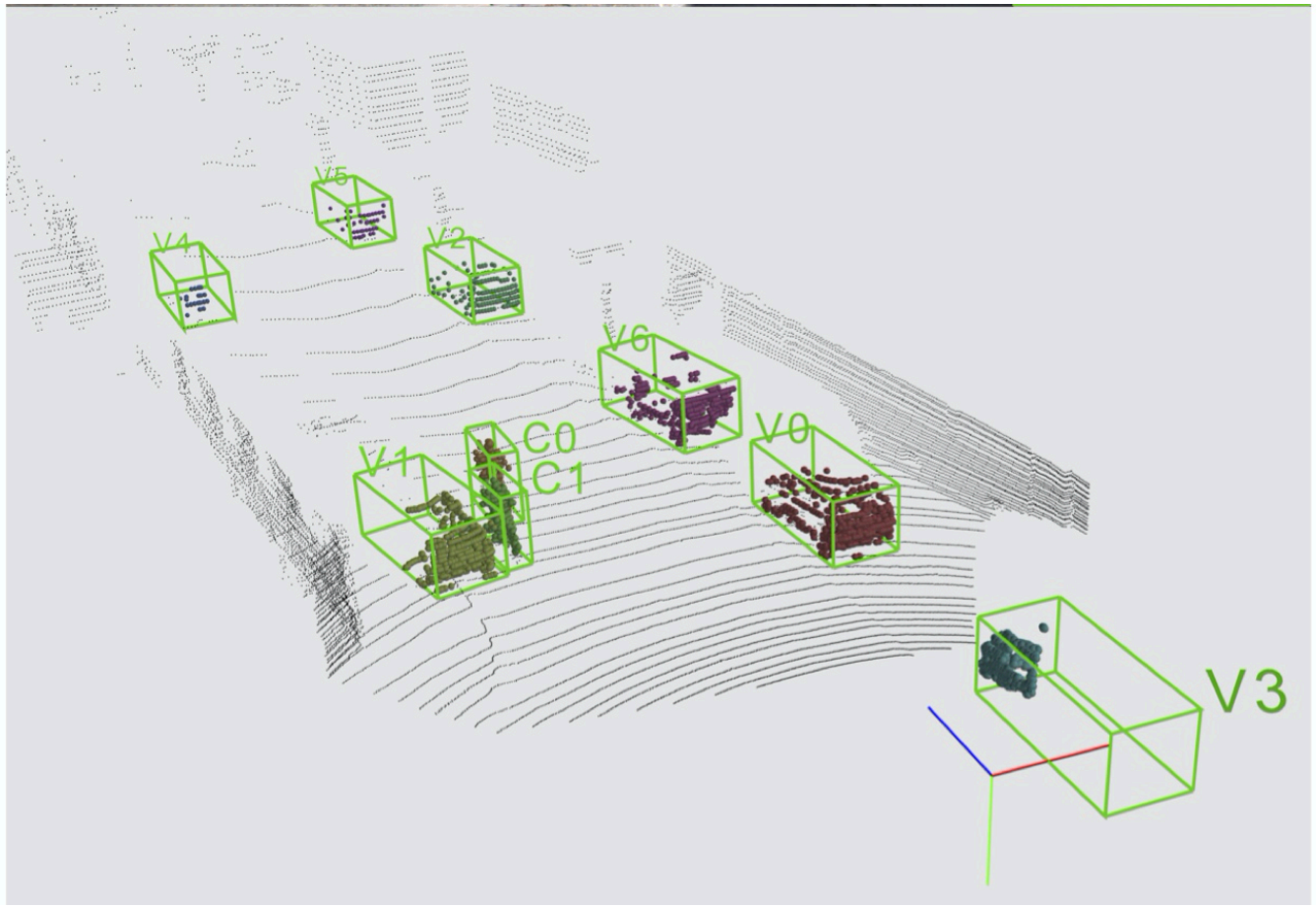
3 works by Charles R. Qi

- **Frustum PointNets for 3D Object Detection from RGB-D Data (CVPR 2018)**
- Deep Hough Voting for 3D Object Detection in Point Clouds (ICCV 2019)
- ImVoteNet: Boosting 3D Object Detection in Point Clouds with Image Votes (CVPR 2020)

Frustum PointNets for 3D Object Detection from RGB-D Data

Charles R. Qi[1*] Wei Liu[2] Chenxia Wu[2] Hao Su[3] Leonidas J. Guibas[1]
[1]Stanford University [2]Nuro, Inc. [3]UC San Diego

Amodal 3D Object Detection



The network estimates the **amodal** 3D bounding box (covering the entire object even if only part of it is visible).

Motivation

- 3D sensor data is often in the form of point clouds

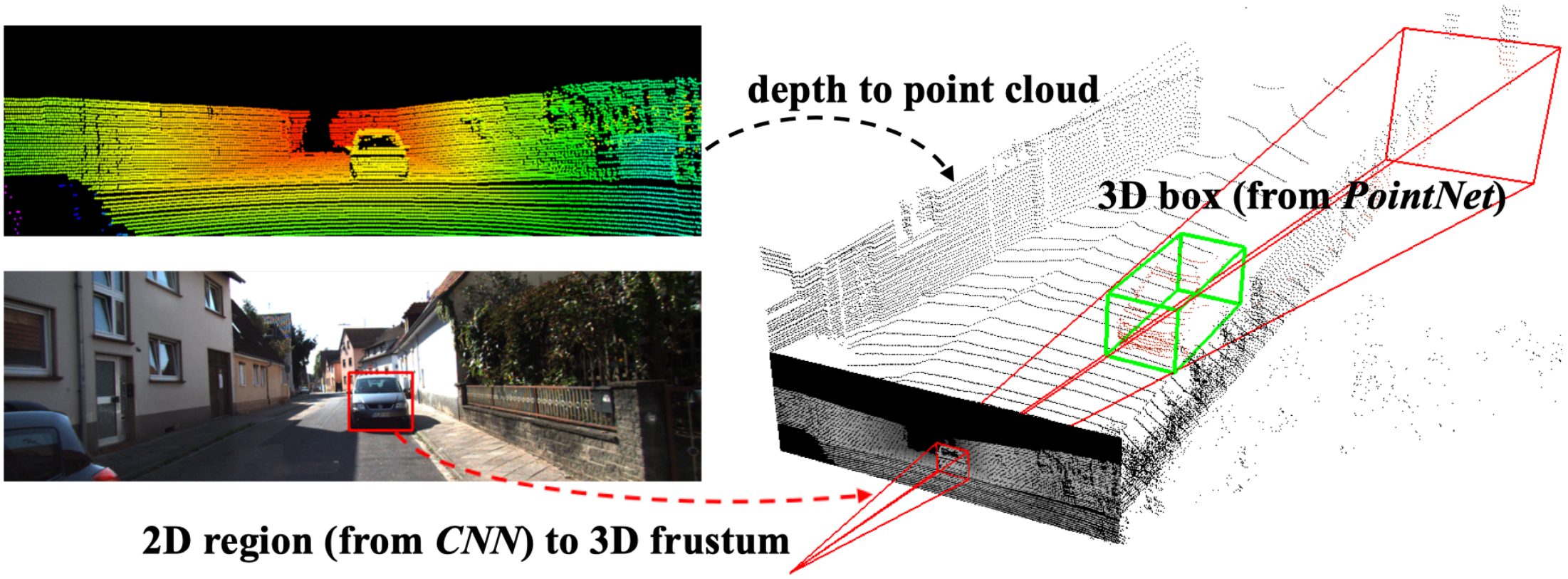
- Previous works are **not 3D-centric**:
  - convert 3D point clouds to images by projection
  - convert 3D point clouds to volumetric grids by quantization
  - treats RGB-D data as 2D maps for CNNs

- PointNets are capable of classifying a whole point cloud or predicting a semantic class for each point in a point cloud

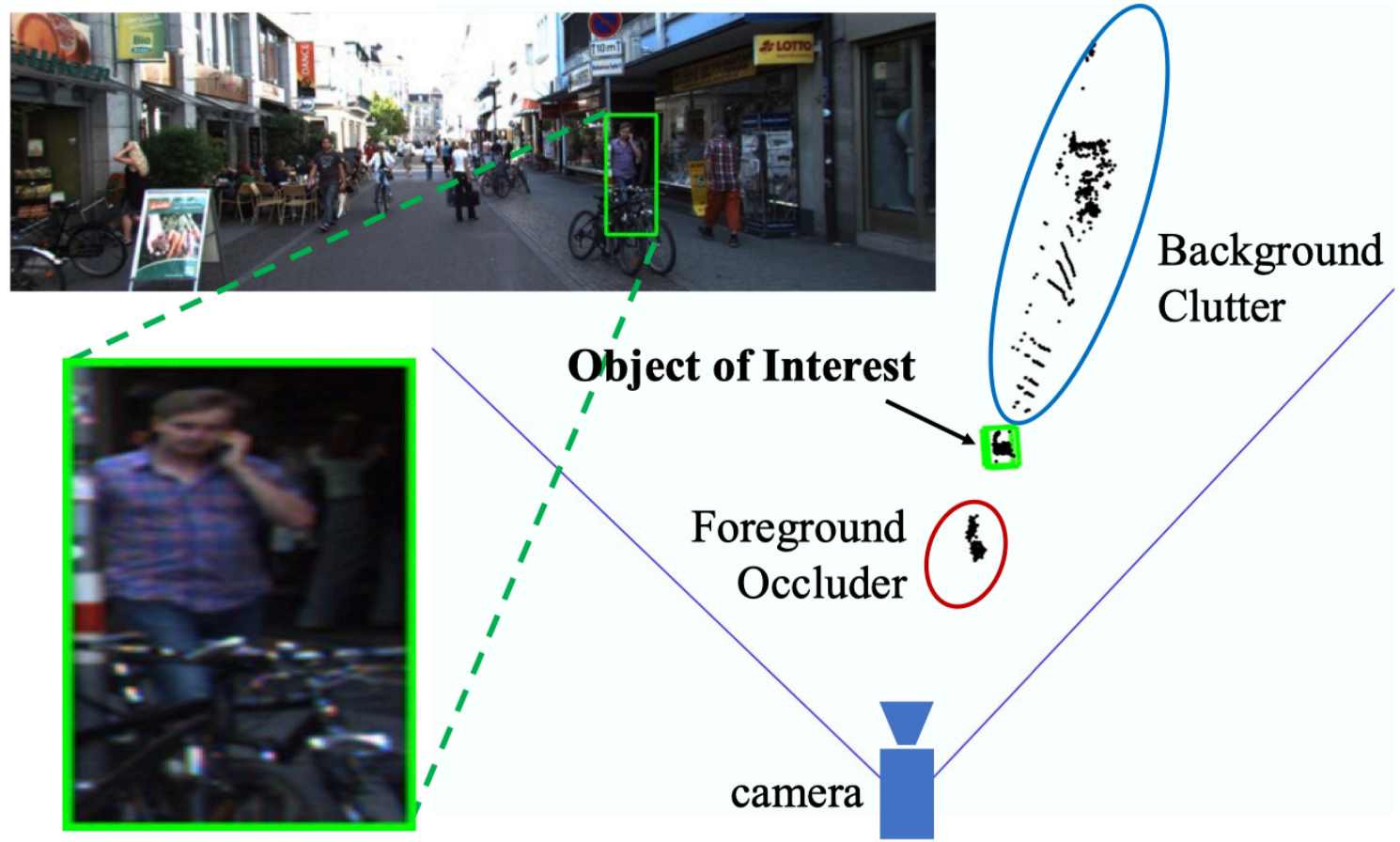
How to apply PointNets ?

