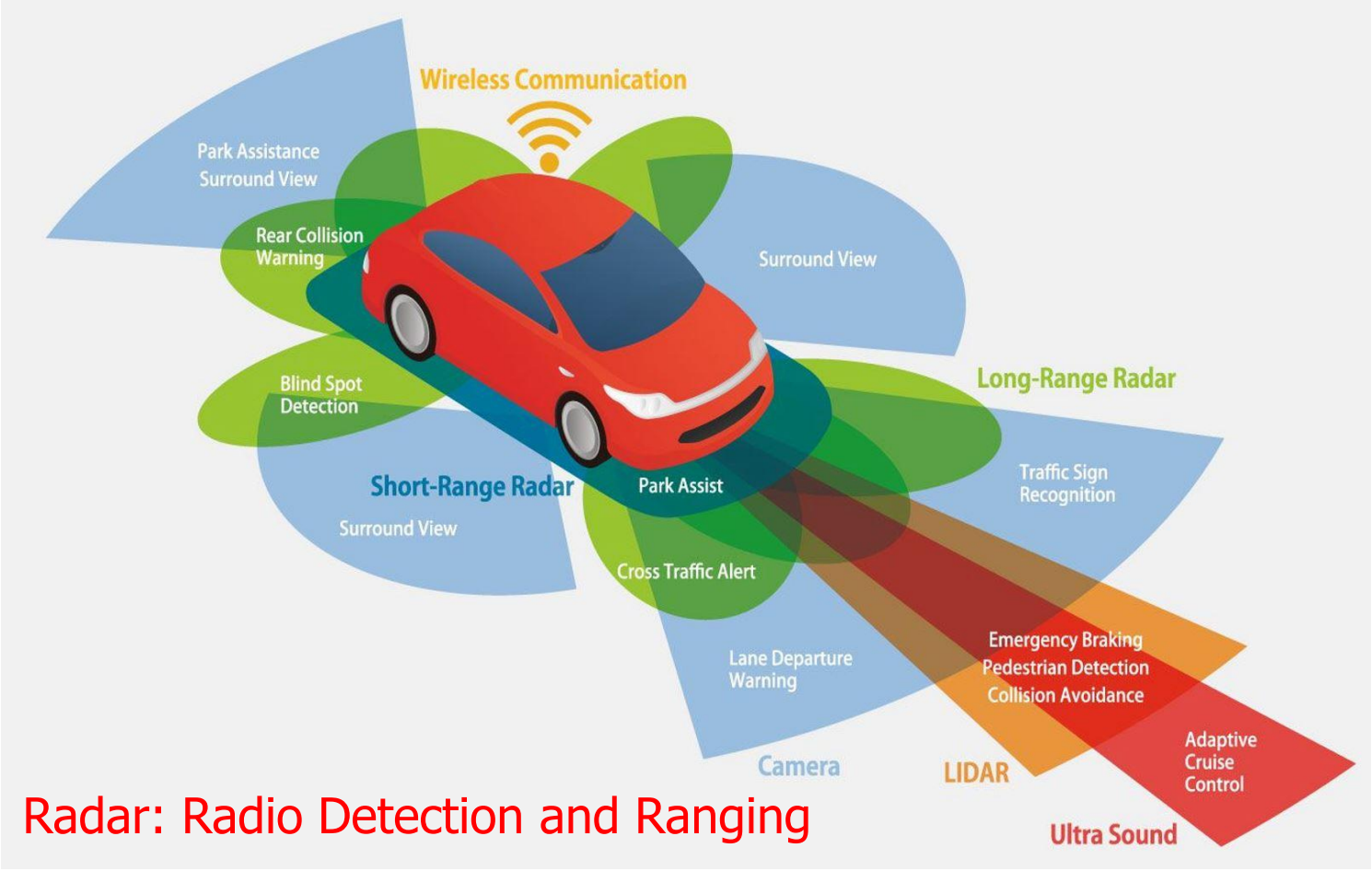
Radar and Lidar Applications

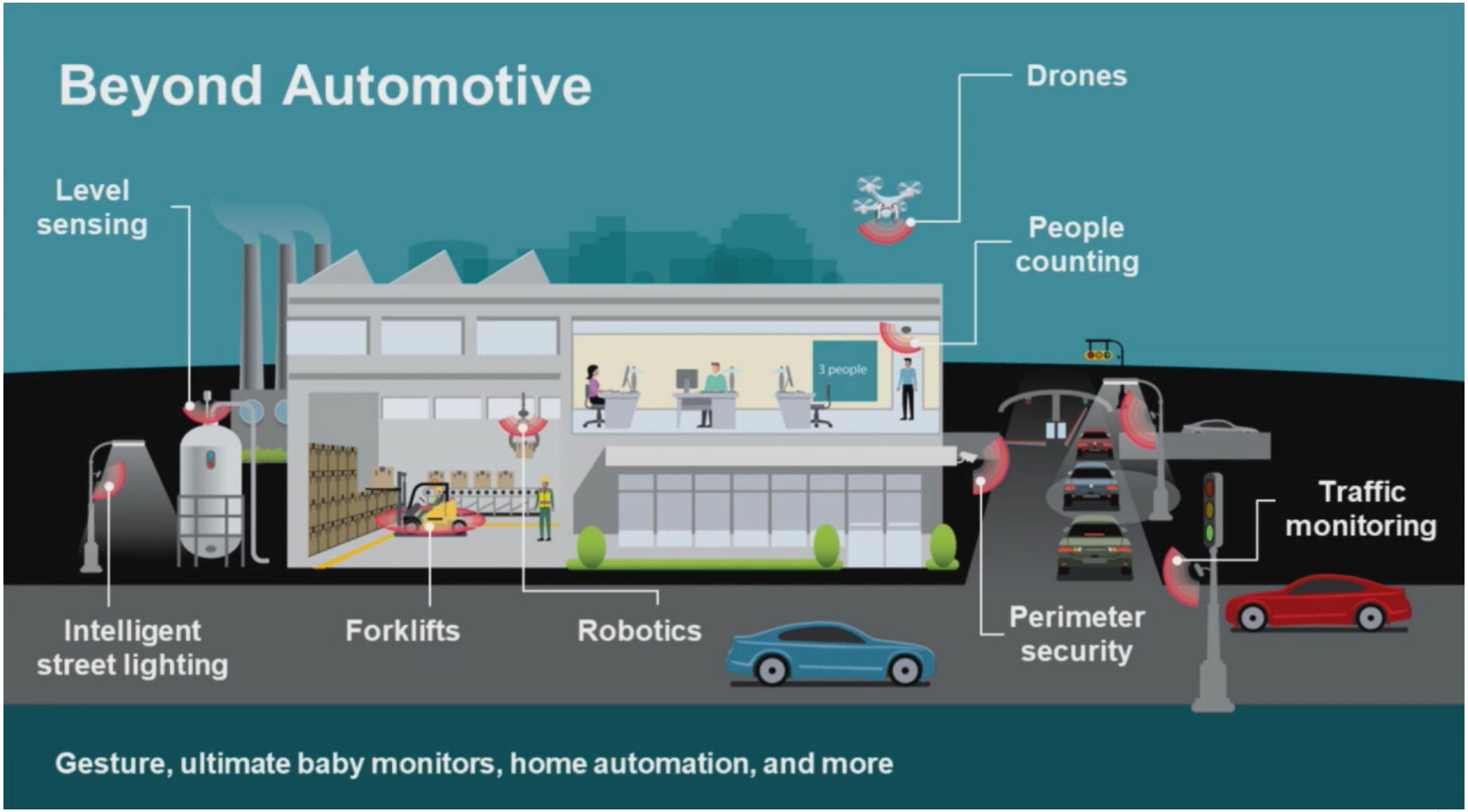
Jenq-Neng Hwang, Professor
Department of Electrical & Computer Engineering
University of Washington, Seattle WA
hwang@uw.edu

Automotive Radar Sensors



Radar: Radio Detection and Ranging

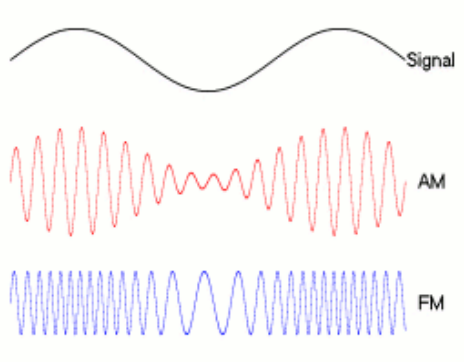
More Applications of Radar



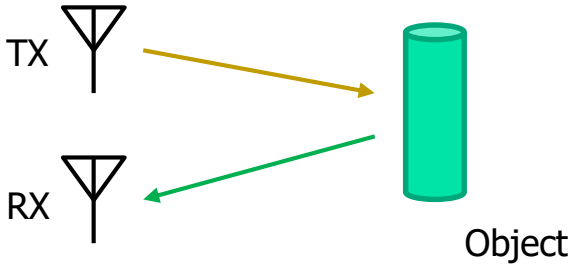
d=

Principles of FMCW Radars

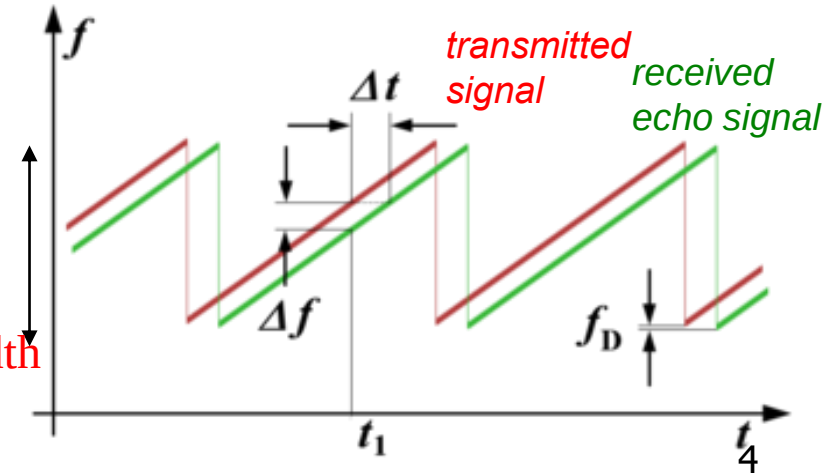
- Ranging with a Frequency-Modulated Continuous-Wave (FMCW) radar (1 TX, 1 RX)



- **AM**: signal amplitude changes with time.
- **FM**: signal frequency changes with time.



The time delay Δt is proportional to the difference of the transmitted and the received signal Δf at any time.

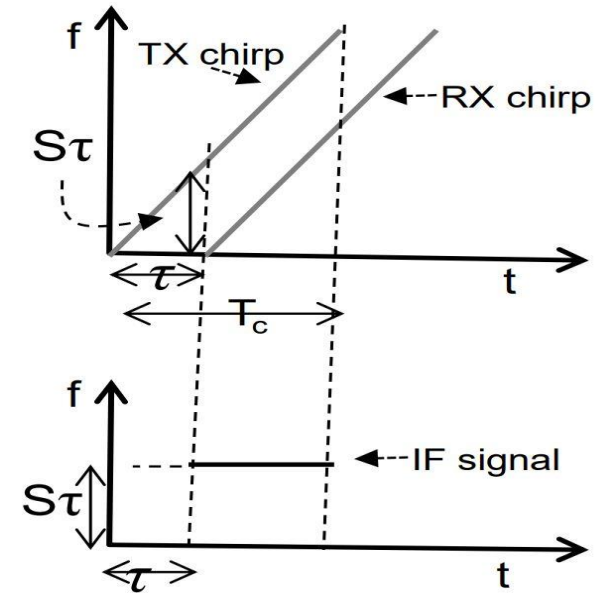


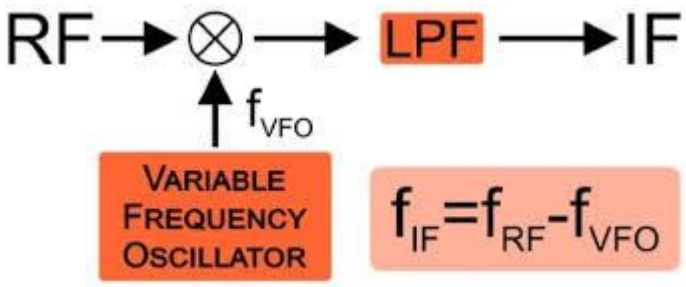
$$\Delta f = s\Delta t = s\frac{2d}{c} \longrightarrow d = \frac{\Delta f c}{2S}$$

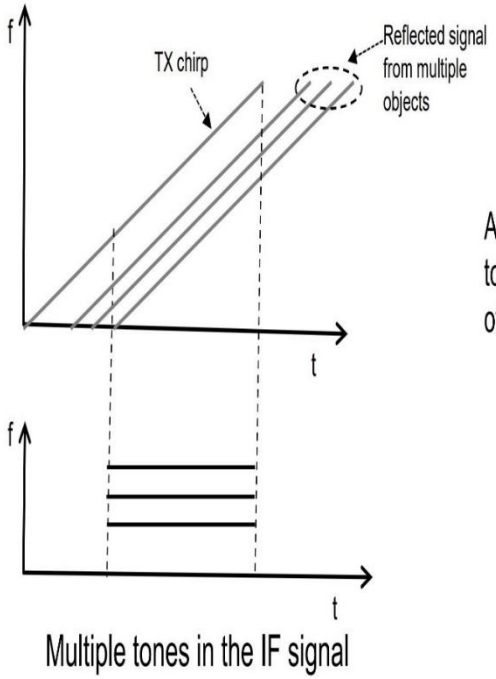
width

s (frequency shift per unit of time)

Range Interpretation

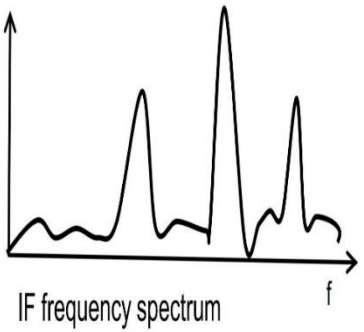






Multiple tones in the IF signal

A frequency spectrum of the IF signal will reveal multiple tones, the frequency of each being proportional to the range of each object from the radar

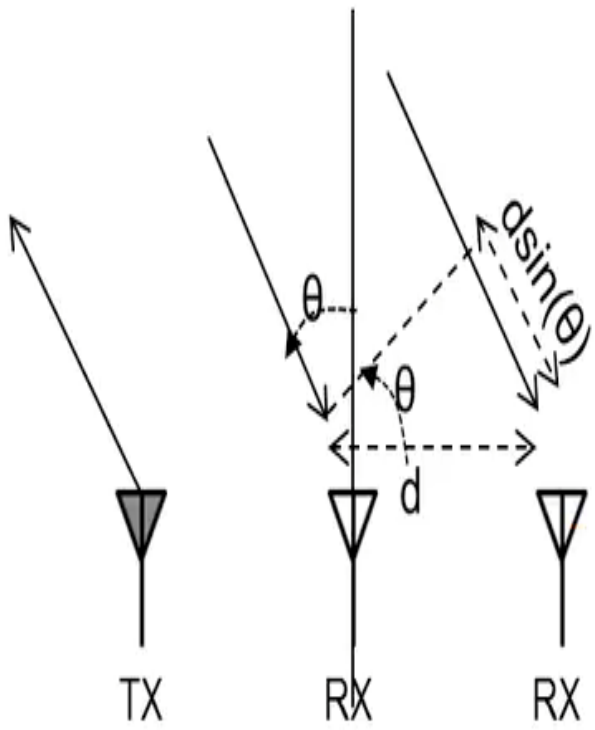


IF frequency spectrum

Frequency difference

Principles of FMCW Radars

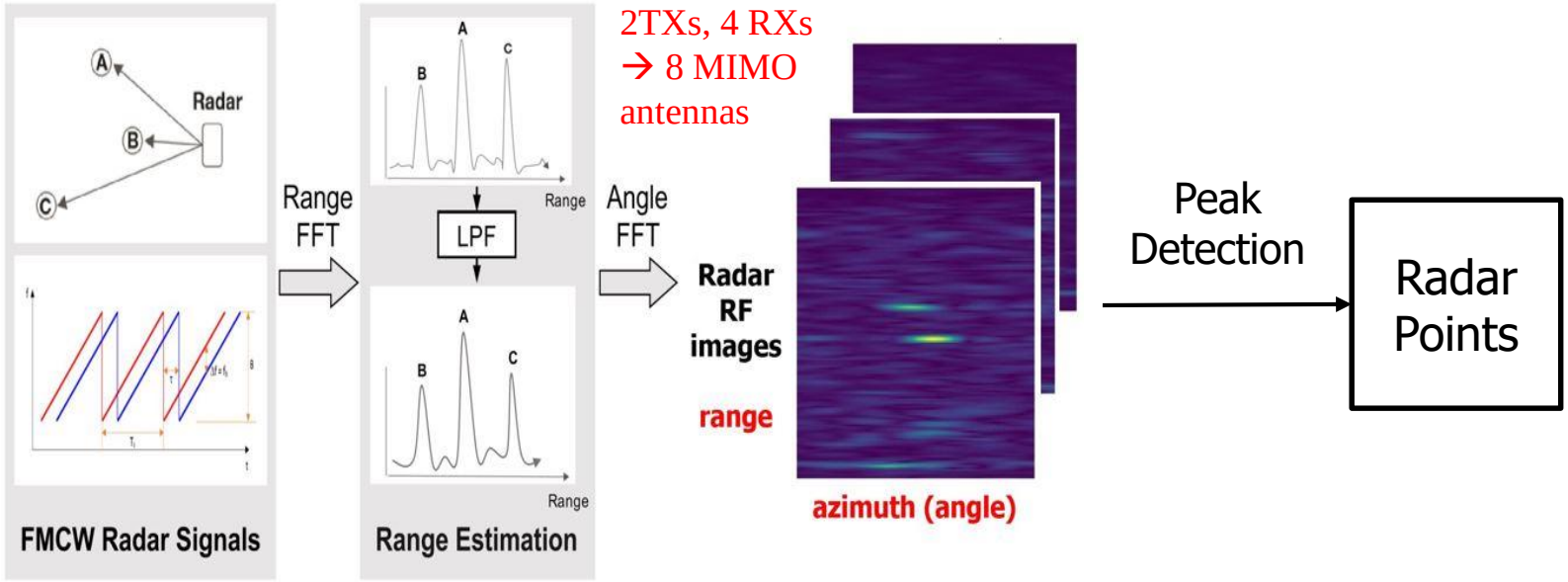
- Angle estimation using multiple RX (1 TX, 2 RX)



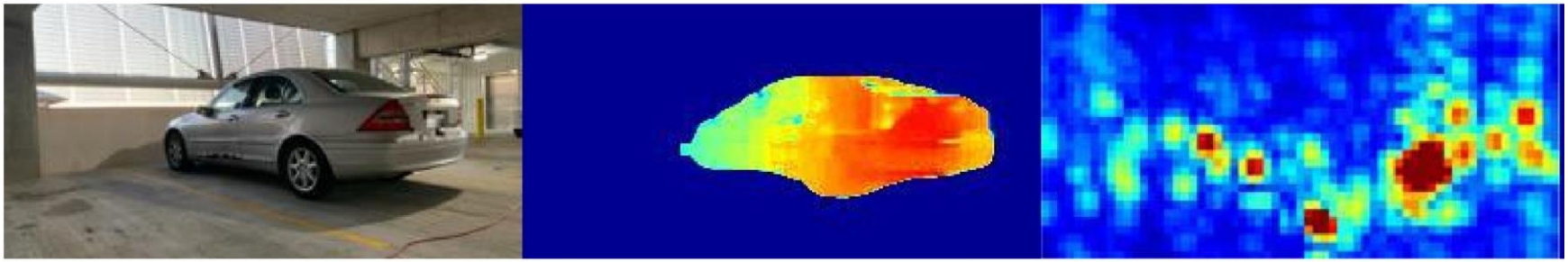
- TX antenna transmits a frame of chirps.
- FFT corresponding to each RX antenna will have peaks about received echo frequency in the same location but with differing phase.
- The measured phase difference ω can be used to estimate the angle of arrival of the target.

$$\omega = \frac{2\pi d\sin(\theta)}{\lambda} \Rightarrow \theta = \sin^{-1}\left(\frac{\lambda\omega}{2\pi d}\right)$$

Radar Radio Frequency (RF) Images

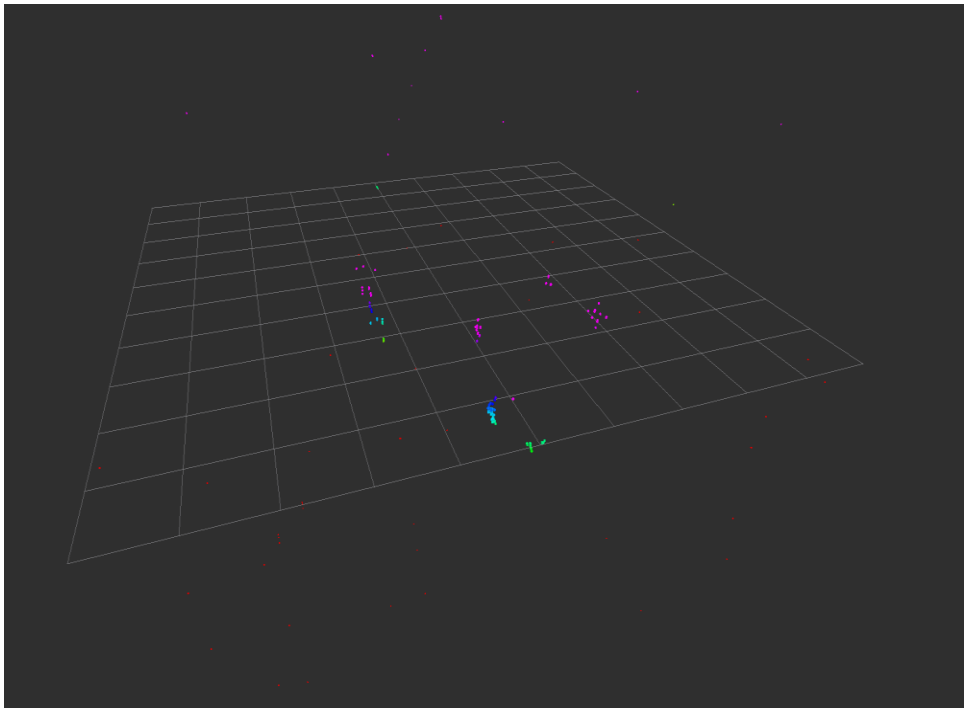


(a) Original scene (b) Ground Truth Depth Map (c) Radar Heatmap Projection

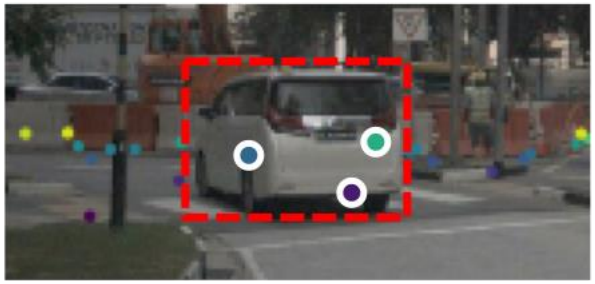


Rrdar Points Data Format

- Strength: Easy to utilize (similar with LiDAR).
- Weakness: Usually very sparse.



