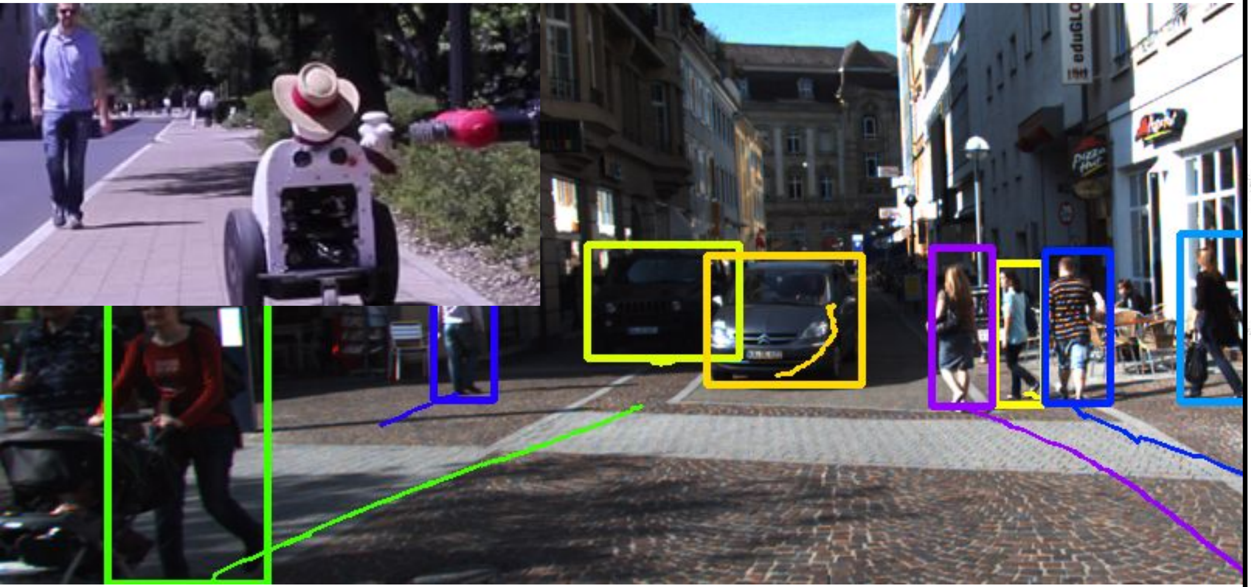
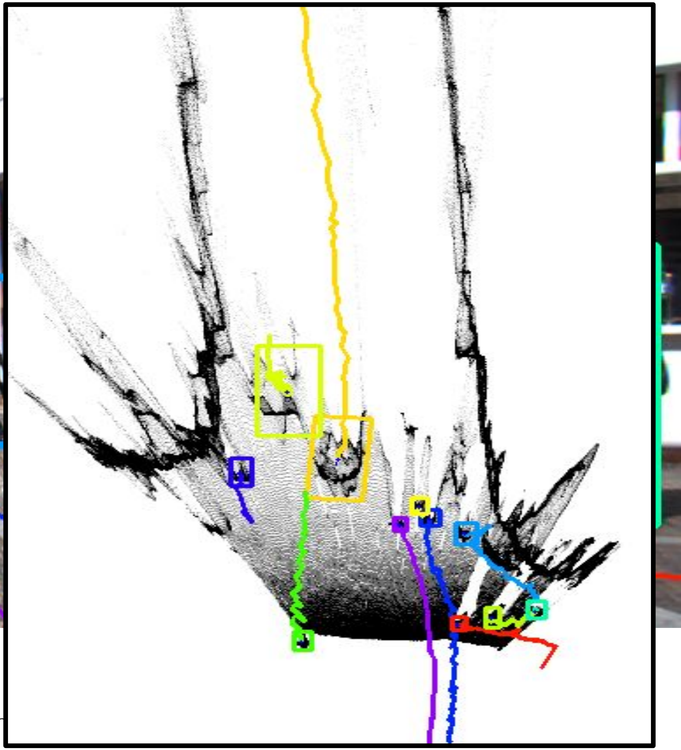
3D (Multi) Object Detection, Tracking and Segmentation

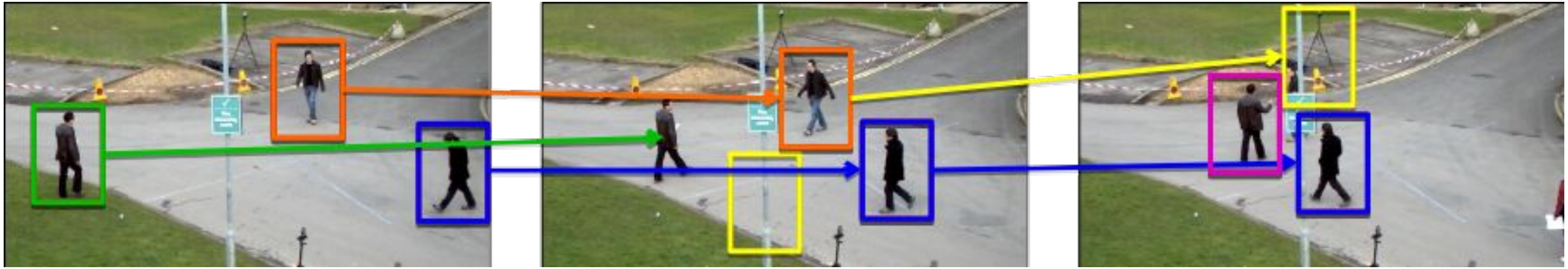
Motivation





Figures from Osep et al, Combined Image- and World-Space Tracking in Street Scenes, ICRA'18; Martín-Martín et al., JRDB: A Dataset and Benchmark for Visual Perception for Navigation in Human Environments

Reminder: Vision-based MOT



- Detect/segment objects
- Associate detections over time

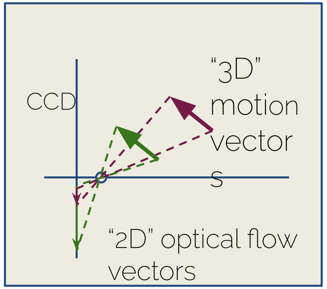
Predictions

Detections

0.9	0.8	0.8	0.3	0.3
0.5	0.4	0.7	0.3	0.3
0.2	0.1	0.4	0.3	0.3
0.1	0.2	0.5	0.3	0.3
0.3	0.3	0.3	0.3	0.3

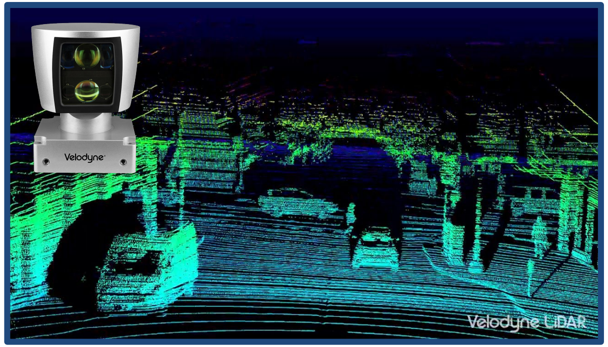
3D Detection and Tracking

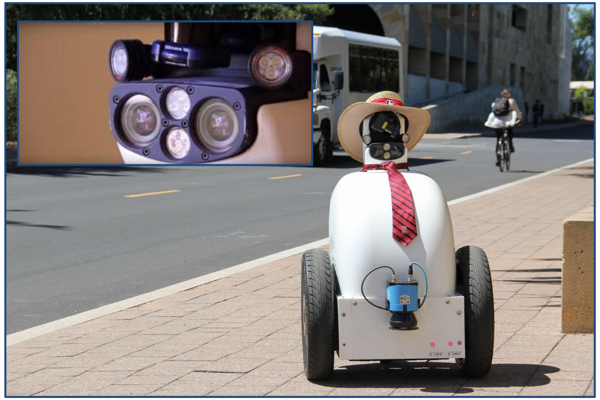
- Variety of sensors
  - Stereo, RGB-D cameras
  - LiDAR
- "Apparent" velocity



- Geometric constraints
  - In 2020, cars don't fly …

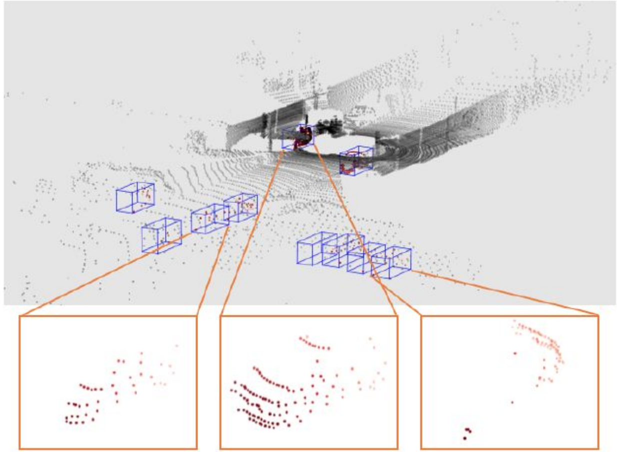
Bottom figure: Martín-Martín et al., JRDB: A Dataset and Benchmark for Visual Perception for Navigation in Human Environments



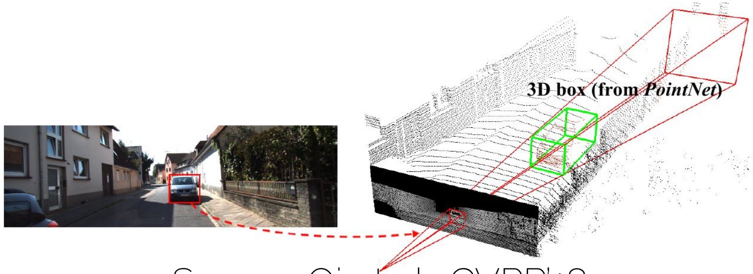


Challenges

- Depth sensor characteristics
  - Limited scan range
  - ``Non-cooperative`` materials
  - Sparse and unstructured signal
- Mobile platform
- Object localization in 3D



Source: Yuan et al., 3DV'19

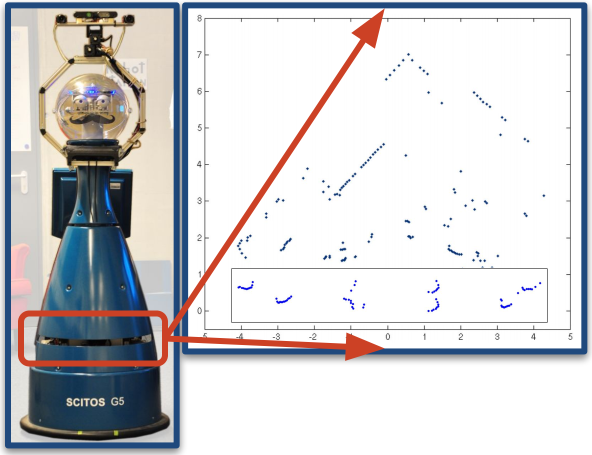


Source: Qi et al., CVPR'18

Historical Perspective

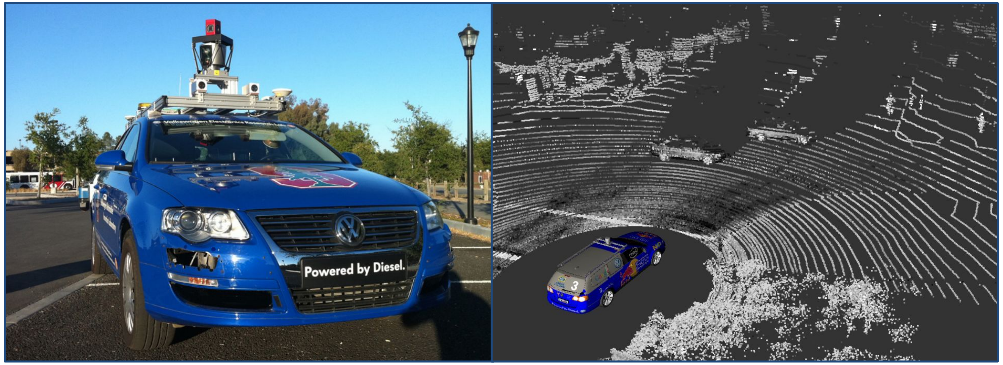
- Aeronautical, naval navigation
- Line laser scanners
- Stanley, '05 DARPA Grand Challenge Winner

Figures taken from:
Beyer et al., DROW: Real-Time Deep Learning based Wheelchair Detection in 2D Range Data, RAL'17;
Arras et al., Efficient People Tracking in Laser Range Data using a Multi-Hypothesis Leg-Tracker with Adaptive Occlusion Probabilities, ICRA'07

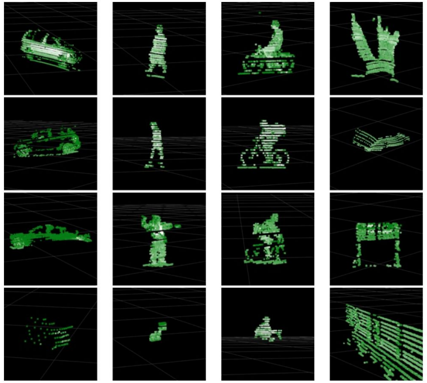


BACK IN MY DAYS
WE DIDNT
HAVE CONVNETS
imgflip.com

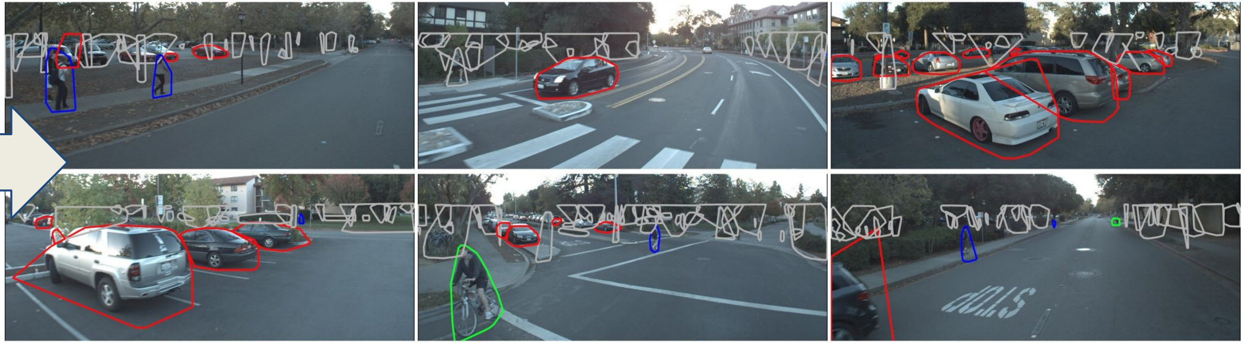
Tracking-before-Detection



Segment & Track



Classify



Teichman et al., Tracking-Based Semi-Supervised Learning, RSS'11

Segmentation is Difficult!