- 3D sliding windows / 3D region proposal networks 😐
- reduce the search space by leveraging mature 2D object detectors 😁

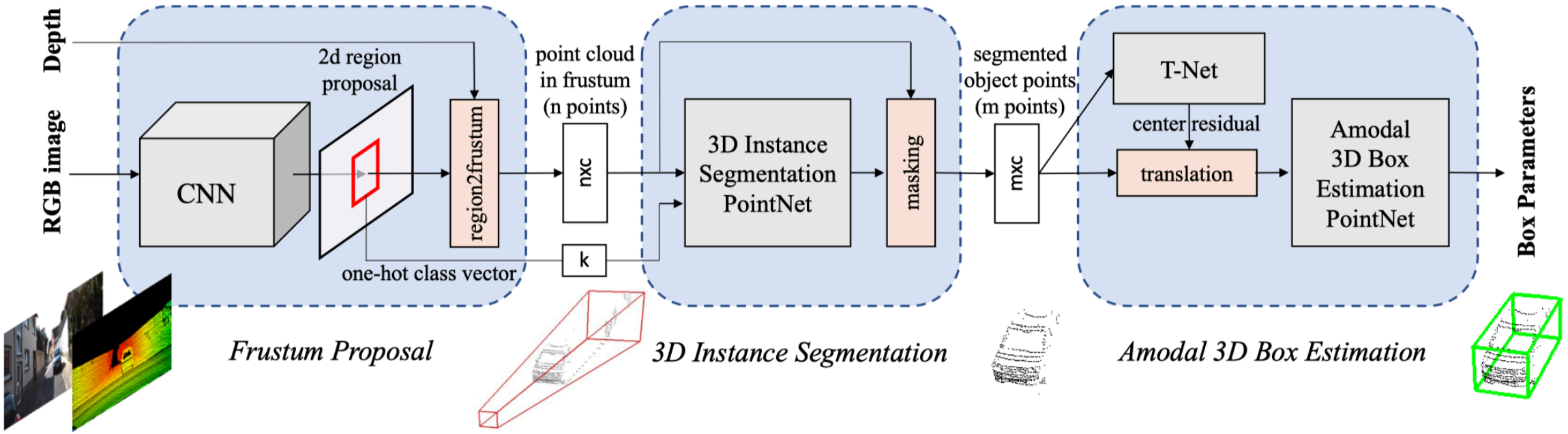
Pipeline of F-PointNet



Challenges for 3D detection in frustum point cloud



Architecture of F-PointNet



Output: 3 + 4 × NS + 2 × NH

center (c_x , c_y , c_z), size (h, w, l) and heading angle θ (along up-axis)

Experiments

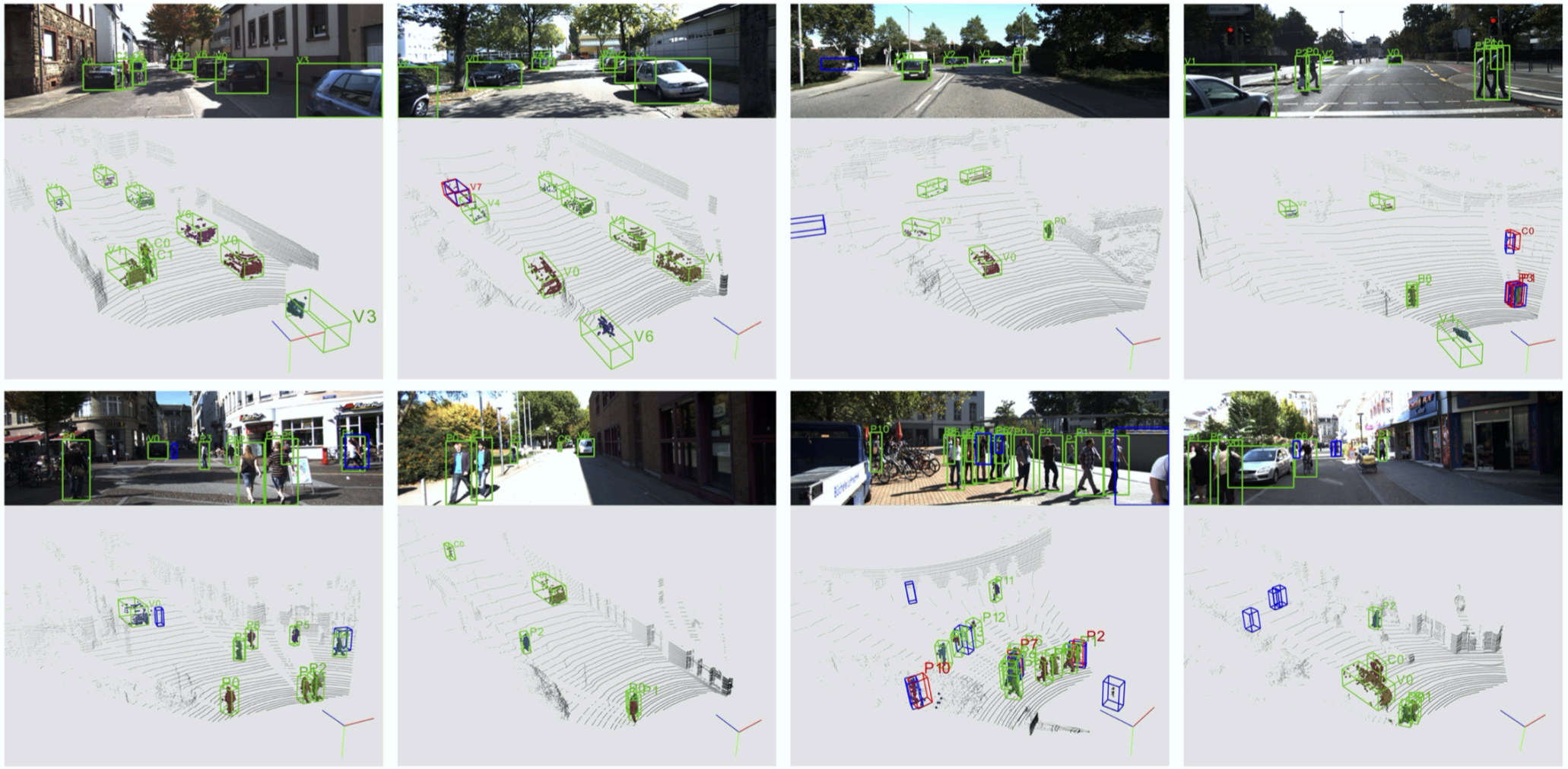
Method	Cars			Pedestrians			Cyclists		
	Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
DoBEM [42]	7.42	6.95	13.45	-	-	-	-	-	-
MV3D [6]	71.09	62.35	55.12	-	-	-	-	-	-
Ours (v1)	80.62	64.70	56.07	50.88	41.55	38.04	69.36	53.50	52.88
Ours (v2)	**81.20**	**70.39**	**62.19**	**51.21**	**44.89**	**40.23**	**71.96**	**56.77**	**50.39**

Table 1. **3D object detection** *3D* AP on KITTI *test* set. DoBEM [42] and MV3D [6] (previous state of the art) are based on 2D CNNs with bird's eye view LiDAR image. Our method, without sensor fusion or multi-view aggregation, outperforms those methods by large margins on all categories and data subsets. 3D bounding box IoU threshold is 70% for cars and 50% for pedestrians and cyclists.

Method	Cars			Pedestrians			Cyclists		
	Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
DoBEM [42]	36.49	36.95	38.10	-	-	-	-	-	-
3D FCN [17]	69.94	62.54	55.94	-	-	-	-	-	-
MV3D [6]	86.02	76.90	68.49	-	-	-	-	-	-
Ours (v1)	87.28	77.09	67.90	55.26	47.56	42.57	73.42	59.87	52.88
Ours (v2)	**88.70**	**84.00**	**75.33**	**58.09**	**50.22**	**47.20**	**75.38**	**61.96**	**54.68**

Table 2. **3D object localization** AP (bird's eye view) on KITTI *test* set. 3D FCN [17] uses 3D CNNs on voxelized point cloud and is far from real-time. MV3D [6] is the previous state of the art. Our method significantly outperforms those methods on all categories and data subsets. Bird's eye view 2D bounding box IoU threshold is 70% for cars and 50% for pedestrians and cyclists.

Experiments



Limitations

- Inaccurate pose and size estimation in a sparse point cloud
- there are multiple instances from the same category in a frustum
- 2D detector misses objects due to dark lighting or strong occlusion

3 works by Charles R. Qi

- Frustum PointNets for 3D Object Detection from RGB-D Data (CVPR 2018)
- **Deep Hough Voting for 3D Object Detection in Point Clouds (ICCV 2019)**
- ImVoteNet: Boosting 3D Object Detection in Point Clouds with Image Votes (CVPR 2020)

Deep Hough Voting for 3D Object Detection in Point Clouds

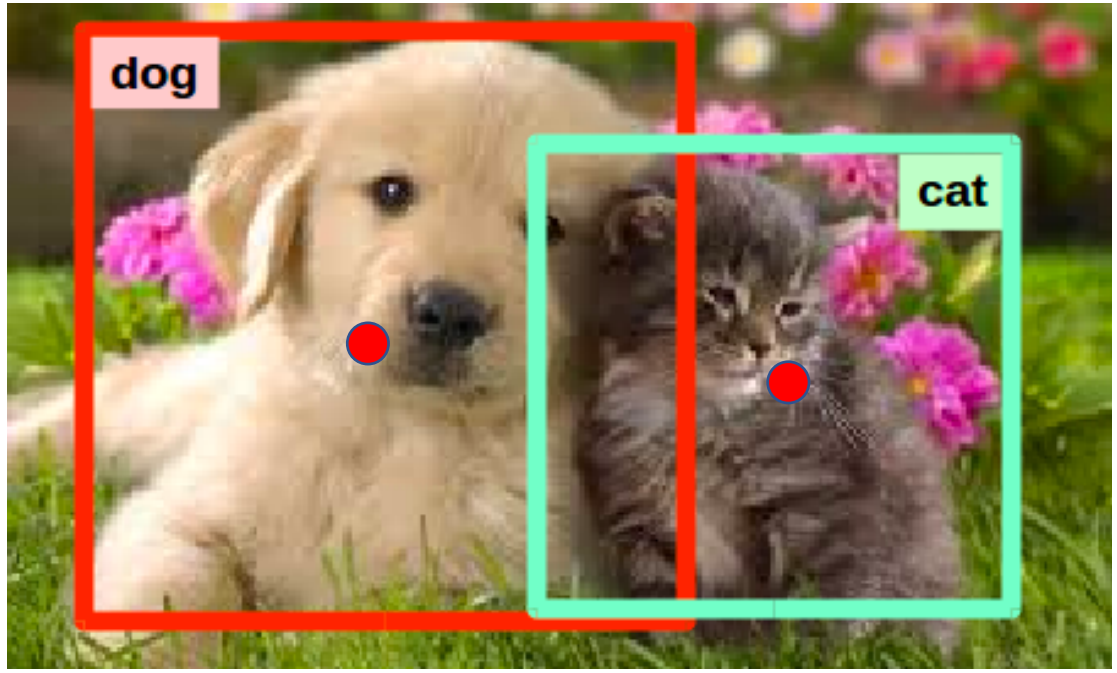
Charles R. Qi[1] Or Litany[1] Kaiming He[1] Leonidas J. Guibas[1,2]