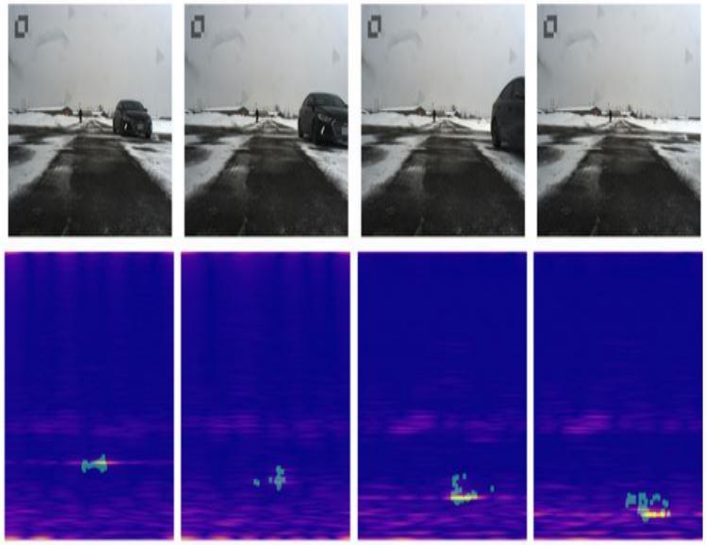
(a) LiDAR points in nuScenes

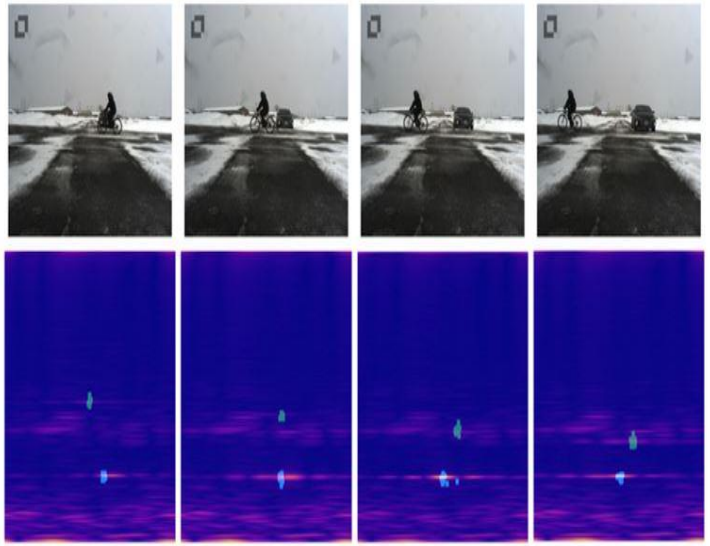


(b) Radar points in nuScenes

RF Image Data Format

- Strength: Contains rich information including location, texture, speed, etc.
- Weakness: Hard to extract semantic features.





Available Radar Datasets

Dataset	Year	Data Format	Annotation	Scale	Public
nuScenes	2019	Radar Points	Human	5.5 hours	Yes
RadarRobotCar	2019	Radar Points	None	280 km	Yes
Qualcomm	2019	RF Images	LiDAR	3 hours	No
ASTYX HIRES2019	2019	Radar Points	Human	546 frames	Yes
CARRARA	2020	RF Images	Camera	21.2 min	No
Xsense.ai	2020	RF Images	LiDAR	34.2 min	No
CRUW (Ours)	**2020**	**RF Images**	**Camera**	**3.5 hours**	**Yes**

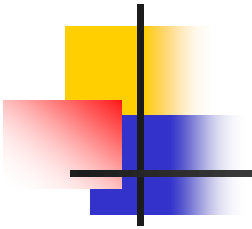
https://www.cruwdataset.org/

Challenges of mmWave Radars: 60-77GHz

- Vulnerability: vulnerable to certain atmospheric and meteorological phenomena
- Over-sensitivity: There have been cases where the alarm for the program activated even when there was no real threat (false positives).
- Limited Range: limited in terms of accuracy and range.
- External Interferences: presence of electrical towers or electromagnetic hotspots can sometimes cause interference with the machine and, in some cases, cause it to malfunction.

Automotive mmWave radar sensors

Industrial mmWave radar sensors

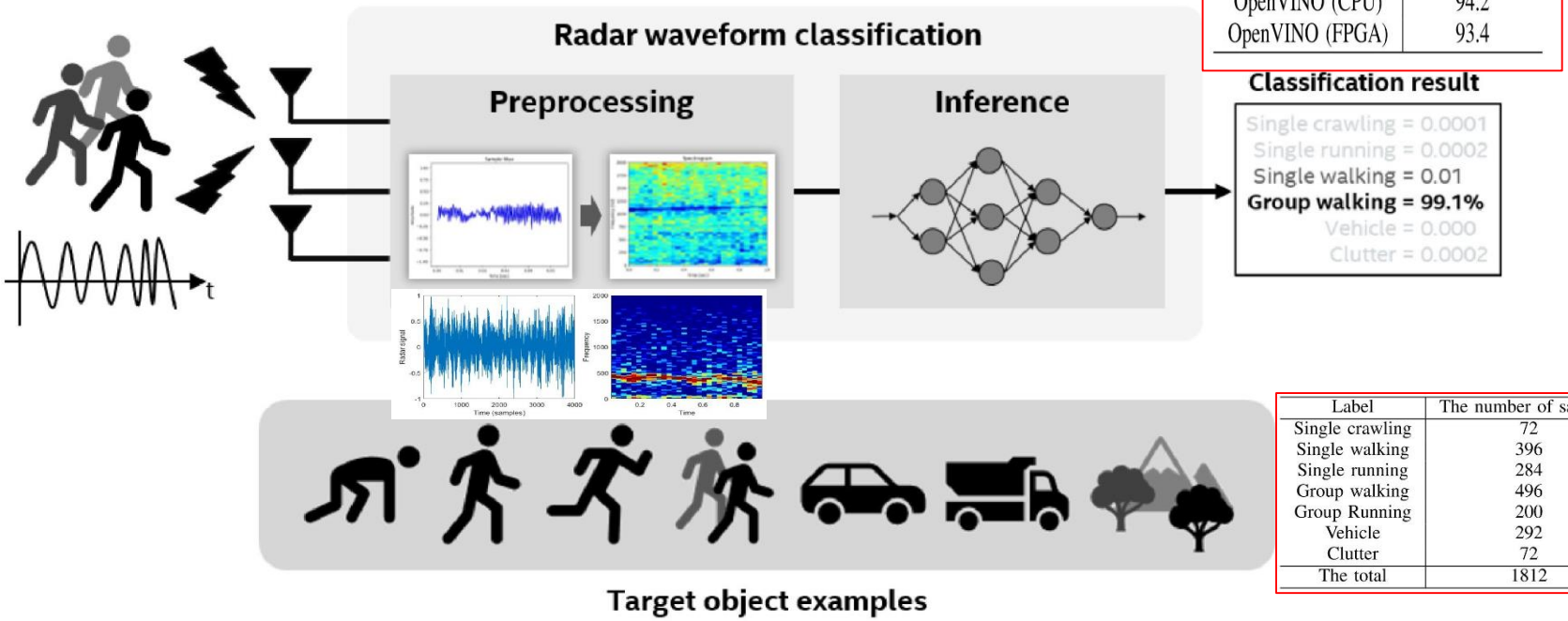


Radar Object Classification

Radar Object Classification

- 1-sec (4000 samples), 256->1024 FFT, 128 stride, 513x96 spectrogram images, AlexNet for classifier.

Platform	Accuracy (%)
Caffe (CPU)	94.2
OpenVINO (CPU)	94.2
OpenVINO (FPGA)	93.4



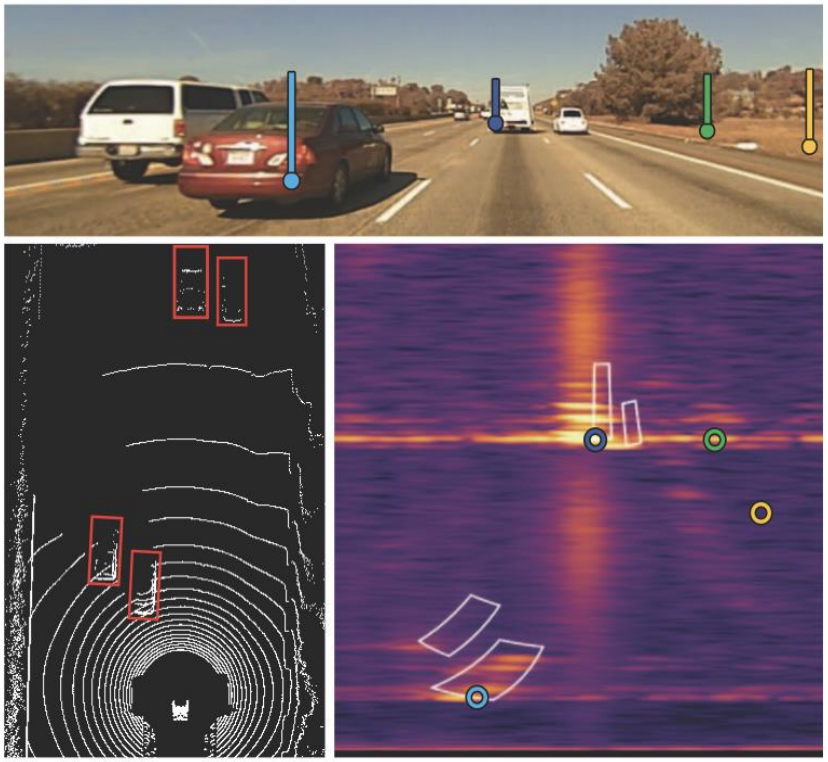
Label	The number of samples
Single crawling	72
Single walking	396
Single running	284
Group walking	496
Group Running	200
Vehicle	292
Clutter	72
The total	1812

Dajung Lee, et al., "Radar-based Object Classification Using An Artificial Neural Network," IEEE NAECON 2019

Radar Object Detection

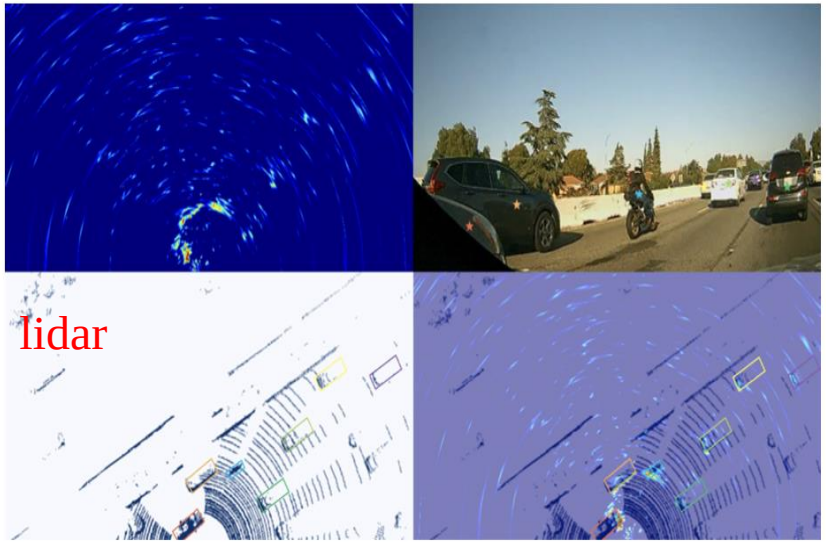
Lidar Annotation for Radar

- **LiDAR provide detection ground truths** (most frequently used)



lidar radar

radar

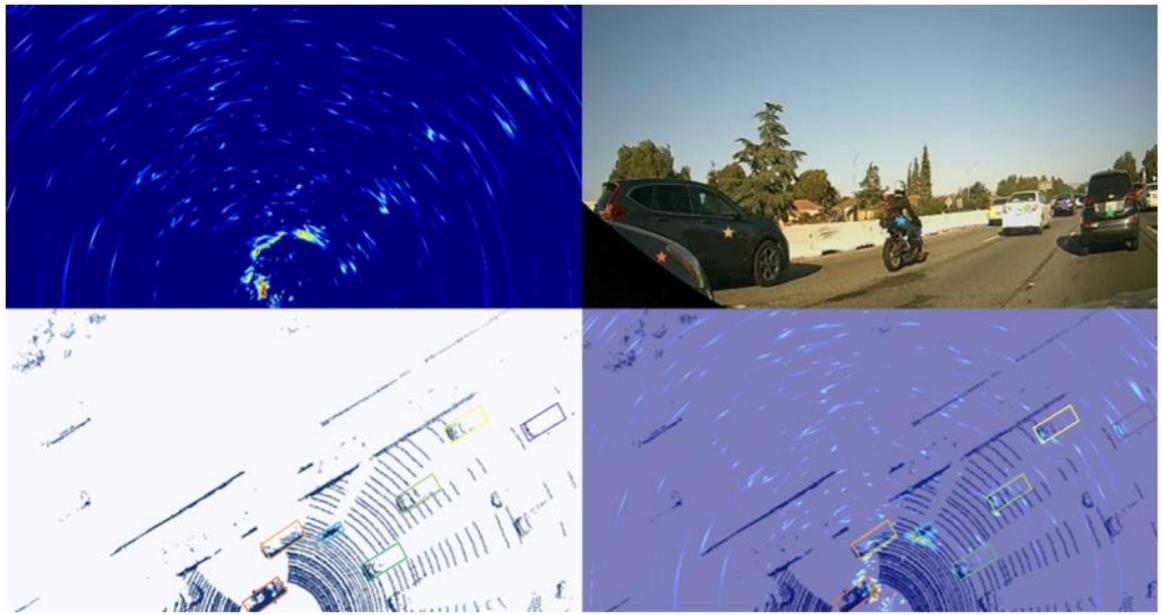


lidar

← Example in Qualcomm dataset.
↑ Example in Xsense.ai dataset.

Oriented BBox Radar Object Detection with Variance

- Radar is mounted at the front left corner at bumper height, thus most of the object bboxes are oriented.



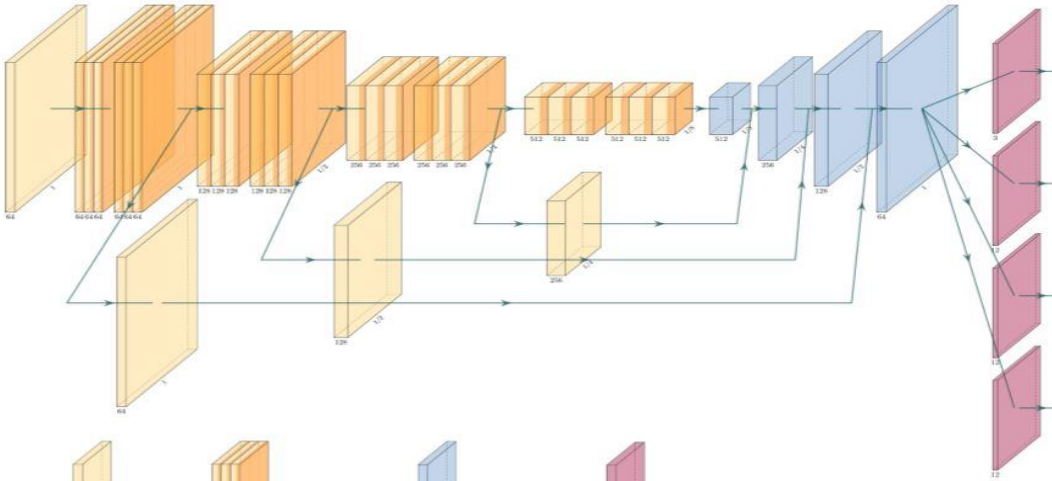
Xu Dong, et al., "Probabilistic Oriented Object Detection in Automotive Radar," IEEE/CVF CVPRW 2020

Oriented BBox Radar Object Detection with Variance

- The network predicts the individual variance for each of the position predictions (x_o,y_o), the size predictions (w_o,h_o), and the orientation predictions ($cos\theta_o$,$sin\theta_o$).

$$L_{tot} = L_{obj} + w_0 \sum_a L_a$$

focal loss



bbox localization predictions (x, y, w, h, θ) also variance

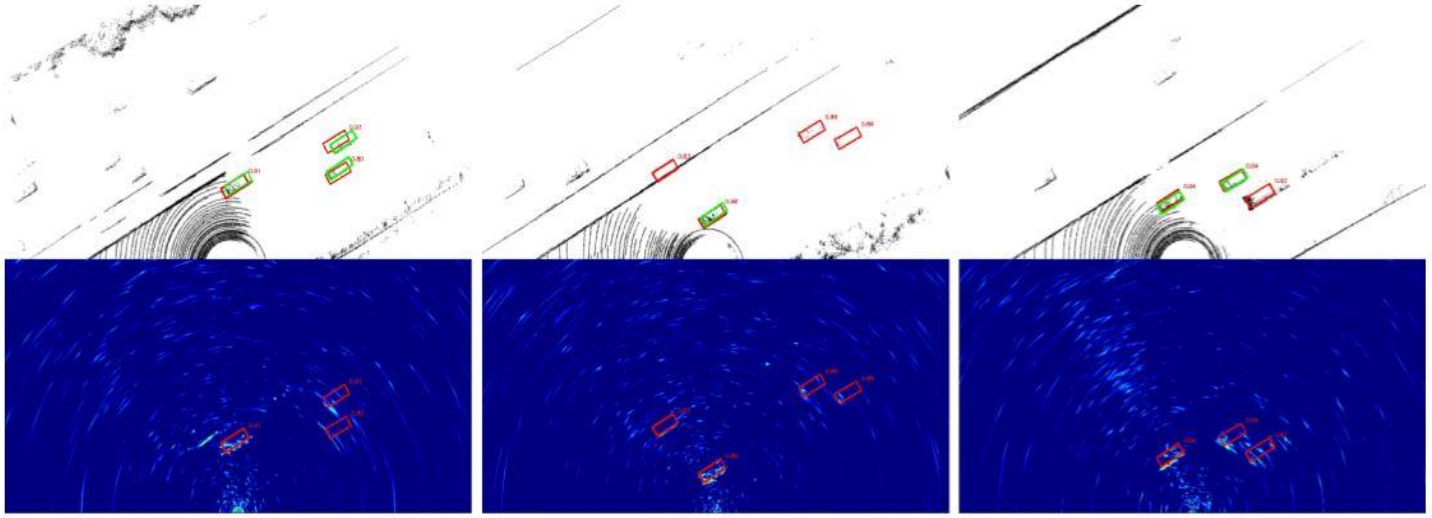
$$\min\ L_a = \sum_{i=0}^{N} \left(\frac{1}{\sigma_a} SL_1(a^{pred} - a^{gt}) + \log \sigma_a \right)$$

Smoothed L_1 (SL_1)

where a represents the prediction of x_o, y_o, w_o, h_o, $\cos\theta_o$, and $\sin\theta_o$; σ_a is the variance of its prediction; and N is the total number of bounding boxes in a image.

Oriented BBox Radar Object Detection with Variance

- Adding variance (uncertainty) prediction is able to improve the performance of the object detector, e.g., under more stringent IoU threshold of 0.7. critical for autonomous driving



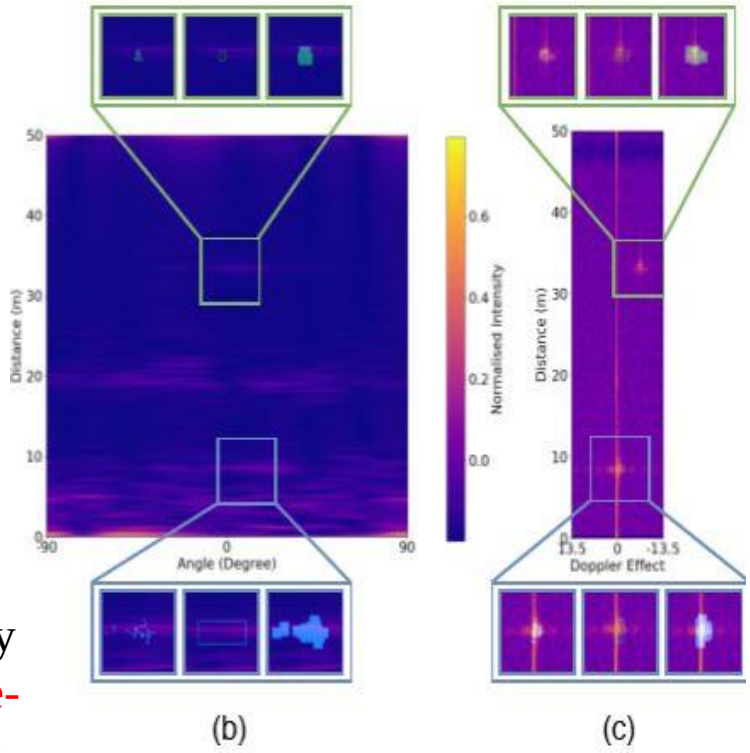
Regression	AP@IoU = 0.3/0.5/0.7
w/o variance	76.81/63.57/31.53
w/ variance	77.28/65.36/34.33

(no distinction of object types)

A. Quaknine, et al., “CARRADA: Camera and Automotive Radar with Range-Angle-Doppler Annotations,” ICPR 2020

Annotation: Point Correspondence between Camera RGB Images and Radar RF Images

(a)



(b) (c)

(a) A pedestrian at approximately 8m from the sensors and a car in the background at approximately 33m; (b)(c) Radar signal at the same instant in range-angle and range-Doppler representation respectively

3D Localization: Camera Perspective Transform

projection matrix $P_w \rightarrow P_c \rightarrow P_{uv}$

$$[u, v, 1]^T \sim \mathbf{F} \cdot [X, Y, Z, 1]^T$$

$$\mathbf{F} = \mathbf{K} \cdot [\mathbf{R}|\mathbf{t}]$$

$$\mathbf{R} = \mathbf{R}_Z \mathbf{R}_X \mathbf{R}_Y$$

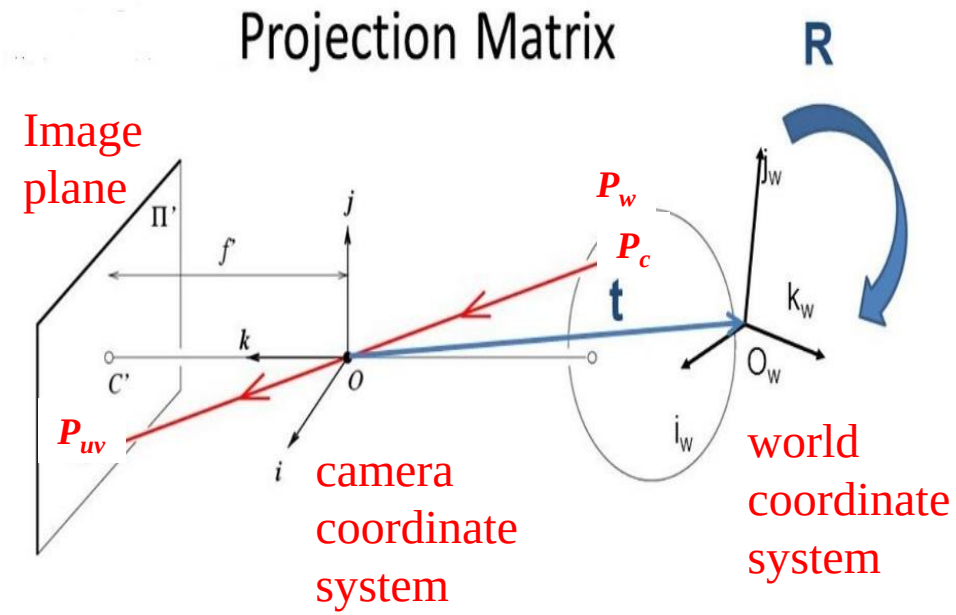
Extrinsic: rotation matrix

$$\mathbf{t} = \begin{bmatrix} t_X \\ t_Y \\ t_Z \end{bmatrix}$$

Extrinsic: translation vector

intrinsic parameter matrix

$$\mathbf{K} = \begin{bmatrix} f_u & s & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{bmatrix}$$