- Interacting objects, crowded scenes
- Sensor resolution decreasing with distance from the sensor, "holes" due to reflective and low-albedo surfaces

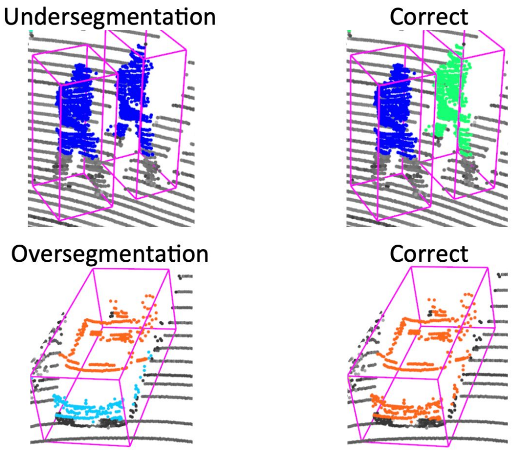


Figure from Held et al., A Probabilistic Framework for Real-time 3D Segmentation using Spatial, Temporal, and Semantic Cues, RSS'16

Stereo-vision Based MOT

- Vision: success of tracking-by-detection paradigm
- How to localize objects in 3D space?
  - Leibe et al., TPAMI'08; Ess et al., CVPR'08

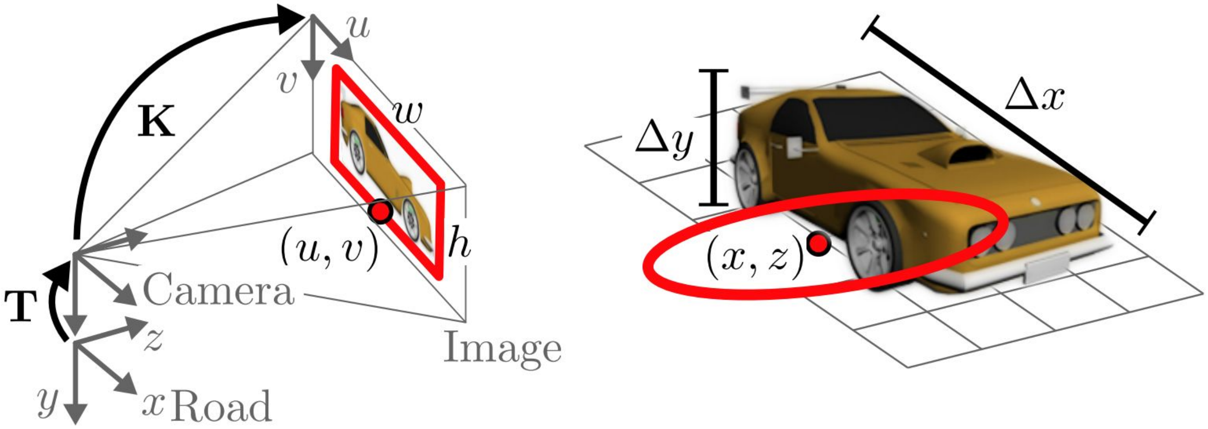
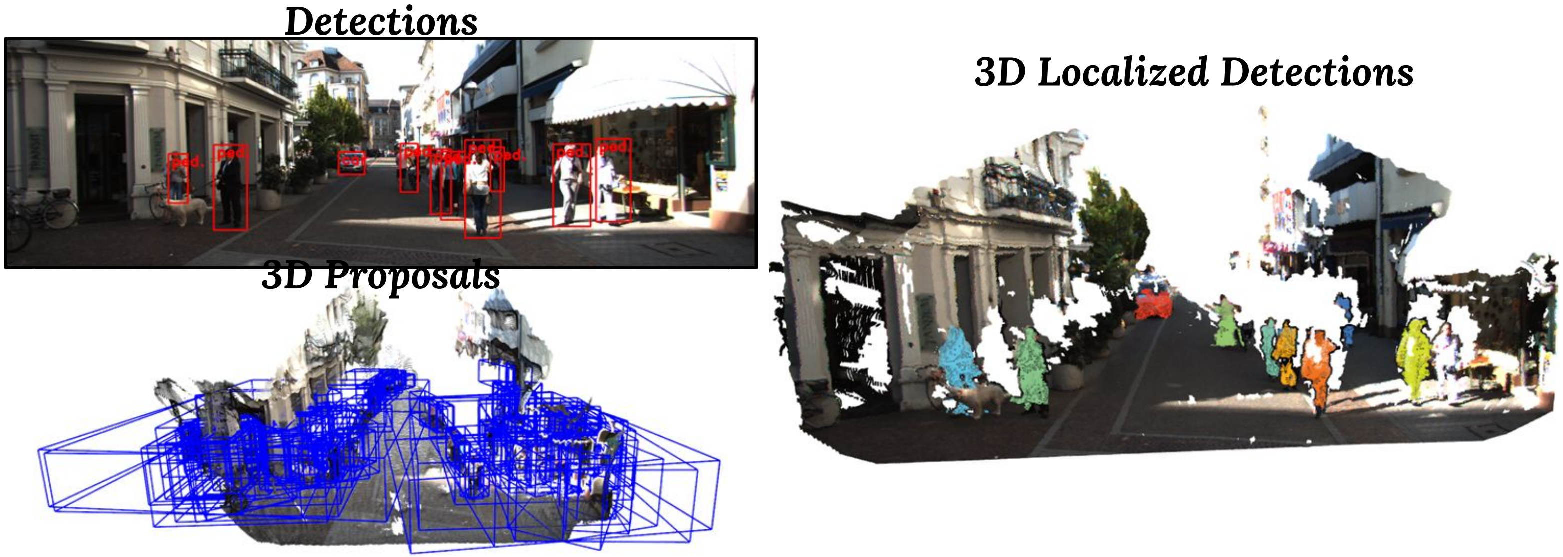


Figure: Andreas Geiger, Probabilistic Models for 3D Urban Scene Understanding from Movable Platforms, PhD thesis, 2013

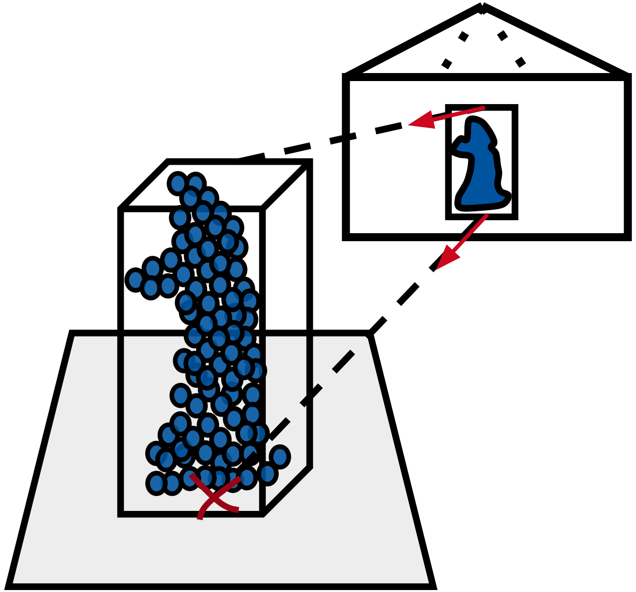
Stereo-vision Based MOT

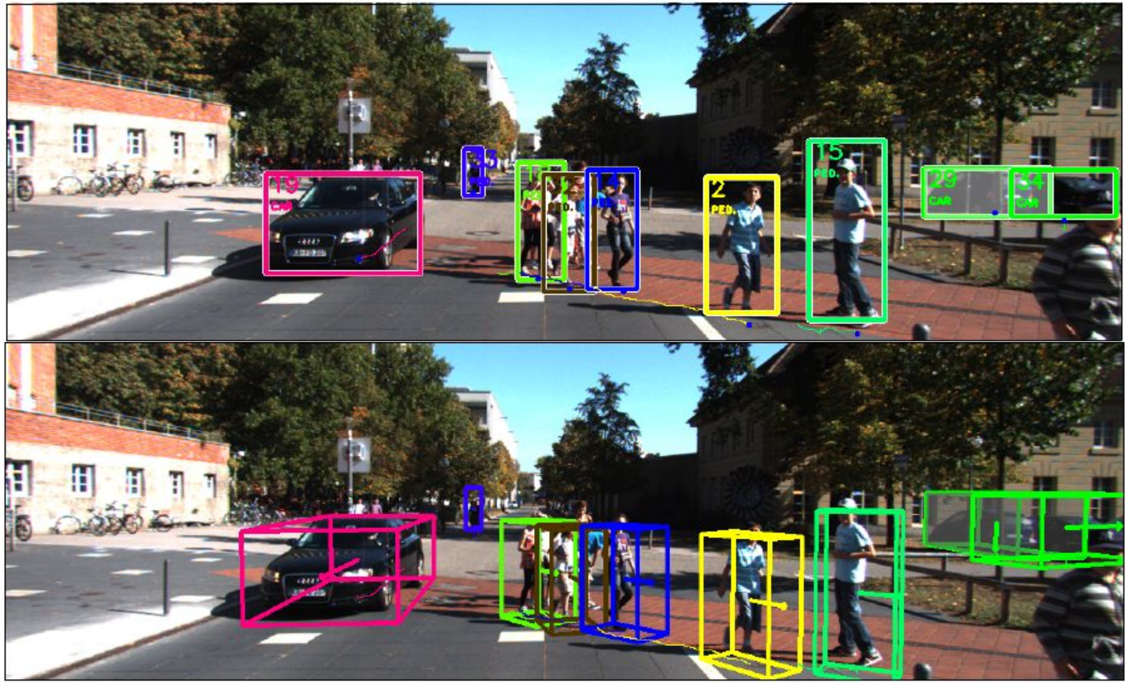
- Vision: success of tracking-by-detection paradigm
- How to localize objects in 3D space?



Osep et al., Combined Image- and World-Space Tracking in Street Scenes, ICRA'17

Stereo-vision Based MOT





Stereo-vision Based MOT

- CIWT still got it (KITTI MOT2D, Regionlets) …

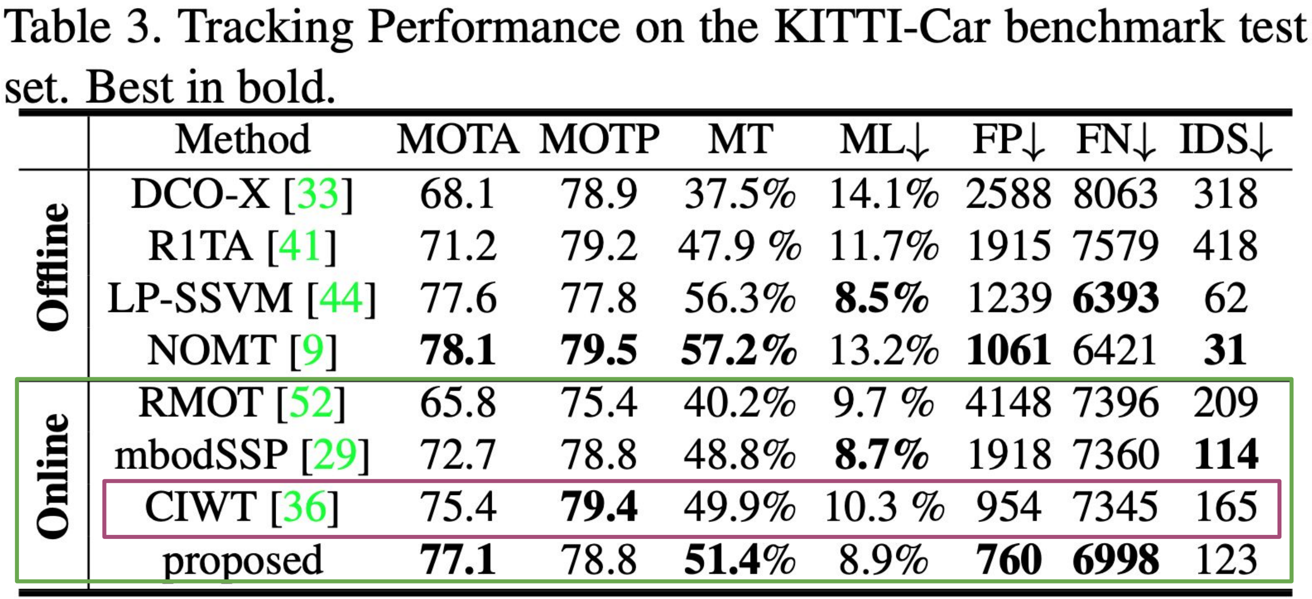


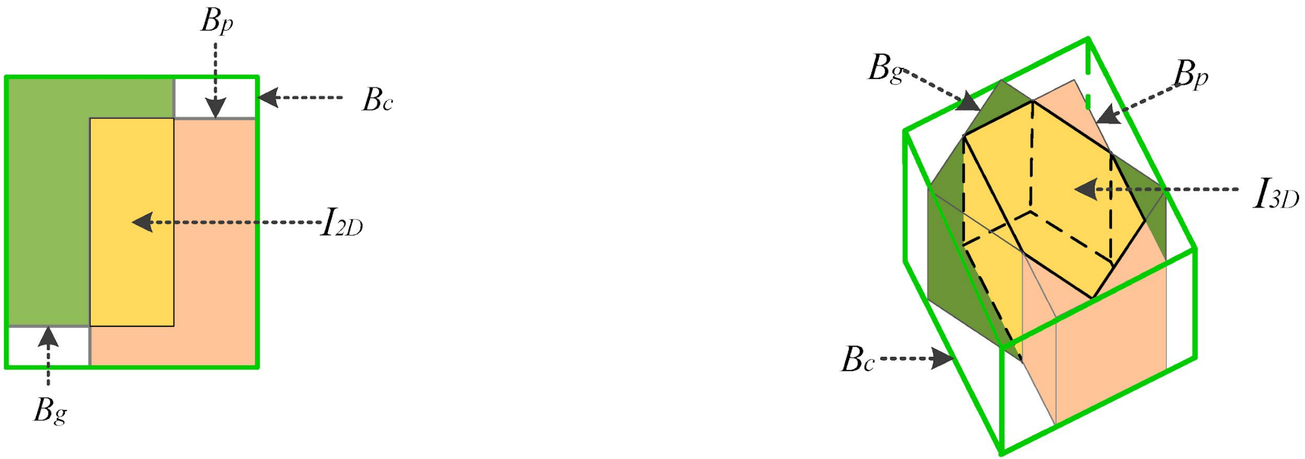
Table 3. Tracking Performance on the KITTI-Car benchmark test set. Best in bold.