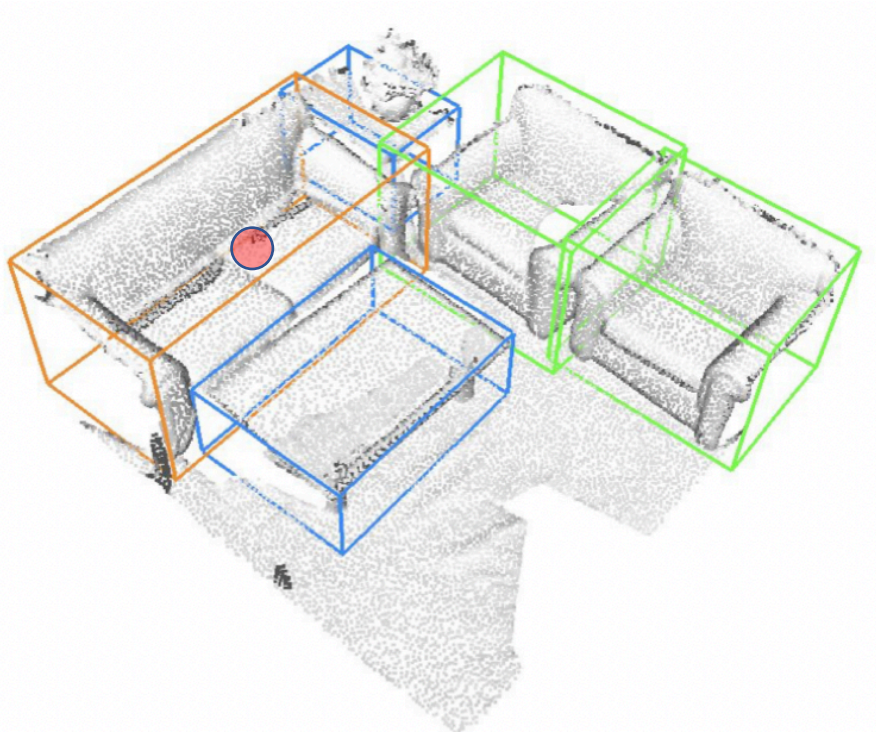
[1]Facebook AI Research [2]Stanford University

Motivation

- a *point cloud focused* 3D detection framework that directly processes **raw data** and does not depend on any 2D detectors neither in architecture nor in object proposal.

Challenges

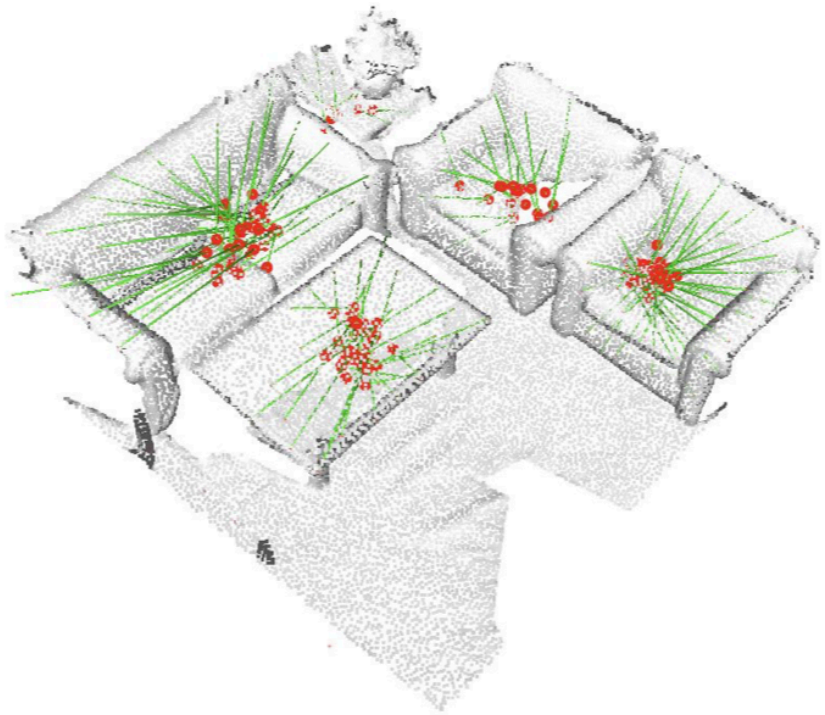




3D object centers are likely to be in empty space, far away from any point

Vote to object centers

Voting from input point cloud



Seed Point: $s_i = [x_i; f_i], x_i \in \mathbb{R}^3, f_i \in \mathbb{R}^C$

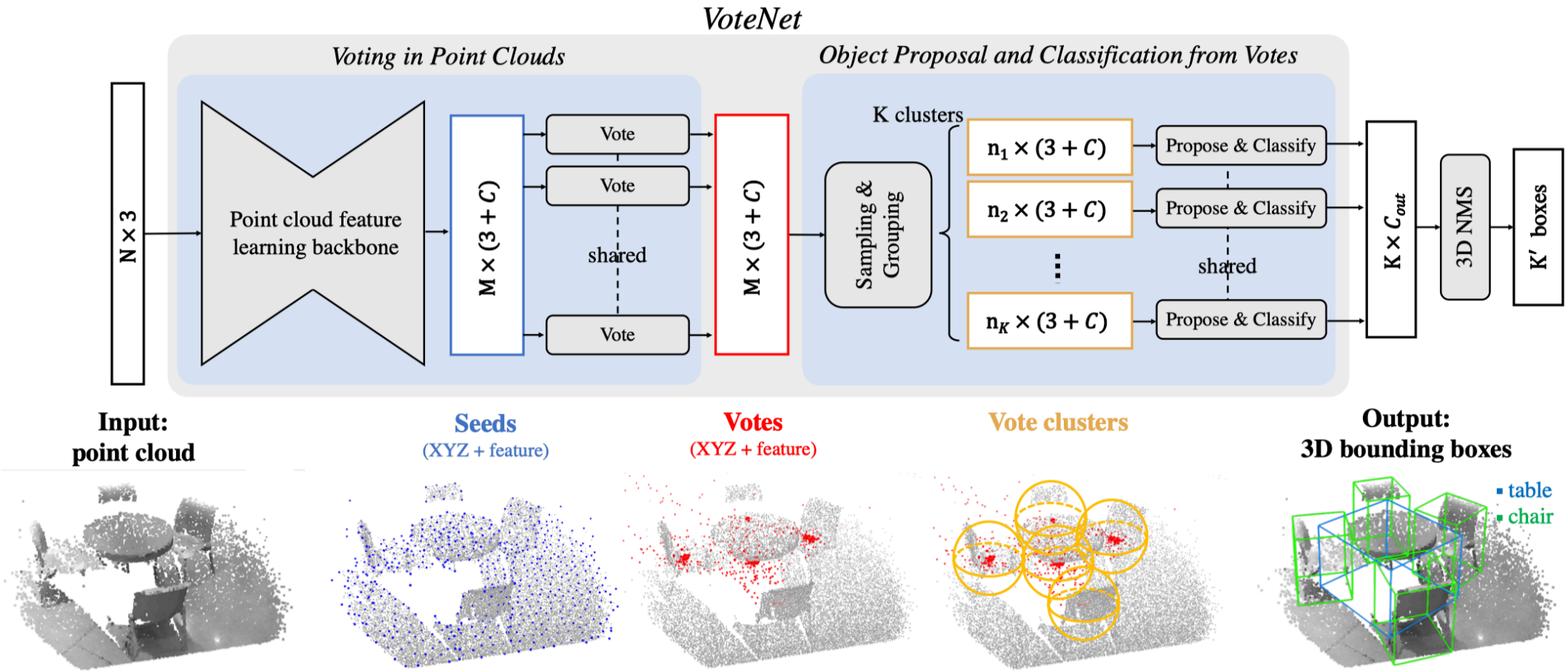
Offset learning: $[x_i; f_i] \rightarrow MLP \rightarrow [\Delta x_i; \Delta f_i]$

Votes: $v_i = [y_i; g_i] = [x_i + \Delta x_i; f_i + \Delta f_i]$

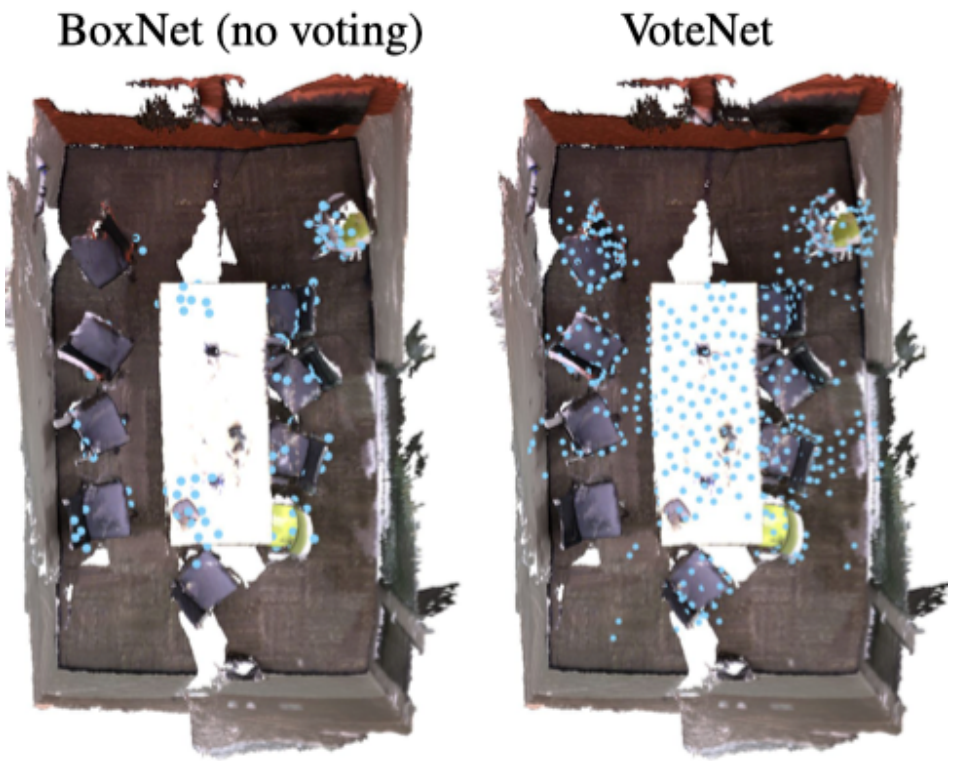
The regression loss for votes

$$L_{\text{vote-reg}} = \frac{1}{M_{\text{pos}}} \sum_i \|\Delta x_i - \Delta x_i^*\| \mathbb{1}[s_i \text{ on object}]$$

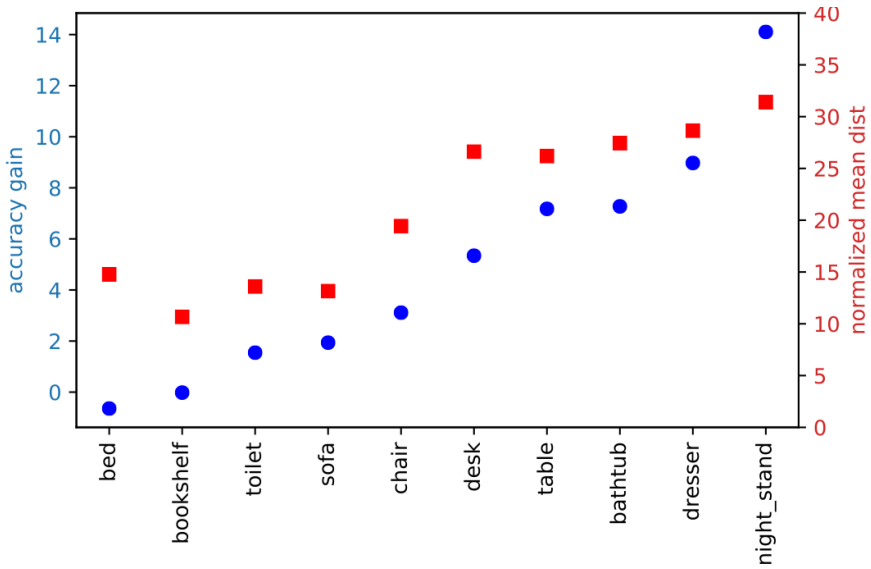
Architecture of VoteNet



To Vote or Not To Vote?