[Peng *et al.*, Pattern Recognition'09]

2D to 3D Backprojection

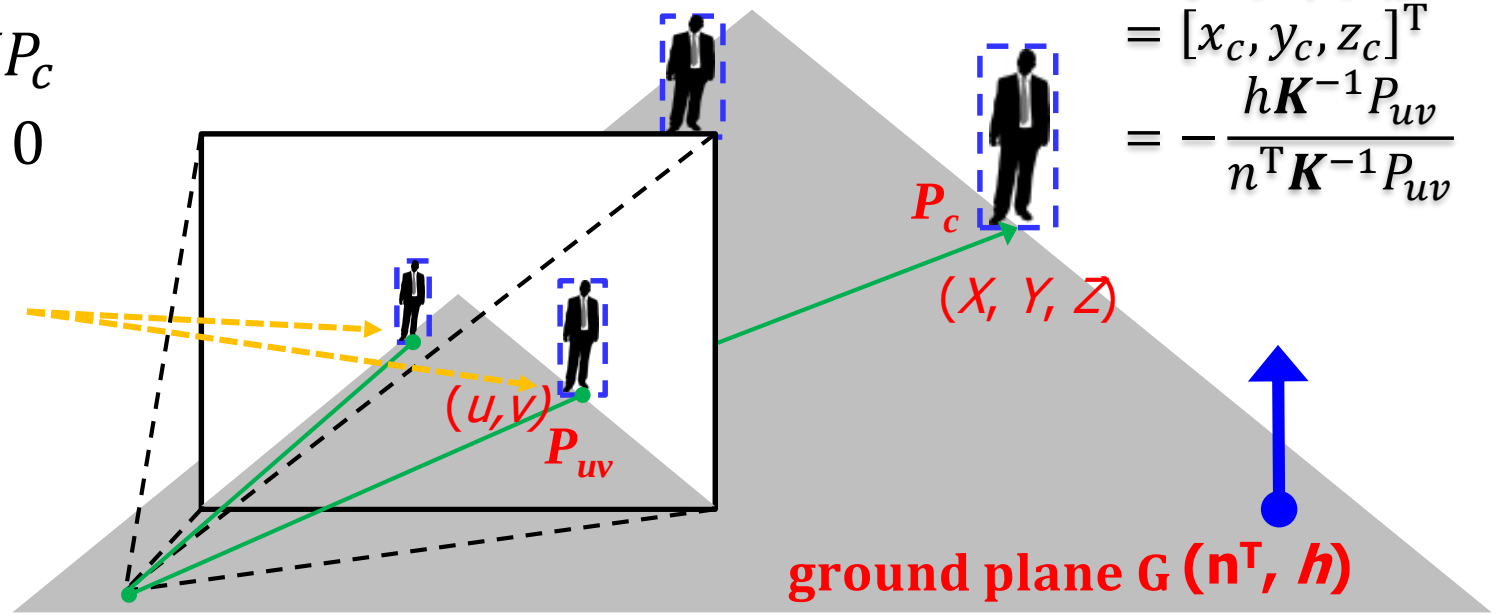
- Recover from 2D P_{uv} in image plan to 3D P_c in camera coordinate:

$$[u, v, 1]^T \sim \mathbf{F} \cdot [X, Y, Z, 1]^T, \qquad P_{uv} = \mathbf{K} \cdot [\mathbf{R}|\mathbf{t}] P_w = \boldsymbol{F}_{\mathbf{3\times4}} P_w = \mathbf{K} \cdot P_c$$

$$\begin{cases} P_{uv} = \boldsymbol{K} P_c \\ G^{\mathrm{T}} P_c = 0 \end{cases}$$

$$P_c = [\mathbf{R}|\mathbf{t}] P_w = [x_c, y_c, z_c]^{\mathrm{T}} = -\frac{h\boldsymbol{K}^{-1} P_{uv}}{n^{\mathrm{T}} \boldsymbol{K}^{-1} P_{uv}}$$

limiting the point on the ground plan



Association of Camera and Radar Objects

- Step 1: From vision to physical measurements
- Mask R-CNN, project center of mass of each segmented instance.

$$s\ \mathbf{p} = A\ B\ \mathbf{c}, \quad \mathbf{p} = [p_x, p_y, 1]^\top \text{ and } \mathbf{c} = [c_x, c_y, c_z, 1]^\top$$

- Fixed environment: scaling factor s is fixed.
- Track the detected instances (δt = 1 sec).

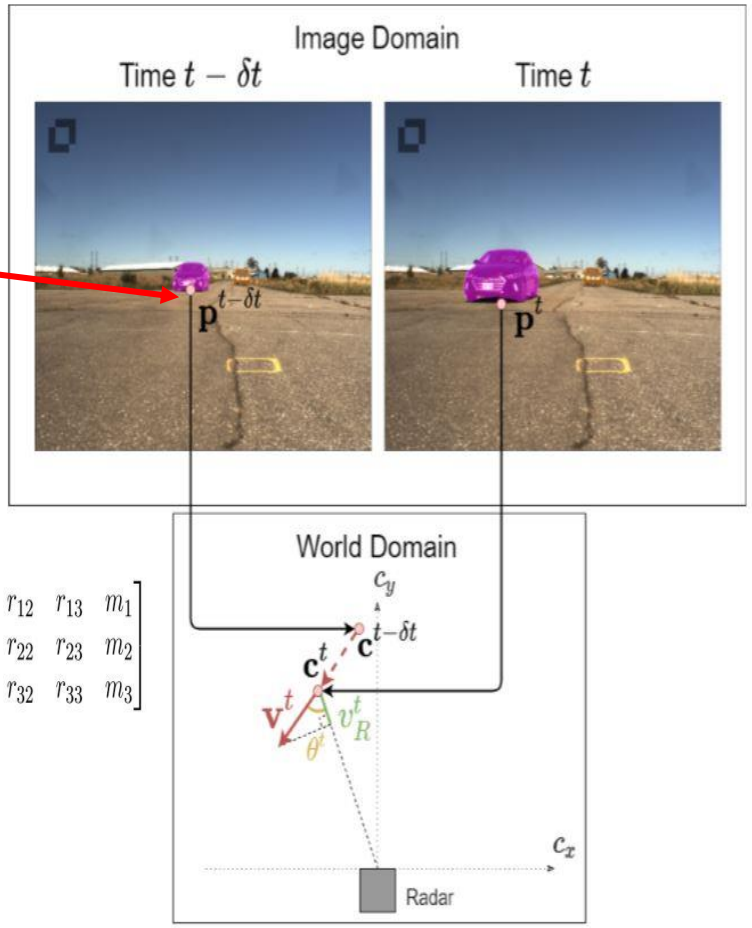
$$\mathbf{v}^t = \mathbf{c}^t - \mathbf{c}^{t-\delta t},$$

$$v_R^t = \cos\theta^t\ \|\mathbf{v}^t\|,$$

$$A = \begin{bmatrix} f_x & 0 & a_x \\ 0 & f_y & a_y \\ 0 & 0 & 1 \end{bmatrix}, B = \begin{bmatrix} r_{11} & r_{12} & r_{13} & m_1 \\ r_{21} & r_{22} & r_{23} & m_2 \\ r_{31} & r_{32} & r_{33} & m_3 \end{bmatrix}$$

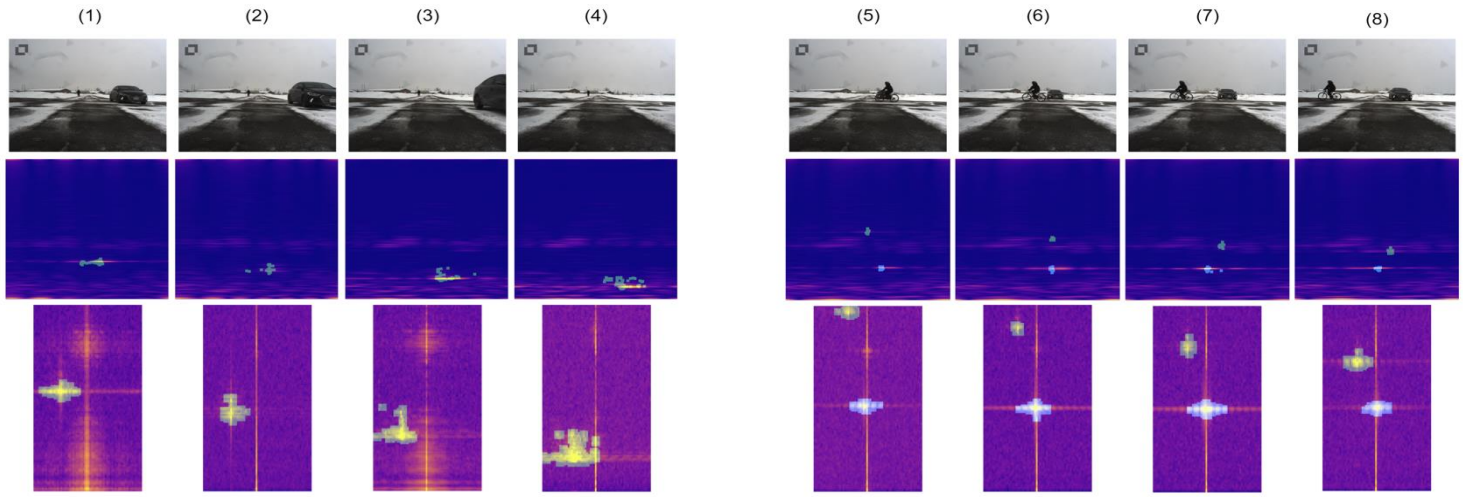
- Each instance detected is characterized by (location and speed)

$$I^t = [\mathbf{c}^t, v_R^t]^\top.$$



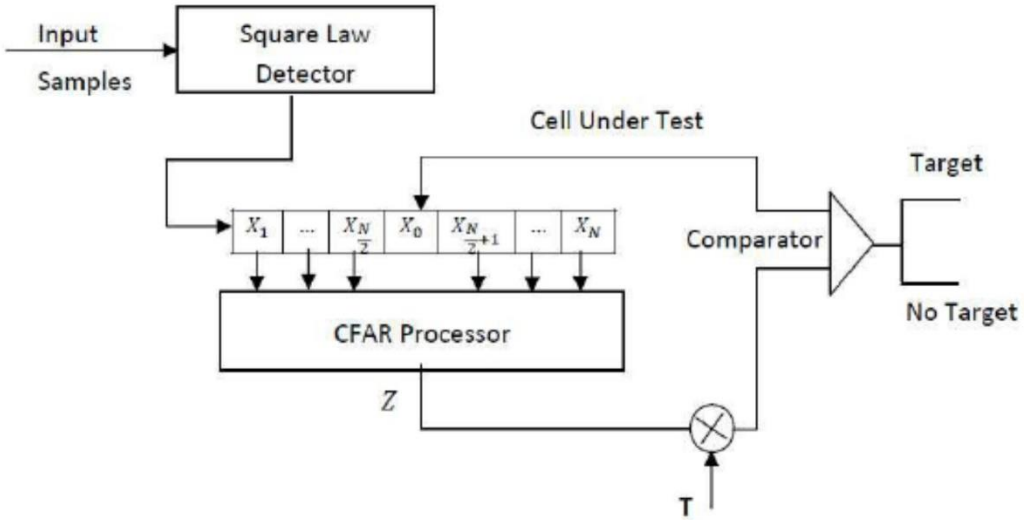
Annotation for Radar

- Step 2: 3D DoA-Doppler Point Cloud Clustering:
  - Use CFAR to detect peaks in DoA-Doppler space.
  - Cluster 3D peaks using Mean Shift clustering.
  - Assign the cluster to the closest detected and tracked object closest I^t
- Step 3: Projections and annotations:
  - Project to range-angle and range-Doppler representations.
  - Sparse annotations to dense segmentation annotations (morphological dilation, circular SE).

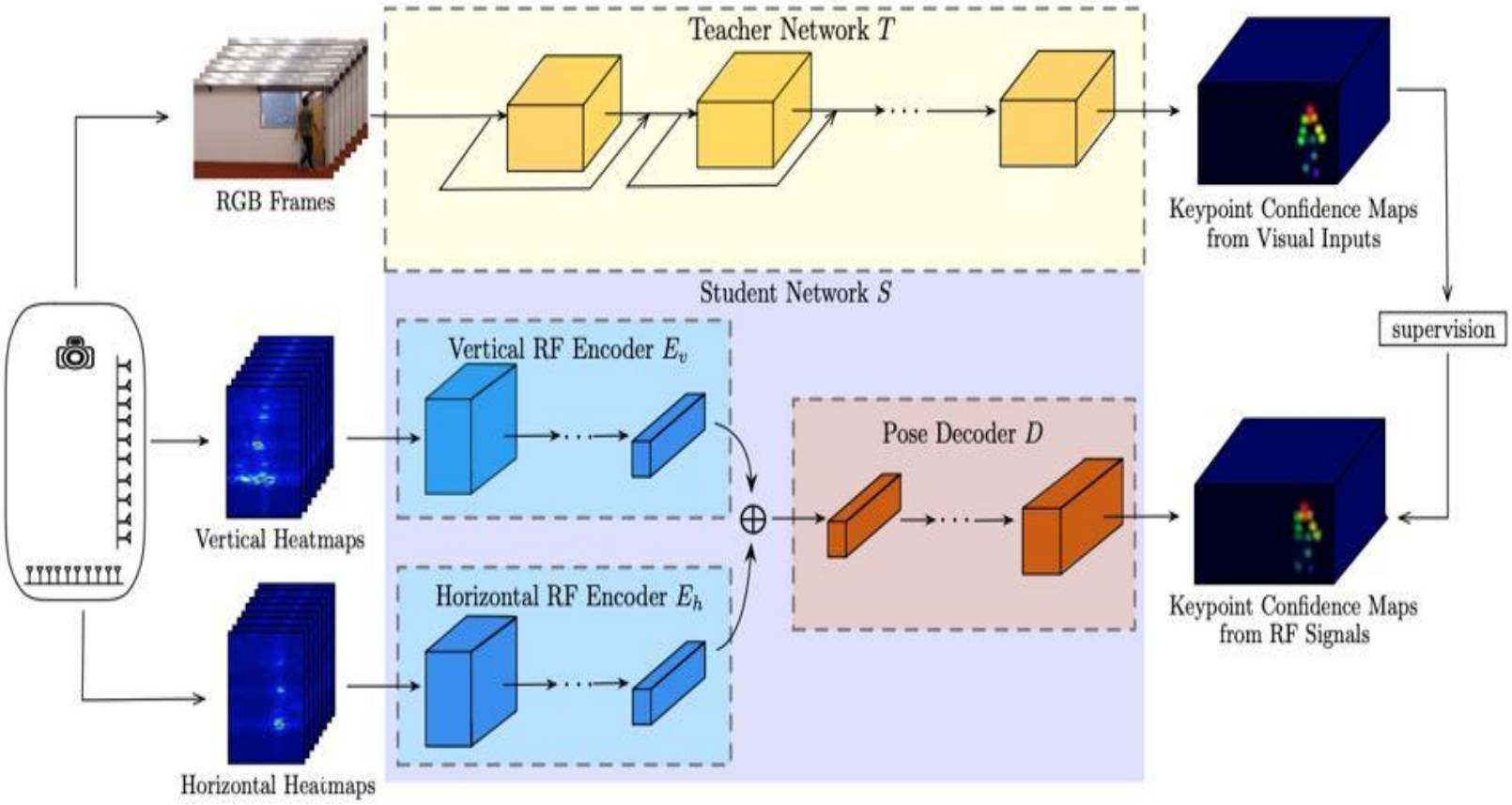


CFAR

- Constant false alarm rate (CFAR) detector implements an adaptive algorithm for detecting target signal reflections in the presence of background noise, clutter and interference, and provides an estimate of the target distance.
- Detection threshold too high, very few false alarms, inhibits detection of valid targets. If the threshold is set too low, the large number of false alarms.
- CFAR detectors employ a "background averaging" technique to dynamically adapt the decision threshold.



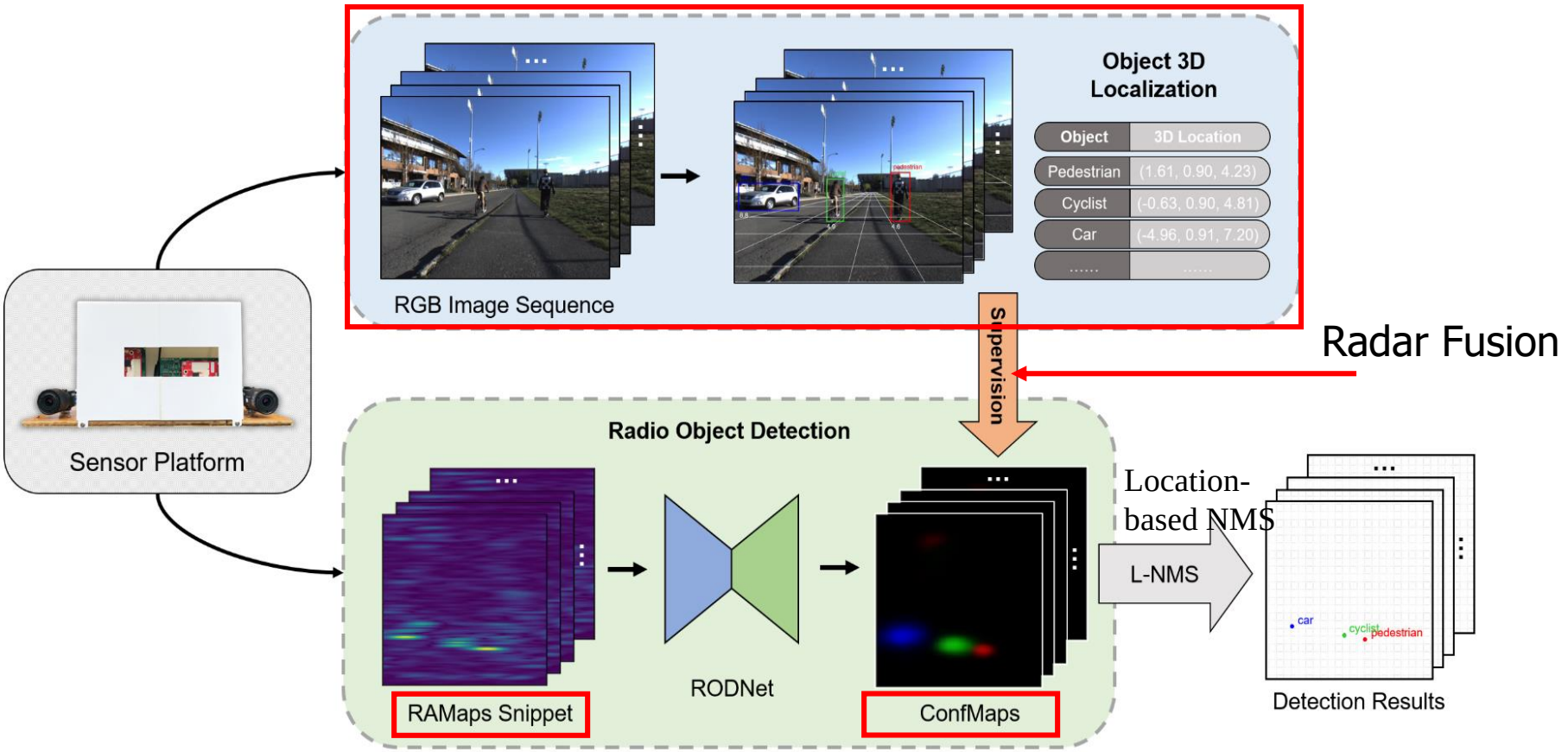
Cross-Modal Supervision: Fully Automatic Annotation



Mingmin Zhao, "Through-Wall Human Pose Estimation Using Radio Signals," CVPR 2018

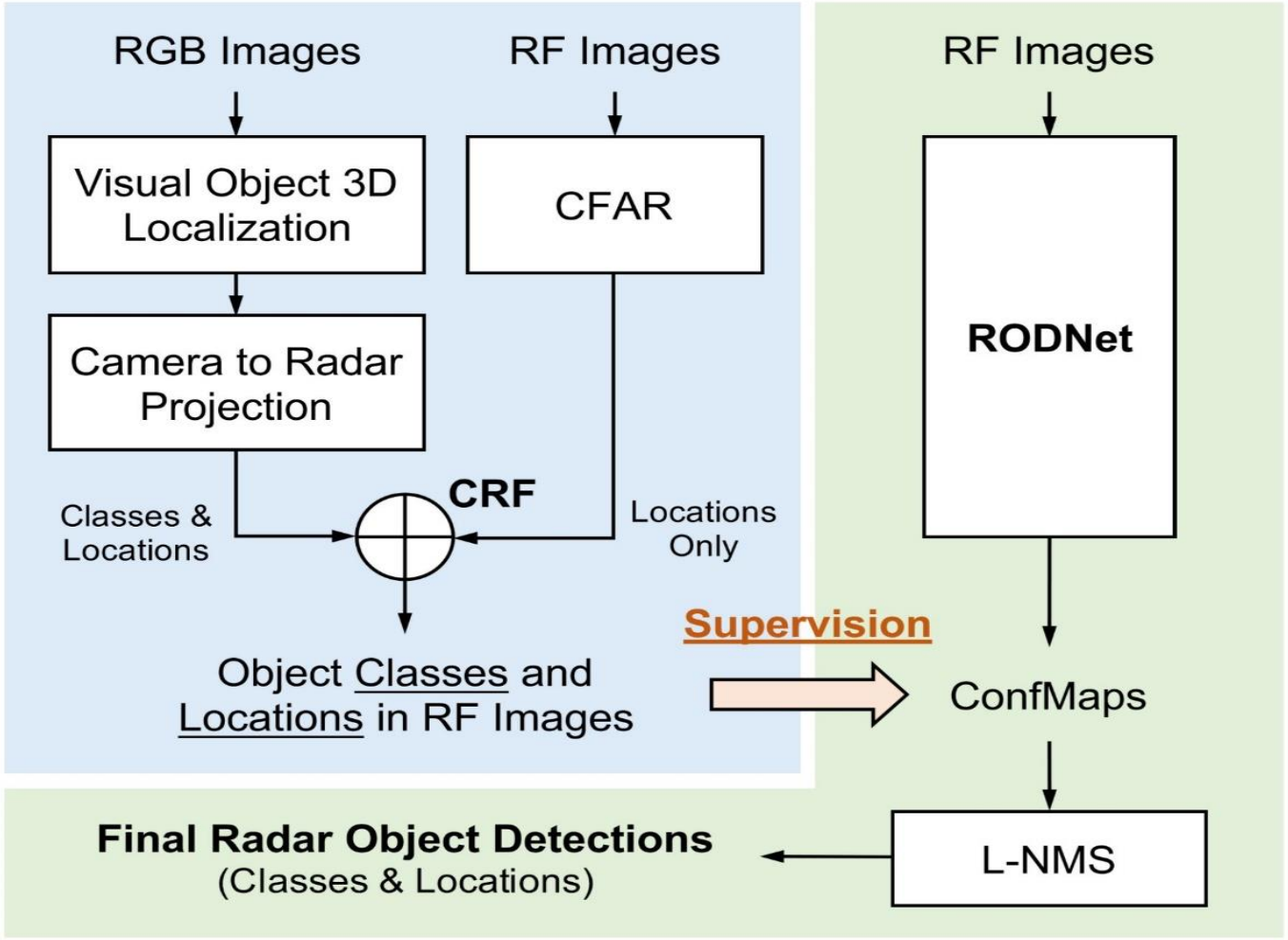
Cross-Modal Supervision: Fully Automatic Annotation

Radar Object Detection Network (RODNet)

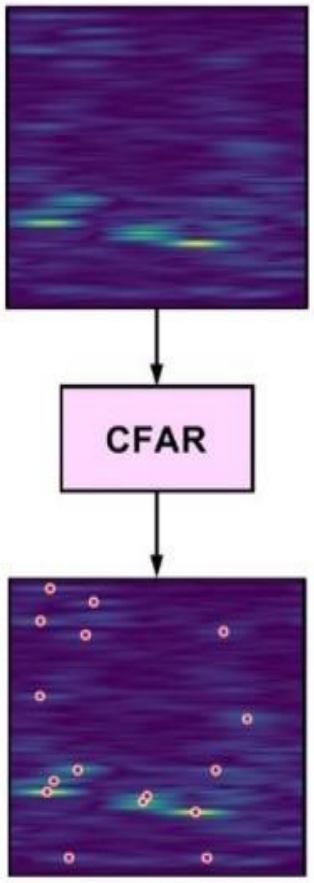


Yizhou Wang, et al., "RODNet: A Real-Time Radar Object Detection Network Cross-Supervised by Camera-Radar Fused Object 3D Localization," IEEE Journal of Selected Topics in Signal Processing, special issue on Recent Advances in Automotive Radar Signal Processing, Feb. 2021.