	Method	MOTA	MOTP	MT	ML↓	FP↓	FN↓	IDS↓
Offline	DCO-X [33]	68.1	78.9	37.5%	14.1%	2588	8063	318
	R1TA [41]	71.2	79.2	47.9 %	11.7%	1915	7579	418
	LP-SSVM [44]	77.6	77.8	56.3%	**8.5%**	1239	**6393**	62
	NOMT [9]	**78.1**	**79.5**	**57.2%**	13.2%	**1061**	6421	**31**
Online	RMOT [52]	65.8	75.4	40.2%	9.7 %	4148	7396	209
	mbodSSP [29]	72.7	78.8	48.8%	**8.7%**	1918	7360	**114**
	CIWT [36]	75.4	**79.4**	49.9%	10.3 %	954	7345	165
	proposed	**77.1**	78.8	**51.4%**	8.9%	**760**	**6998**	123

Chu et al.., FAMNet: Joint Learning of Feature, Affinity and Multi-dimensional Assignment for Online Multiple Object Tracking, ICCV'19

A Note on the Evaluation

- As before: mAP, MOTA
- 3D IoU



(a) 2D

(b) 3D

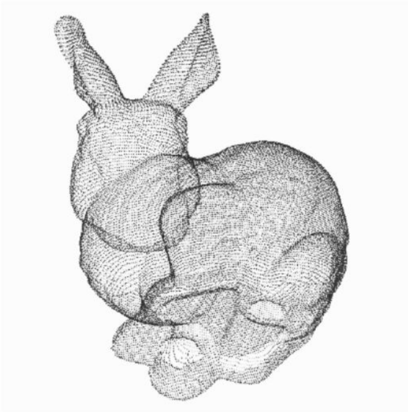
Figure taken from Xu et al., 3D-GIoU: 3D Generalized Intersection over Union for Object Detection in Point Cloud, Sensors'19

3D Object Detection

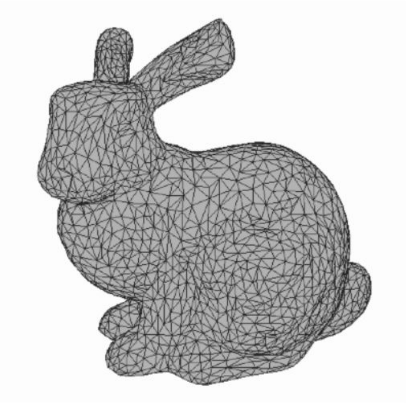
Part I.

Deep Learning on Point Clouds

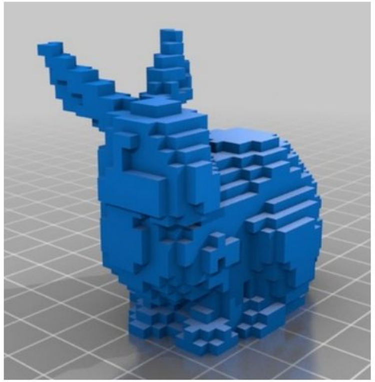
- Signal representation?



Point Cloud



Mesh

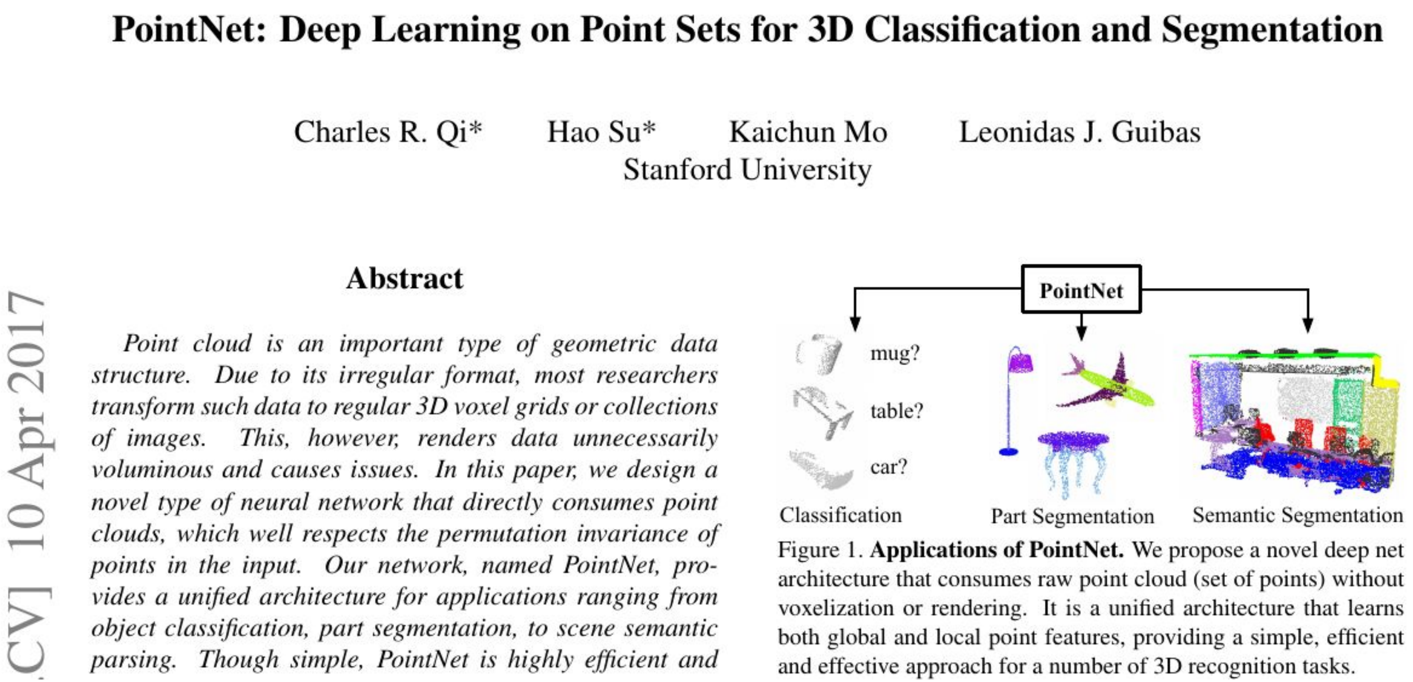


Volumetric

Projected View RGB(D)

Slides adapted from Charles Qi CVPR presentation slides (https://web.stanford.edu/~rqi/pointnet/docs/cvpr17_pointnet_slides.pdf)

Deep Learning on Unordered Sets



PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation

Charles R. Qi* Hao Su* Kaichun Mo Leonidas J. Guibas
Stanford University

.CV] 10 Apr 2017

Abstract

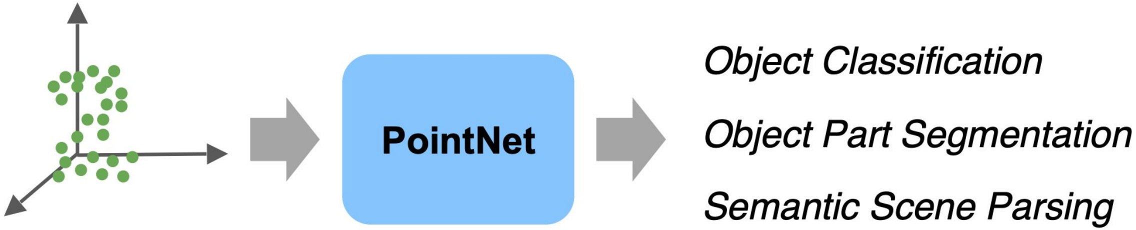
Point cloud is an important type of geometric data structure. Due to its irregular format, most researchers transform such data to regular 3D voxel grids or collections of images. This, however, renders data unnecessarily voluminous and causes issues. In this paper, we design a novel type of neural network that directly consumes point clouds, which well respects the permutation invariance of points in the input. Our network, named PointNet, provides a unified architecture for applications ranging from object classification, part segmentation, to scene semantic parsing. Though simple, PointNet is highly efficient and

Figure 1. **Applications of PointNet.** We propose a novel deep net architecture that consumes raw point cloud (set of points) without voxelization or rendering. It is a unified architecture that learns both global and local point features, providing a simple, efficient and effective approach for a number of 3D recognition tasks.

- Seminal paper by Qi et al., CVPR'17
- Game-changer

Deep Learning on Point Clouds

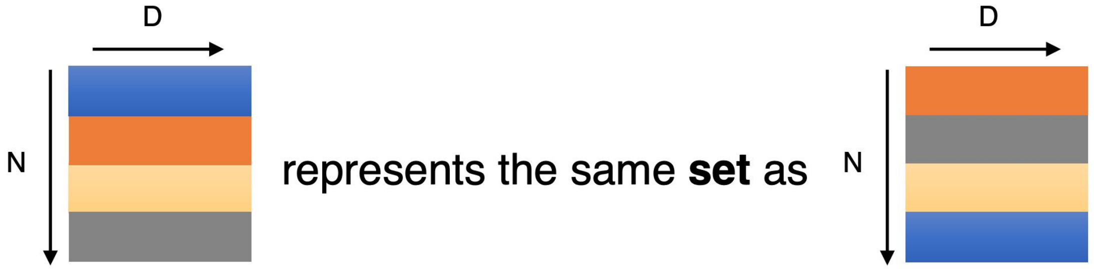
- End-to-end learning for scattered, unordered point data



- Challenges:
  - Unordered: Model needs to be invariant to N! permutations.
  - Invariance under geometric transformations: Point cloud rotations should not alter classification results.

Slides adapted from Charles Qi CVPR presentation slides (https://web.stanford.edu/~rqi/pointnet/docs/cvpr17_pointnet_slides.pdf)

Permutation Invariance



represents the same **set** as

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

Examples:

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

...

- How can we construct a family of symmetric functions by neural networks?

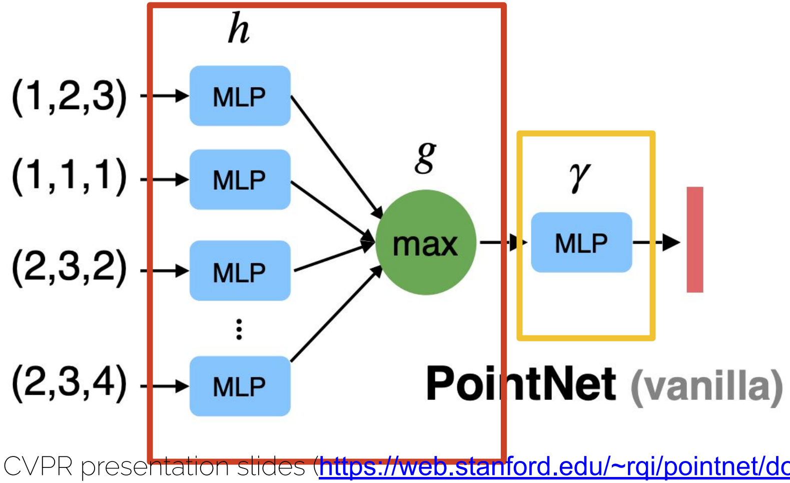
Slides adapted from Charles Qi CVPR presentation slides (https://web.stanford.edu/~rqi/pointnet/docs/cvpr17_pointnet_slides.pdf)

Vanilla PointNet

- Observe:

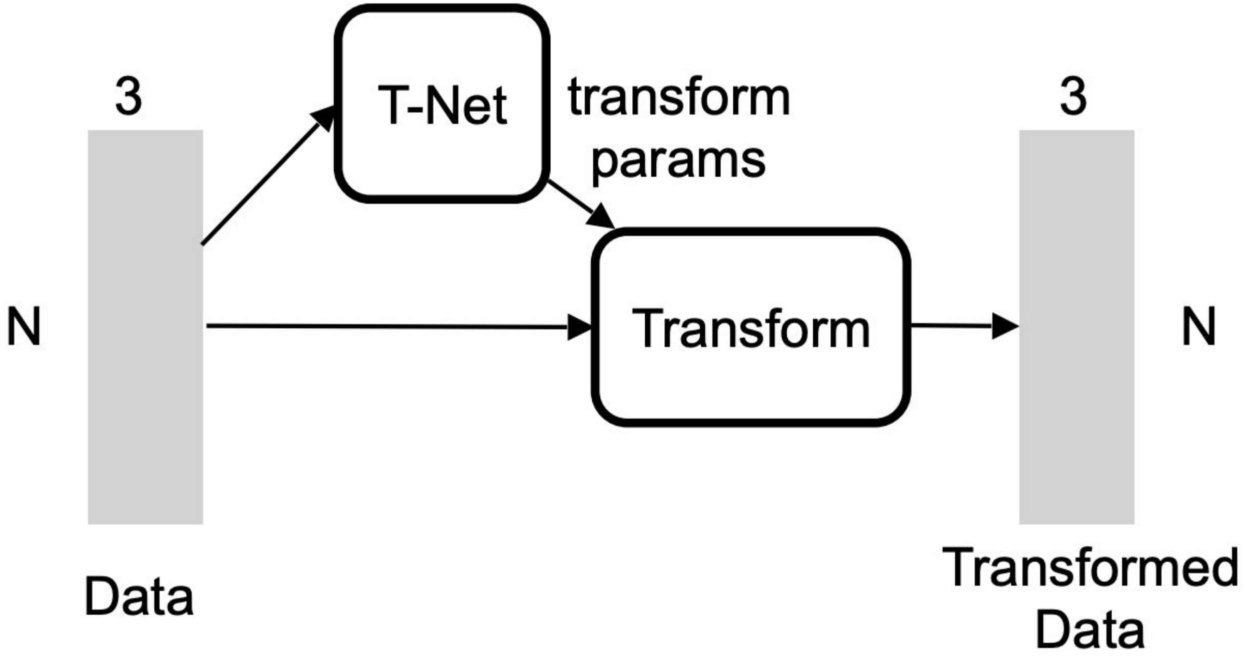
$$f(x_1,x_2,\ldots,x_n) = \boxed{\gamma} \circ \boxed{g(h(x_1),\ldots,h(x_n))} \text{ is symmetric if } g \text{ is symmetric}$$