Method	mAP@0.25	
	SUN RGB-D	ScanNet
BoxNet (ours)	53.0	45.4
VoteNet (ours)	**57.7**	**58.6**



Experiments

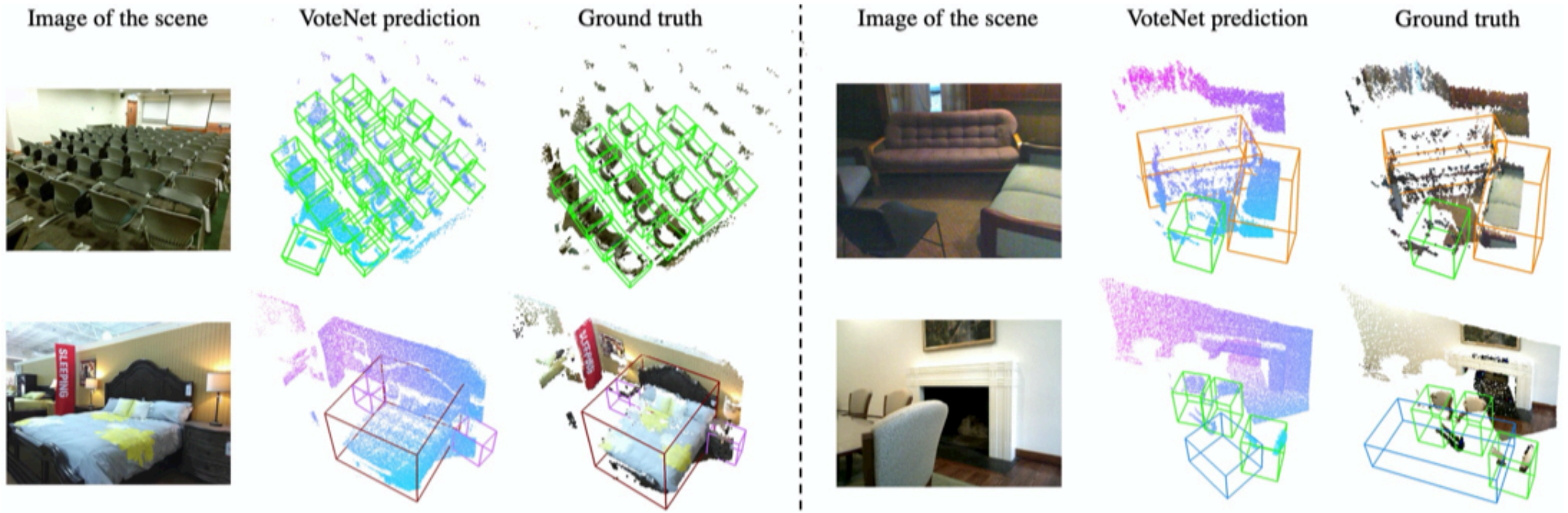
	Input	bathtub	bed	bookshelf	chair	desk	dresser	nightstand	sofa	table	toilet	mAP
DSS [42]	Geo + RGB	44.2	78.8	11.9	61.2	20.5	6.4	15.4	53.5	50.3	78.9	42.1
COG [38]	Geo + RGB	58.3	63.7	31.8	62.2	**45.2**	15.5	27.4	51.0	**51.3**	70.1	47.6
2D-driven [20]	Geo + RGB	43.5	64.5	31.4	48.3	27.9	25.9	41.9	50.4	37.0	80.4	45.1
F-PointNet [34]	Geo + RGB	43.3	81.1	**33.3**	64.2	24.7	**32.0**	58.1	61.1	51.1	**90.9**	54.0
VoteNet (ours)	**Geo only**	**74.4**	**83.0**	28.8	**75.3**	22.0	29.8	**62.2**	**64.0**	47.3	90.1	**57.7**

Table 1. **3D object detection results on SUN RGB-D val set.** Evaluation metric is average precision with 3D IoU threshold 0.25 as proposed by [40]. Note that both COG [38] and 2D-driven [20] use room layout context to boost performance. To have fair comparison with previous methods, the evaluation is on the SUN RGB-D V1 data.

	Input	mAP@0.25	mAP@0.5
DSS [42, 12]	Geo + RGB	15.2	6.8
MRCNN 2D-3D [11, 12]	Geo + RGB	17.3	10.5
F-PointNet [34, 12]	Geo + RGB	19.8	10.8
GSPN [54]	Geo + RGB	30.6	17.7
3D-SIS [12]	Geo + 1 view	35.1	18.7
3D-SIS [12]	Geo + 3 views	36.6	19.0
3D-SIS [12]	Geo + 5 views	40.2	22.5
3D-SIS [12]	Geo only	25.4	14.6
VoteNet (ours)	Geo only	**58.6**	**33.5**

Table 2. **3D object detection results on ScanNetV2 val set.** DSS and F-PointNet results are from [12]. Mask R-CNN 2D-3D results are from [54]. GSPN and 3D-SIS results are up-to-date numbers provided by the original authors.

Experiments



Limitations

- Inaccurate pose and size estimation in a sparse point cloud

- ~~there are multiple instances from the same category in a frustum~~

- ~~2D detector misses objects due to dark lighting or strong occlusion~~

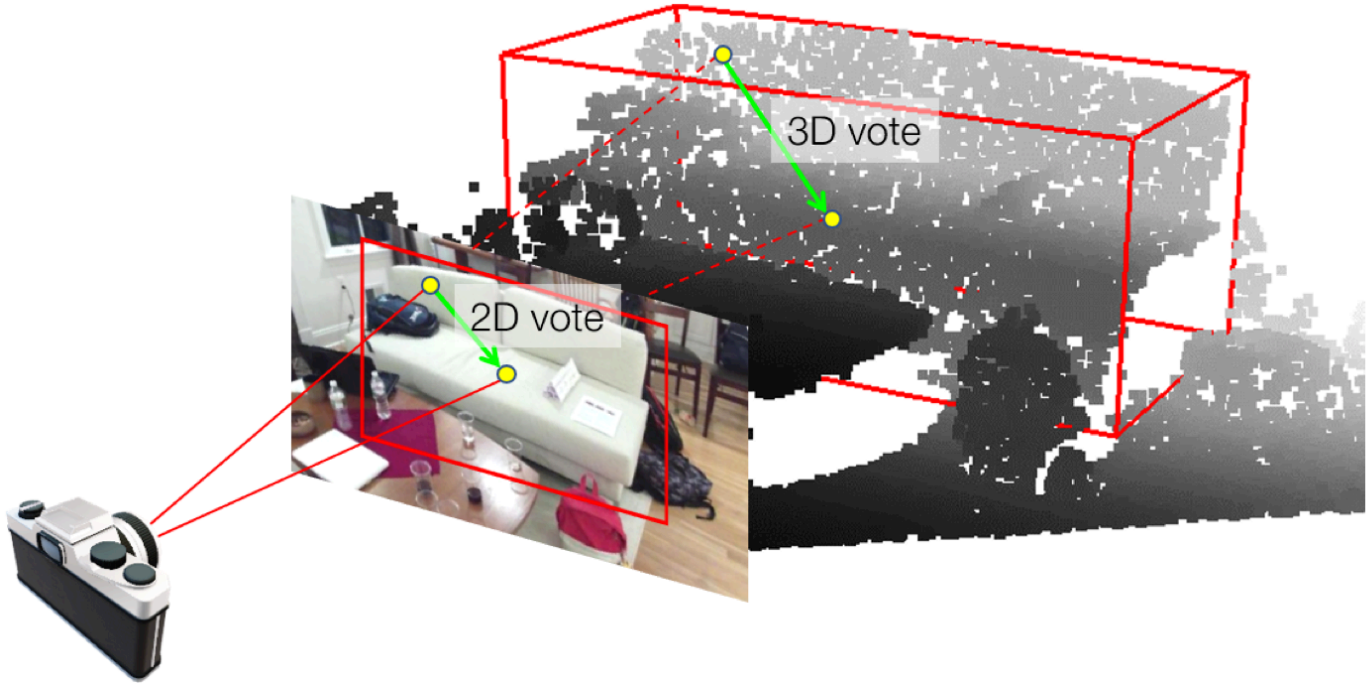
3 works by Charles R. Qi

- Frustum PointNets for 3D Object Detection from RGB-D Data (CVPR 2018)
- Deep Hough Voting for 3D Object Detection in Point Clouds (ICCV 2019)
- **ImVoteNet: Boosting 3D Object Detection in Point Clouds with Image Votes (CVPR 2020)**

ImVoteNet: Boosting 3D Object Detection in Point Clouds with Image Votes

Charles R. Qi*† Xinlei Chen*[1] Or Litany[1,2] Leonidas J. Guibas[1,2]
[1]Facebook AI [2]Stanford University

Motivation



The 2D vote reduces the search space of the 3D object center to a ray while the color texture in image provides a strong semantic prior.

points + images

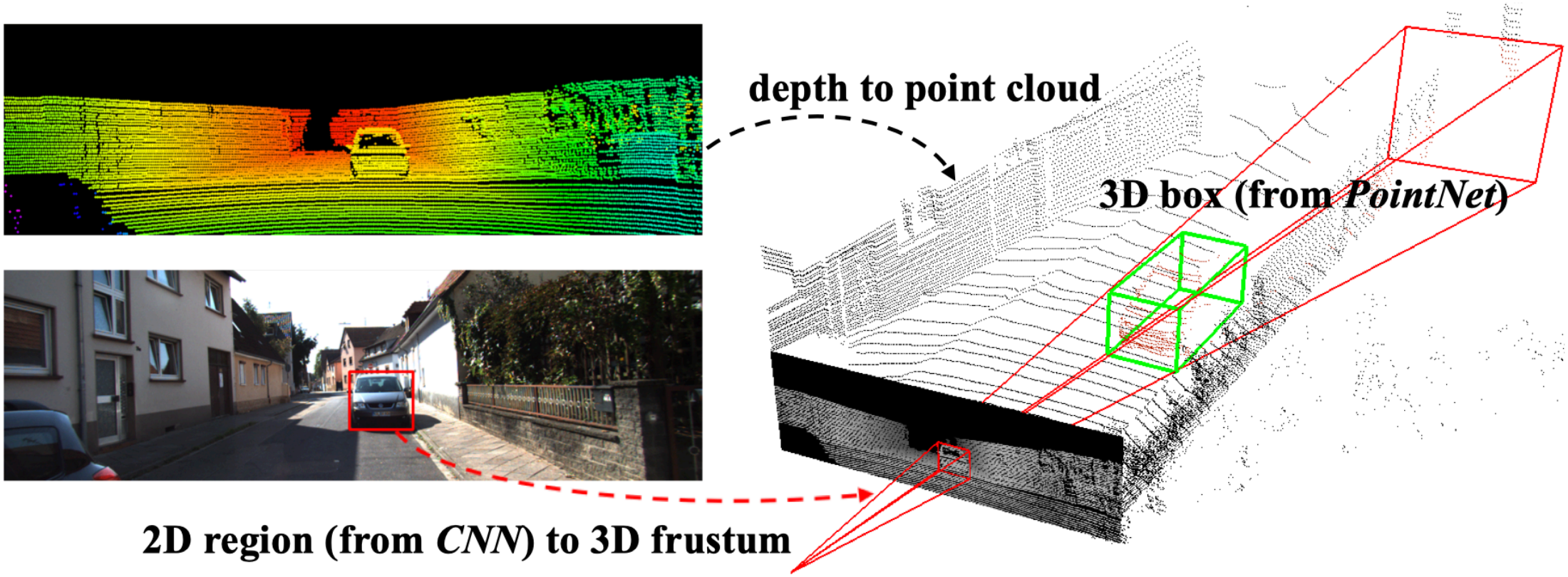
Motivation

- RGB images have value in 3D object detection
  - have **higher resolution** than depth images or LiDAR point clouds
  - contain **rich textures** that are not available in the point domain.
  - can cover "**blind regions**" of active depth sensors which often occur due to reflective surfaces.