Camera-Radar Fused Supervision



Cross-Modality Training



CFAR

Teacher

Student

RGB Images

RF Images

Visual Object 3D Localization

CFAR Peak Detection

Camera to Radar Projection

Classes & Locations

CRF

Locations Only

Object Classes and Locations in RF Images

RF Images

RODNet

Supervision

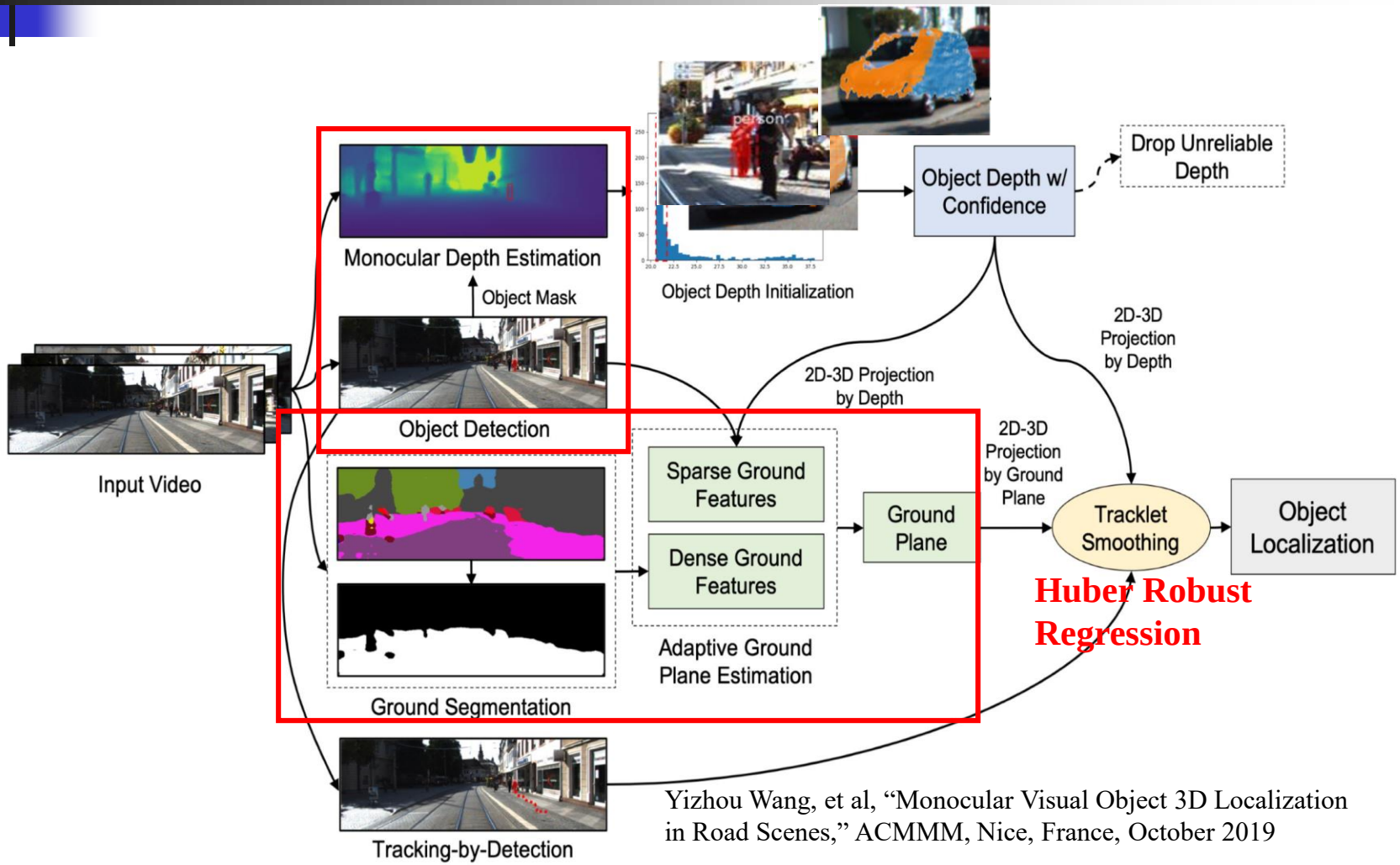
ConfMaps

L-NMS

Final Radar Object Detections
(Classes & Locations)

Location-based NMS

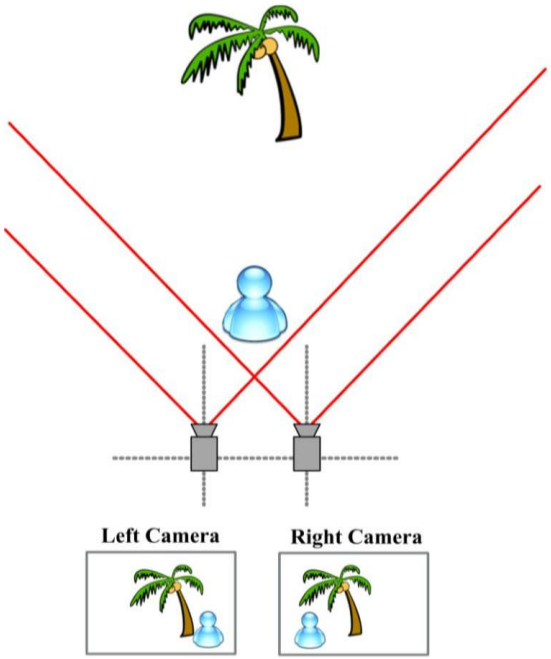
Object 3D Localization from A Moving Camera

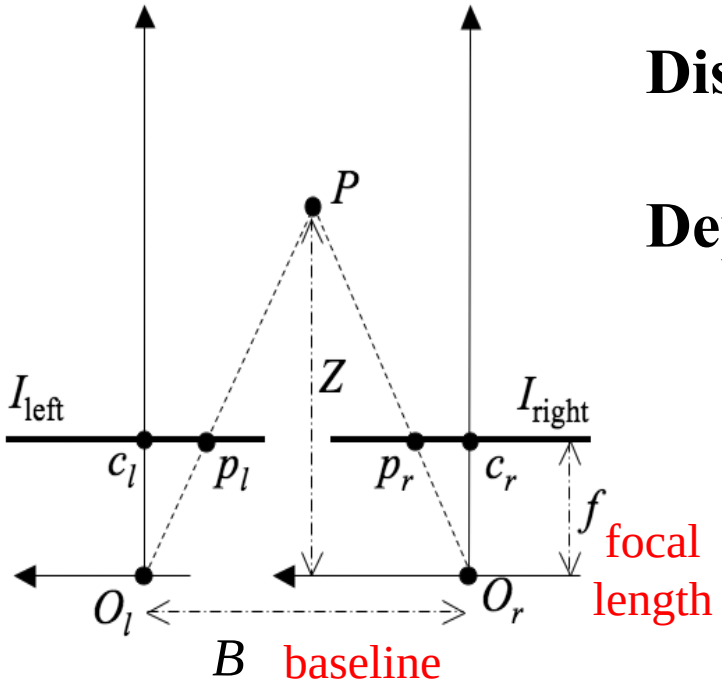


Yizhou Wang, et al, "Monocular Visual Object 3D Localization in Road Scenes," ACMMM, Nice, France, October 2019

Stereo Depth Estimation

Stereo camera: Estimate depth from disparity





Disparity:

$$D = p_l - p_r$$

Depth:

$$d = Z = \frac{fB}{D} = \frac{fB}{p_l - p_r}$$

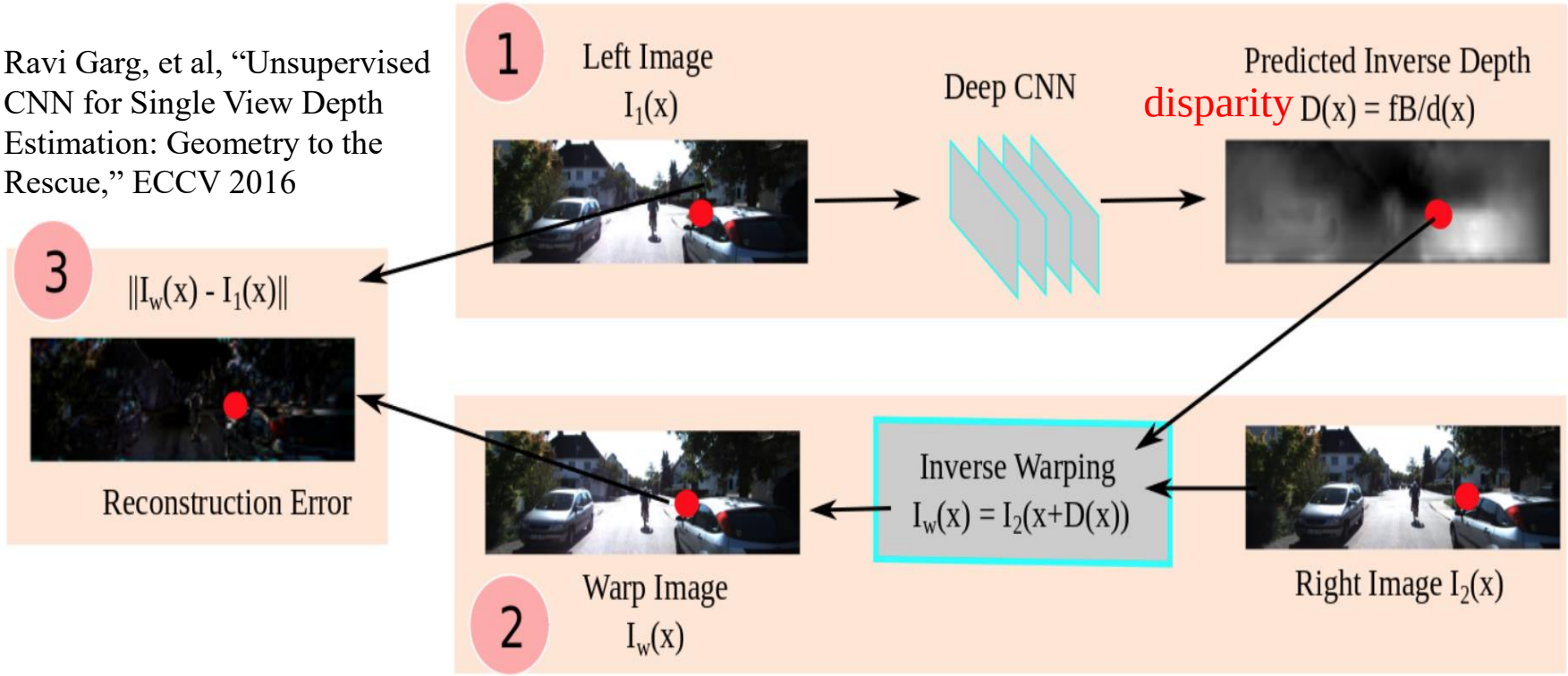
Depth is evaluated from the disparity of the corresponding points.

Figure credit to https://www.cs.auckland.ac.nz/courses/compsci773s1t/lectures/773-GG/topCS773.htm

Unsupervised CNN for Single View Depth Estimation

- **Training**: minimize the reconstruction loss with a simple smoothness prior.
- **Testing**: CNN performs single-view disparity (inverse depth) prediction, up to the scene scale given in form of fB at the time of training.

Ravi Garg, et al, "Unsupervised CNN for Single View Depth Estimation: Geometry to the Rescue," ECCV 2016



Unsupervised CNN for Single View Depth Estimation

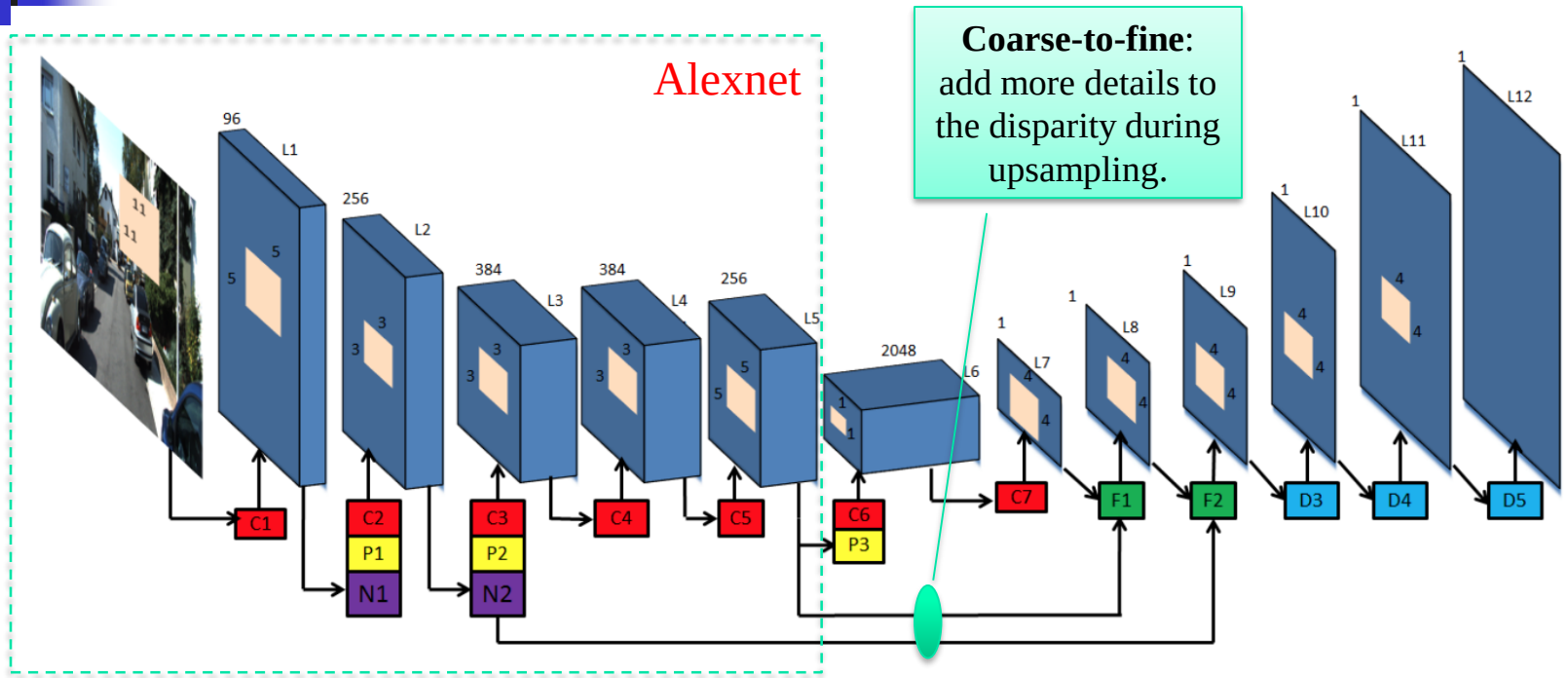


Fig. 3. Network architecture: The blocks C (red), P (yellow), L (dark blue), F (green), D (blue) correspond to convolution, pooling, local response normalization, FCN and upsampling layers respectively. The FCN blocks F1 and F2 upsample the predictions from layers (L7, L8) and combine it with the input of the pooling layers P3 and P2 respectively.

Unsupervised CNN for Single View Depth Estimation

Reconstruction loss (a generalization of the multi-frame optic flow):

$$E^i_{recons} = \int_\Omega \|I^i_w(x) - I^i_1(x)\|^2 dx = \int_\Omega \|I^i_2(x + \underbrace{D^i(x)}_{fB/d^i(x)}) - I^i_1(x)\|^2 dx$$

Disparity regularization:

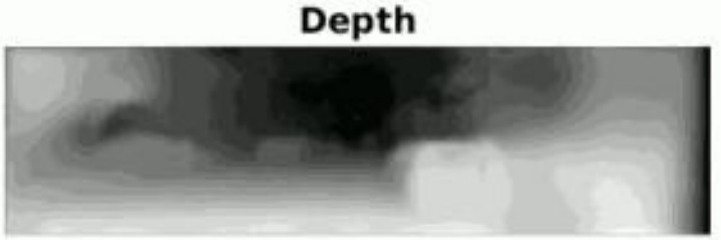
$$E^i_{smooth} = \|\nabla D^i(x)\|^2$$

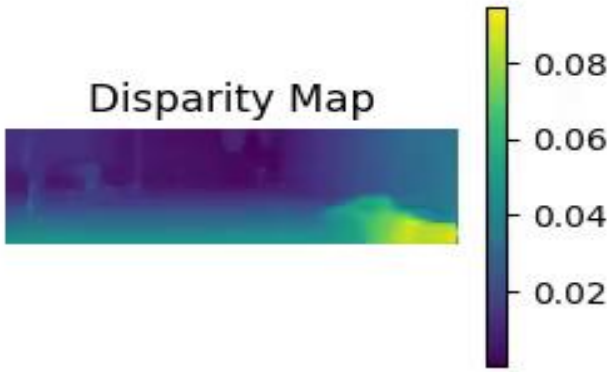
Total loss:

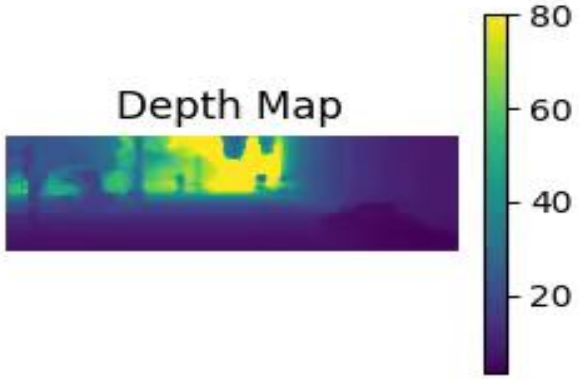
$$E = \sum_{i=1}^{N} E^i_{recons} + \gamma E^i_{smooth}$$

Mono Depth Map Demo

Ravi Garg et al., “Unsupervised CNN for Single View Depth Estimation: Geometry to the Rescue, “ ECCV 2016.







