Computer Vision

Kerem Yildirir

Autonmous driving.
Everywhere.

Overview of the presentation

- **Task Description**
- **Literature Review**
    - Possible methods for 3D object detection
    - Popular datasets to test the methods
    - Limitations and constraints
- **LDLS:** Label Diffusion Lidar Segmentation
    - Advantages and disadvantages
    - YOLACT: Real Time Instance Segmentation
- **LDLS-YOLACT** for real-time performance
- **SORT**: Simple, online, real-time tracking
- **Conclusion and Future Work**

Task Description

LiDAR & SLAM give us an offline global map, but we need to detect dynamic obstacles, most importantly pedestrians, but also cars, cyclists and other objects not present when recording the global map. Based on live imagery of Intel RealSense and the Livox Mid-100 LiDAR, we need to determine the position and class of obstacles in 3D.

Literature Review

Date	Name	Modalities	Map data	External data	mAP	mATE (m)	mASE (1-IOU)	mAOE (rad)	mAVE (m/s)	mAAE (1-acc)	NDS	FPS (Hz)	Stats
		Any	All	no									
2020-05-27	CenterPoint	Lidar, Radar	no	no	0.611	0.259	0.238	0.368	0.309	0.137	0.675	n/a	
2020-05-28	CVCNet ensemble	Lidar	no	no	0.582	0.284	0.241	0.372	0.224	0.126	0.666	n/a	
2020-05-28	CVCNet single mod	Lidar	no	no	0.558	0.300	0.248	0.431	0.269	0.119	0.642	n/a	
2020-05-21	DTIF	Lidar	no	no	0.538	0.301	0.253	0.389	0.239	0.159	0.635	n/a	
2019-06-18	MEGVII	Lidar	no	no	0.528	0.300	0.247	0.379	0.245	0.140	0.633	n/a	
2020-05-22	CRIPAC	Lidar	no	no	0.527	0.326	0.246	0.384	0.242	0.121	0.632	n/a	
2020-04-13	PanoNet3D	Lidar	no	no	0.545	0.298	0.247	0.393	0.338	0.136	0.631	n/a	
2020-05-09	SSN v2	Lidar	no	no	0.506	0.339	0.245	0.429	0.266	0.087	0.616	n/a	
2020-05-09	Chirouhua	Lidar	no	no	0.500	0.336	0.244	0.423	0.267	0.086	0.614	n/a	
2020-05-26	LRCF360	Camera, Lidar, Radar	no	no	0.541	0.350	0.261	0.543	0.394	0.133	0.603	n/a	

Literature Review

Pedestrian

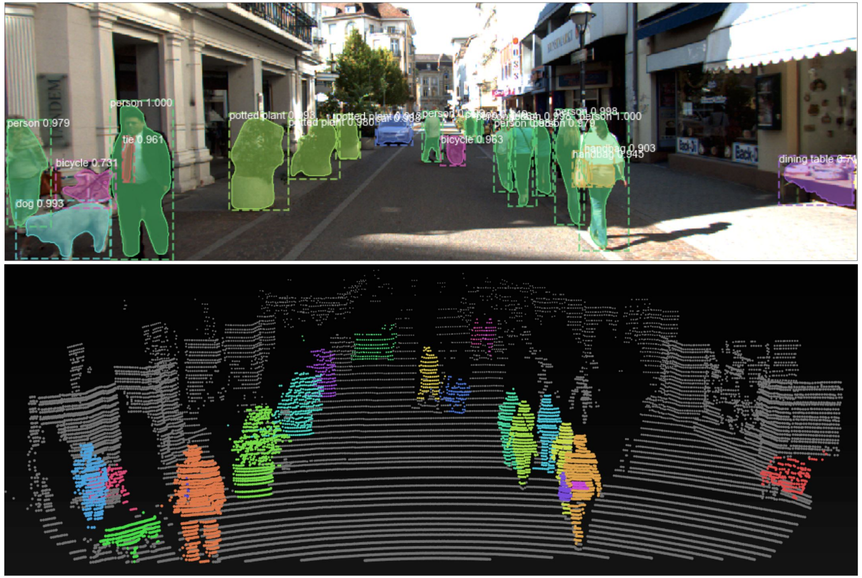
	Method	Setting	Code	Moderate	Easy	Hard	Runtime	Environment
1	HotSpotNet			**45.37 %**	53.10 %	41.47 %	0.04 s	1 core @ 2.5 Ghz (Python + C/C++)
Q. Chen, L. Sun, Z. Wang, K. Jia and A. Yuille: object as hotspots. Proceedings of the European Conference on Computer Vision (ECCV) 2020.								
2	Noah CV Lab - SSL			45.23 %	52.85 %	41.28 %	0.1 s	GPU @ 2.5 Ghz (Python)
3	TANet		code	44.34 %	53.72 %	40.49 %	0.035s	GPU @ 2.5 Ghz (Python + C/C++)
Z. Liu, X. Zhao, T. Huang, R. Hu, Y. Zhou and X. Bai: TANet: Robust 3D Object Detection from Point Clouds with Triple Attention. AAAI 2020.								
4	3DSSD		code	44.27 %	**54.64 %**	40.23 %	0.04 s	GPU @ 2.5 Ghz (Python + C/C++)
Z. Yang, Y. Sun, S. Liu and J. Jia: 3DSSD: Point-based 3D Single Stage Object Detector. CVPR 2020.								
5	PPBA			44.08 %	52.65 %	**41.54 %**	NA s	GPU @ 2.5 Ghz (Python)
6	CentrNet-FG			44.02 %	53.51 %	40.53 %	0.03 s	1 core @ 2.5 Ghz (C/C++)
7	Point-GNN		code	43.77 %	51.92 %	40.14 %	0.6 s	GPU @ 2.5 Ghz (Python)
W. Shi and R. Rajkumar: Point-GNN: Graph Neural Network for 3D Object Detection in a Point Cloud. CVPR 2020.								
8	PP-3D			43.77 %	51.92 %	40.14 %	0.1 s	1 core @ 2.5 Ghz (Python)