- Images lack absolute measures of object depth and scale, which are exactly what 3D point clouds can provide

points + images: 1+1 > 2

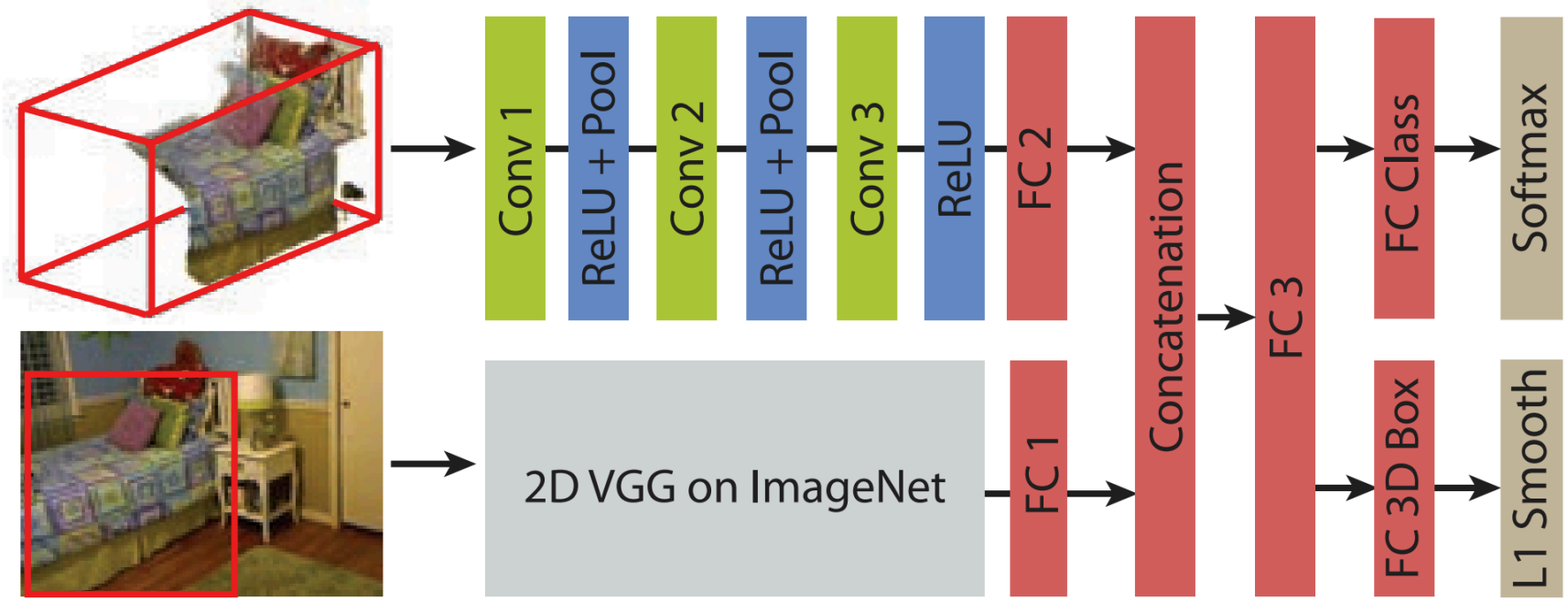
Previous works



Cascaded design. If an object is missed in 2D, it will be missed in 3D as well

Previous works

Deep Sliding Shapes for Amodal 3D Object Detection in RGB-D Images (CVPR 2016)



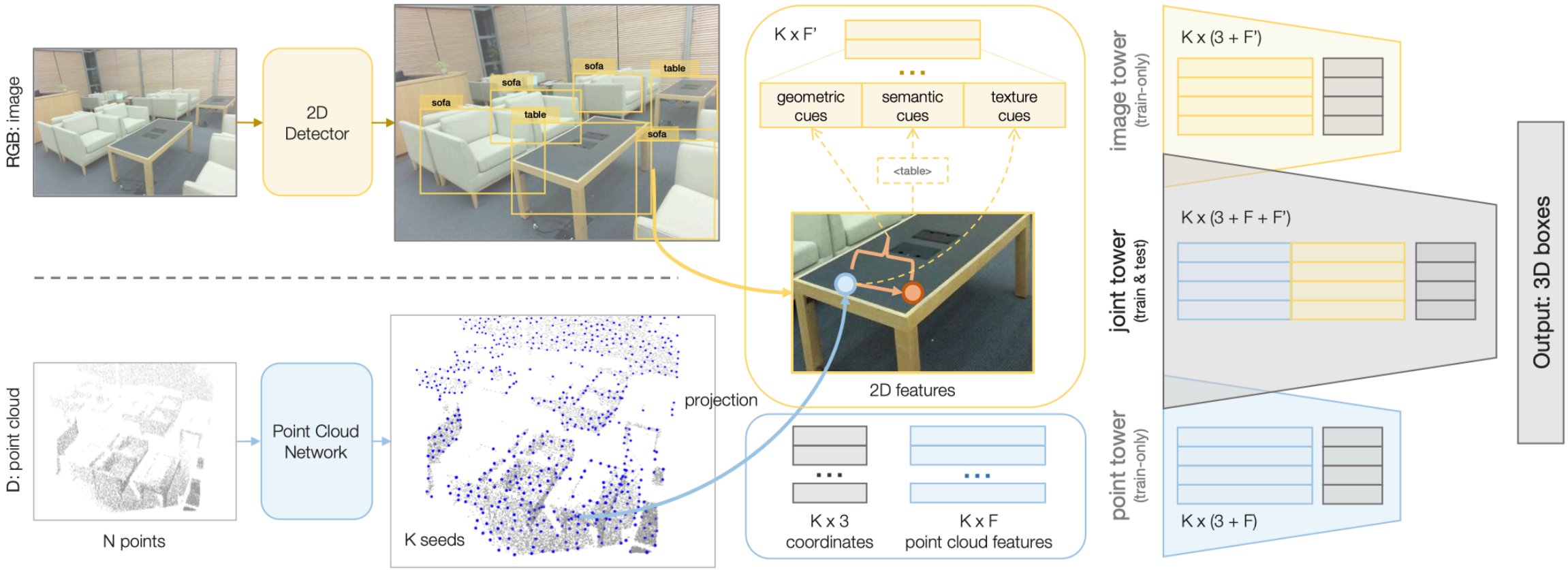
Do not use 2D images directly for localization,
which can provide helpful guidance for detection objects in 3D

ImVoteNet

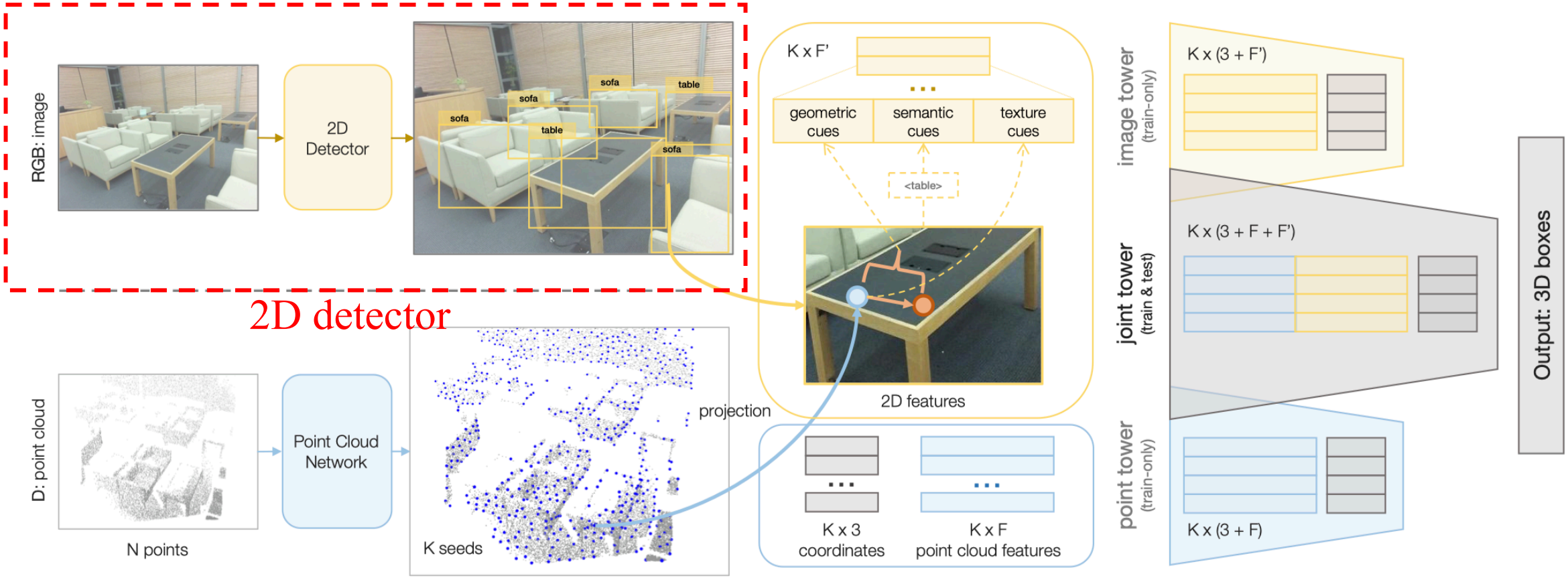
- Takes advantage of the more mature 2D detectors

- Reserves the ability to propose objects from the full point cloud itself .