Unsupervised CNN for Single View Depth Estimation

Comparison with State-of-the-art Methods on KITTI Dataset

Methods	Resolution	RMS	*log* RMS	Absolute relative	Square relative	Accuracies $\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Ours L12	176 × 608	5.285	0.282	0.177	1.169	0.727	0.896	0.958
Ours L12, Aug 8x		**5.104**	0.273	**0.169**	**1.080**	**0.740**	**0.904**	0.962
Mean	-	9.635	0.444	0.412	5.712	0.556	0.752	0.870
Make3D [29]	Dense	8.734	0.361	0.280	3.012	0.601	0.820	0.926
Eigen *et al* (c^7) [8]	28 × 144	7.216	0.273	0.194	1.531	0.679	0.897	**0.967**
Eigen *et al* (f) [8]	27 × 142	7.156	**0.270**	0.190	1.515	0.692	0.899	**0.967**
Fayao *et al* (pt) [24]	superpix	7.421	-	-	-	0.613	0.858	0.949
Fayao *et al* (ft) [24]	superpix	7.046	-	-	-	0.656	0.881	0.958

Comparisons with Baseline Supervised Networks and Stereo

Methods	Coverage	RMS	*log* RMS	Absolute relative	Square relative	Accuracies $\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Ours L12	100%	**5.285**	**0.282**	**0.177**	**1.169**	**0.727**	**0.896**	**0.958**
HS→ CNN,$\gamma = .01$	100%	6.691	0.385	0.309	2.657	0.476	0.750	0.891
HS→ CNN	100%	6.292	0.338	0.238	1.639	0.573	0.841	0.941
SGM→ CNN	100%	5.680	0.300	0.185	1.370	0.703	0.886	0.955
HS-Stereo, $\gamma = .01$	**100%**	6.077	0.381	0.299	3.264	0.677	0.822	0.90
HS-Stereo	**100%**	6.760	0.366	0.254	4.040	0.754	0.872	0.928
SGM-Stereo	87%	3.030	0.150	0.064	0.506	0.955	0.979	0.989

RMS: $\sqrt{\frac{1}{T}\sum_{i \in T} \|d_i - d_i^{gt}\|^2}$ log RMS: $\sqrt{\frac{1}{T}\sum_{i \in T} \|log(d_i) - log(d_i^{gt})\|^2}$

abs. relative: $\frac{1}{T}\sum_{i \in T} \frac{|d_i - d_i^{gt}|}{d_i^{gt}}$ sq. relative: $\frac{1}{T}\sum_{i \in T} \frac{\|d_i - d_i^{gt}\|^2}{d_i^{gt}}$

Accuracies: % of d_i s.t. $max(\frac{d_i}{d_i^{gt}}, \frac{d_i^{gt}}{d_i}) = \delta < thr$

Depth Estimation with Left-Right Consistency

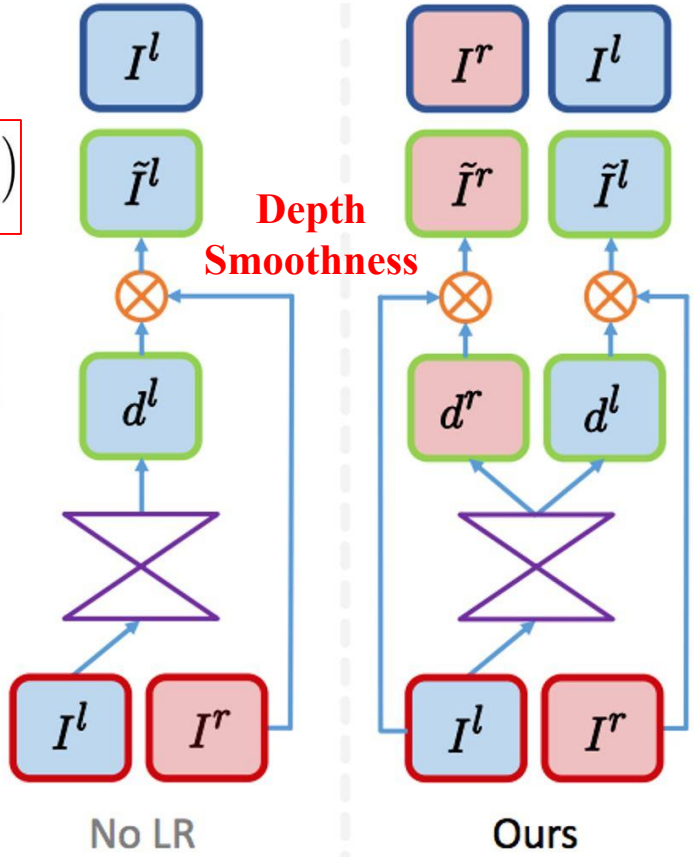
$$C = \sum_{s=1}^{4} C_s$$

$$C_s = \alpha_{ap}(C_{ap}^l + C_{ap}^r) + \alpha_{ds}(C_{ds}^l + C_{ds}^r) + \alpha_{lr}(C_{lr}^l + C_{lr}^r)$$

$$C_{ap}^l = \frac{1}{N}\sum_{i,j} \alpha \frac{1 - \mathrm{SSIM}(I_{ij}^l, \tilde{I}_{ij}^l)}{2} + (1-\alpha)\left\| I_{ij}^l - \tilde{I}_{ij}^l \right\|$$

$$C_{ds}^l = \frac{1}{N}\sum_{i,j} \left|\partial_x d_{ij}^l\right| e^{-\left\|\partial_x I_{ij}^l\right\|} + \left|\partial_y d_{ij}^l\right| e^{-\left\|\partial_y I_{ij}^l\right\|}$$

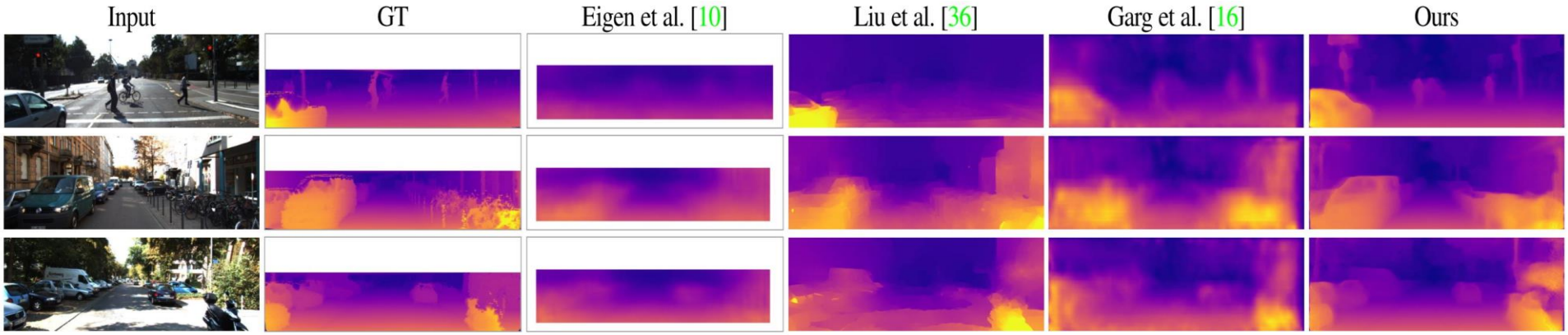
$$C_{lr}^l = \frac{1}{N}\sum_{i,j} \left| d_{ij}^l - d_{ij+d_{ij}^l}^r \right|$$

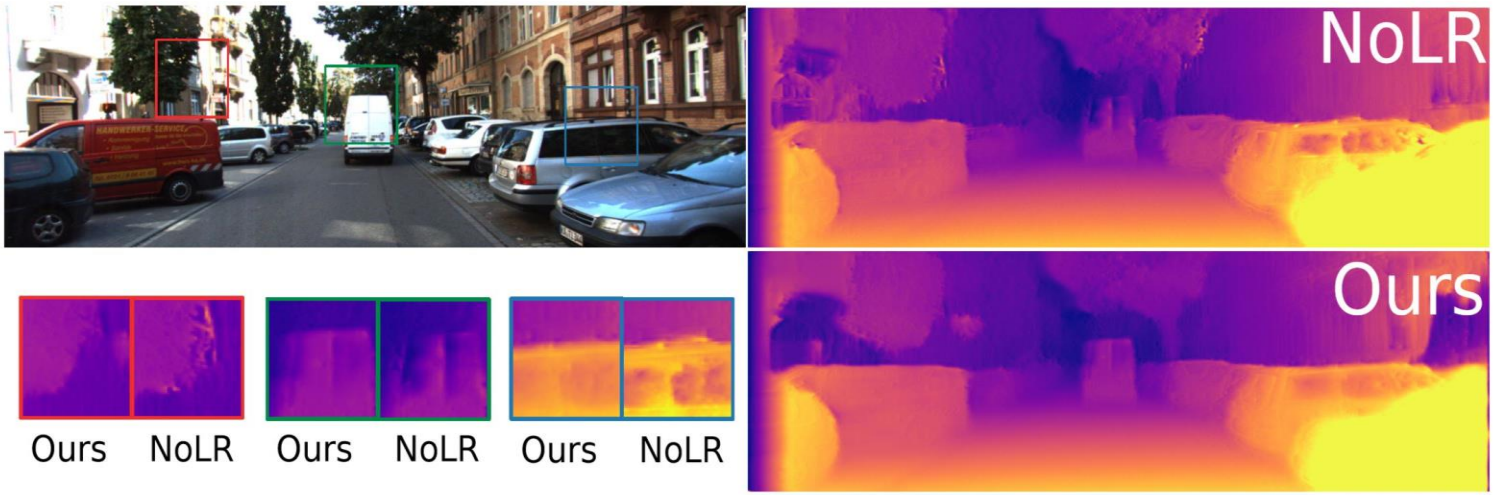


C_{ap} encourages the reconstructed image to appear similar to the corresponding training input, C_{ds} enforces smooth disparities, and C_{lr} prefers the predicted left and right disparities to be consistent.

C. Godard, et al., CVPR 2017

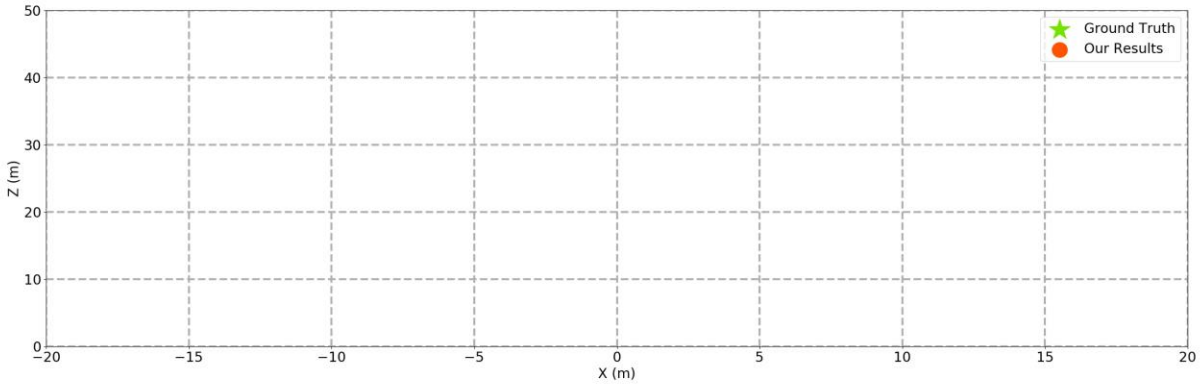
Depth Estimation with Left-Right Consistency





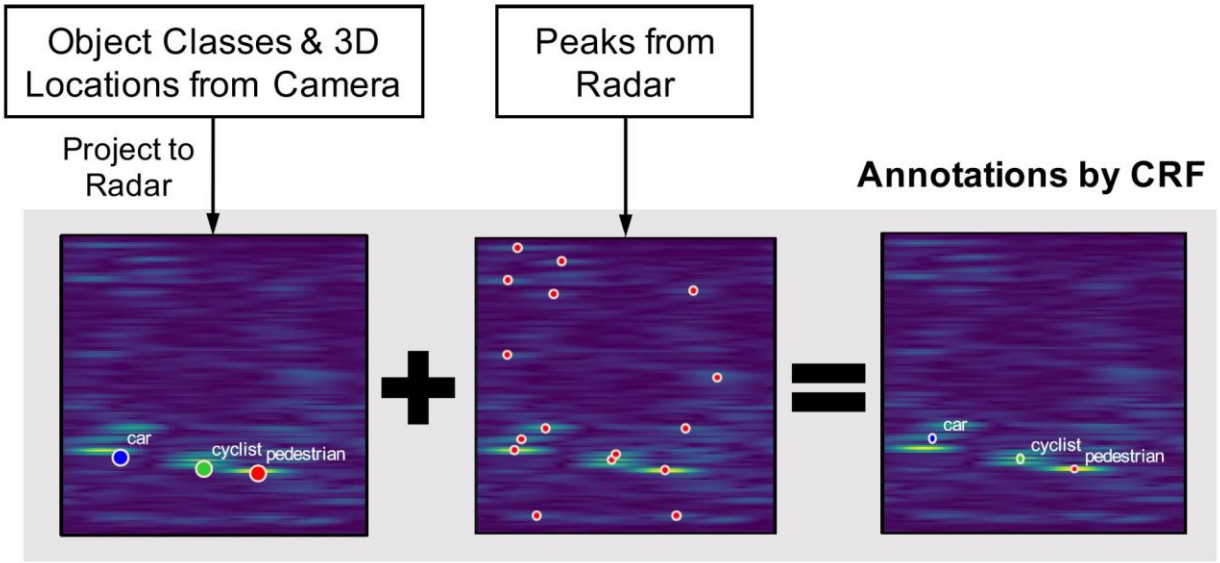
Qualitative Performance

Methods	Mean localization error (m)
Depth histogram	1.19 (±0.90)
3D point cloud	0.83 (±0.73)



Methods	Overall (m)	≤ 15m	≤ 30m	> 30m	Running speed
Murthy et al. [27]	2.61 (±2.23)	1.59 (±0.96)	2.52 (±2.16)	4.30 (±2.83)	–
Ansari et al. [2]	1.00 (±0.77)	0.67 (±0.50)	0.94 (±0.69)	2.19 (±1.18)	–
Ansari et al. (Opt) [2]	0.86 (±0.87)	0.55 (±0.50)	0.79 (±0.79)	2.16 (±1.18)	–
Ours (DHist+AGPE+TS)	**0.69 (±0.51)**	0.42 (±0.33)	**0.68 (±0.53)**	**1.22 (±0.74)**	2.0 FPS

Camera-Radar Fusion (CRF)



$$P^{c}_{(cls)}(\mathbf{x}) = \max_i \left\{ \text{normalize} \left(\frac{1}{2\pi\sqrt{|\mathbf{\Sigma}^{c}_{i(cls)}|}} \exp\left\{ -\frac{1}{2}(\mathbf{x}-\mu^{c}_{i})^{\top}(\mathbf{\Sigma}^{c}_{i(cls)})^{-1}(\mathbf{x}-\mu^{c}_{i}) \right\} \right) \right\}$$

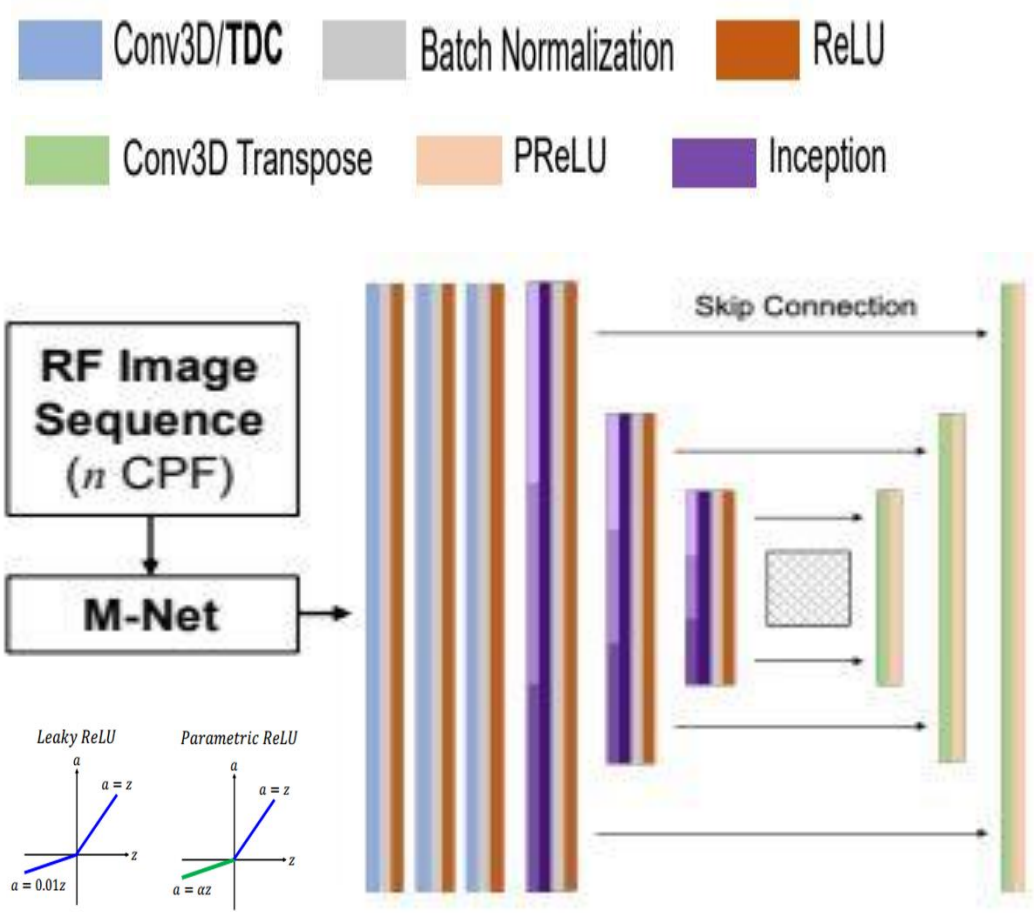
$$\mu^{c}_{i} = \begin{bmatrix} \rho^{c}_{i} \\ \theta^{c}_{i} \end{bmatrix}, \mathbf{\Sigma}^{c}_{i(cls)} = \begin{bmatrix} \left(d_i s_{(cls)}/c_i\right)^2 & 0 \\ 0 & \delta_{(cls)} \end{bmatrix}.$$

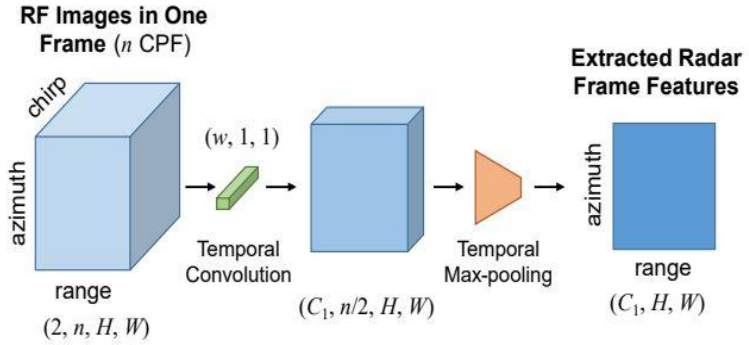
$$P^{CRF}_{(cls)}(\mathbf{x}) = P^{c}_{(cls)}(\mathbf{x}) * P^{r}(\mathbf{x})$$

$$P^{r}(\mathbf{x}) = \max_j \left\{ \text{normalize} \left(\frac{1}{2\pi\sqrt{|\mathbf{\Sigma}^{r}_{j}|}} \exp\left\{ -\frac{1}{2}(\mathbf{x}-\mu^{r}_{j})^{\top}(\mathbf{\Sigma}^{r}_{j})^{-1}(\mathbf{x}-\mu^{r}_{j}) \right\} \right) \right\}$$

$$\mu^{r}_{j} = \begin{bmatrix} \rho^{r}_{j} \\ \theta^{r}_{j} \end{bmatrix}, \mathbf{\Sigma}^{r}_{j} = \begin{bmatrix} \delta^{r}_{j} & 0 \\ 0 & \epsilon(\theta^{r}_{j}) \end{bmatrix}.$$

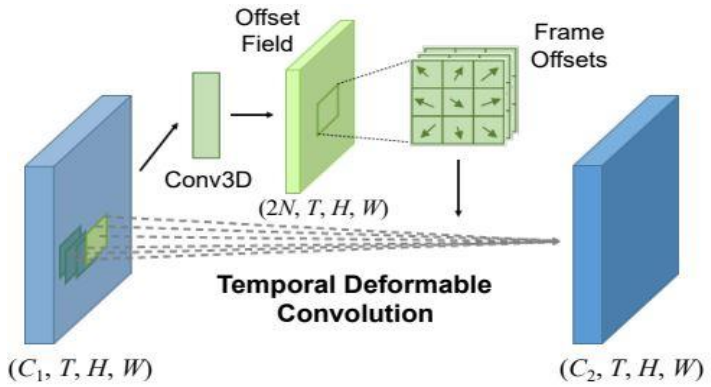
Radar Object Detection Network (RODNet)





(d) M-Net Module for Chirp Merging

N=8 chirps (uniform sample from 255)



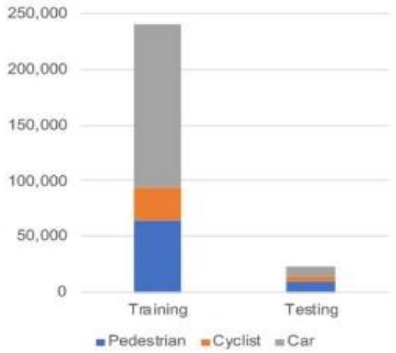
(e) Temporal Deformable Convolution (TDC) Module

T=16 frames (snippet)

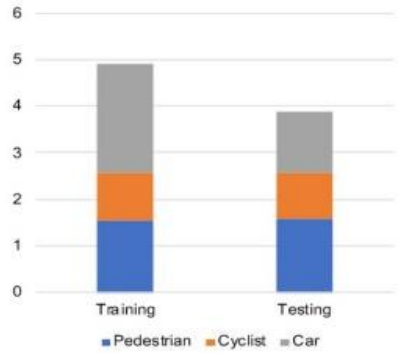
CRUW DataSet

- Sensor platform: Stereo cameras + two RADAR antenna arrays.
- Scenarios: Campus road, city street, highway, parking lot, etc.
- Scale: More than 3 hours of synchronized vision-radio data.
- Views: front-view and side-view.

