Frustum-PointPillars

A multi-stage approach for 3D object detection using RGB camera and LiDAR

Anshul Paigwar, David Sierra-Gonzalez, Ozgur Erkent, Christian Laugier

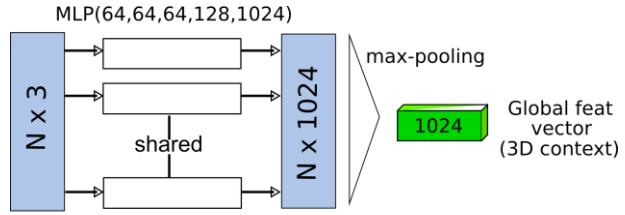
Team CHROMA, Inria Grenoble, France.

Agenda

1. Introduction – 3D object detection
2. Difficulties with pedestrian detection
3. Sensor fusion approaches -- RGB + LiDAR
4. Our method
5. Results
6. Conclusions

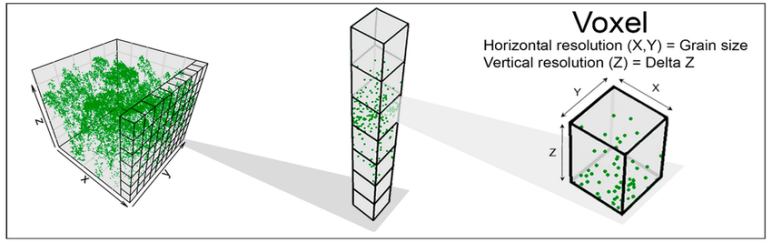
3D Object Detection in point cloud

feature extraction
512 to 1024 points

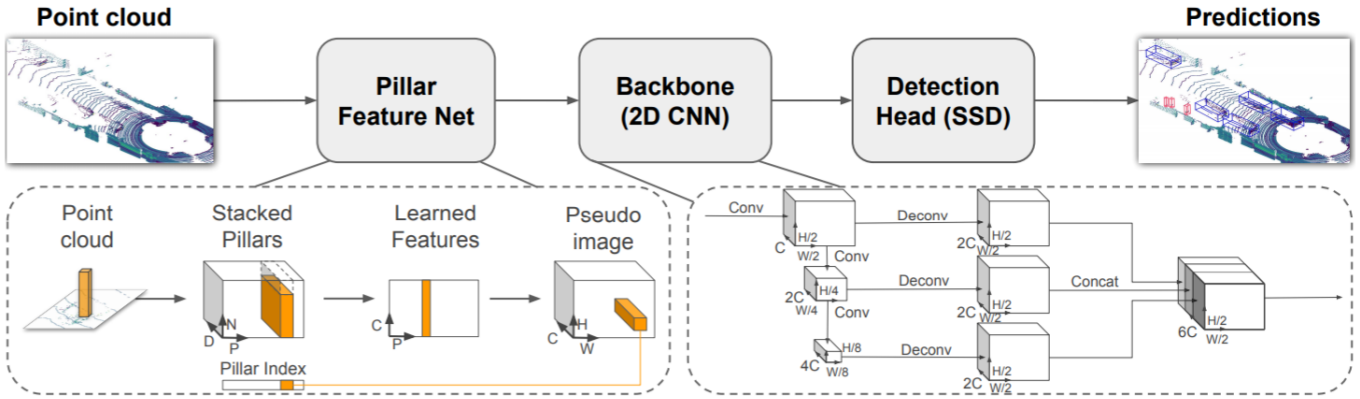


PointNet

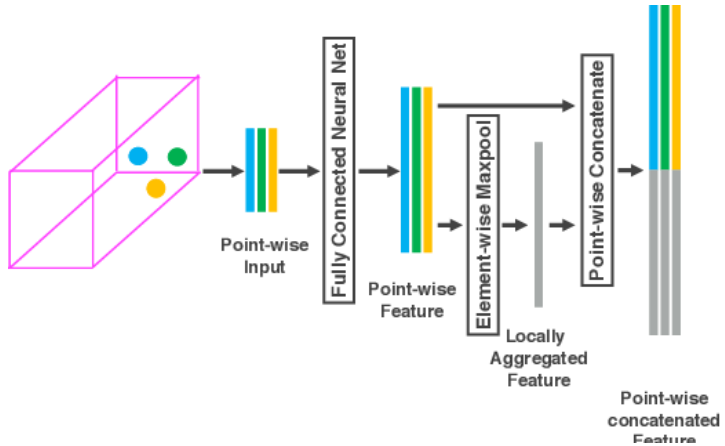
1. LiDARs are widely popular sensor in autonomous vehicle, as they are very robust to varying environmental conditions.
2. Point cloud data which is huge but sparse and unstructured. PointNet for feature extraction.
3. Good performance for large objects like cars and vans, but suffer at localizing smaller objects like pedestrians.



Voxelisation



PointPillars architecture ~14 to 16 Hz