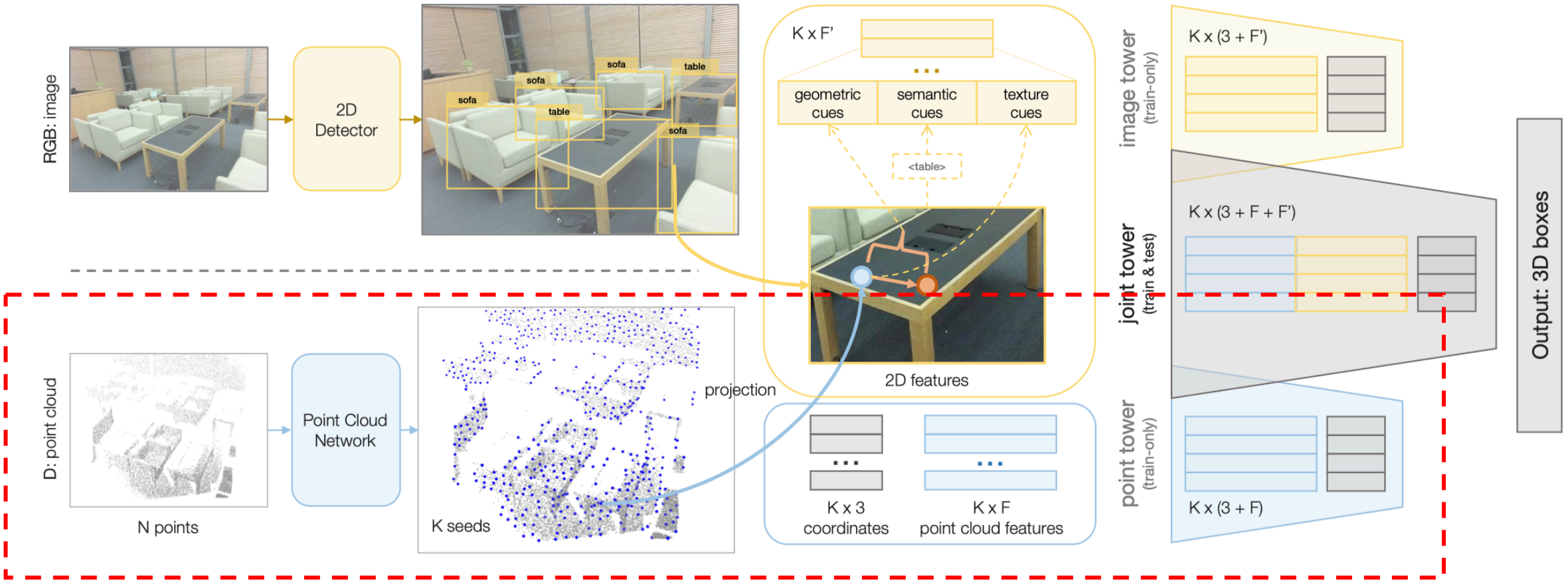
Architecture



Architecture

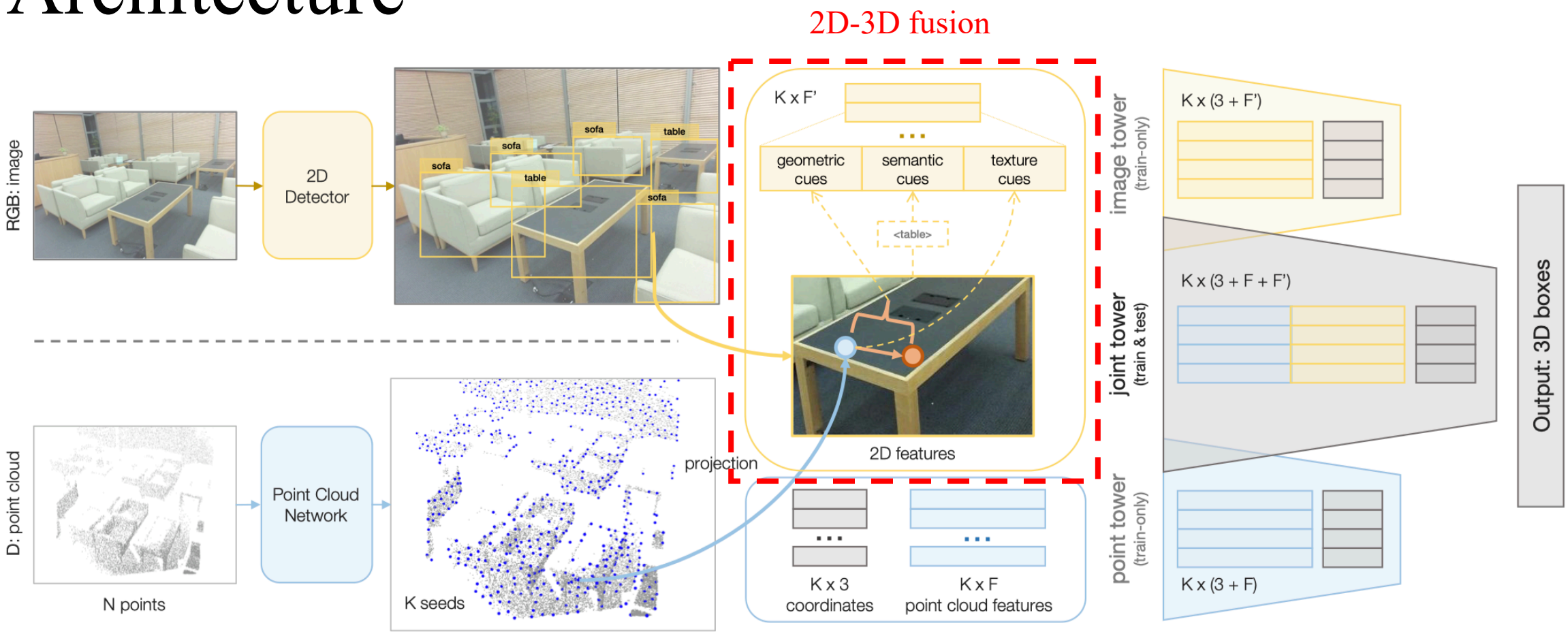


Architecture

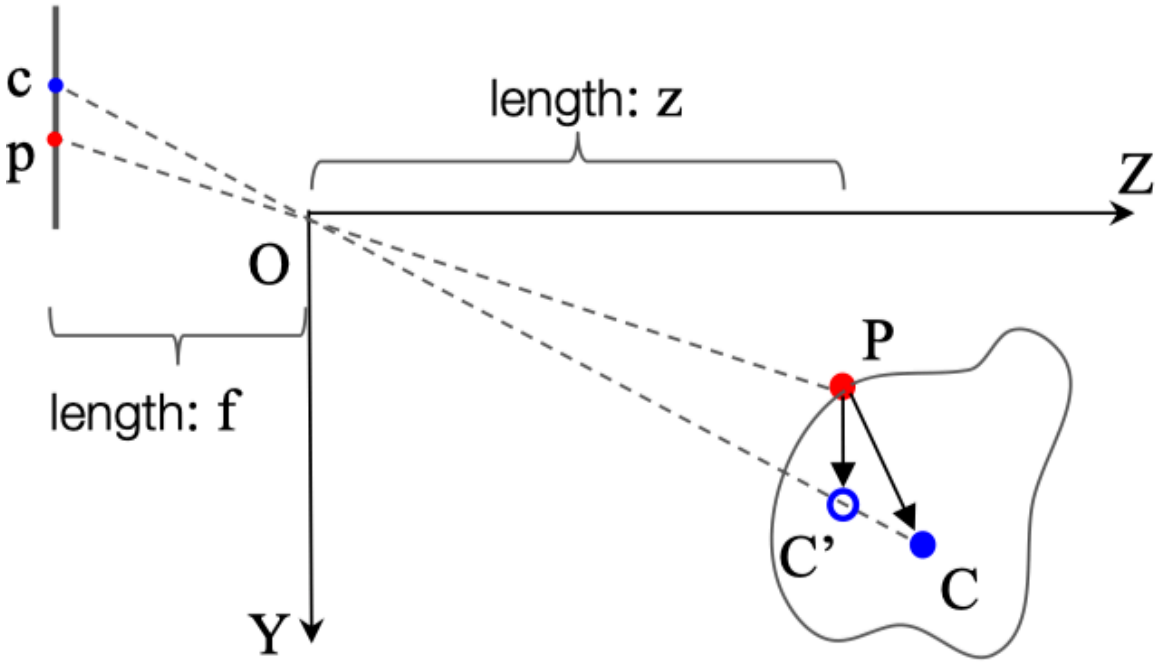


VoteNet

Architecture



Geometric cues

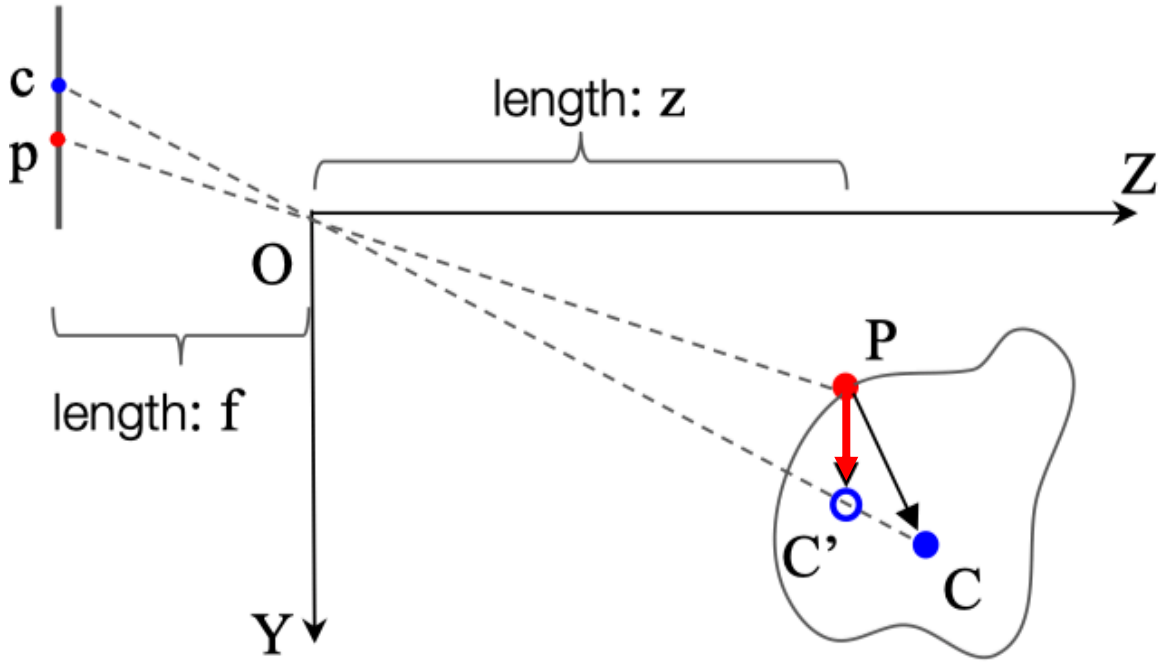


$$P = (x_1, y_1, z_1),$$
$$C = (x_2, y_2, z_2),$$
$$p = (u_1, v_1),$$
$$c = (u_2, v_2)$$

Known 2D vote: $\overrightarrow{pc} = (u_2 - u_1, v_2 - v_1) = (\Delta u, \Delta v)$

Target 3D vote: $\overrightarrow{PC} = (x_2 - x_1, y_2 - y_1, z_2 - z_1)$

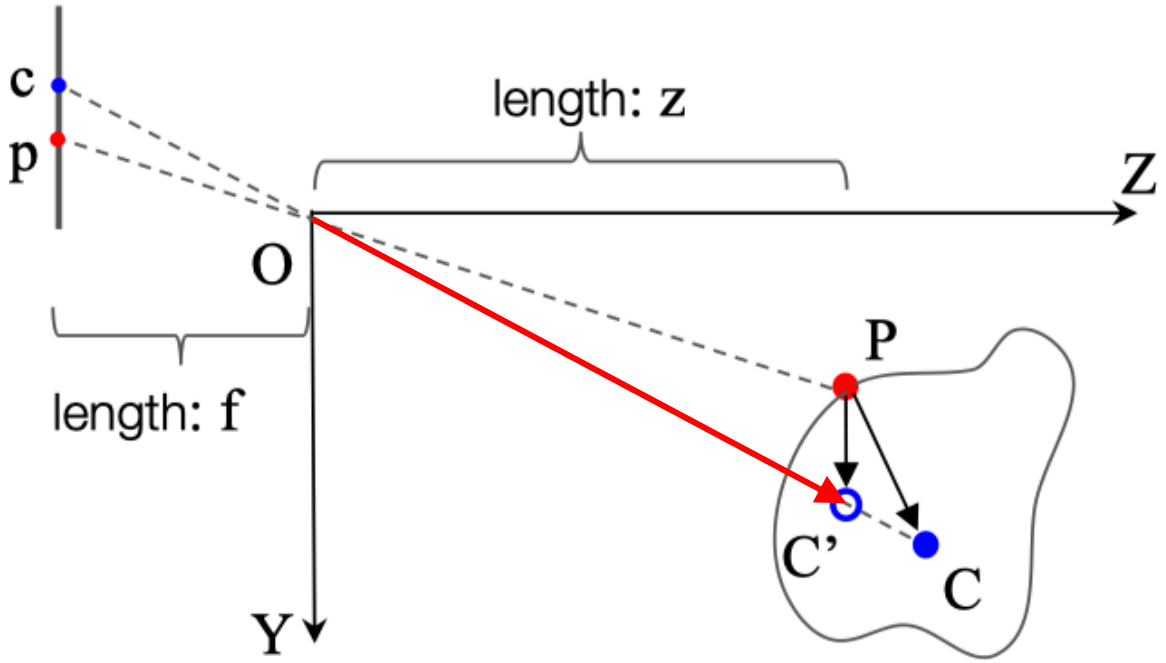
Geometric cues - Pseudo vote



$$z_1 \approx z_2$$

Pseudo vote: $\overrightarrow{PC'} = \left(\frac{\Delta u}{f} z_1, \frac{\Delta v}{f} z_1, 0\right)$