- PointNet: MLP + max pooling



Slides adapted from Charles Qi CVPR presentation slides (https://web.stanford.edu/~rqi/pointnet/docs/cvpr17_pointnet_slides.pdf)

Invariance to Transformations



Slides adapted from Charles Qi CVPR presentation slides (https://web.stanford.edu/~rqi/pointnet/docs/cvpr17_pointnet_slides.pdf)

PointNet++

- Ok cool, but:
  - PointNet does not capture local structures
  - Global representation depend on absolute coordinates -- poor generalization
- Idea:
  - Apply PointNet recursively on a nested partitioning of the input point set
  - Learn local features with increasing contextual scales
  - "Multi-scale point-net"

PointNet++

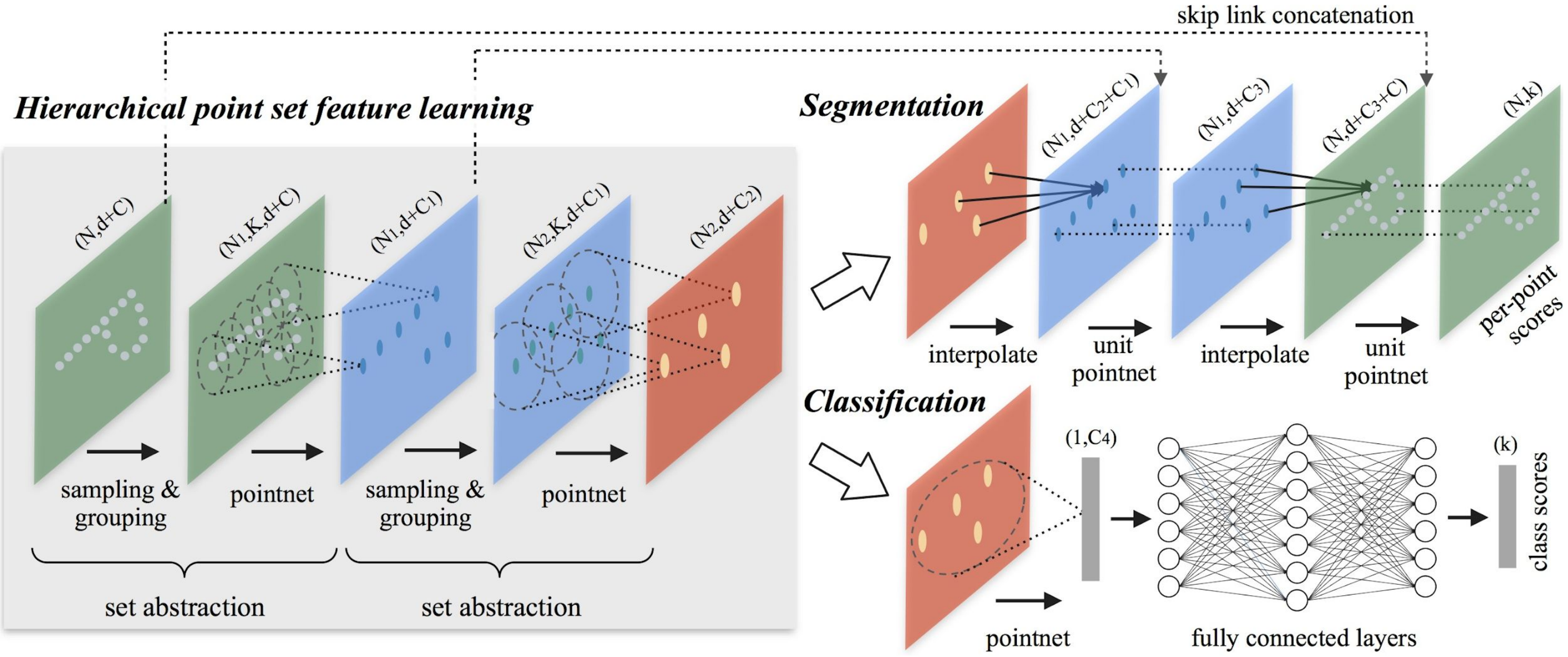
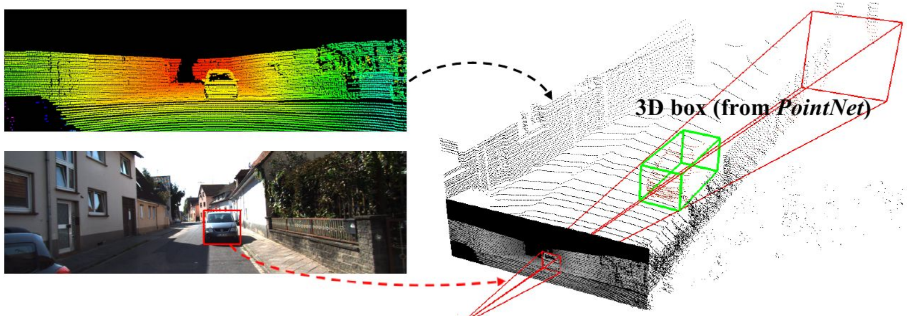
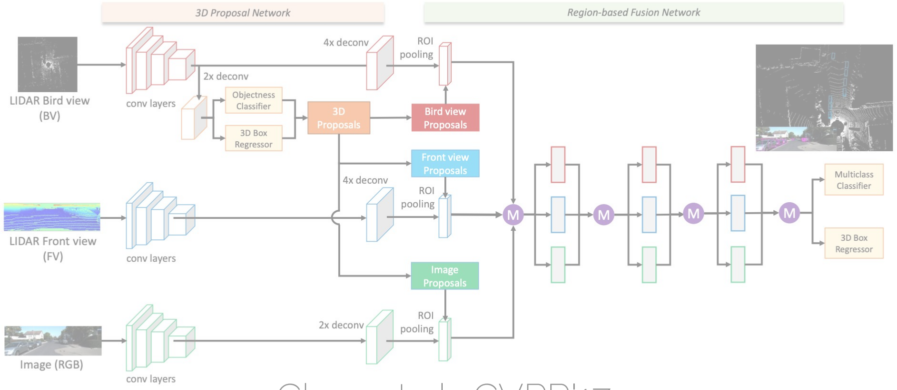


Figure from Qi et al., PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space, NIPS'17

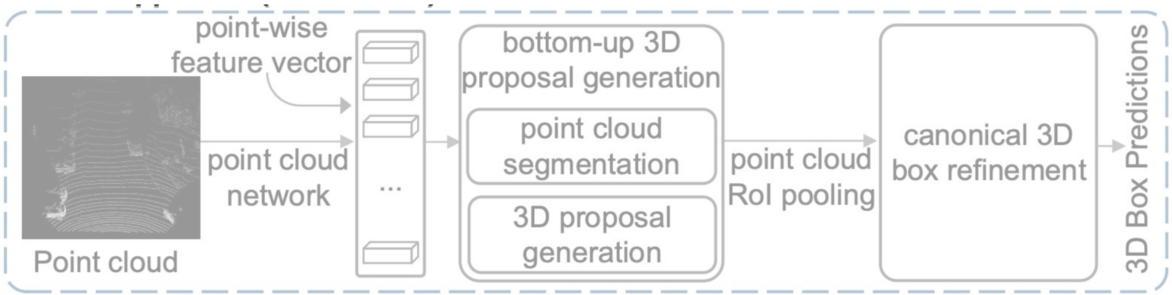
3D Object Detection Landscape



Qi et al., CVPR'18



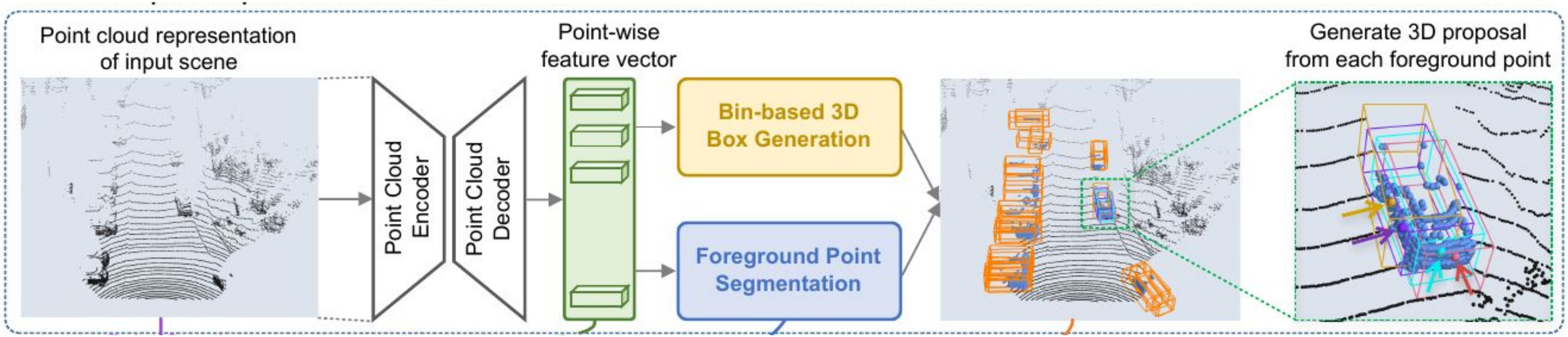
Chen et al., CVPR'17



Shi et al., CVPR'19

Point RCNN

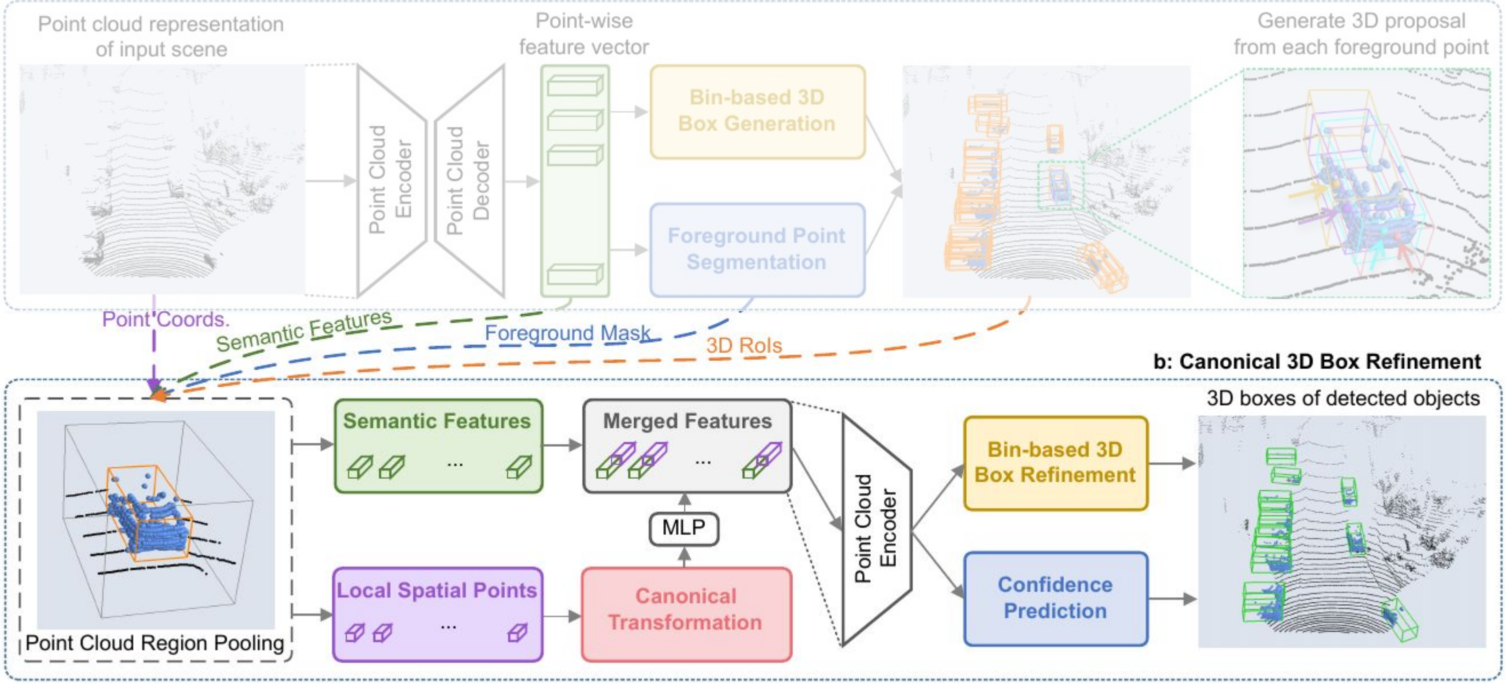
- Two-stage detector (Faster R-CNN!)
- Stage-1: proposal generation


Point cloud representation of input scene
Point Cloud Encoder
Point Cloud Decoder
Point-wise feature vector
Bin-based 3D Box Generation
Foreground Point Segmentation
Generate 3D proposal from each foreground point