~3 to 4 Hz VoxelNet

Pedestrian detection in point cloud

1. Point perturbations of a pedestrian do not have a distinctive geometric structure and have fewer data points.
2. Unlike cars, pedestrians are not rigid bodies

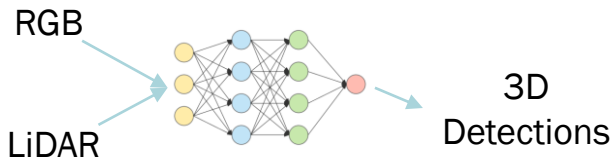
Pole Pedestrian Pedestrian walking

Pedestrian Detection	SOTA	Accuracy
2D RGB image	TuSimple	78.40%
3D point cloud	HotSpotNet	45.36%

Sensor Fusion
RGB + LiDAR

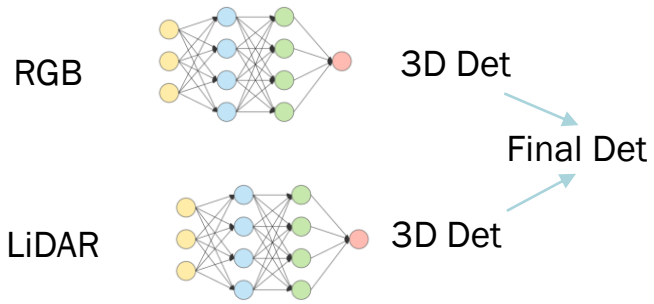
Early Fusion

- A single network takes input from two or more sensor modalities.
- Failure of one sensor can lead to the total failure of the network.



Late Fusion

- Independent networks, one for each sensor modality, output detections in a redundant manner.
- 3D detections with RGB have poor performace.

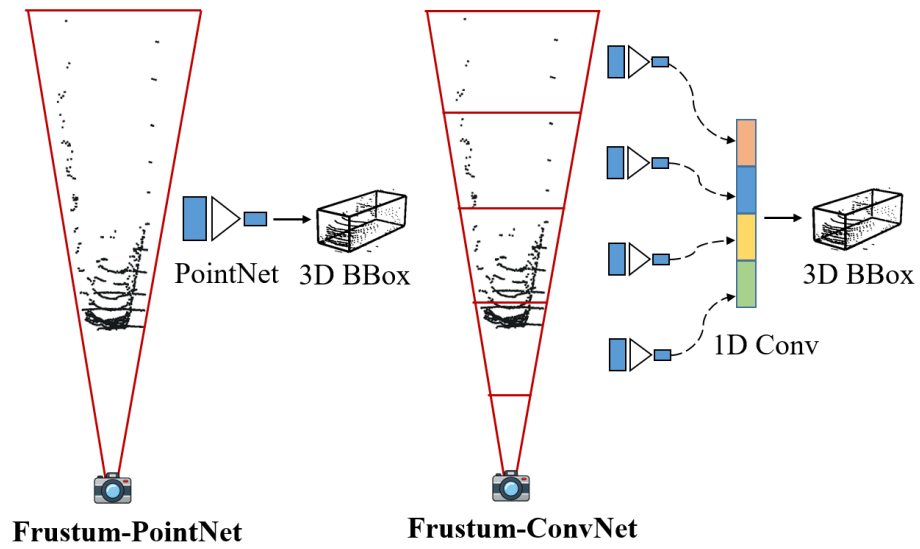


Multi Stage approach

- Independent networks, one for each modality, are stacked together. The output of the first network (Stage-I) constitutes the input to the second network (Stage-II).