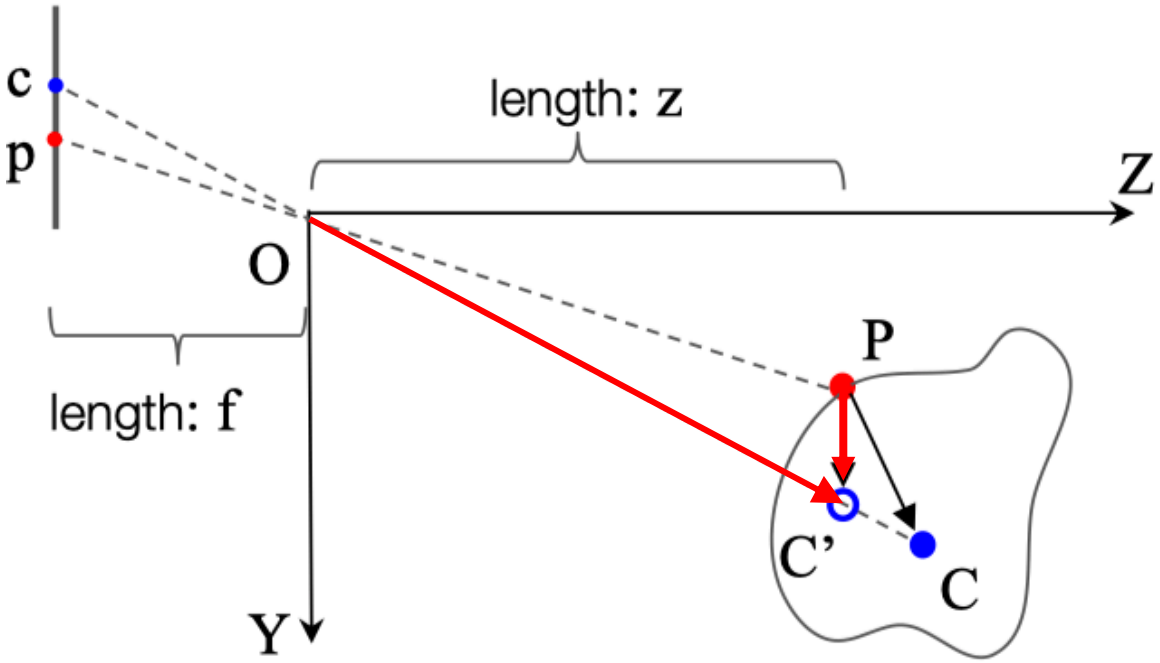
Geometric cues - Ray direction



Ray direction is useful

$$\overrightarrow{OC'} = \overrightarrow{OP} + \overrightarrow{PC'} = \left(x_1 + \frac{\Delta u}{f} z_1, y_1 + \frac{\Delta v}{f} z_1, z_1\right)$$

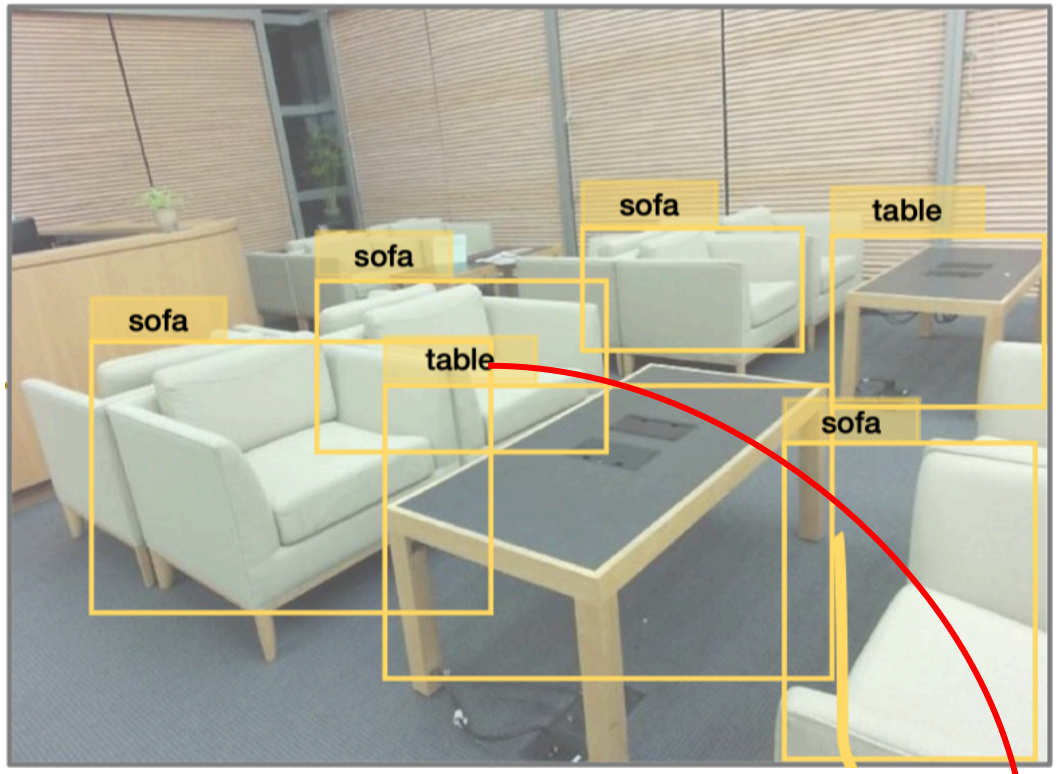
Geometric cues – Final vote



Final vote: pseudo vote + ray direction

$$\left(\frac{\Delta u}{f} z_1, \frac{\Delta v}{f} z_1, \frac{\overrightarrow{OC'}}{|\overrightarrow{OC'}|}\right)$$

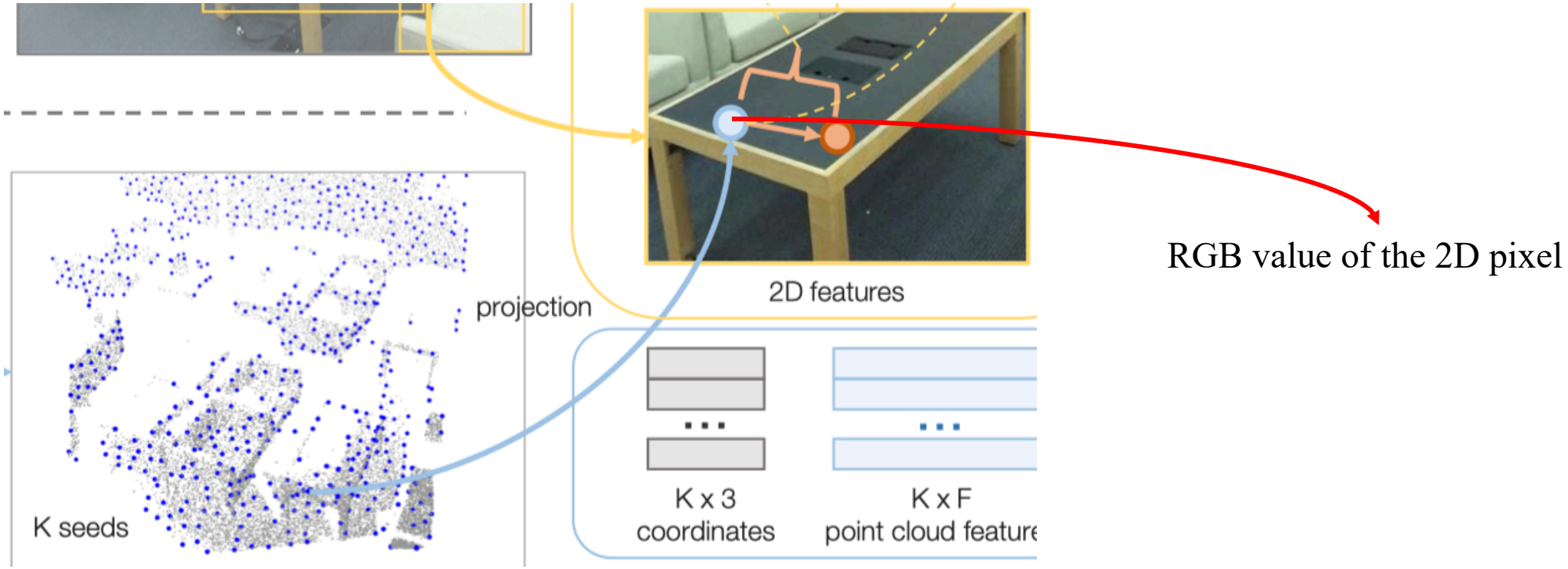
Semantic cues – What's inside the box?



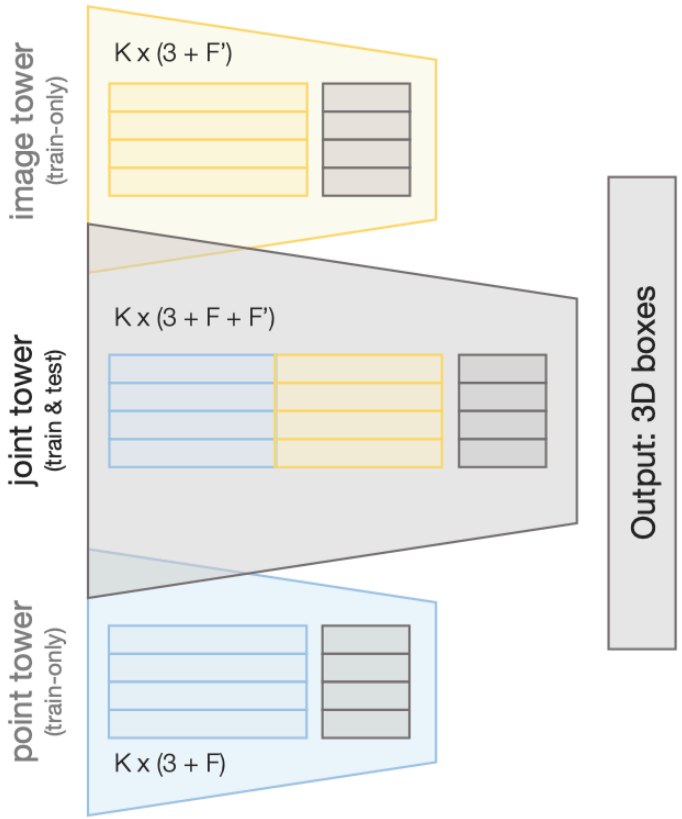
Help to distinguish between classes that are **geometrically very similar** (such as table vs. desk or nightstand vs. dresser).