Shi et al., PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud, CVPR'19

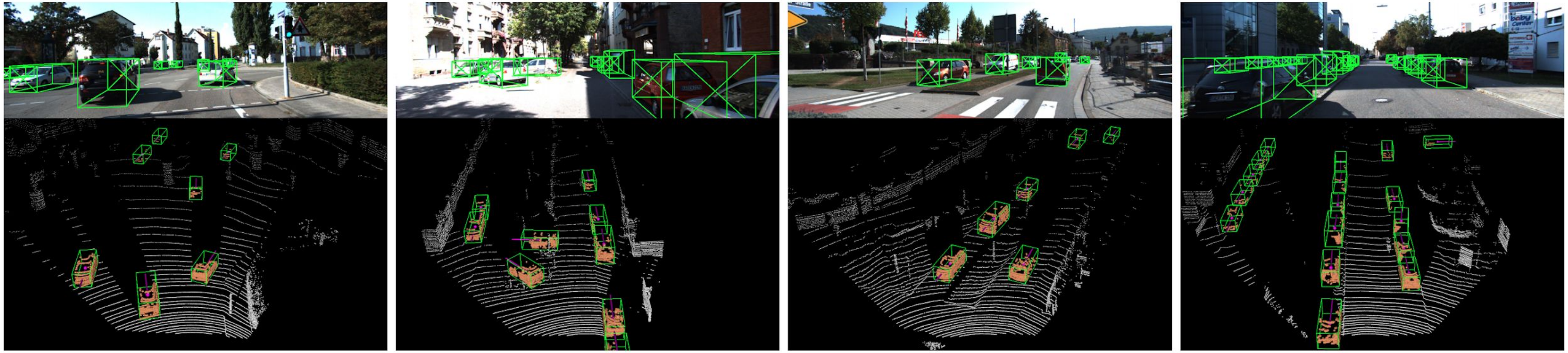
Point RCNN

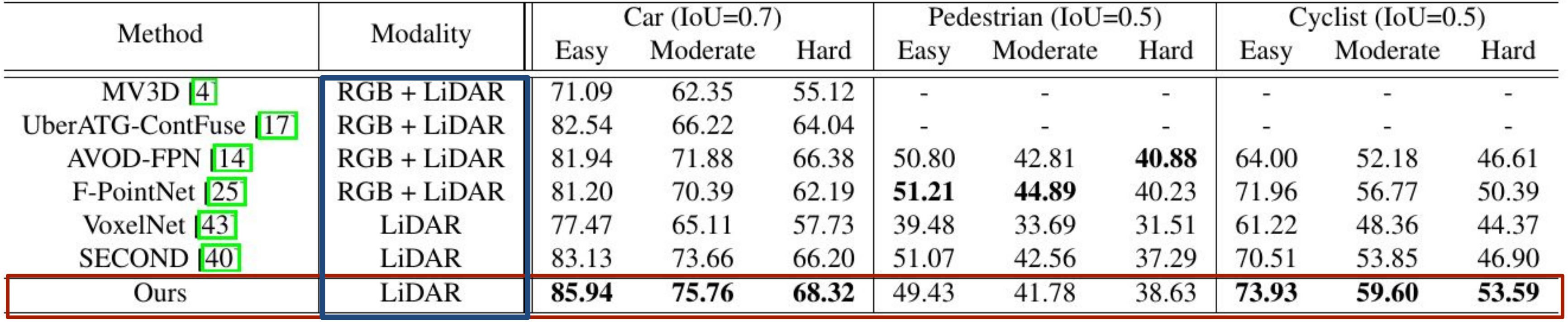
- Stage-II


Point cloud representation of input scene
Point-wise feature vector
Generate 3D proposal from each foreground point
Point Cloud Encoder
Point Cloud Decoder
Bin-based 3D Box Generation
Foreground Point Segmentation
Point Coords.
Semantic Features
Foreground Mask
3D RoIs
b: Canonical 3D Box Refinement
3D boxes of detected objects
Semantic Features
...
Merged Features
...
MLP
Local Spatial Points
...
Canonical Transformation
Point Cloud Encoder
Bin-based 3D Box Refinement
Confidence Prediction
Point Cloud Region Pooling

Shi et al., PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud, CVPR'19

Point RCNN





Method	Modality	Car (IoU=0.7)			Pedestrian (IoU=0.5)			Cyclist (IoU=0.5)		
		Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
MV3D [4]	RGB + LiDAR	71.09	62.35	55.12	-	-	-	-	-	-
UberATG-ContFuse [17]	RGB + LiDAR	82.54	66.22	64.04	-	-	-	-	-	-
AVOD-FPN [14]	RGB + LiDAR	81.94	71.88	66.38	50.80	42.81	**40.88**	64.00	52.18	46.61
F-PointNet [25]	RGB + LiDAR	81.20	70.39	62.19	**51.21**	**44.89**	40.23	71.96	56.77	50.39
VoxelNet [43]	LiDAR	77.47	65.11	57.73	39.48	33.69	31.51	61.22	48.36	44.37
SECOND [40]	LiDAR	83.13	73.66	66.20	51.07	42.56	37.29	70.51	53.85	46.90
Ours	LiDAR	**85.94**	**75.76**	**68.32**	49.43	41.78	38.63	**73.93**	**59.60**	**53.59**

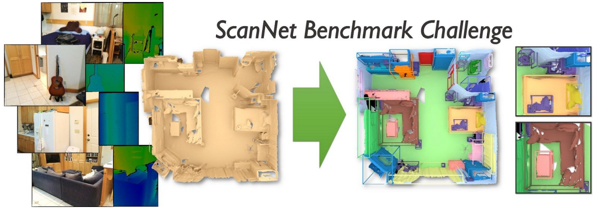
Shi et al., PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud, CVPR'19

3D Segmentation

Part II.

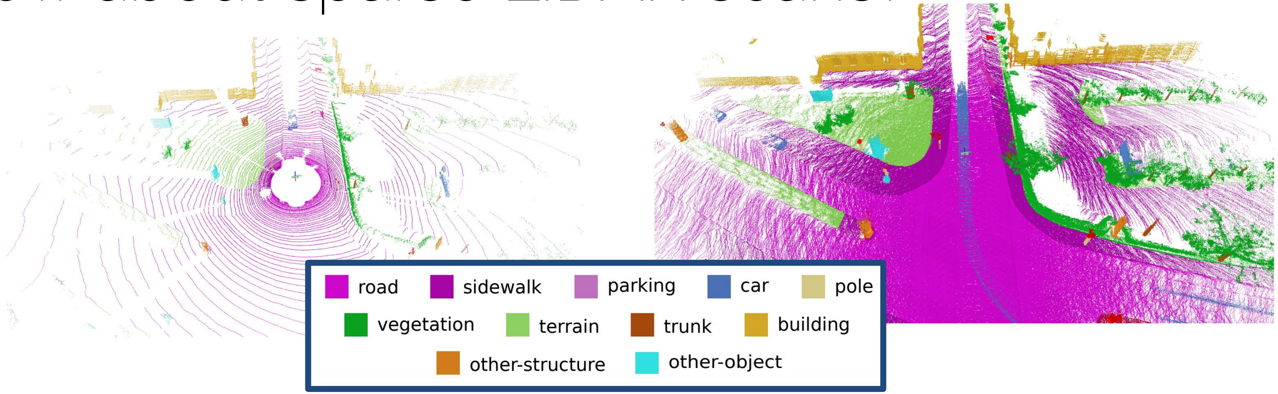
3D Semantic Segmentation

- Existing datasets (Dense, pre-aligned RGB-D)



Dai et al., ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes, CVPR'17

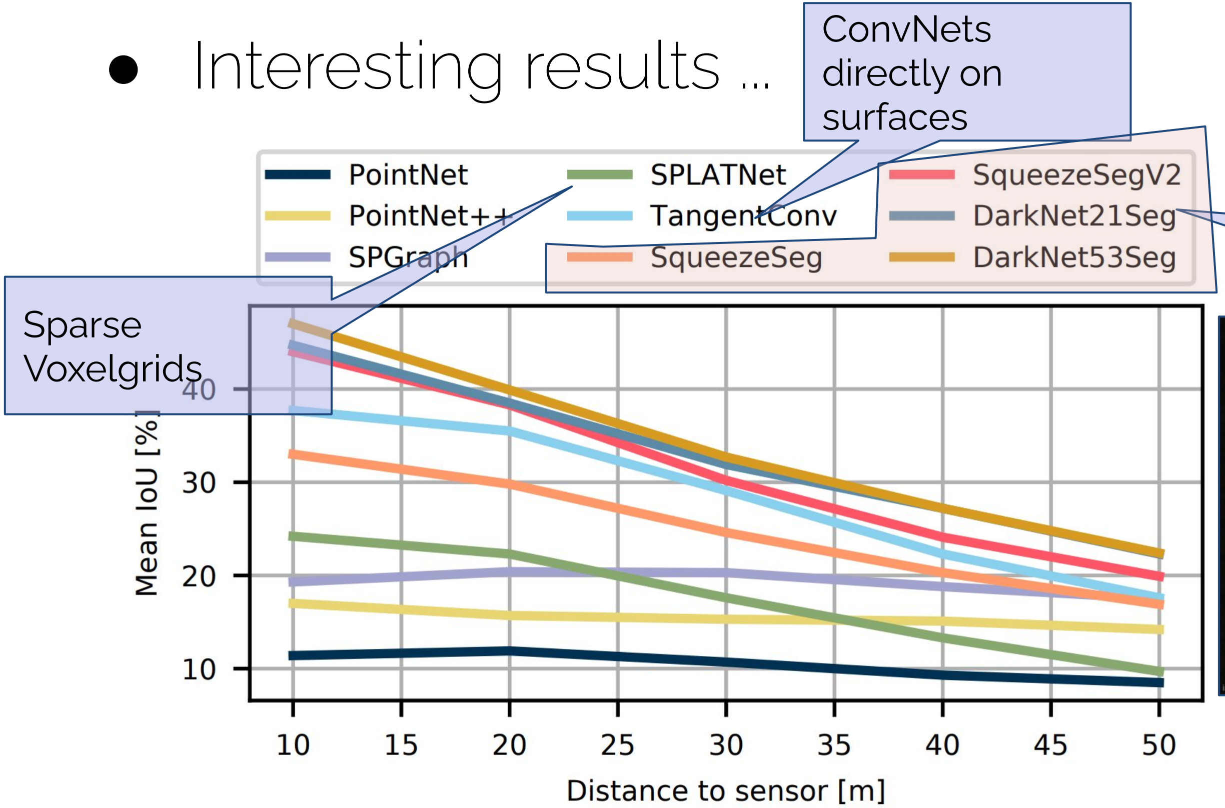
- How about sparse LiDAR scans?



Behley et al., SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences, ICCV'19

Signal Representation?

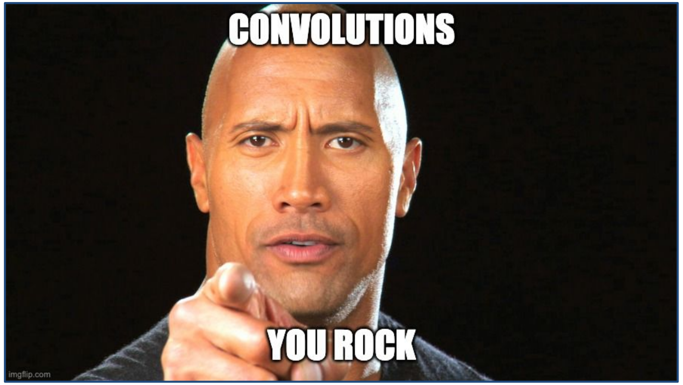
- Interesting results ...



ConvNets directly on surfaces

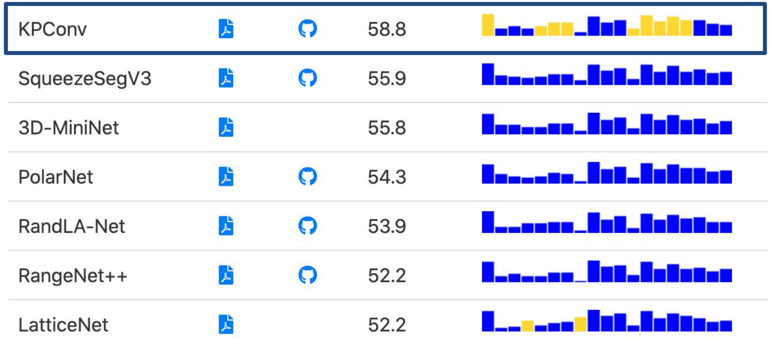
Sparse Voxelgrids

Spherical projection + CNNs

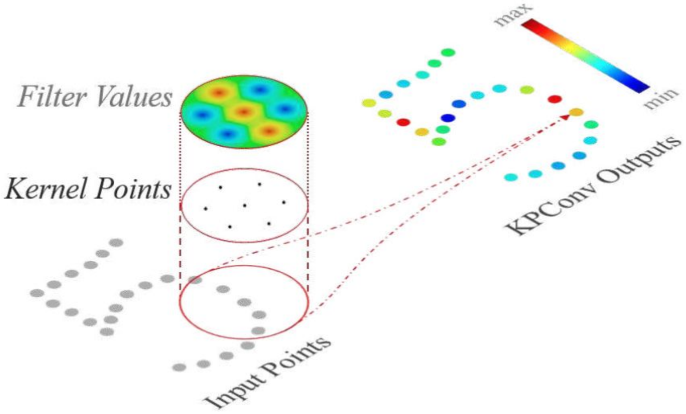


Comeback for Raw Point Clouds + Convolutions

- Kernel Point Convolution

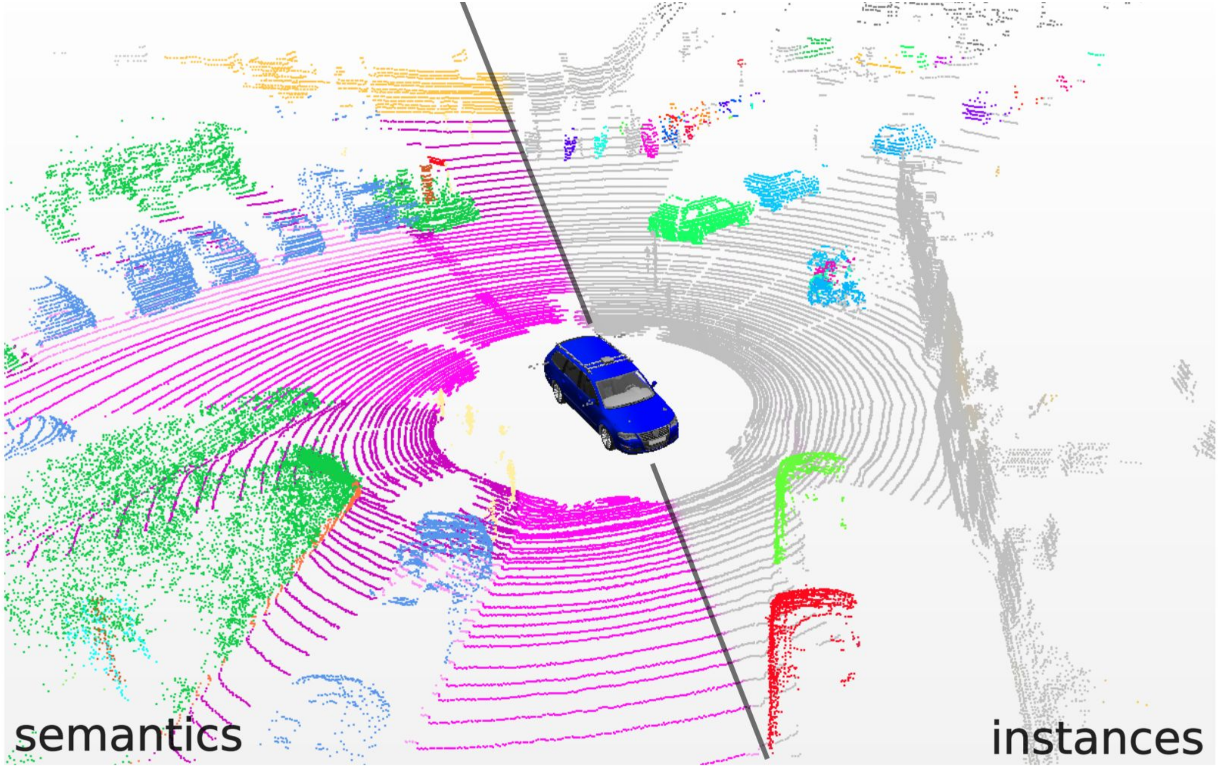


			mIoU	per-class
KPConv	📄	ⓖ	58.8	
SqueezeSegV3	📄	ⓖ	55.9	
3D-MiniNet	📄		55.8	
PolarNet	📄	ⓖ	54.3	
RandLA-Net	📄	ⓖ	53.9	
RangeNet++	📄	ⓖ	52.2	
LatticeNet	📄		52.2	