Multi Stage methods 3D Detections

1. Leverage the mature field of 2D object detection to reduce the search space in the 3D space.
2. Given 2D region proposals in RGB images, first find local points corresponding to pixels inside the 2D regions.
3. Use PointNet on these local points to predict amodal bounding box.



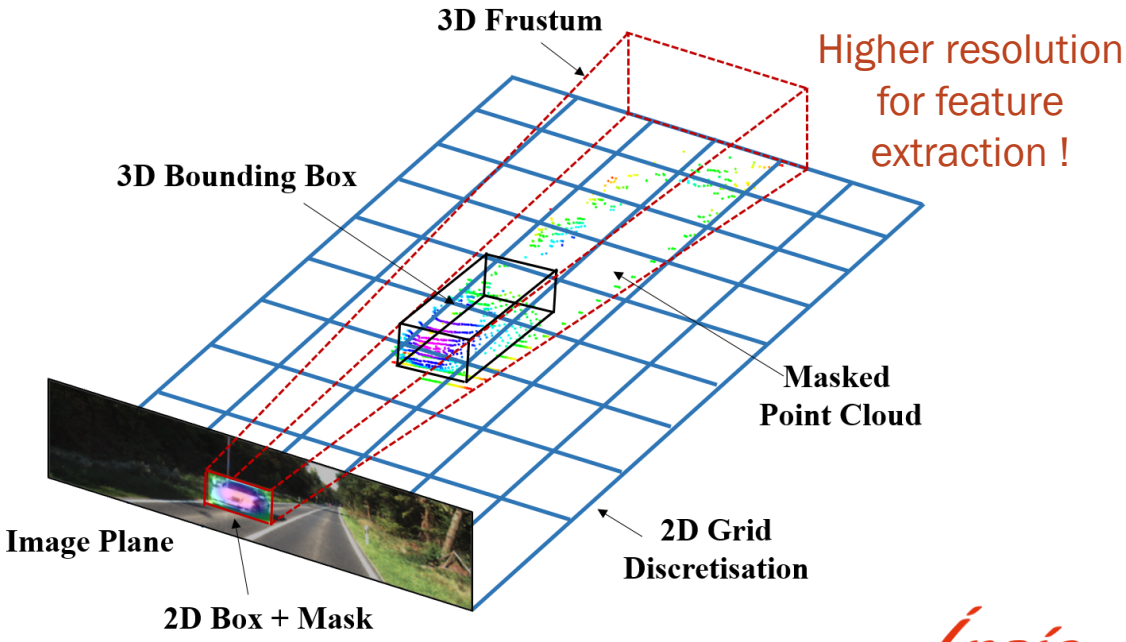
Stage - I

2D detections
RGB Camera

Filtering & Masking of point cloud

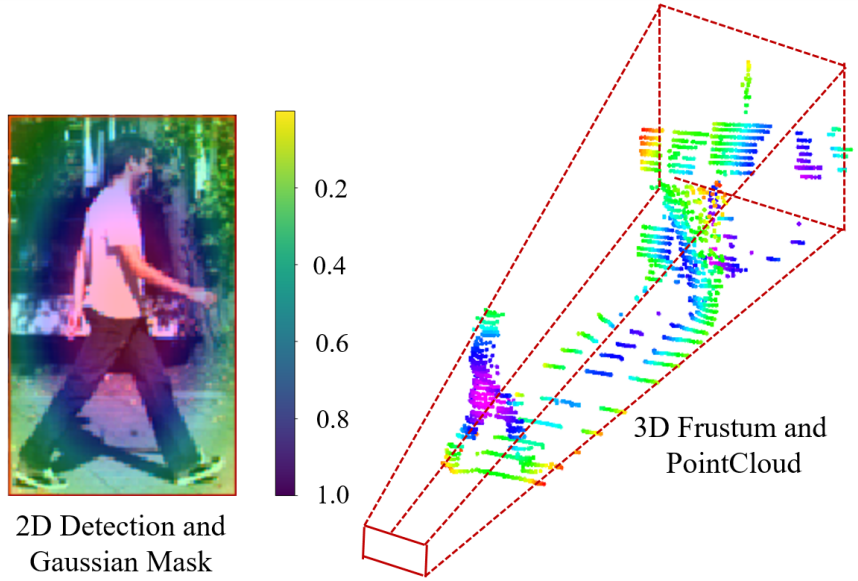
Stage - II

3D detections
in point cloud

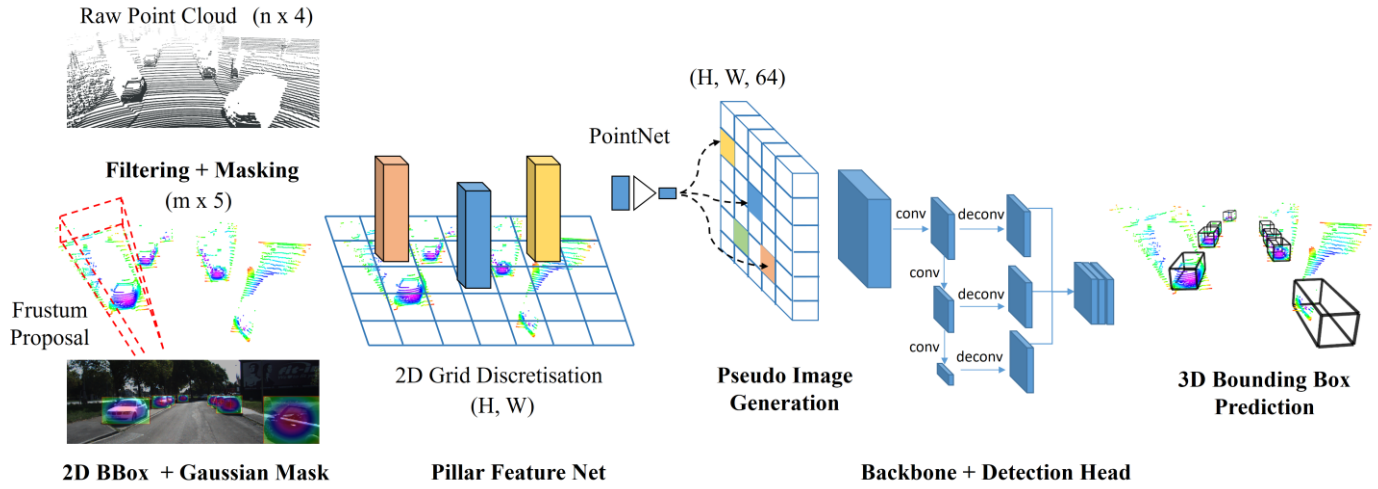


Frustum PointPillars

1. We extend PointPillars architecture by the addition of RGB camera and the use of a multi-stage approach.
2. We extend the data augmentation for training to work with multi-stage network architecture.
3. Gaussian-based masking of 3D points to distinguish foreground from background clutter (improves localization of objects in 3D)



Gaussian Masking (likelihood of object)



Why not semantic mask ?

1. Semantic masking is computationally expensive.
2. SOTA 2D detection >> Segmentation
3. Point cloud data augmentation not possible as it will require corresponding augmentation with semantic mask.

Results

Method	3D detection			BEV detection			Runtime
	Easy	Mod.	Hard	Easy	Mod.	Hard	
STD [19]	53.29	42.47	38.35	60.02	48.72	44.55	0.08 s
PointPillars [8]	51.45	41.92	38.89	57.60	48.64	45.78	**0.07 s**
AVOD-FPN [11]	50.46	42.27	39.04	58.49	50.32	46.98	0.1 s
F-PointNet [14]	50.53	42.15	38.08	57.13	49.57	45.48	0.17 s
F-ConvNet [15]	**52.16**	**43.38**	38.80	57.04	48.96	44.33	0.47 s
F-PointPillars (Ours)	51.22	42.89	**39.28**	**60.98**	**52.23**	**48.30**	**0.07 s**

TABLE I: AP (%) on KITTI test set for pedestrian detection.