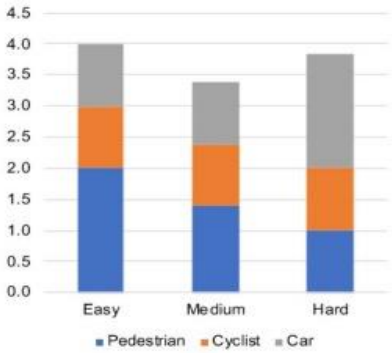
CRUW DataSet



(a) # of objects in total



(b) # of objects per frame

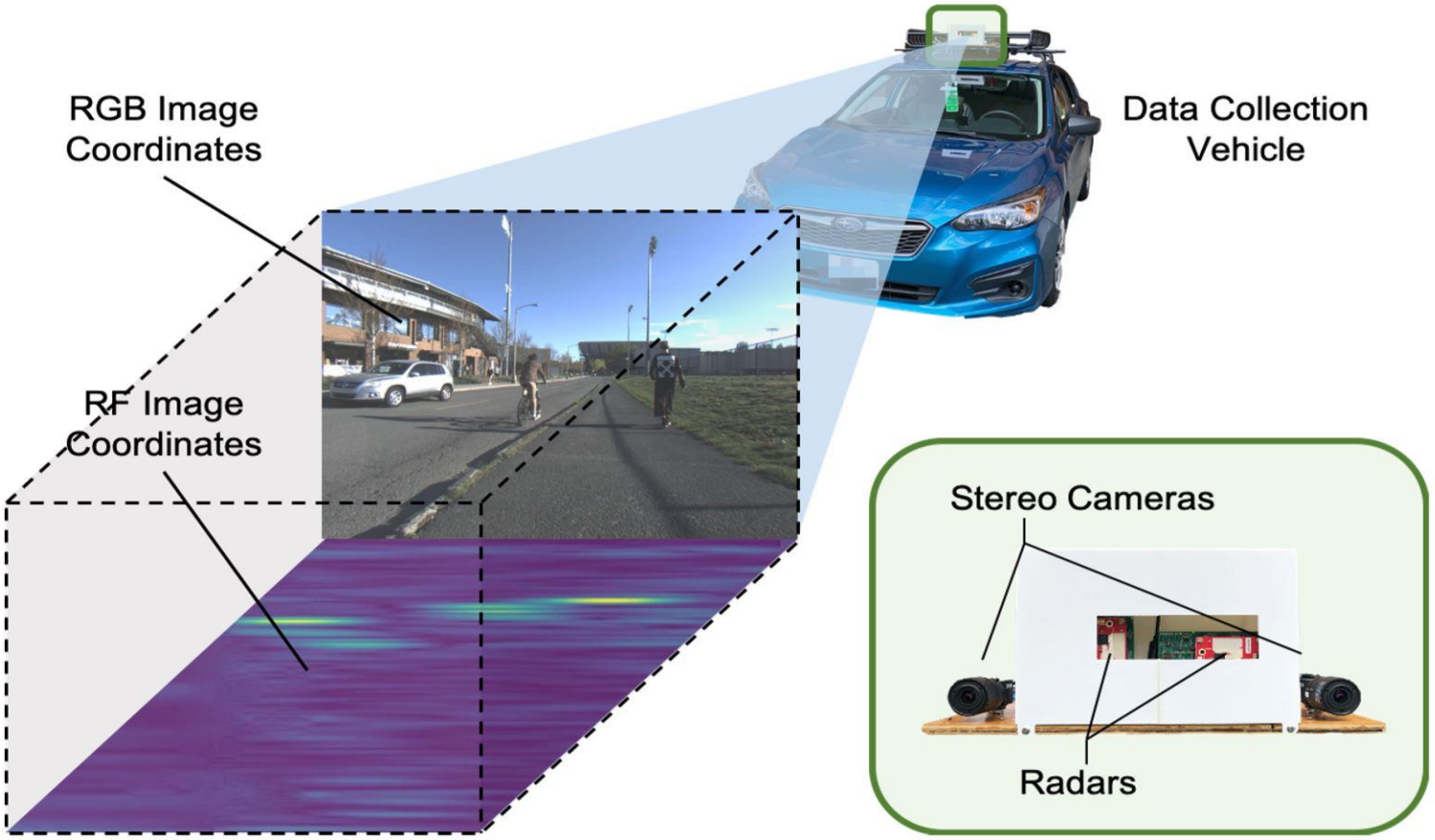


(c) # of objects per frame in the testing set

Scenarios	# of Seqs	# of Frames	Vision-Fail %
Parking Lot	124	106,057	15%
Campus Road	112	94,416	11%
City Street	216	175,392	6%
Highway	12	20,376	0%
Overall	464	396,241	9%

(d) Driving scenarios statistics for CRUW dataset

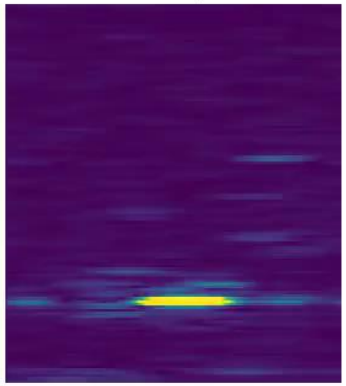
CRUW Capturing



RODNet Detection

Image

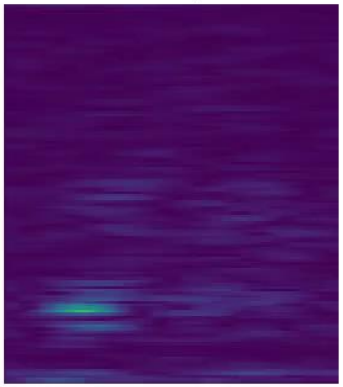
RAMap



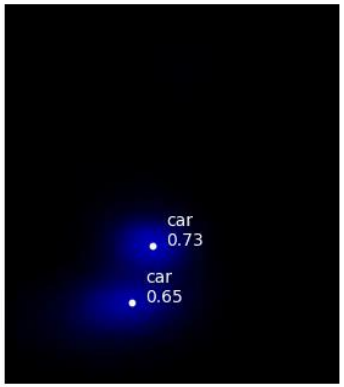
RODNet Detections

Image

RAMap

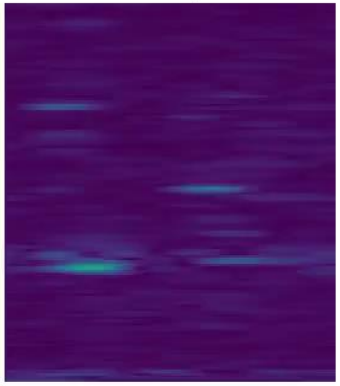


RODNet Detections

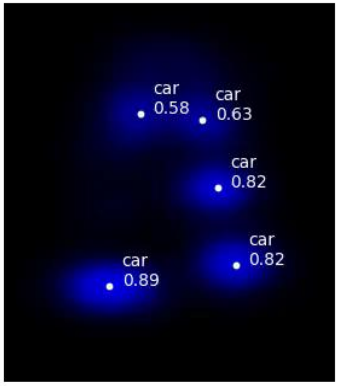


Image

RAMap



RODNet Detections



Evaluation Metric: Object Location Similarity OLS

- An IoU based non-maximum suppression (NMS) is needed to remove the redundant bounding boxes from the detections.
- No bounding box definition in RF images and the ConfMaps
- Object Location Similarity (OLS)

$$\mathrm{OLS} = \exp\left\{\frac{-d^2}{2(s\kappa_{cls})^2}\right\}$$

- *d* is the distance (in meters) between the two points in an RF image; *s* is the object distance from the sensors, representing object scale information; and κ_{cls} is a per-class constant that represents the error tolerance for class *cls*, which can be determined by the object average size of the corresponding class.

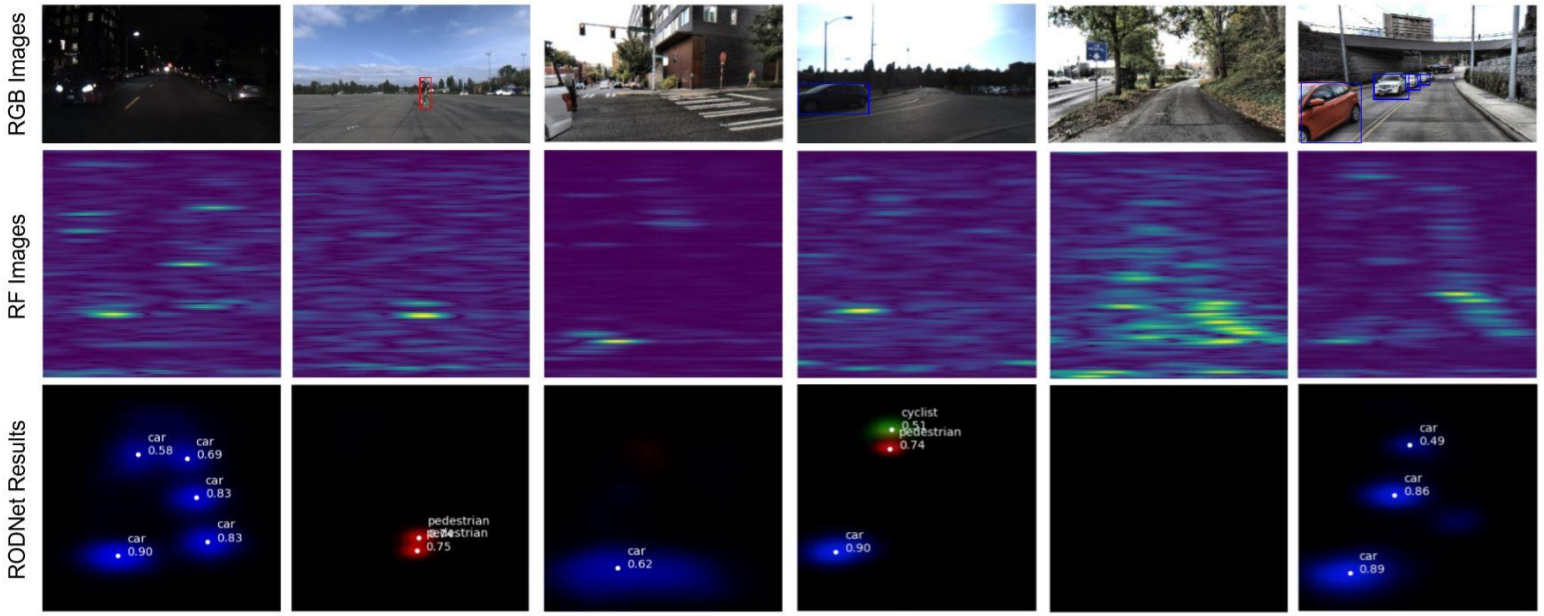
Experimental Results

Methods	Overall		Easy		Medium		Hard	
	AP	AR	AP	AR	AP	AR	AP	AR
Decision Tree [33]	4.70	44.26	6.21	47.81	4.63	43.92	3.21	37.02
CFAR+ResNet [29]	40.49	60.56	78.92	85.26	11.00	33.02	6.84	36.65
CFAR+VGG-16 [33]	40.73	72.88	85.24	88.97	47.21	62.09	10.97	45.03
RODNet (Ours)	85.98	87.86	96.97	98.02	76.11	78.57	67.28	72.60

Supervision	Pedestrian	Cyclist	Car
CO	0.69 (±0.77)	0.87 (±0.89)	1.57 (±1.12)
CRF	0.67 (±0.55)	0.82 (±0.59)	1.26 (±0.64)

mean localization error (meters)

we use different thresholds from 0.5 to 0.9 with a step of 0.05 for OLS and calculate the average precision (AP)

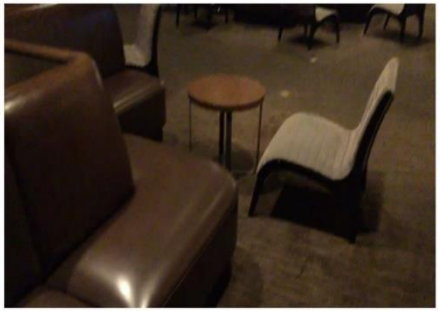


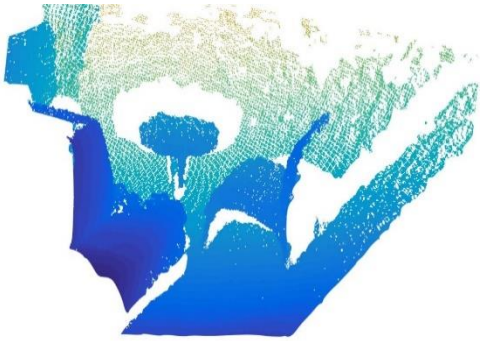
3D Point Cloud Data

- Rich 3D Representations
- Feature Extraction from Point Clouds
- 3D Object Classification and Segmentation
- 3D Object Detection from Point Clouds

Rich 3D Representations

- Point Cloud (PC)
- Mesh
- Volumetric (Voxel)
- Projected View (RGB-D, BEV)





Point Cloud

Mesh

Volumetric

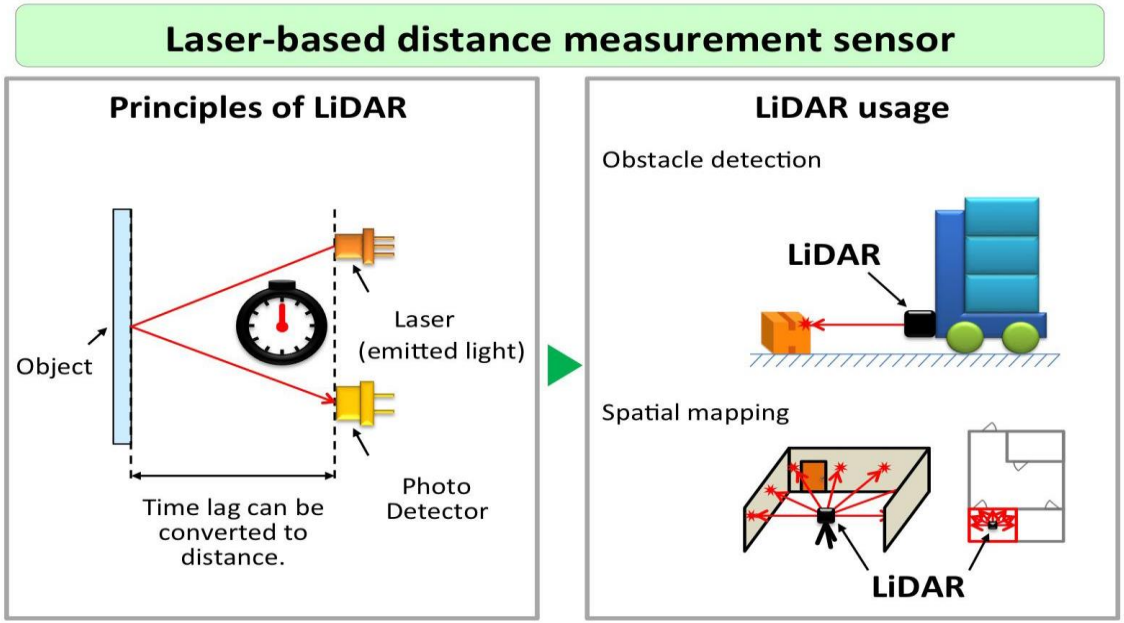
Projected View
RGB(D)

Pros and Cons for PC Representations

- Pros
  - Close to raw sensor data (with rich geometric information).
  - Canonical and preserves geometric cues.
  - Memory efficiency is high.
- Cons
  - Unordered point set as the input.
  - Invariance under geometric transformations.

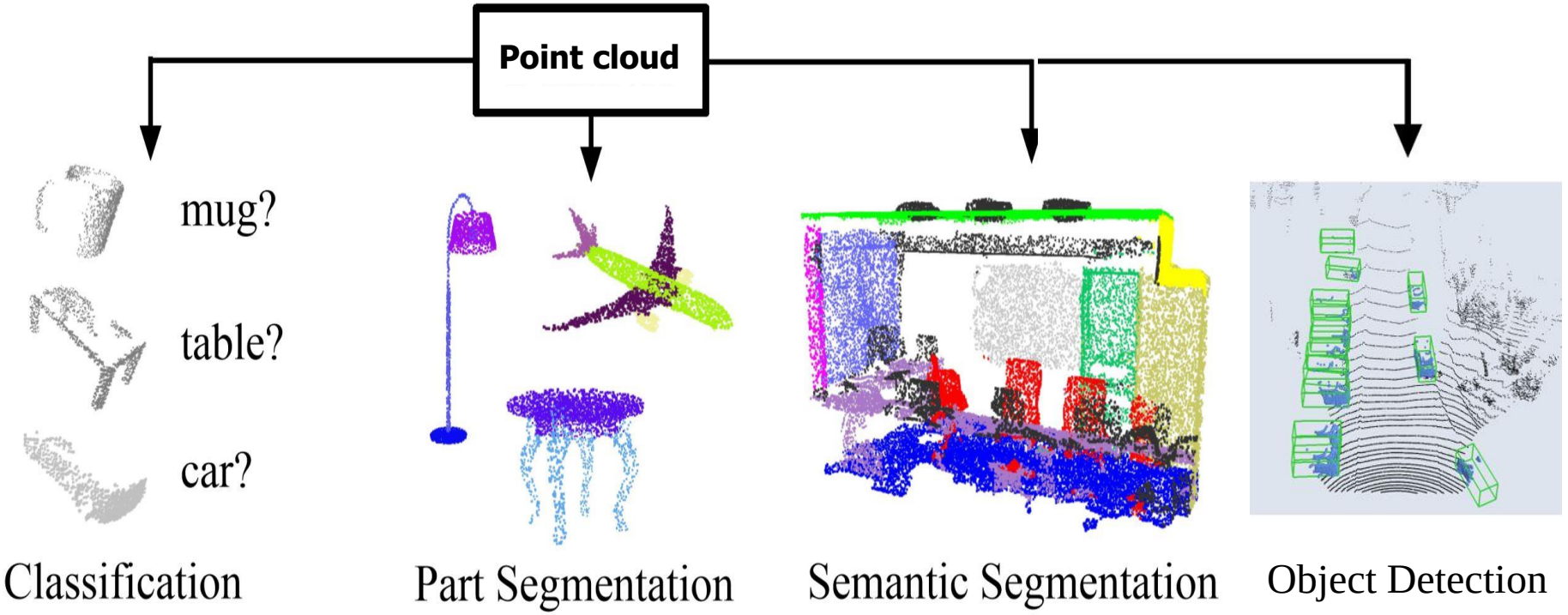
PC Data from Lidar

- Lidar: light detection and ranging, which uses eye-safe laser beams to create a 3D representation of the surveyed environment.



Applications of Point Cloud

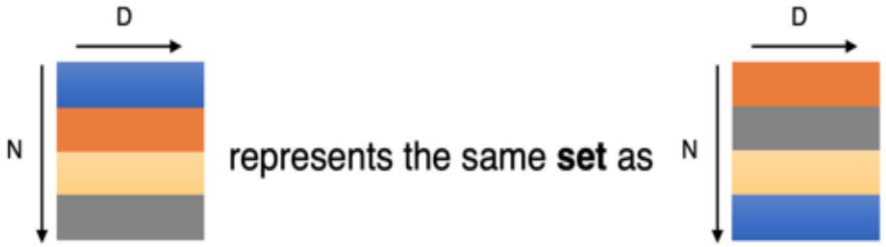
- Close to raw sensor data (with rich geometric information).
- Unordered point set as the input, partial views



Feature Extraction

- Point cloud: *N* orderless points, each represented by a *m*-dim (3-D) vector.
- Model needs to be invariant to *N!* permutations.

$$f(x_1, x_2, \ldots, x_n) \equiv f(x_{\pi_1}, x_{\pi_2}, \ldots, x_{\pi_n}), \quad x_i \in \mathbb{R}^D$$



- Permutation Invariant by Symmetric Functions
- Examples:

$$f(x_1, x_2, \ldots, x_n) = \max\{x_1, x_2, \ldots, x_n\}$$

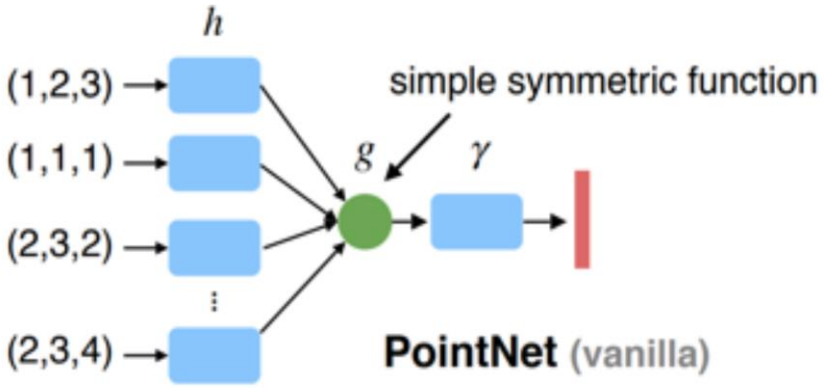
$$f(x_1, x_2, \ldots, x_n) = x_1 + x_2 + \ldots + x_n$$

$$\ldots$$

PointNet

- Highly robust to small perturbation of input points as well as to corruption through point insertion (outliers) or deletion (missing data) (via theoretical analysis).

$$f(x_1, x_2, ..., x_n) = \gamma \circ g(h(x_1), h(x_2), ..., h(x_n))$$

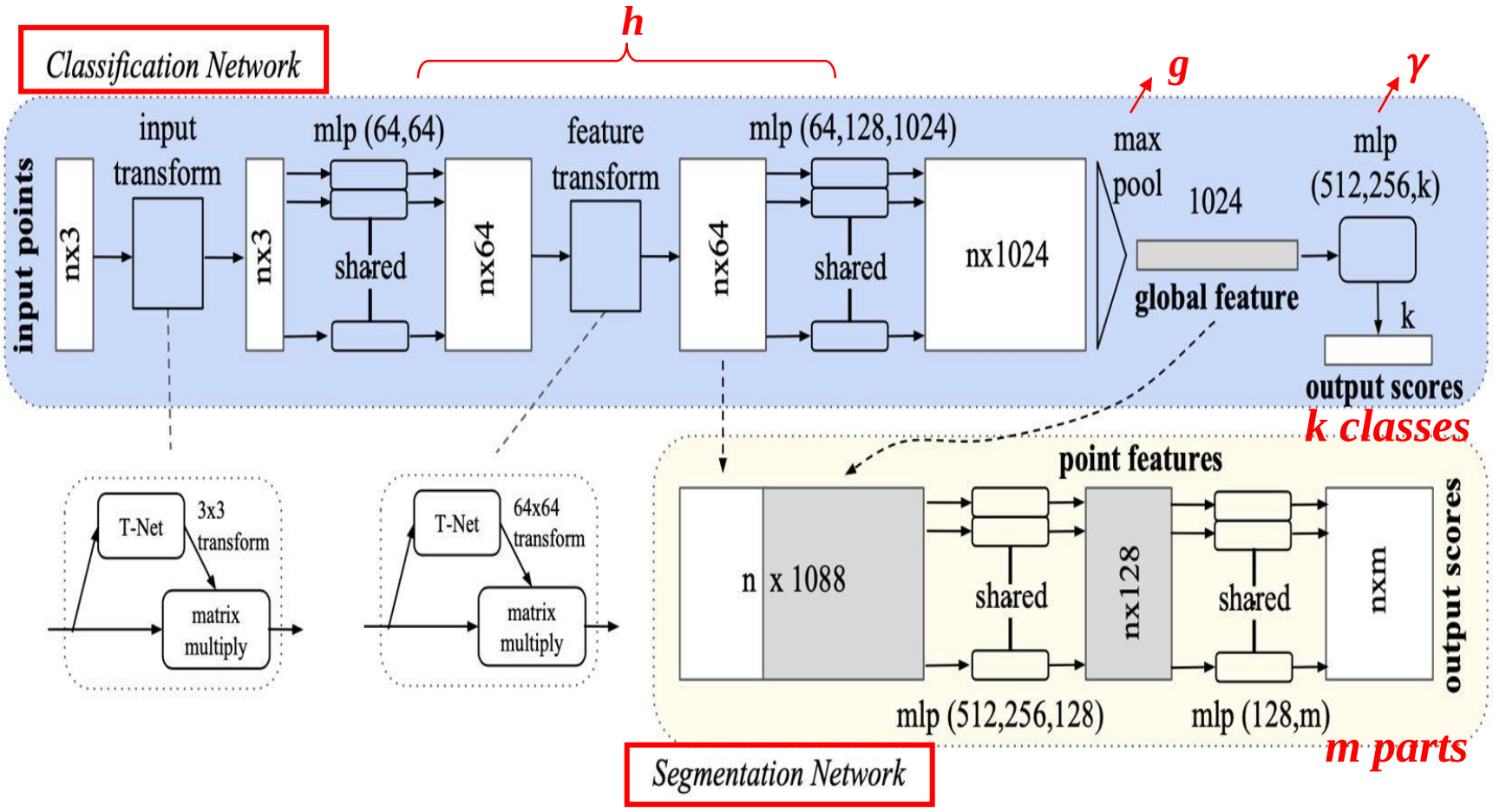


$$\left| f(S) - \gamma \left(\underset{x_i \in S}{MAX} \{h(x_i)\} \right) \right| < \epsilon$$

PointNet

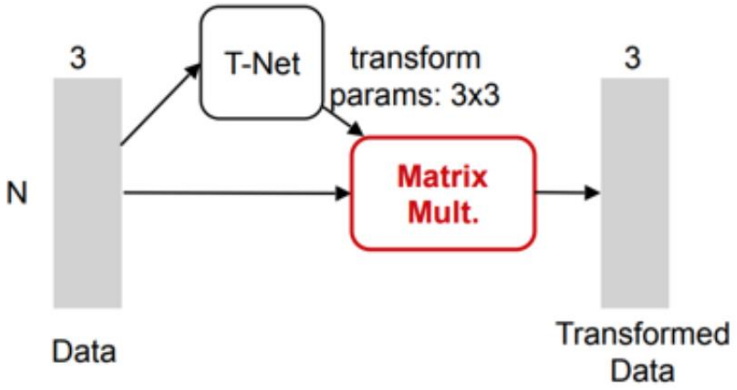
[Qi et al. "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," CVPR 2017]