Thomas et al.,, KPConv: Flexible and Deformable Convolution for Point Clouds, ICCV'19

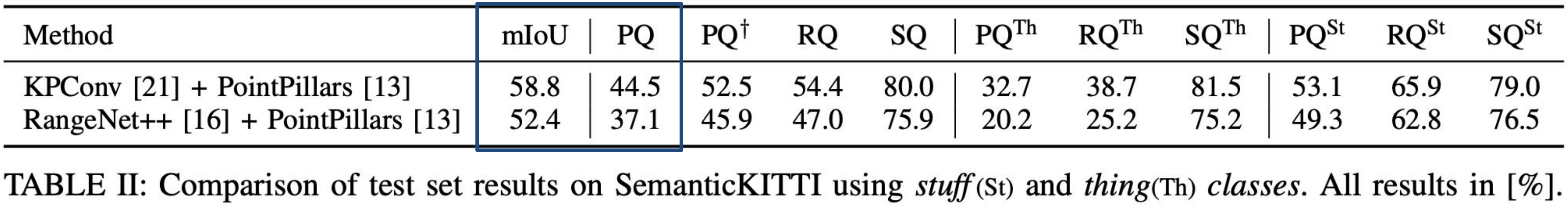
LiDAR Panoptic Segmentation



Behley et al., A Benchmark for LiDAR-based Panoptic Segmentation based on KITTI, arXiv:2003.02371

LiDAR Panoptic Segmentation

- Simple baseline
  - Compute semantic segmentation, object detections
  - Fuse the results (heuristic postprocessing)



Method	mIoU	PQ	PQ†	RQ	SQ	PQ$^{\mathrm{Th}}$	RQ$^{\mathrm{Th}}$	SQ$^{\mathrm{Th}}$	PQ$^{\mathrm{St}}$	RQ$^{\mathrm{St}}$	SQ$^{\mathrm{St}}$
KPConv [21] + PointPillars [13]	58.8	44.5	52.5	54.4	80.0	32.7	38.7	81.5	53.1	65.9	79.0
RangeNet++ [16] + PointPillars [13]	52.4	37.1	45.9	47.0	75.9	20.2	25.2	75.2	49.3	62.8	76.5

TABLE II: Comparison of test set results on SemanticKITTI using *stuff* (St) and *thing*(Th) *classes*. All results in [%].

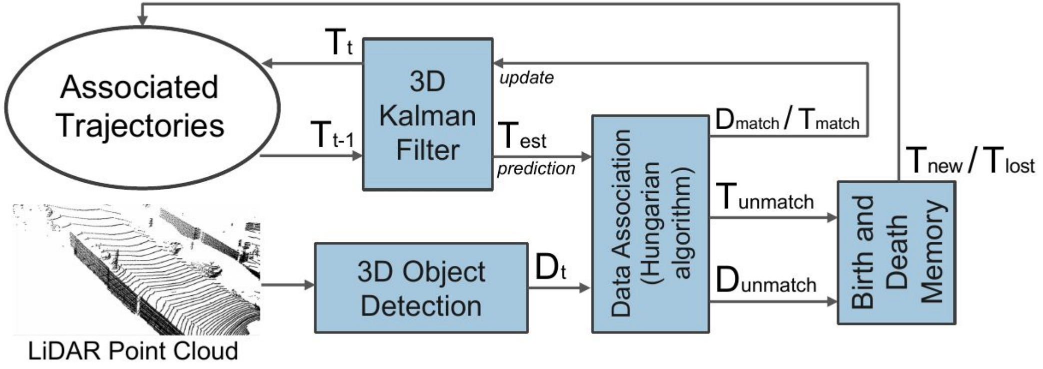
- Cool research opportunities
  - End-to-end learning
  - 3D Panoptic segmentation and tracking

3D MOT

Part II.

AB3D-MOT

- ``Embarrassingly simple``, great performance!
  - Bi-partite matching, 3D IoU
  - Dynamics model: const-velocity Kalman Filter
  - Why does this simple approach work so well in this case?
    - => Strong 3D detectors, motion models reliable in 3D



$$\mathbf{x}^k = [x, y, z, \theta, w, h, l, s, v_x, v_y, v_z]$$

$$f(\mathbf{x}):$$

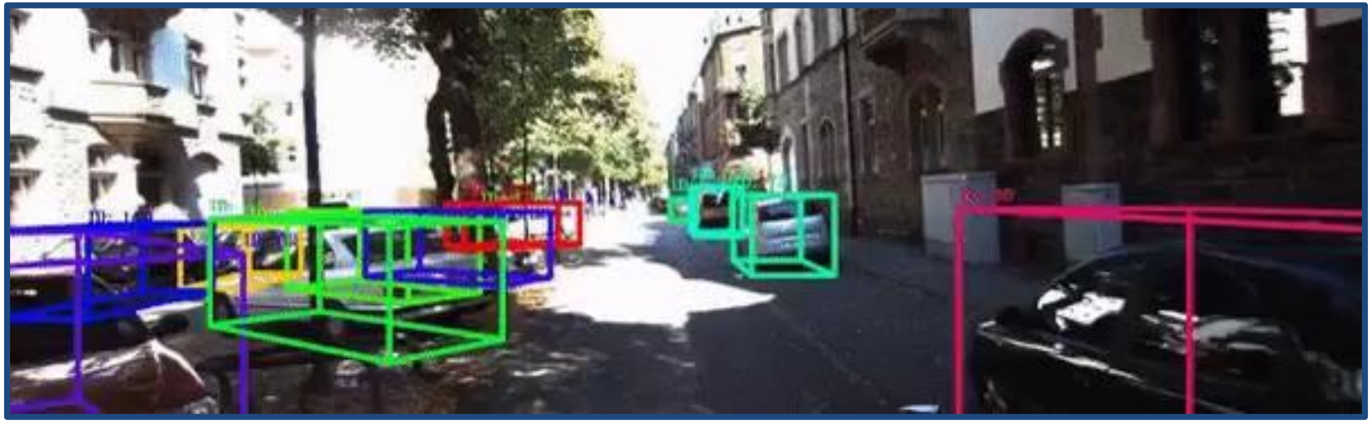
$$\hat{x}^{k+1} = x + v_x,$$

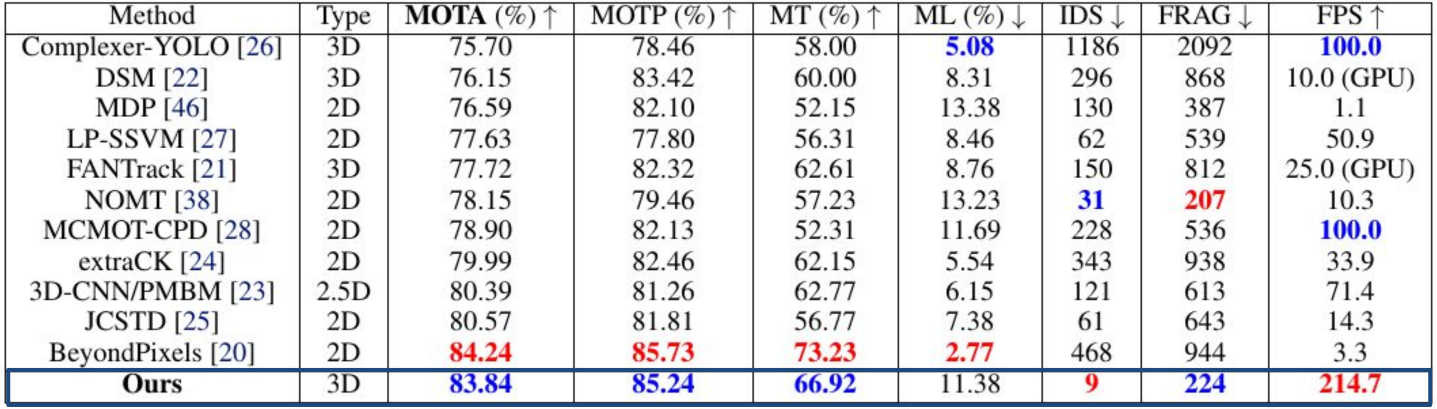
$$\hat{y}^{k+1} = y + v_y,$$

$$\hat{z}^{k+1} = z + v_z.$$

Weng et al., A Baseline for 3D Multi-Object Tracking, IROS'20

AB3D-MOT

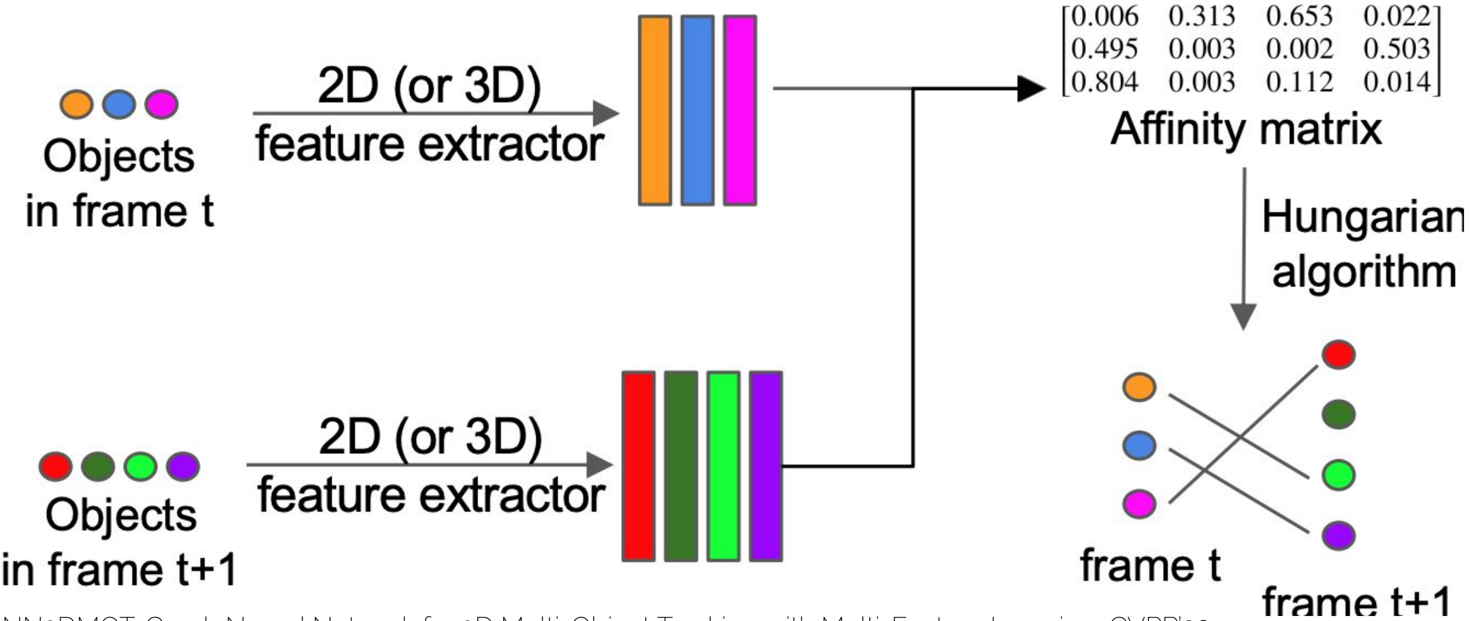




Method	Type	**MOTA** (%) ↑	MOTP (%) ↑	MT (%) ↑	ML (%) ↓	IDS ↓	FRAG ↓	FPS ↑
Complexer-YOLO [26]	3D	75.70	78.46	58.00	**5.08**	1186	2092	**100.0**
DSM [22]	3D	76.15	83.42	60.00	8.31	296	868	10.0 (GPU)
MDP [46]	2D	76.59	82.10	52.15	13.38	130	387	1.1
LP-SSVM [27]	2D	77.63	77.80	56.31	8.46	62	539	50.9
FANTrack [21]	3D	77.72	82.32	62.61	8.76	150	812	25.0 (GPU)
NOMT [38]	2D	78.15	79.46	57.23	13.23	**31**	**207**	10.3
MCMOT-CPD [28]	2D	78.90	82.13	52.31	11.69	228	536	**100.0**
extraCK [24]	2D	79.99	82.46	62.15	5.54	343	938	33.9
3D-CNN/PMBM [23]	2.5D	80.39	81.26	62.77	6.15	121	613	71.4
JCSTD [25]	2D	80.57	81.81	56.77	7.38	61	643	14.3
BeyondPixels [20]	2D	**84.24**	**85.73**	**73.23**	**2.77**	468	944	3.3
Ours	3D	**83.84**	**85.24**	**66.92**	11.38	**9**	**224**	**214.7**

GNN3DMOT - Idea

- AB3DMOT (and existing):



Objects in frame t

2D (or 3D) feature extractor

Objects in frame t+1

2D (or 3D) feature extractor

$$\begin{bmatrix} 0.006 & 0.313 & 0.653 & 0.022 \\ 0.495 & 0.003 & 0.002 & 0.503 \\ 0.804 & 0.003 & 0.112 & 0.014 \end{bmatrix}$$