Ranked 28th pedestrian detection

Ranked 1st BEV pedestrian detection

Ablation studies

Method	Car			Pedestrian			Cyclist		
	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
F-PointNet [14]	83.76	70.92	63.65	**70.00**	61.32	53.59	77.15	56.49	53.37
F-ConvNet [15]	**89.31**	79.08	77.17	-	-	-	-	-	-
PointPillars	84.06	75.13	69.43	62.57	57.52	51.17	81.96	62.11	57.39
F-PointPillars (no mask)	88.02	77.87	76.15	67.24	60.69	54.71	82.12	63.06	60.54
F-PointPillars	88.90	**79.28**	**78.07**	66.11	**61.89**	**56.91**	**87.54**	**72.78**	**66.07**

TABLE II: AP (%) on KITTI val set for 3D object detection.

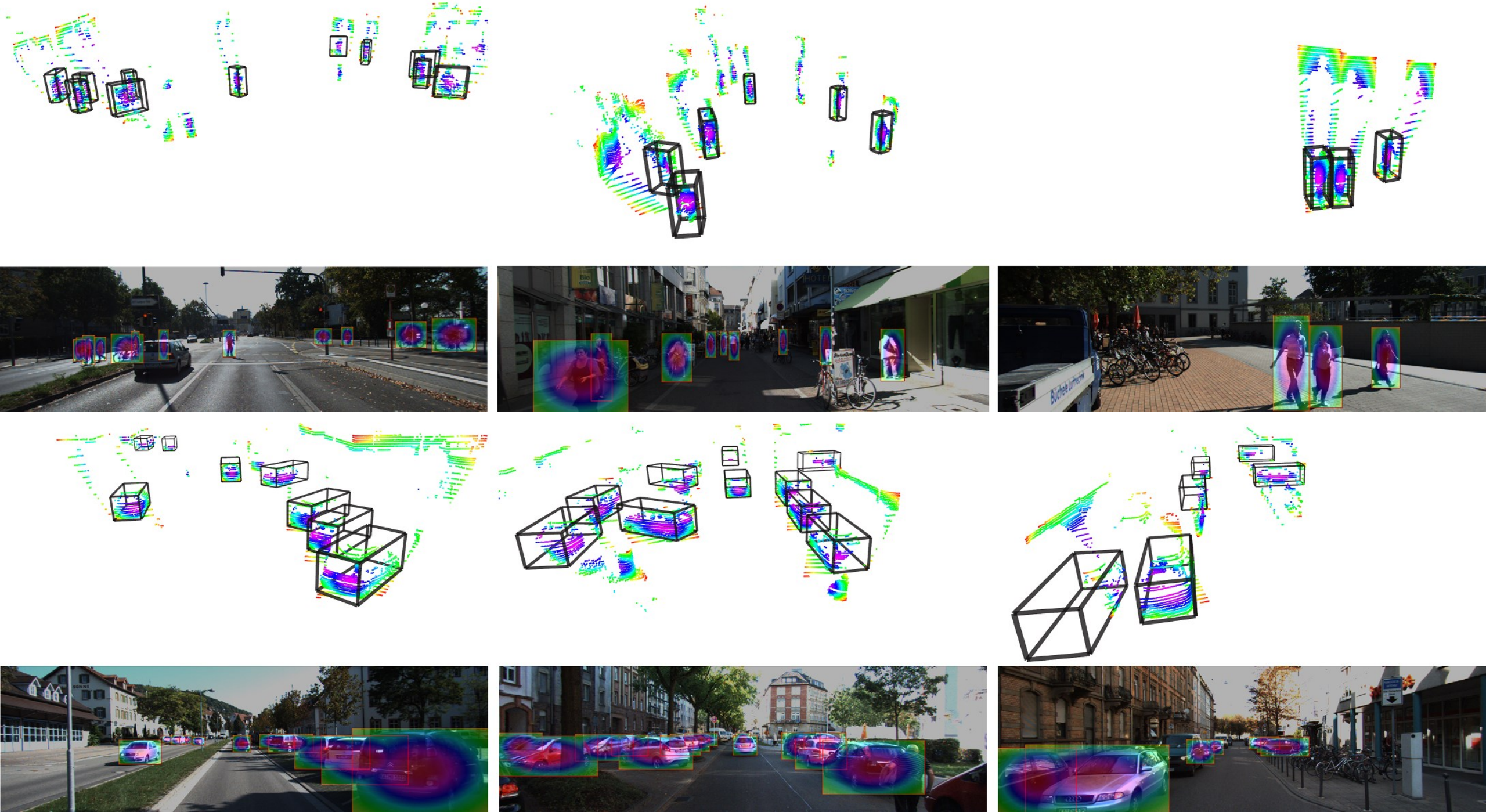
Method	Car			Pedestrian			Cyclist		
	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
F-PointNet [14]	88.16	84.02	76.44	**72.38**	66.39	59.57	81.82	60.03	56.32
F-ConvNet [15]	**90.42**	88.99	86.88	-	-	-	-	-	-
PointPillars	89.99	87.13	85.15	70.54	65.70	60.18	85.07	65.06	61.72
F-PointPillars (no mask)	89.95	88.38	87.19	71.69	66.20	60.95	83.17	65.76	63.03
F-PointPillars	90.20	**89.43**	**88.77**	72.17	**67.89**	**63.46**	**88.58**	**76.79**	**74.80**