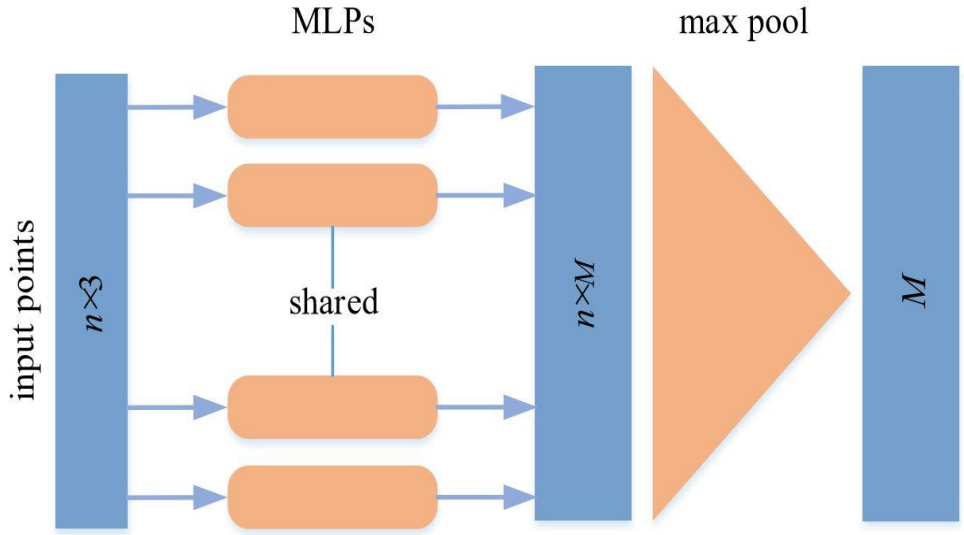
PointNet

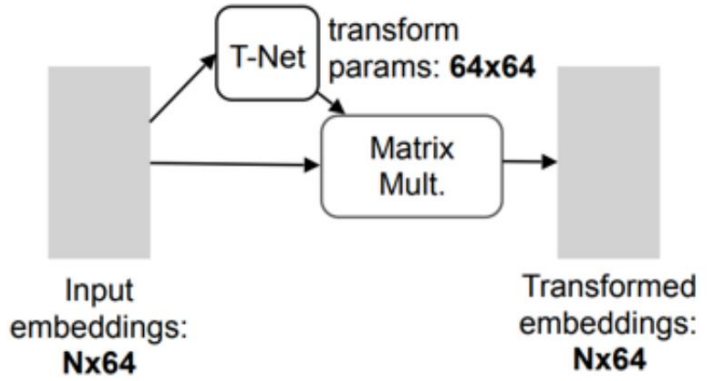

h
Classification Network
g
γ
input points
nx3
input transform
nx3
mlp (64,64)
shared
nx64
feature transform
nx64
mlp (64,128,1024)
shared
nx1024
max pool
1024
global feature
mlp (512,256,k)
k
output scores
k classes
T-Net
3x3 transform
matrix multiply
T-Net
64x64 transform
matrix multiply
point features
n x 1088
shared
nx128
shared
nxm
output scores
mlp (512,256,128)
mlp (128,m)
m parts
Segmentation Network

Transformation Invariant by T-Nets

- PointNet should be stable for different orientations of the point cloud.

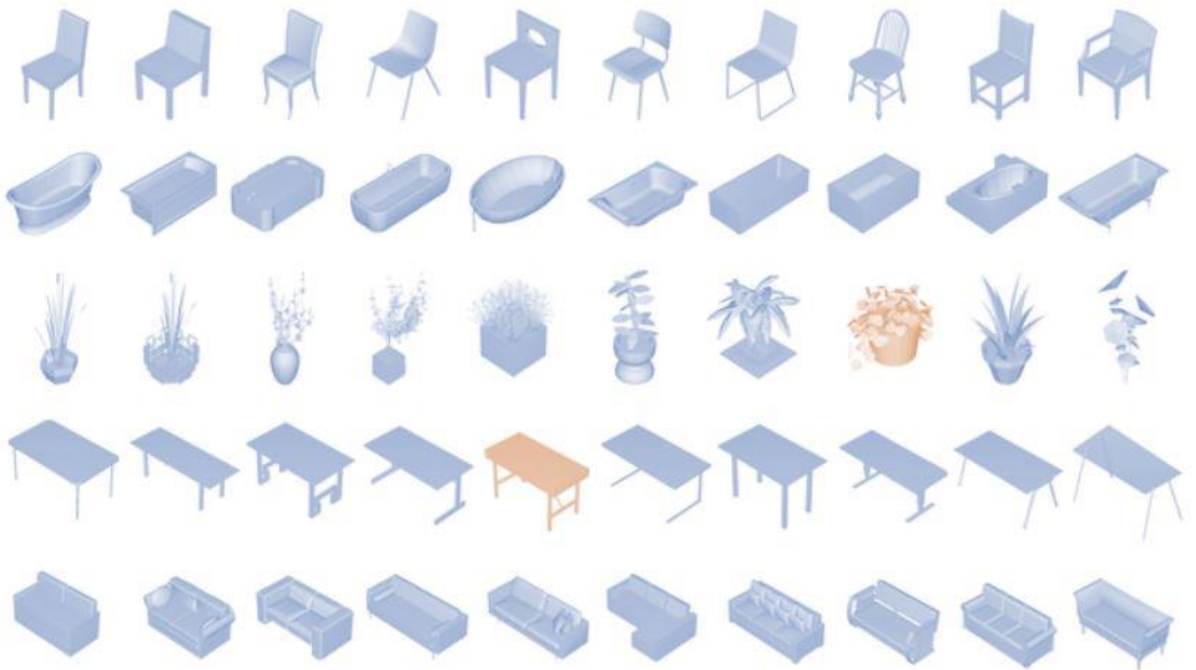






3D Object Classification Dataset

- Princeton ModelNet40 shape classification benchmark: 12,311 CAD models from 40 man-made object categories, 9,843 for training and 2,468 for testing



Z. Wu, et al., "3d shapenets: A deep representation for volumetric shapes," IEEE CVPR 2015

PointNet Classification Performance

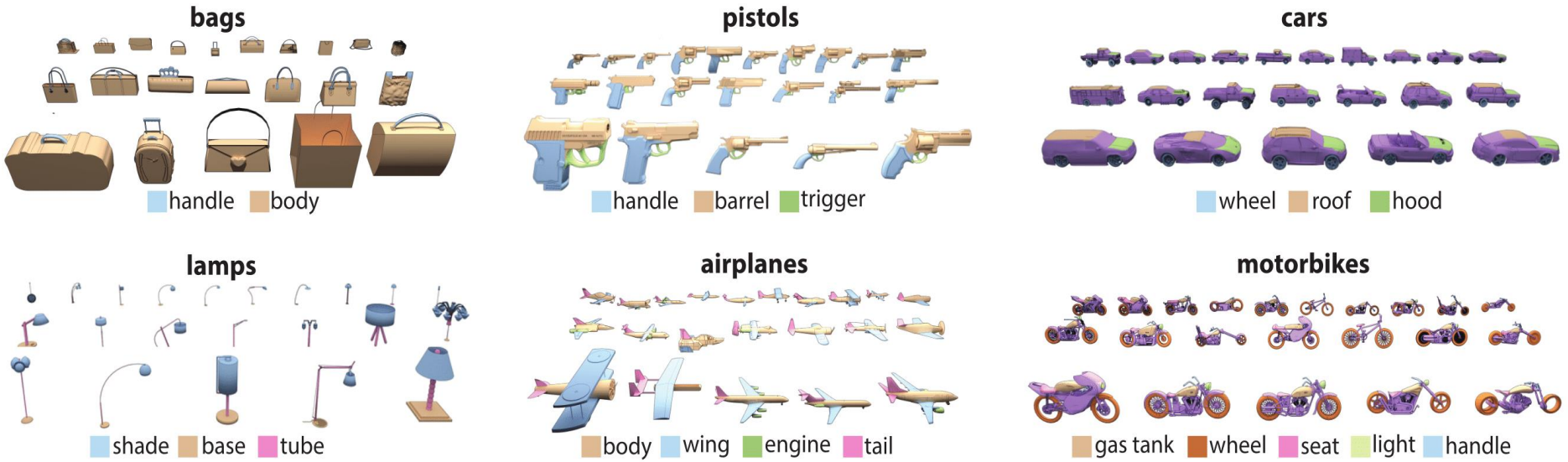
	input	#views	accuracy avg. class	accuracy overall
SPH [11]	mesh	-	68.2	-
3DShapeNets [28]	volume	1	77.3	84.7
VoxNet [17]	volume	12	83.0	85.9
Subvolume [18]	volume	20	86.0	**89.2**
LFD [28]	image	10	75.5	-
MVCNN [23]	image	80	**90.1**	-
Ours baseline	point	-	72.6	77.4
Ours PointNet	point	1	86.2	**89.2**

Table 1. **Classification results on ModelNet40.** Our net achieves state-of-the-art among deep nets on 3D input.

Algorithm	ModelNet40 Classification (Accuracy)	ModelNet40 Retrieval (mAP)	ModelNet10 Classification (Accuracy)	ModelNet10 Retrieval (mAP)
RS-CNN[63]	93.6%	-	-	-
LP-3DCNN[62]	92.1%	-	94.4%	-
LDGCNN[61]	92.9%	-	-	-
Primitive-GAN[60]	86.4%	-	92.2%	-
3DCapsule [59]	92.7%	-	94.7%	-
3D2SeqViews [58]	93.40%	90.76%	94.71%	92.12%
OrthographicNet [57]	-	-	88.56%	86.85%
Ma et al. [56]	91.05%	84.34%	95.29%	93.19%
MLVCNN [55]	94.16%	92.84%	-	-
iMHL [54]	97.16%	-	-	-
HGNN [53]	96.6%	-	-	-
SPNet [52]	92.63%	85.21%	97.25%	94.20%
MHBN [51]	94.7	-	95.0	-
VIPGAN [50]	91.98	89.23	94.05	90.69
Point2Sequence [49]	92.60	-	95.30	-
Triplet-Center Loss [48]	-	88.0%	-	-
PVNet[47]	93.2%	89.5%	-	-
GVCNN[46]	93.1%	85.7%	-	-
MLH-MV[45]	93.11%		94.80%	
MVCNN-New[44]	95.0%			
SeqViews2SeqLabels[43]	93.40%	89.09%	94.82%	91.43%
G3DNet[42]	91.13%		93.1%	
VSL [41]	84.5%		91.0%	
3D-CapsNets[40]	82.73%	70.1%	93.08%	88.44%
KCNet[39]	91.0%		94.4%	
FoldingNet[38]	88.4%		94.4%	
binVoxNetPlus[37]	85.47%		92.32%	
DeepSets[36]	90.3%			
3D-DescriptorNet[35]			92.4%	
SO-Net[34]	93.4%		95.7%	
Minto et al.[33]	89.3%		93.6%	
RotationNet[32]	97.37%		98.46%	
LonchaNet[31]			94.37	
Achlioptas et al. [30]	84.5%		95.4%	
PANORAMA-ENN [29]	95.56%	86.34%	96.85%	93.28%
3D-A-Nets [28]	90.5%	80.1%		
Soltani et al. [27]	82.10%			
Arvind et al. [26]	86.50%			
LonchaNet [25]			94.37%	
3DmFV-Net [24]	91.6%		95.2%	
Zanuttigh and Minto [23]	87.8%		91.5%	
Wang et al. [22]	93.8%			
ECC [21]	83.2%		90.0%	
PANORAMA-NN [20]	90.7%	83.5%	91.1%	87.4%
MVCNN-MultiRes [19]	91.4%			
FPNN [18]	88.4%			
PointNet[17]	89.2%			
Klokov and Lempitsky[16]	91.8%		94.0%	
LightNet[15]	88.93%		93.94%	
Xu and Todorovic[14]	81.26%		88.00%	
Geometry Image [13]	83.9%	51.3%	88.4%	74.9%
Set convolution [11]	90%			

3D Object Part Segmentation

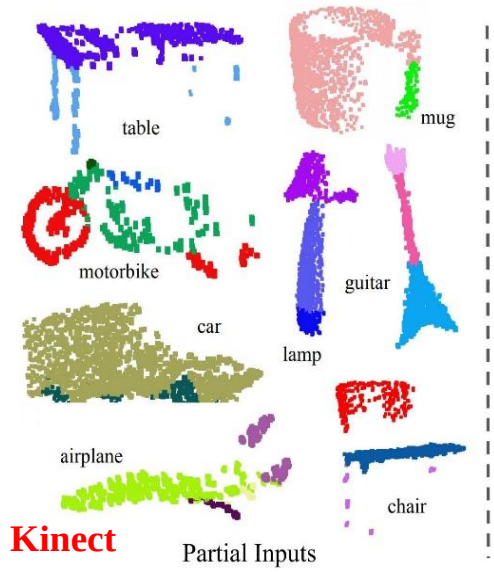
- ShapeNet Part dataset, 16,881 shapes from 16 categories, annotated with 50 parts in total.
- Most object categories are labeled with two to five parts, part segmentation as a per-point classification problem.



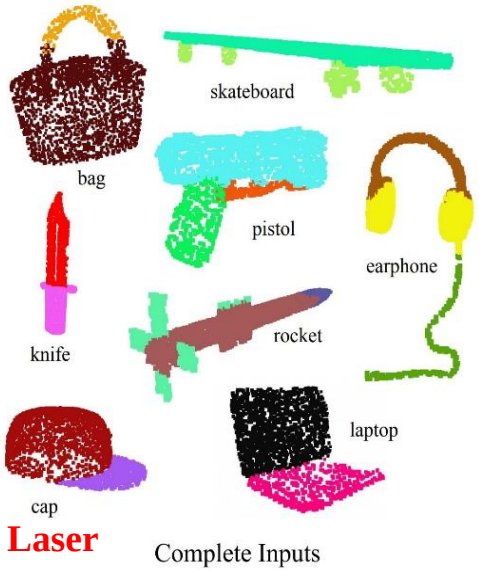
L. Yi, "A scalable active framework for region annotation in 3d shape collections," SIGGRAPH Asia, 2016.

Part Segmentation Performance

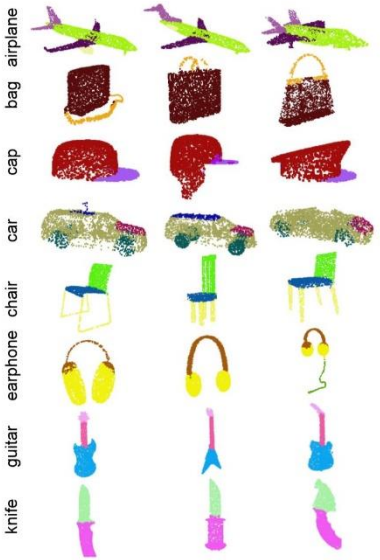
	mean	aero	bag	cap	car	chair	ear phone	guitar	knife	lamp	laptop	motor	mug	pistol	rocket	skate board	table
# shapes		2690	76	55	898	3758	69	787	392	1547	451	202	184	283	66	152	5271
Wu [27]	-	63.2	-	-	-	73.5	-	-	-	74.4	-	-	-	-	-	-	74.8
Yi [29]	81.4	81.0	78.4	77.7	**75.7**	87.6	61.9	**92.0**	85.4	**82.5**	**95.7**	**70.6**	91.9	**85.9**	53.1	69.8	75.3
3DCNN	79.4	75.1	72.8	73.3	70.0	87.2	63.5	88.4	79.6	74.4	93.9	58.7	91.8	76.4	51.2	65.3	77.1
Ours	**83.7**	**83.4**	**78.7**	**82.5**	74.9	**89.6**	**73.0**	91.5	**85.9**	80.8	95.3	65.2	**93.0**	81.2	**57.9**	**72.8**	**80.6**

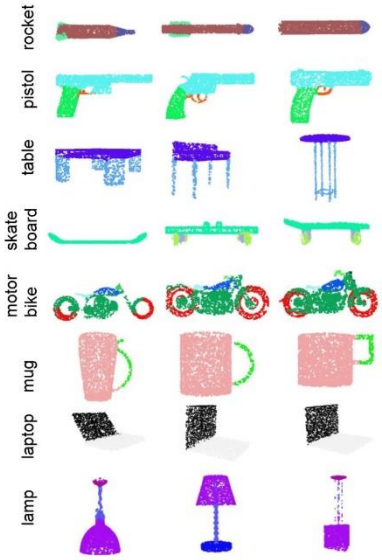


Kinect scan



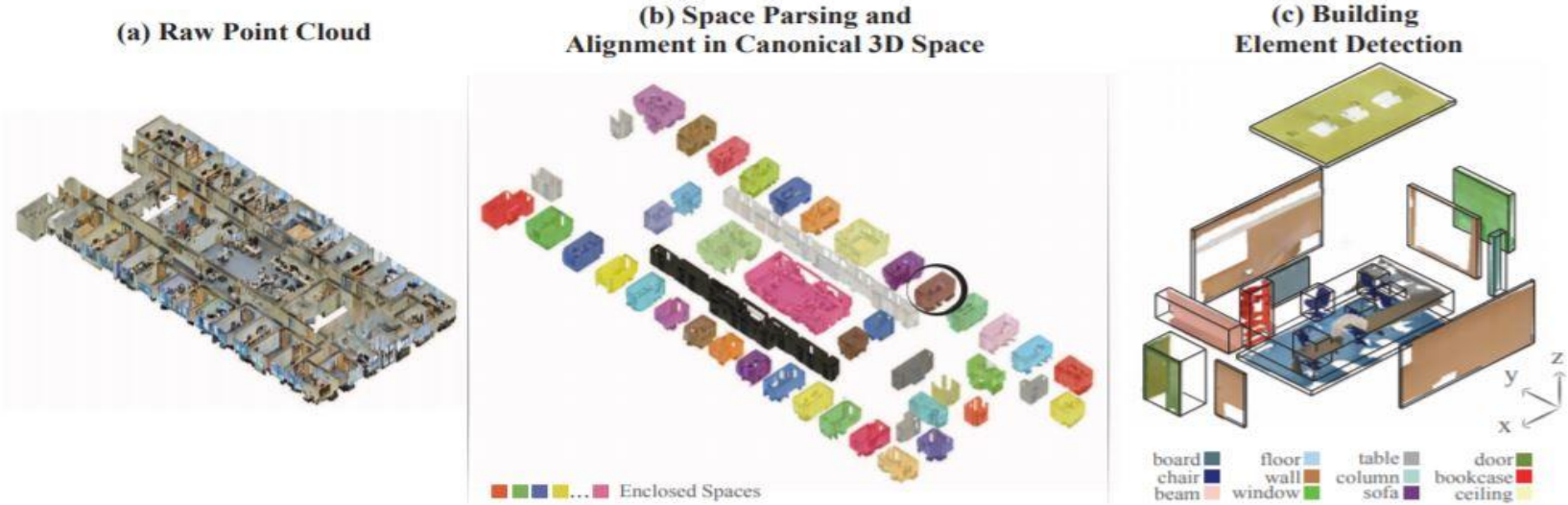
Laser scan





Semantic Segmentation in Scenes

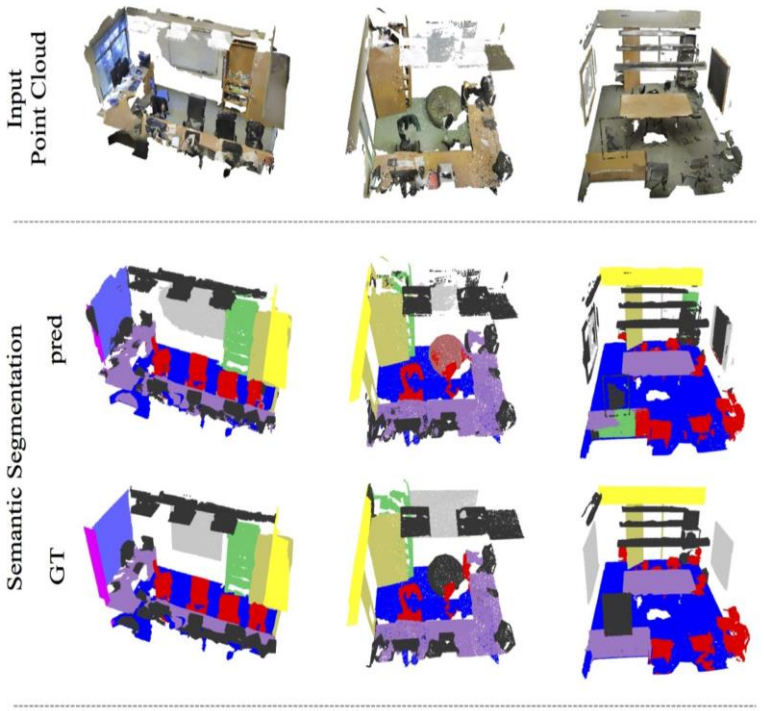
- Stanford 3D semantic parsing dataset: 3D scans from Matterport scanners in 6 areas including 271 rooms. Each point in the scan is annotated with one of the semantic labels from 13 categories (chair, table, floor, wall etc. plus clutter).