One-hot class vector with confidence

Texture cues - Low-level, texture-rich representations

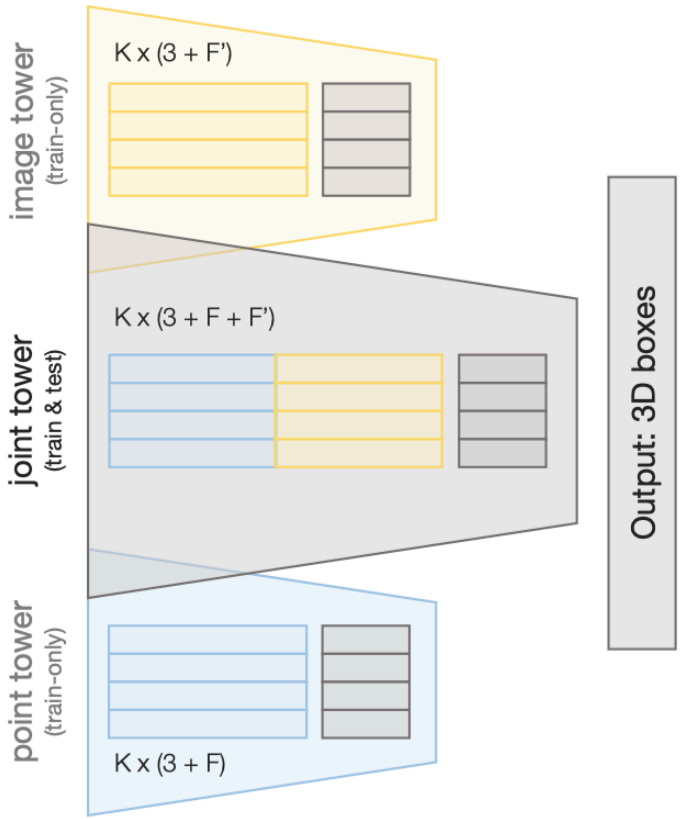


Multi-tower Training



Different modalities may learn to solve the task at **different rates** so, without attention, certain features may dominate the learning and result in overfitting

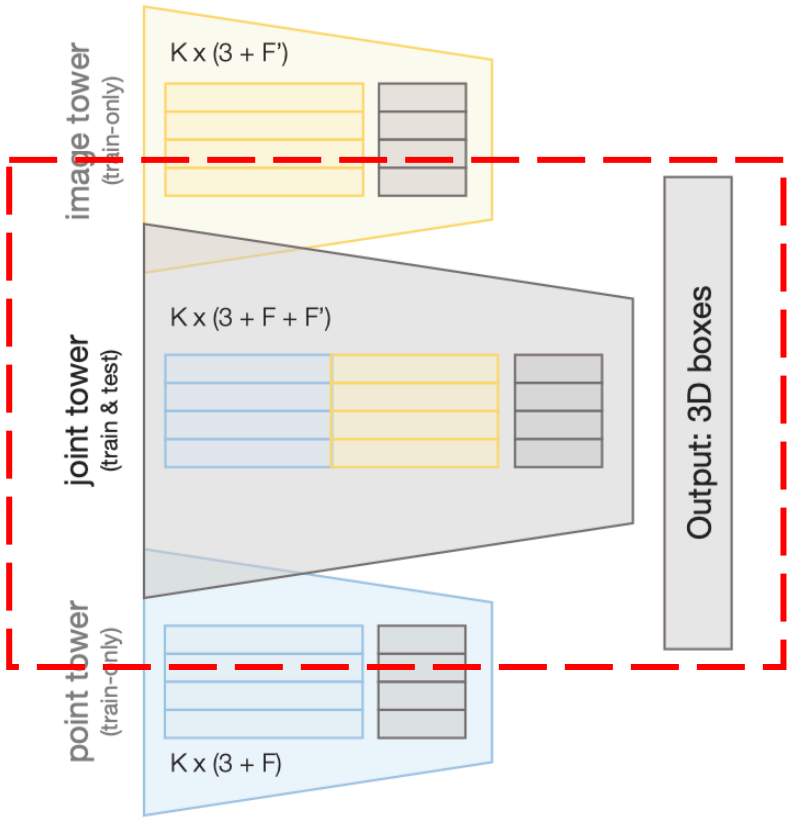
Multi-tower Training



3 towers with their **separate** 3D voting and box proposal network parameters as well as their separate losses.

$$L = w_{\text{img}} L_{\text{img}} + w_{\text{point}} L_{\text{point}} + w_{\text{joint}} L_{\text{joint}}$$

Multi-tower Training



Only joint tower is kept at inference time.

Experiments

methods	RGB	bathtub	bed	bookshelf	chair	desk	dresser	nightstand	sofa	table	toilet	mAP
DSS [43]	✓	44.2	78.8	11.9	61.2	20.5	6.4	15.4	53.5	50.3	78.9	42.1
COG [37]	✓	58.3	63.7	31.8	62.2	**45.2**	15.5	27.4	51.0	**51.3**	70.1	47.6
2D-driven [17]	✓	43.5	64.5	31.4	48.3	27.9	25.9	41.9	50.4	37.0	80.4	45.1
PointFusion [48]	✓	37.3	68.6	37.7	55.1	17.2	23.9	32.3	53.8	31.0	83.8	45.4
F-PointNet [32]	✓	43.3	81.1	33.3	64.2	24.7	32.0	58.1	61.1	51.1	**90.9**	54.0
VOTENET [31]	✗	74.4	83.0	28.8	75.3	22.0	29.8	62.2	64.0	47.3	90.1	57.7
+RGB	✓	70.0	82.8	27.6	73.1	23.2	27.2	60.7	63.7	48.0	86.9	56.3
+region feature	✓	71.7	86.1	34.0	74.7	26.0	34.2	64.3	66.5	49.7	88.4	59.6
IMVOTENET	✓	**75.9**	**87.6**	**41.3**	**76.7**	28.7	**41.4**	**69.9**	**70.7**	51.1	90.5	**63.4**

Table 1. **3D object detection results on SUN RGB-D v1 val set.** Evaluation metric is average precision with 3D IoU threshold 0.25 as proposed by [41]. Note that both COG [37] and 2D-driven [17] use room layout context to boost performance. The evaluation is on the SUN RGB-D v1 data for fair comparisons.

Experiments

geometric cues		mAP
2D vote	ray angle	
✓	✓	63.4
✓	✗	62.2
✗	✗	61.2

(a) Ablation studies on 2D **geometric** cues. 2D vote means the lifted 2D vote (2-dim) as in Eq. 6 and ray angle means the direction of $\overrightarrow{OC'}$ (3-dim). Both geometric cues helped our model.

semantic cues		mAP
region feature	# dims	
one-hot score	10	63.4
RoI [36]	64	62.4
	1024	59.5
✗	-	58.9

(b) Ablation studies on 2D **semantic** cues. Different region features are experimented. This includes simple one-hot class score vector and rich RoI features. The former (default) works best.

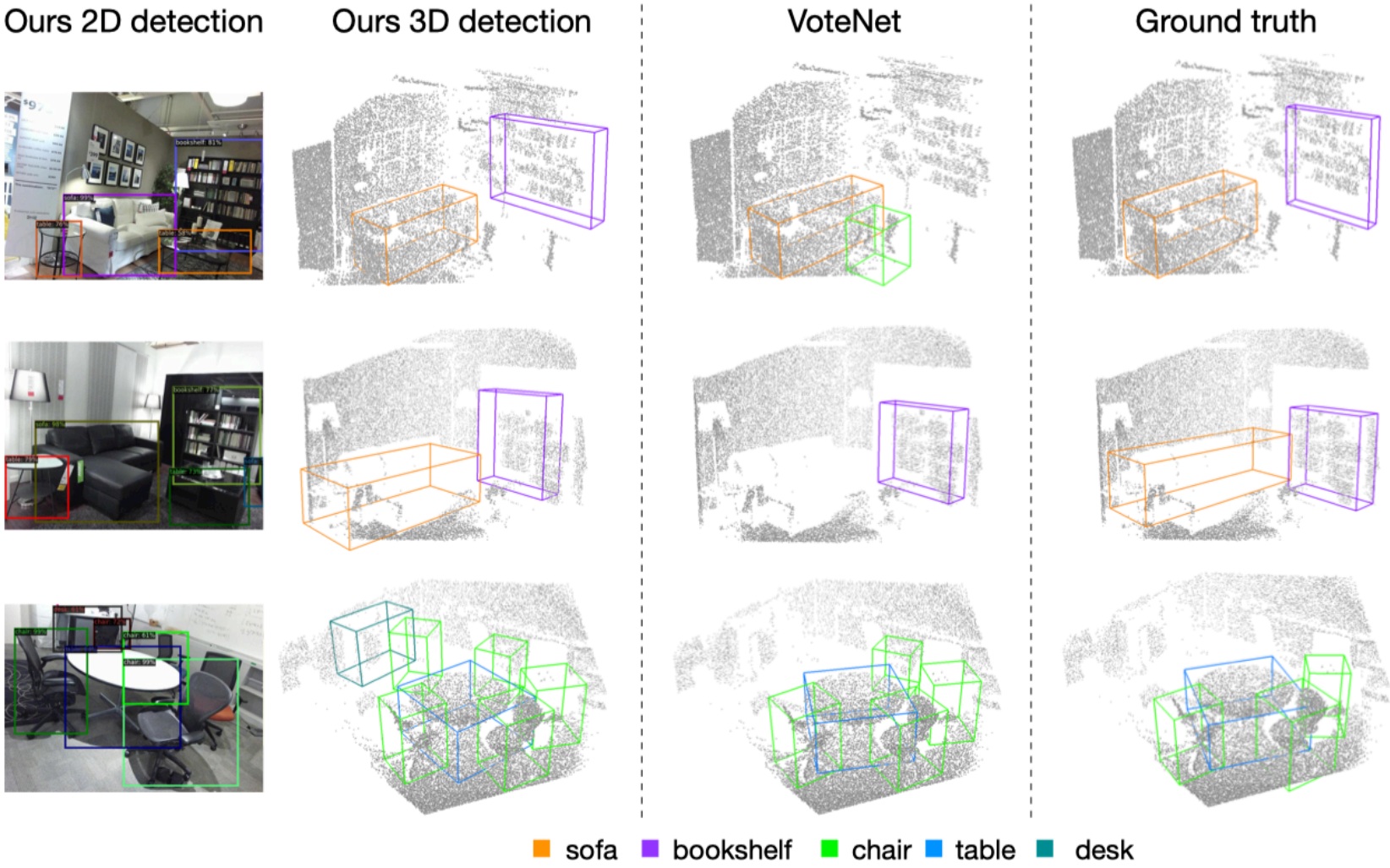
texture cues		mAP
pixel feature	# dims	
RGB	3	63.4
FPN-P_2 [21]	256	62.0
FPN-P_3	256	62.0
✗	-	62.4

(c) Ablation studies on 2D **texture** cues. We experiment with different pixel-level features including RGB values (default) and learned representations from the feature pyramid.

tower weights			mAP		
w_{img}	w_{point}	w_{joint}	image	point cloud	joint
-	-	-	46.8	57.4	62.1
0.1	0.8	0.1	**46.9**	57.8	62.7
0.8	0.1	0.1	46.8	**58.2**	63.3
0.1	0.1	0.8	46.1	56.8	62.7
0.3	0.3	0.4	46.6	57.9	**63.4**

Table 3. Analysis on **multi-tower training**. In the first block we show performance *without* blending in gray. Then we show the setting where each of the tower dominates (0.8) the overall training. Finally we show our default setting where weights are more balanced.

Experiments



Thanks