Literature Review

- For the task of 3D detection, best performing methods are LiDAR based and require 3D annotated data for fine-tuning
- Type of LiDAR we use is different from the one’s used in popular datasets

LDLS: Label Diffusion Lidar Segmentation

- 3D Instance Segmentation of the point cloud using 2D detections provided from Mask R-CNN
    - Does not require any annotated 3D data!
    - Original implementation is not suitable for real time due to slow inference speed of Mask R-CNN



Images are from the [paper]

Autonmous driving.
Everywhere.

Kerem Yildirir

WARP

YOLACT: Real Time Instance Segmentation

Images are from the [paper]

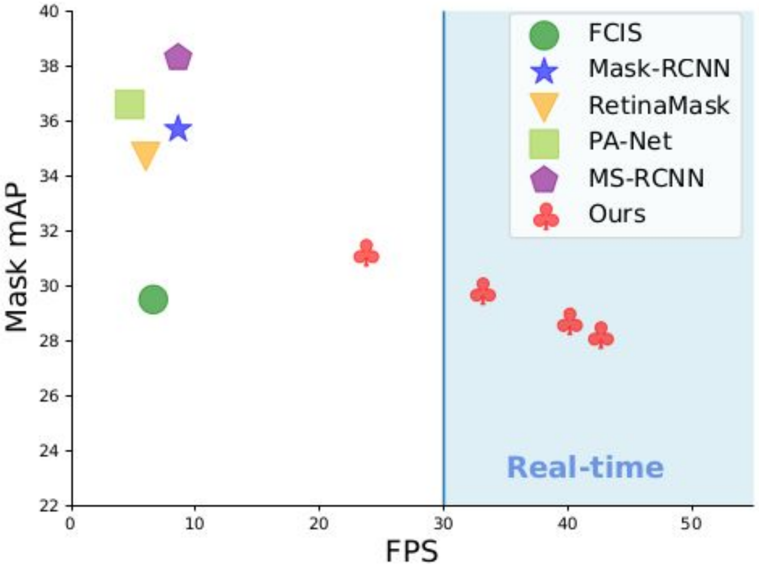


Figure 1: Speed-performance trade-off for various instance segmentation methods on COCO. To our knowledge, ours is the first *real-time* (above 30 FPS) approach with around 30 mask mAP on COCO `test-dev`.

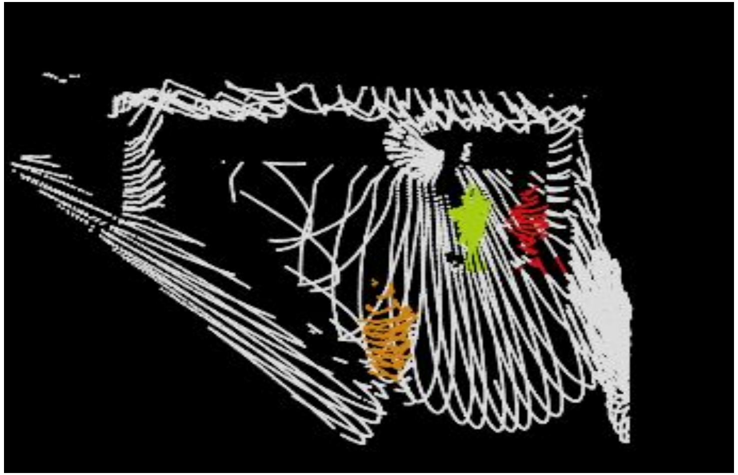
Autonmous driving.
Everywhere.