I. Armeni, "3d semantic parsing of large-scale indoor spaces,: IEEE CVPR 2016

PointNet Performance

	mean IoU	overall accuracy
Ours baseline	20.12	53.19
Ours PointNet	**47.71**	**78.62**



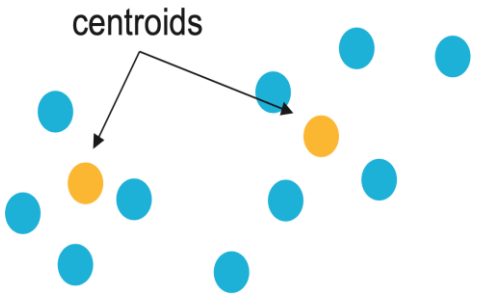
PointNet++

- PointNet Limitations:
  - Original PointNet lacks hierarchical feature learning like modern CNN, i.e., does not capture local structures (every point is independent).
  - Leveraging multi-scales with PointNet.
- The basic abstraction layer of PointNet++:
  - Sampling centroids
  - Grouping points by centroids (clustering), set abstraction
  - Apply PointNet on each group

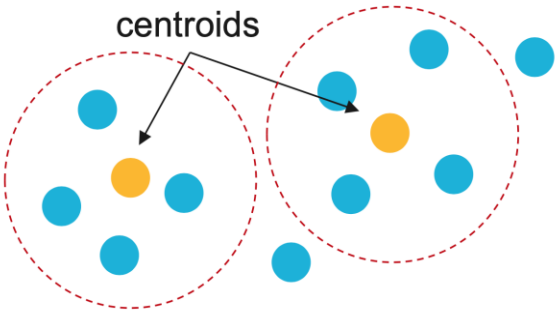
[Qi et al., PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space, NIPS 2017]

PointNet++

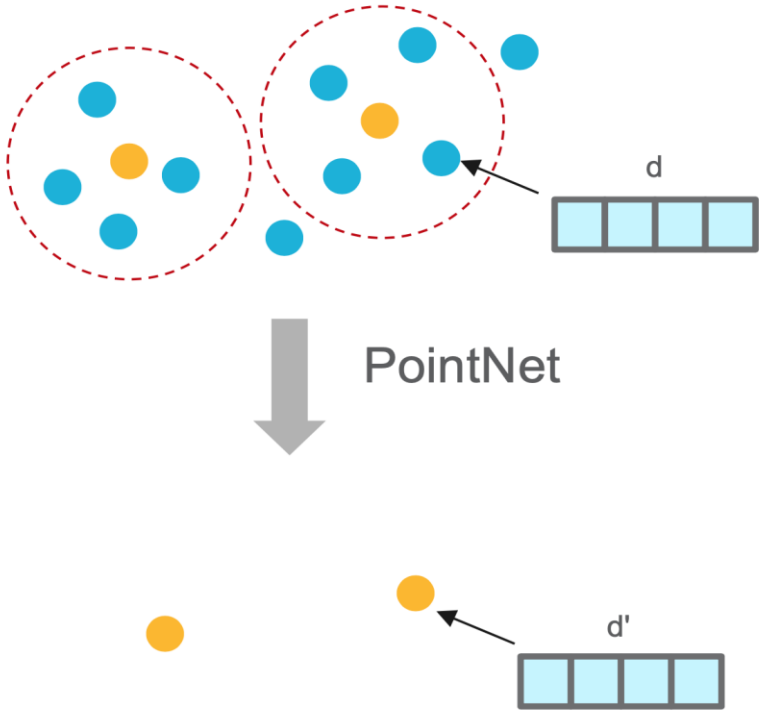
- Sampling (uniform/**farthest**)



- Grouping (*kNN*/**ball-query**)

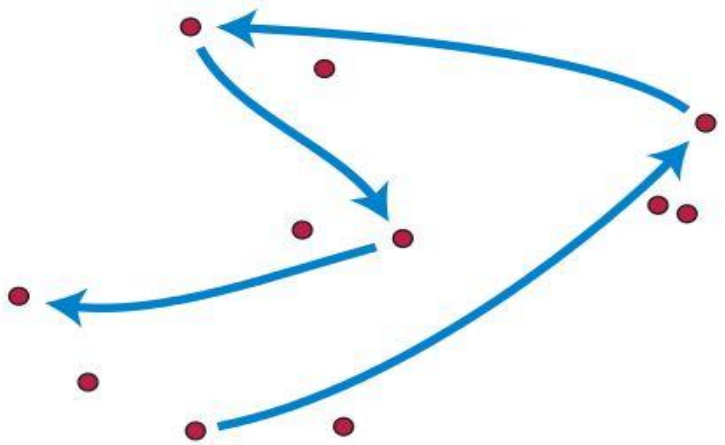


- Apply PointNet to each group

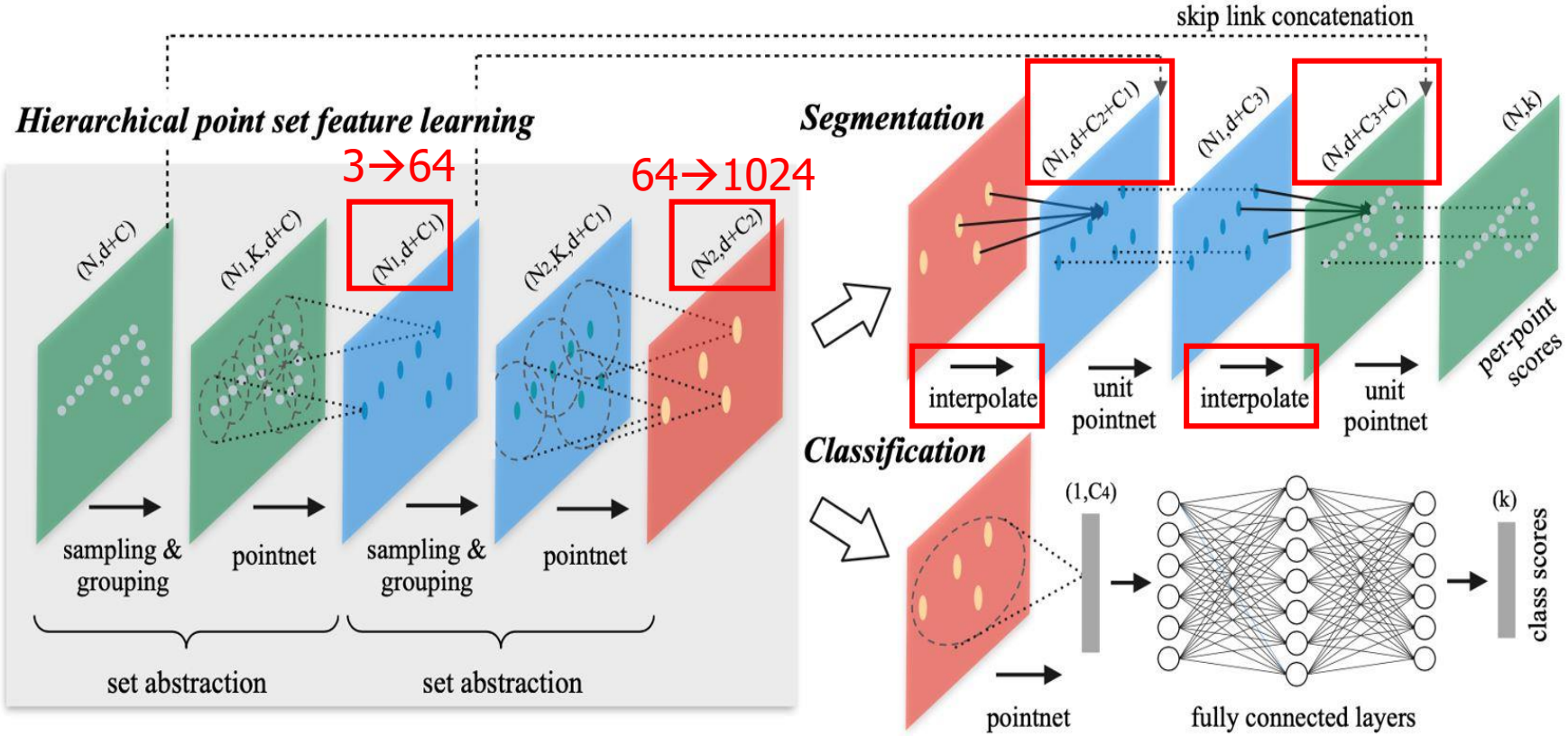


Farthest Point Sampling (FPS)

- The first five steps of the farthest-first traversal of a planar point set.
- The first point is chosen arbitrarily and each successive point is as far as possible from all previously chosen points.
- The new point which has maximum average distance is the farthest.

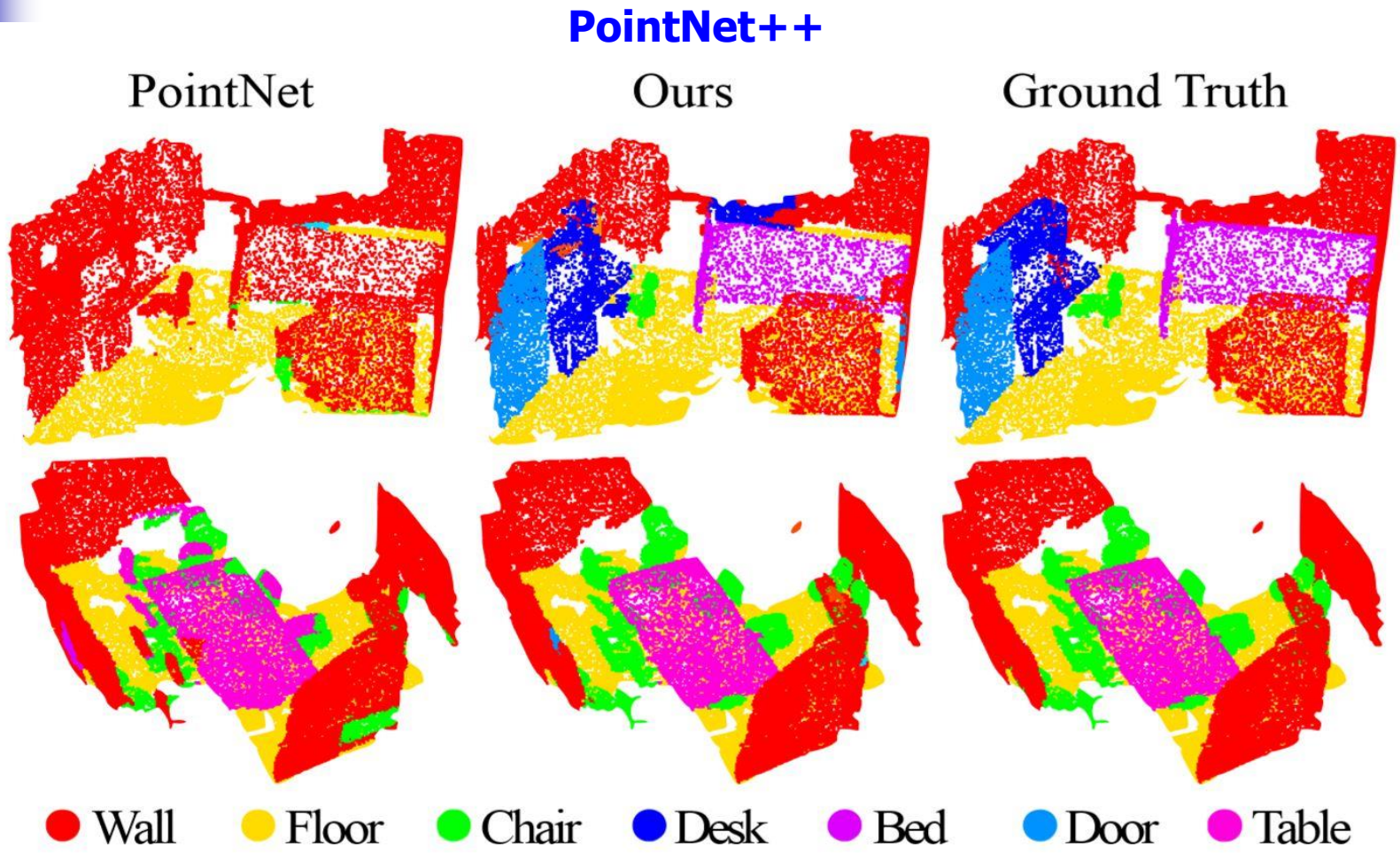


PointNet++



A set abstraction level takes an N × (d + C) matrix as input that is from N points with d-dim coordinates and C-dim point feature. It outputs an N1 × (d + C1) matrix of N1 subsampled points with d-dim coordinates and new C1-dim feature vectors summarizing local context.

Semantic Segmentation



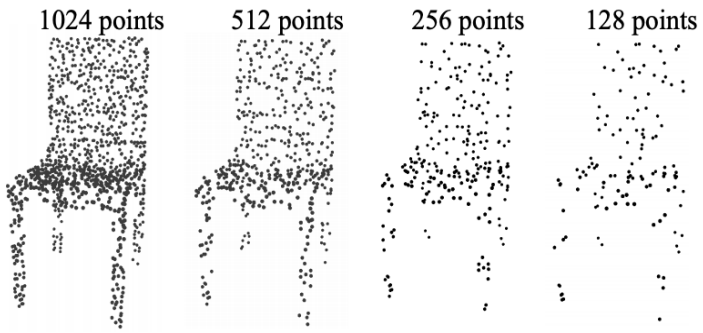
PointNet++

Method	Error rate (%)
Multi-layer perceptron [24]	1.60
LeNet5 [11]	0.80
Network in Network [13]	**0.47**
PointNet (vanilla) [20]	1.30
PointNet [20]	0.78
Ours	0.51

Table 1: MNIST digit classification.

Method	Input	Accuracy (%)
Subvolume [21]	vox	89.2
MVCNN [26]	img	90.1
PointNet (vanilla) [20]	pc	87.2
PointNet [20]	pc	89.2
Ours	pc	90.7
Ours (with normal)	pc	**91.9** face normal

Table 2: ModelNet40 shape classification.



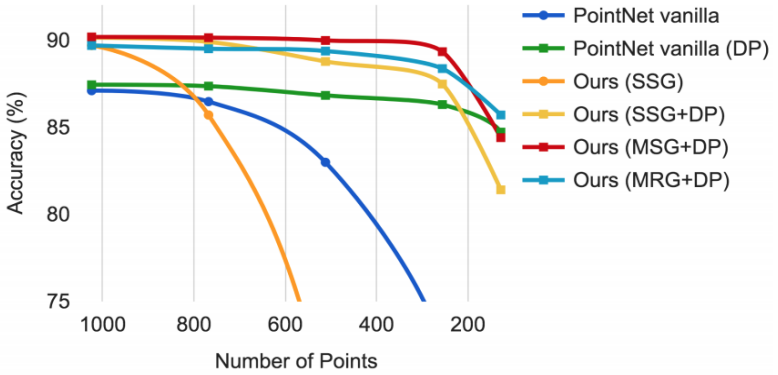
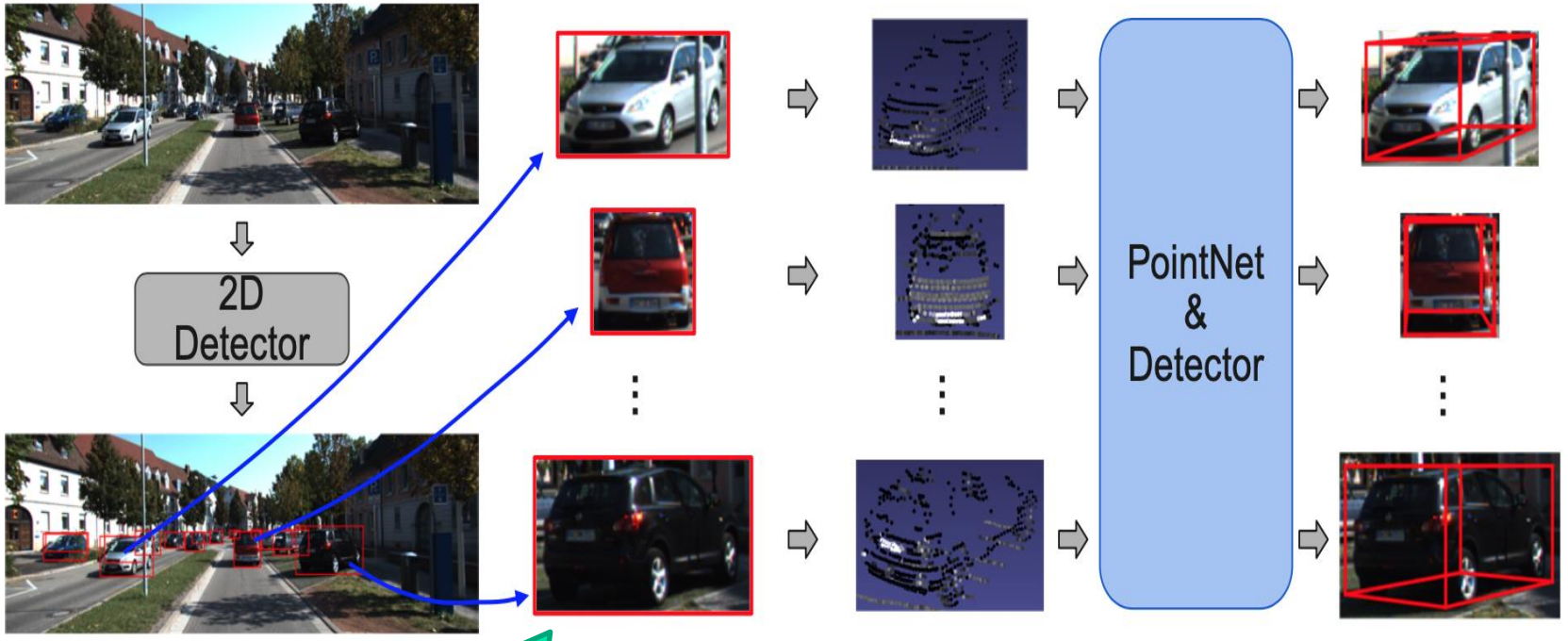


Figure 4: Left: Point cloud with random point dropout. Right: Curve showing advantage of our density adaptive strategy in dealing with non-uniform density. DP means random input dropout during training; otherwise training is on uniformly dense points. See Sec.3.3 for details.

Point Cloud Object Detection

Detection of Point Cloud: 2D Detections as Proposals



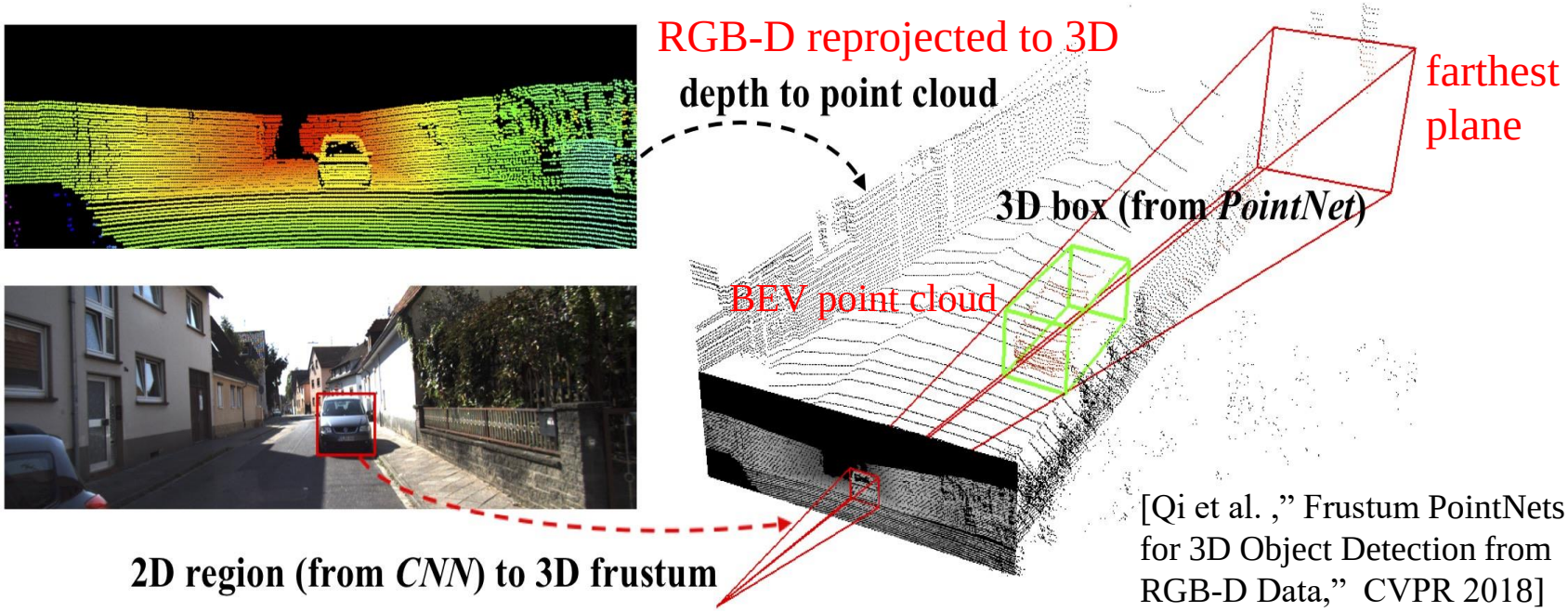
Highly reply on accurate 2D detections.

For each ROI, crop associate points.

3D Bbox Prediction

Frustum PointNets

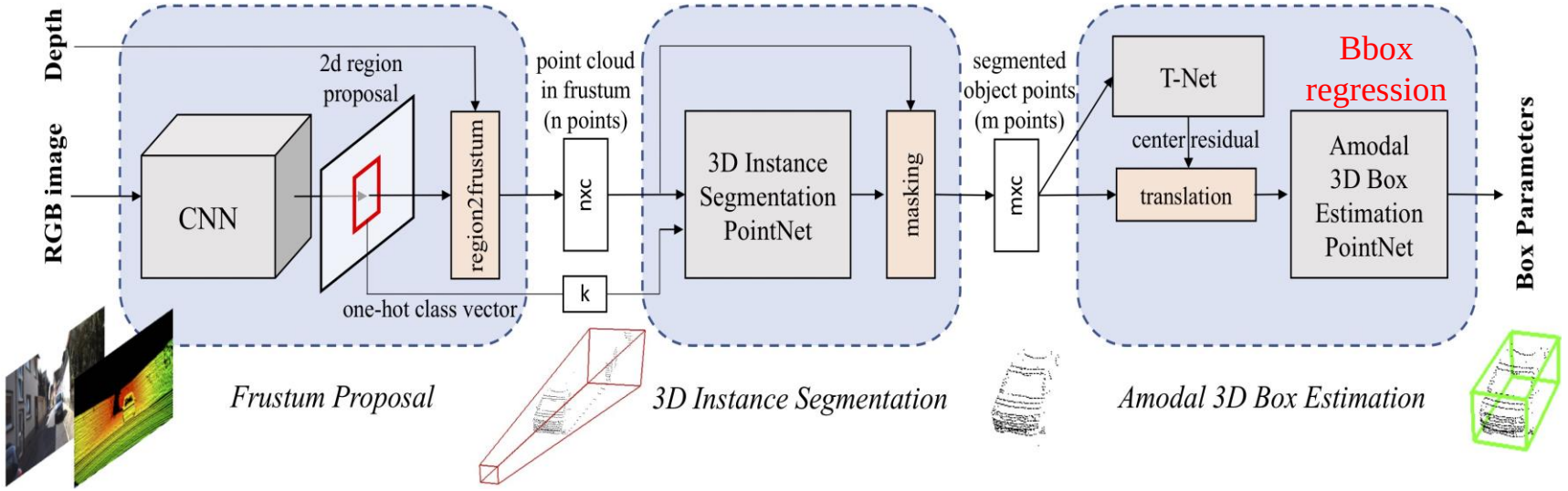
- Generating 2D object region proposals.
- 2D region extruded to a 3D viewing for related point clouds.
- Predicting a 3D bounding box from the points in frustum.



[Qi et al. ," Frustum PointNets for 3D Object Detection from RGB-D Data," CVPR 2018]

Frustum PointNets

- Frustum Proposal
  - Start with 2D object detection.
  - Using Feature Pyramid Network.
  - Adjust coordinate to frustum center.
- 3D Instance Segmentation
  - Point cloud segmentation in frustum.
  - Producing a point mask.
  - Using PointNet or PointNet++
  - Adjusting coordinate to point mask centroid.
- Amodal 3D Box Estimation
  - T-Net: Predicting object center & orientation.
  - PointNet or PointNet++ to estimate residual size & residual angle.



Frustum PointNets

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.