Affinity matrix

Hungarian algorithm

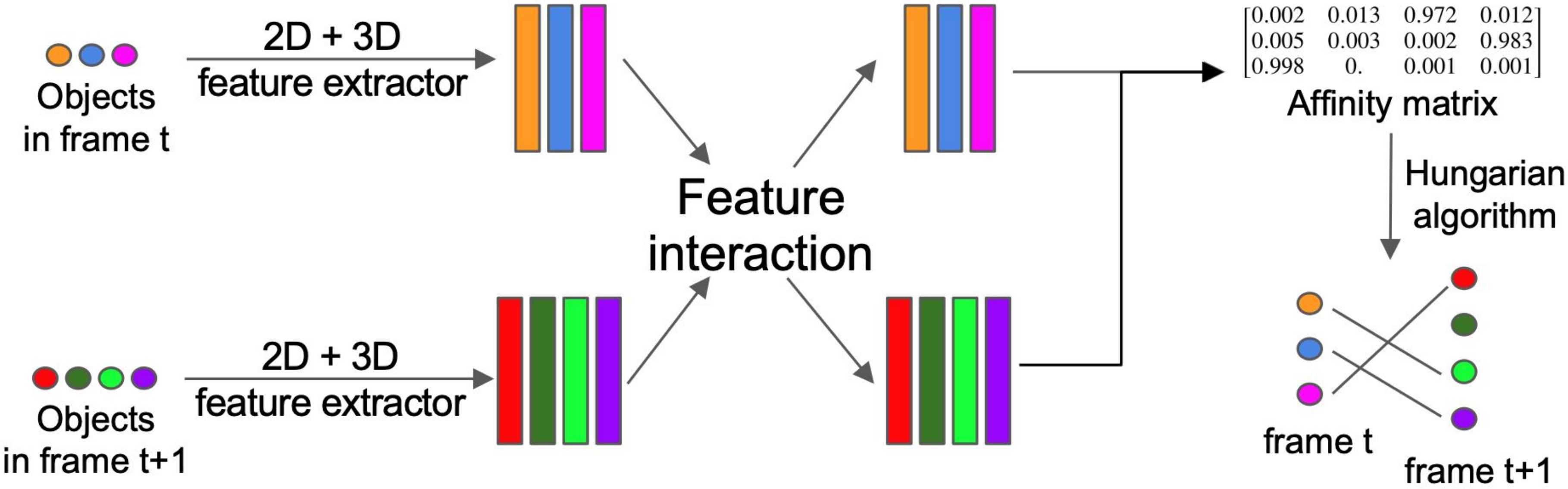
frame t

frame t+1

Weng et al., GNN3DMOT: Graph Neural Network for 3D Multi-Object Tracking with Multi-Feature Learning, CVPR'20

GNN3DMOT - Idea

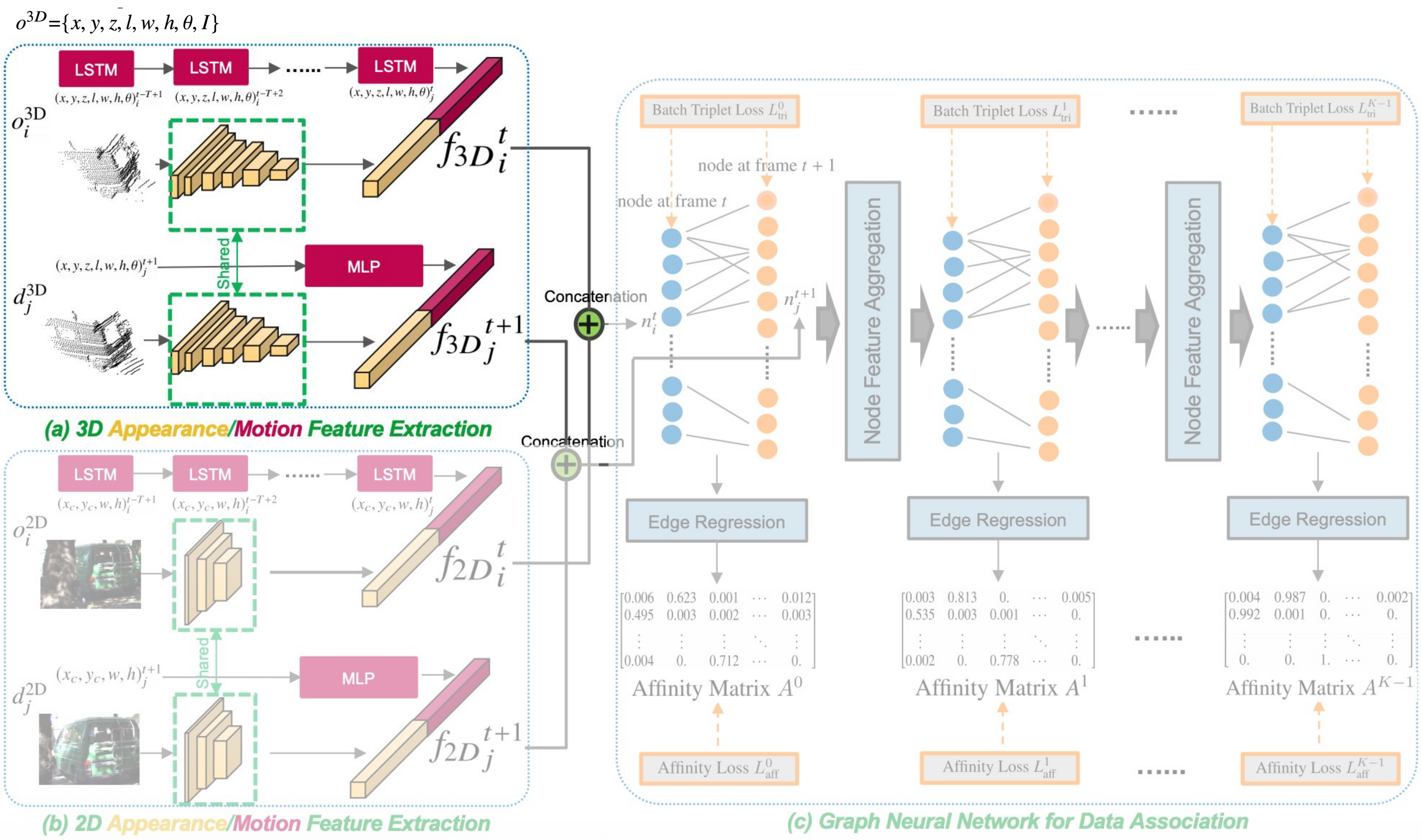
- New here:



$$\begin{bmatrix} 0.002 & 0.013 & 0.972 & 0.012 \\ 0.005 & 0.003 & 0.002 & 0.983 \\ 0.998 & 0. & 0.001 & 0.001 \end{bmatrix}$$

Weng et al., GNN3DMOT: Graph Neural Network for 3D Multi-Object Tracking with Multi-Feature Learning, CVPR'20

GNN3DMOT - Method

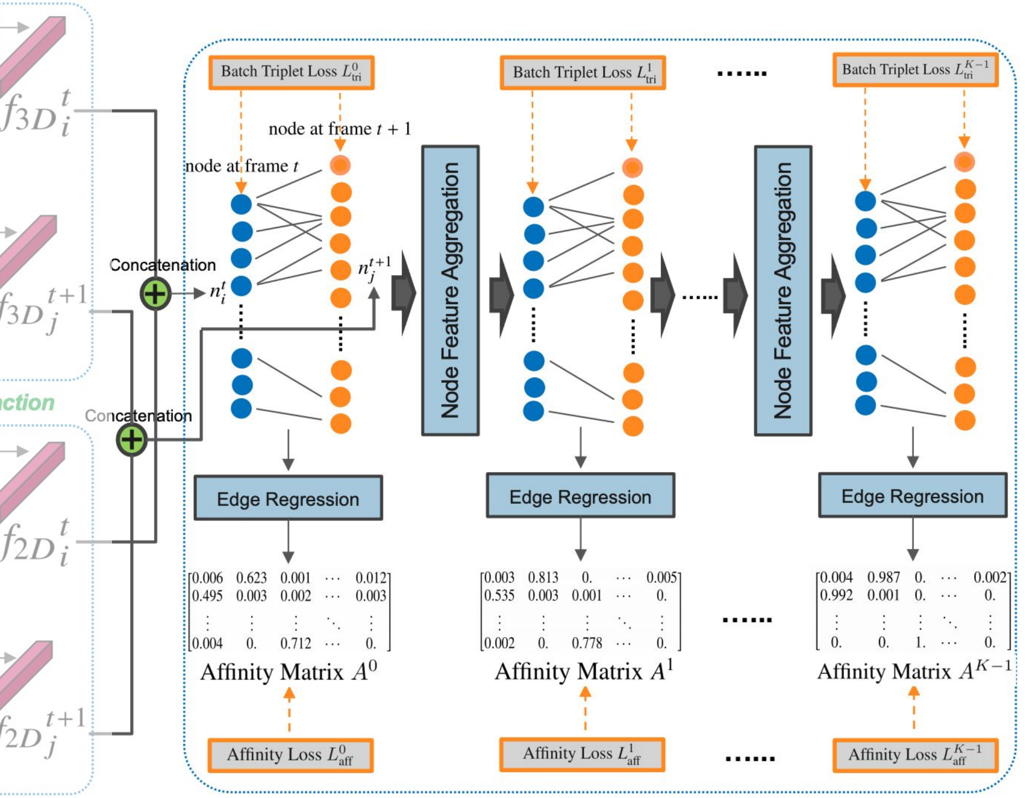

$o^{3D}=\{x, y, z, l, w, h, \theta, I\}$
LSTM
LSTM
......
LSTM
$(x,y,z,l,w,h,\theta)_i^{t-T+1}$
$(x,y,z,l,w,h,\theta)_i^{t-T+2}$
$(x,y,z,l,w,h,\theta)_j^{t}$
o_i^{3D}
${f_{3D}}_i^t$
Shared
$(x,y,z,l,w,h,\theta)_j^{t+1}$
MLP
d_j^{3D}
${f_{3D}}_j^{t+1}$
Concatenation
n_i^t
n_j^{t+1}
Concatenation
LSTM
LSTM
......
LSTM
$(x_c, y_c, w, h)_i^{t-T+1}$
$(x_c, y_c, w, h)_i^{t-T+2}$
$(x_c, y_c, w, h)_j^{t}$
o_i^{2D}
${f_{2D}}_i^t$
Shared
$(x_c, y_c, w, h)_j^{t+1}$
MLP
d_j^{2D}
${f_{2D}}_j^{t+1}$
Batch Triplet Loss L_{tri}^0
Batch Triplet Loss L_{tri}^1
......
Batch Triplet Loss L_{tri}^{K-1}
node at frame $t+1$
node at frame t
Node Feature Aggregation
Node Feature Aggregation
......
Edge Regression
Edge Regression
Edge Regression
$\begin{bmatrix} 0.006 & 0.623 & 0.001 & \cdots & 0.012 \\ 0.495 & 0.003 & 0.002 & \cdots & 0.003 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0.004 & 0. & 0.712 & \cdots & 0. \end{bmatrix}$
Affinity Matrix A^0
$\begin{bmatrix} 0.003 & 0.813 & 0. & \cdots & 0.005 \\ 0.535 & 0.003 & 0.001 & \cdots & 0. \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0.002 & 0. & 0.778 & \cdots & 0. \end{bmatrix}$
Affinity Matrix A^1
......
$\begin{bmatrix} 0.004 & 0.987 & 0. & \cdots & 0.002 \\ 0.992 & 0.001 & 0. & \cdots & 0. \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0. & 0. & 1. & \cdots & 0. \end{bmatrix}$
Affinity Matrix A^{K-1}
Affinity Loss L_{aff}^0
Affinity Loss L_{aff}^1
......
Affinity Loss L_{aff}^{K-1}

(a) 3D Appearance/Motion Feature Extraction

(b) 2D Appearance/Motion Feature Extraction

(c) Graph Neural Network for Data Association

GNN3DMOT - Method



(c) Graph Neural Network for Data Association

$$A_{ij} = \text{Sigmoid}(\sigma_2(\text{ReLU}(\sigma_1(n^t_i - n^{t+1}_j))))$$

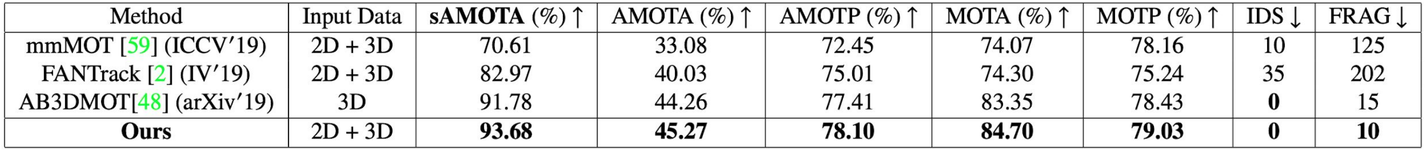
Linear layers

Features at time *t, t+1*

- Trained using triplet loss, cross-entropy (“affinity”) loss

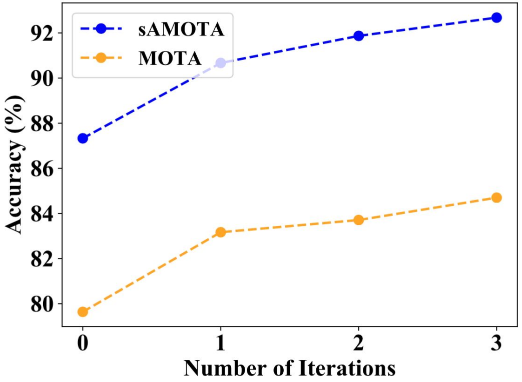
GNN3DMOT - Results

- Final results on the KITTI-val split:
  - MOTA/AMOTA/sAMOTA improves (+ 1.35 MOTA)