Kerem Yildirir

WARP

3D Real-Time Instance Segmentation

- We solve the performance issue of LDLS by swapping Mask R-CNN with YOLACT, a faster instance segmentation model.
- For any given image and LiDAR data pair, we first perform 2D Instance Object Segmentation using YOLACT
- We then apply LDLS to label the point cloud data with the detected classes in 2D



3D Instance Segmentation ~12 FPS with Titan V

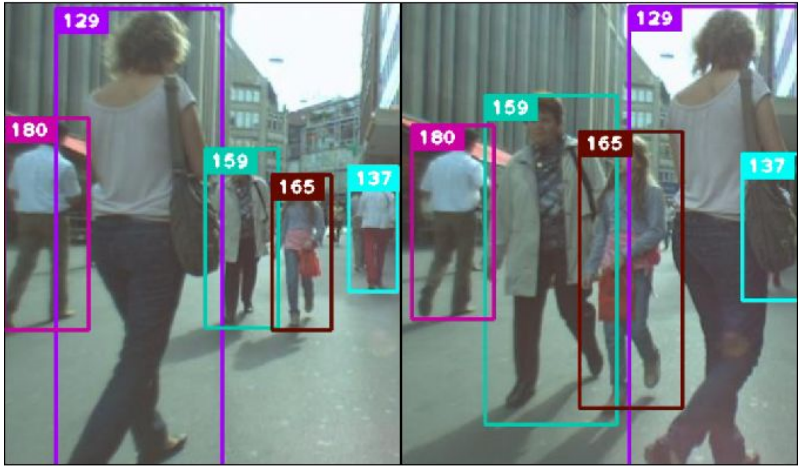
Autonmous driving.
Everywhere.

Kerem Yildirir

WARP

SORT: Simple Online Realtime Tracking

- Detections at every frame
- Kalman filter for predicting the next position of an object
- Discard frames which intersects with current tracks
- Match the frames at frame t-1 with frame t to extend the trajectory



Kerem Yildirir

Autonmous driving.
Everywhere.

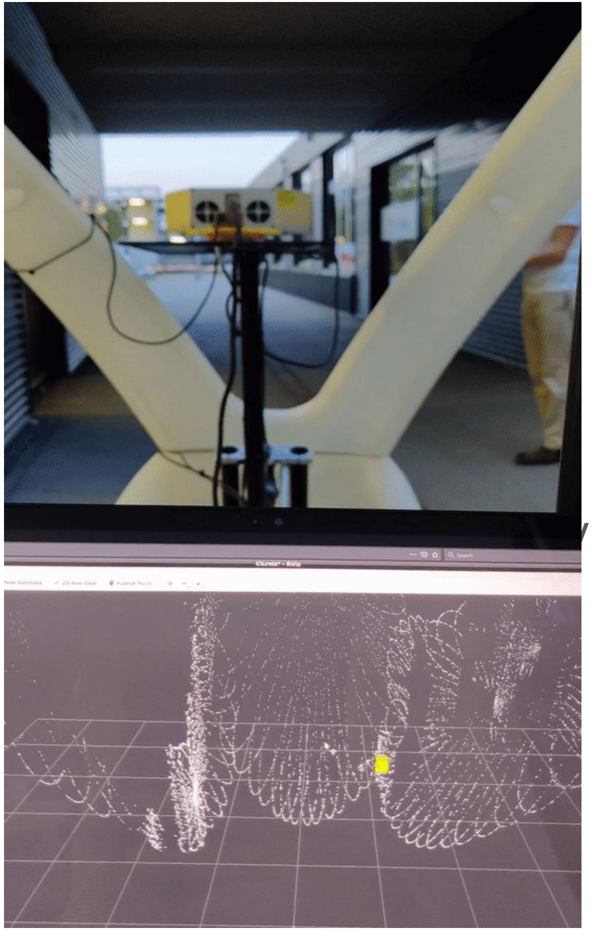
WARP

Autonmous drivin[redacted]
Everywhere.

[redacted] R P

Conclusion

- 2D segmentation is really fast and accurate
- We can get accurate and fast results by subsampling the point cloud



Future Work

- Replacing the camera with a larger field of view alternative
- Implementation of LDLS using C++ and CUDA
- Using LDLS and a 3D bounding box regressor for generating 3D Bounding Boxes, then train LiDAR based methods with the new data
- Using 3D trackers which take 3D motion into account or better 2D trackers which are more complex and also [predict velocity and orientations