TABLE III: AP (%) on KITTI val set for BEV detection.

2D detections (Mod.)			3D detections (Mod.)		
Car	Ped.	Cyclist	Car	Ped.	Cyclist
89.47	62.47	72.70	76.79	58.76	64.75
90.30	76.39	81.35	77.39	59.27	65.36
100	100	100	79.28	61.89	72.78

TABLE IV: Influence of 2D region proposal on F-Pointpillars. Left: Each row represent results from a different 2D detector on KITTI val set (Mod. difficulty). Right: Corresponding output of F-Pointpillar 3D detection AP (%).

Inria



Conclusions & Future work

1. We proposed Frustum-PointPillars, a multi-stage design approach that uses both RBG and LiDAR data for 3D detection.
2. We also proposed a novel approach for masking 3D point clouds with likelihood values to distinguish foreground from background clutter.
3. F-PointPillars outperforms other multi-stage approaches for 3D pedestrian detection in hard difficulty level and BEV detection in all difficulty levels.
4. Our method achieves a run-time of 14 Hz and is significantly faster than other multi-stage approaches.

1. Improve the runtime of our method by using sparse convolutions in the backbone of our network.
2. Leverage the object class information provided by 2D detectors to improve multi-class 3D detection.

Inria

References:

1. C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “Pointnet: Deep learning on point sets for 3d classification and segmentation,” Proc. Computer Vision and Pattern Recognition (CVPR), IEEE, vol. 1, no. 2, p. 4, 2017.
2. Y. Zhou and O. Tuzel, “Voxelnet: End-to-end learning for point cloud based 3d object detection,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4490–4499.
3. Y. Zhou and O. Tuzel, “Voxelnet: End-to-end learning for point cloud based 3d object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4490–4499.
4. A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, “Pointpillars: Fast encoders for object detection from point clouds,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 697–12 705.
5. A. Paigwar, O. Erkent, C. Wolf, and C. Laugier, “Attentional PointNet ¨ for 3D-Object Detection in Point Clouds,” in *CVPR 2019 – Workshop on Autonomous driving*, Long Beach, California, United States, June 2019, pp. 1–10. [Online]. Available: https://hal.inria.fr/hal-02156555

Thank you

This work has been conducted within the scope of ES3CAP (Embedded Smart Safe Secure Computing Autonomous Platform) project funded by European Union.

Contact:

anshul.paigwar@inria.fr