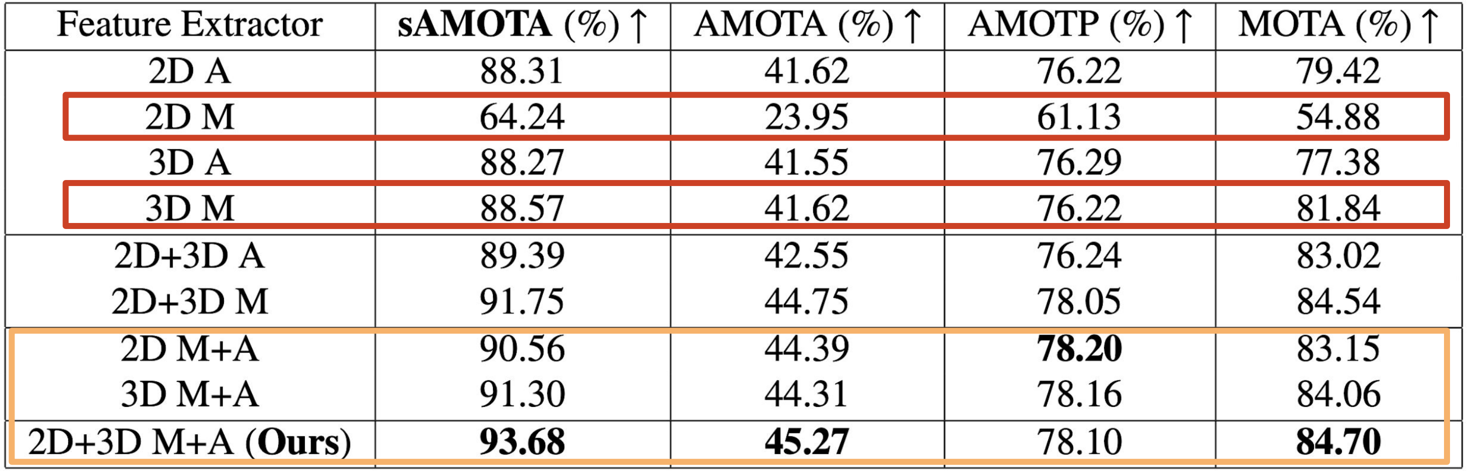
Method	Input Data	**sAMOTA** (%) ↑	AMOTA (%) ↑	AMOTP (%) ↑	MOTA (%) ↑	MOTP (%) ↑	IDS ↓	FRAG ↓
mmMOT [59] (ICCV'19)	2D + 3D	70.61	33.08	72.45	74.07	78.16	10	125
FANTrack [2] (IV'19)	2D + 3D	82.97	40.03	75.01	74.30	75.24	35	202
AB3DMOT[48] (arXiv'19)	3D	91.78	44.26	77.41	83.35	78.43	**0**	15
Ours	2D + 3D	**93.68**	**45.27**	**78.10**	**84.70**	**79.03**	**0**	**10**

- The effect of the feature aggregation:



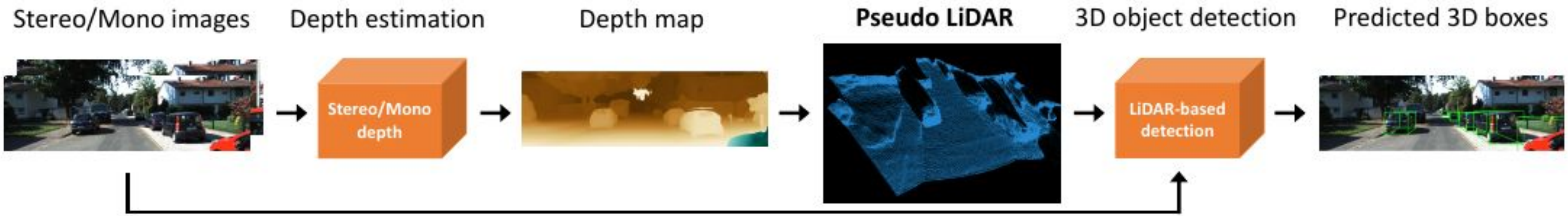
GNN3DMOT - Ablation

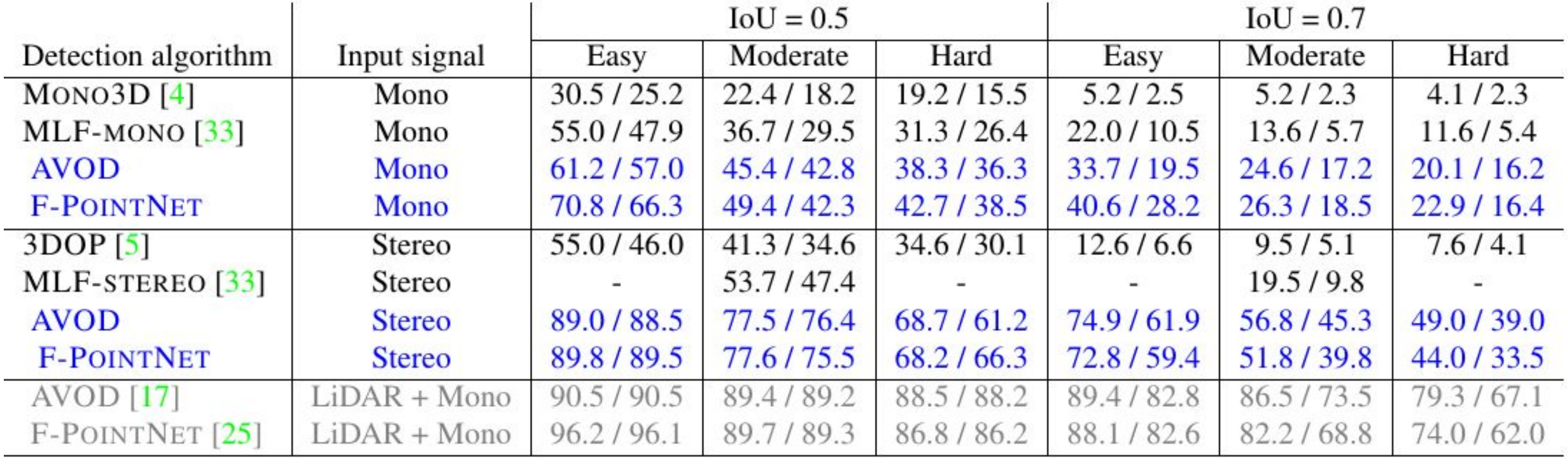
- Large gap between 2D and 3D motion model
- 3D motion > 2D appearance > 3D appearance
  - => Motion cues are super-important!
- Performance gain when combining 2D+3D



Feature Extractor	**sAMOTA** (%) ↑	AMOTA (%) ↑	AMOTP (%) ↑	MOTA (%) ↑
2D A	88.31	41.62	76.22	79.42
2D M	64.24	23.95	61.13	54.88
3D A	88.27	41.55	76.29	77.38
3D M	88.57	41.62	76.22	81.84
2D+3D A	89.39	42.55	76.24	83.02
2D+3D M	91.75	44.75	78.05	84.54
2D M+A	90.56	44.39	**78.20**	83.15
3D M+A	91.30	44.31	78.16	84.06
2D+3D M+A (**Ours**)	**93.68**	**45.27**	78.10	**84.70**

Hey, How About Stereo? :(





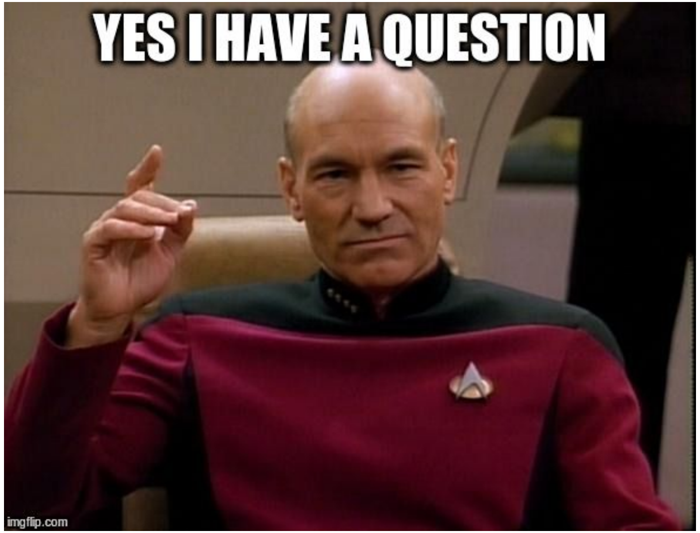
Detection algorithm	Input signal	IoU = 0.5			IoU = 0.7		
		Easy	Moderate	Hard	Easy	Moderate	Hard
Mono3D [4]	Mono	30.5 / 25.2	22.4 / 18.2	19.2 / 15.5	5.2 / 2.5	5.2 / 2.3	4.1 / 2.3
MLF-Mono [33]	Mono	55.0 / 47.9	36.7 / 29.5	31.3 / 26.4	22.0 / 10.5	13.6 / 5.7	11.6 / 5.4
AVOD	Mono	61.2 / 57.0	45.4 / 42.8	38.3 / 36.3	33.7 / 19.5	24.6 / 17.2	20.1 / 16.2
F-PointNet	Mono	70.8 / 66.3	49.4 / 42.3	42.7 / 38.5	40.6 / 28.2	26.3 / 18.5	22.9 / 16.4
3DOP [5]	Stereo	55.0 / 46.0	41.3 / 34.6	34.6 / 30.1	12.6 / 6.6	9.5 / 5.1	7.6 / 4.1
MLF-Stereo [33]	Stereo	-	53.7 / 47.4	-	-	19.5 / 9.8	-
AVOD	Stereo	89.0 / 88.5	77.5 / 76.4	68.7 / 61.2	74.9 / 61.9	56.8 / 45.3	49.0 / 39.0
F-PointNet	Stereo	89.8 / 89.5	77.6 / 75.5	68.2 / 66.3	72.8 / 59.4	51.8 / 39.8	44.0 / 33.5
AVOD [17]	LiDAR + Mono	90.5 / 90.5	89.4 / 89.2	88.5 / 88.2	89.4 / 82.8	86.5 / 73.5	79.3 / 67.1
F-PointNet [25]	LiDAR + Mono	96.2 / 96.1	89.7 / 89.3	86.8 / 86.2	88.1 / 82.6	82.2 / 68.8	74.0 / 62.0

Wang et al., Pseudo-LiDAR from Visual Depth Estimation, CVPR'19

Takeaways

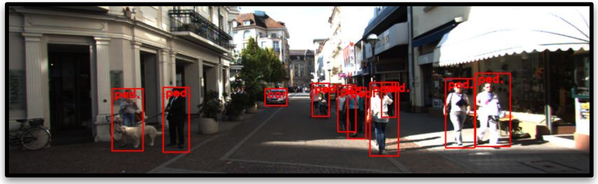
- Nowadays, we know how to **learn representations** from unstructured point clouds, yay!
    - 3D object detection, semantic/instance segmentation
- 3D detection/tracking/segmentation vibrant and **exciting** area of research!
- Surprisingly, we can turn any depth map to a point cloud and apply techniques we learned about -- unifying framework!

Thank you for your attention!



CIWT: Stereo-Vision Based 3D MOT

- Input: stereo images
- Object detections
  - 2013 - 2016 rapid progress in the field of (image-based) object detection (R-CNN family)



- Goal: 2D MOT, but:
  - Utilize stereo
  - Infer 3D trajectories of objects

Osep et. al., Combined Image- and World-Space Tracking, ICRA'17

KPConv

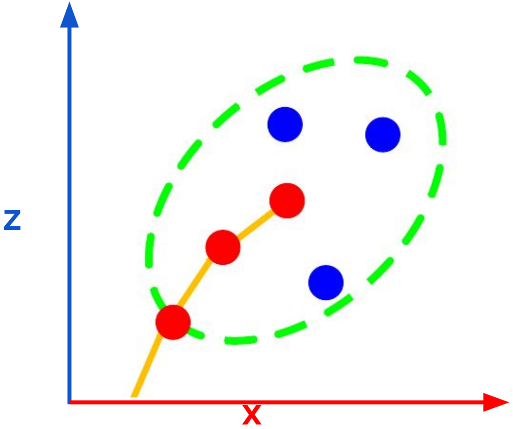
- General point convolution

$$(\mathcal{F} * g)(x) = \sum_{x_i \in \mathcal{N}_x} g(x_i - x) f_i$$

Kernel function
(domain: r-Ball

Dynamics-based Tracking

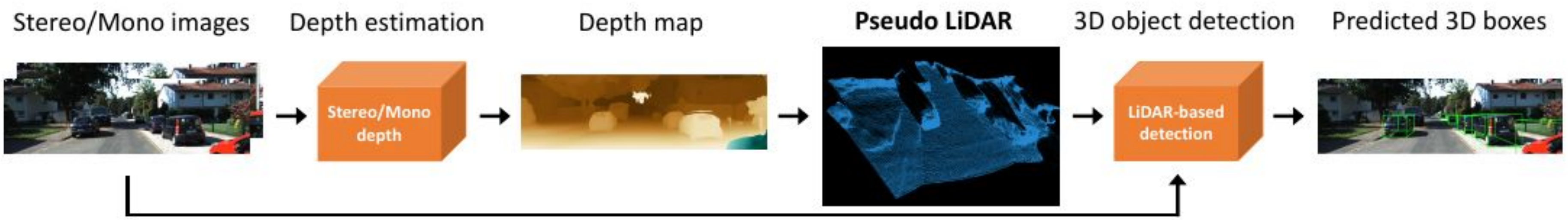
- SONAR, RADAR



BACK IN MY DAYS
WE DIDNT
HAVE CONVNETS
imgflip.com

Work of Jens?

Pseudo LiDAR -- DOWE?



Detection algorithm	Input signal	IoU = 0.5			IoU = 0.7		
		Easy	Moderate	Hard	Easy	Moderate	Hard
Mono3D [4]	Mono	30.5 / 25.2	22.4 / 18.2	19.2 / 15.5	5.2 / 2.5	5.2 / 2.3	4.1 / 2.3
MLF-mono [33]	Mono	55.0 / 47.9	36.7 / 29.5	31.3 / 26.4	22.0 / 10.5	13.6 / 5.7	11.6 / 5.4
AVOD	Mono	61.2 / 57.0	45.4 / 42.8	38.3 / 36.3	33.7 / 19.5	24.6 / 17.2	20.1 / 16.2
F-PointNet	Mono	70.8 / 66.3	49.4 / 42.3	42.7 / 38.5	40.6 / 28.2	26.3 / 18.5	22.9 / 16.4
3DOP [5]	Stereo	55.0 / 46.0	41.3 / 34.6	34.6 / 30.1	12.6 / 6.6	9.5 / 5.1	7.6 / 4.1
MLF-stereo [33]	Stereo	-	53.7 / 47.4	-	-	19.5 / 9.8	-
AVOD	Stereo	89.0 / 88.5	77.5 / 76.4	68.7 / 61.2	74.9 / 61.9	56.8 / 45.3	49.0 / 39.0
F-PointNet	Stereo	89.8 / 89.5	77.6 / 75.5	68.2 / 66.3	72.8 / 59.4	51.8 / 39.8	44.0 / 33.5
AVOD [17]	LiDAR + Mono	90.5 / 90.5	89.4 / 89.2	88.5 / 88.2	89.4 / 82.8	86.5 / 73.5	79.3 / 67.1
F-PointNet [25]	LiDAR + Mono	96.2 / 96.1	89.7 / 89.3	86.8 / 86.2	88.1 / 82.6	82.2 / 68.8	74.0 / 62.0

Wang et al., Pseudo-LiDAR from Visual Depth Estimation, CVPR'19