3D Object Detection Frameworks

Joseph Sepich, Shahriar Hossain, Sanjay Mohan, Sean Steinle

Abstract

Over the years, numerous frameworks have emerged that make 3D object detection in the autonomous scenario easier and more robust. Despite the introduction of such frameworks, there are always areas that require improvement. In this project, we have explored a variety of end-to-end 3D object–detection pipelines on the KITTI benchmark to understand how different representations, network backbones, and native CUDA-accelerated modules impact 3D object detection accuracy and speed. The KITTI dataset consists of image and pointcloud data, which can be transformed into structured inputs in the form of voxels, pillars, or bird's-eye-view (BEV) images, that can be processed by 2D or 3D convolutional backbones. We experimented with different frameworks and finalized three of these to be included in the final report. These are PointPillar, SECOND and PointRCNN.

Introduction

@Sanjay

Related Work

@Sanjay

Technical Approach

We began our review by examining and attempting to implement 9 different 3D object detection models, before narrowing our focus to PointRCNN, SECOND, and PointPillar. We trained these models on two machines. Both machines utilized a 3080 RTX GPU, one using Linux and the other using Windows. PointRCNN took about 4 hours to train, SECOND took about 3 hours, and PointPillar took at 7 hours.

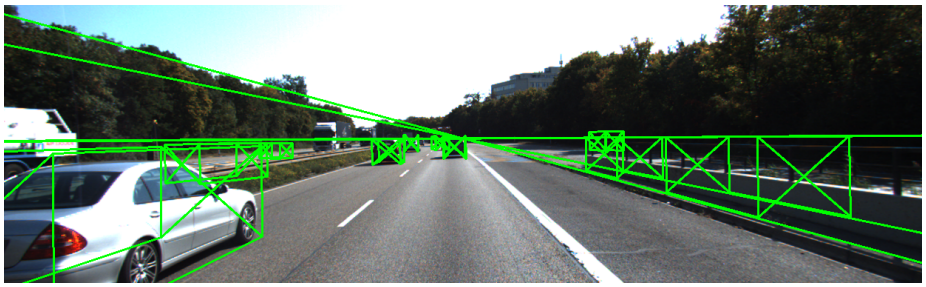
We compare the three models across a number of dimensions. First, we consider the models' performance across easy, moderate, and hard samples. The harder samples tend to be lower visibility, with challenges like weather, occlusion, further distance, and so on. Second, we consider the models' performance across different levels of specificity. The levels of specificity are as follows, in order of increasing difficulty to maximize:

1. *2D BBox AP* - 2D bounding-box AP in image space.
2. *AOS* - Average Orientation Similarity (combines detection and correct orientation) at the same IoU threshold.
3. *BEV AP* - Bird's-Eye View detection AP.

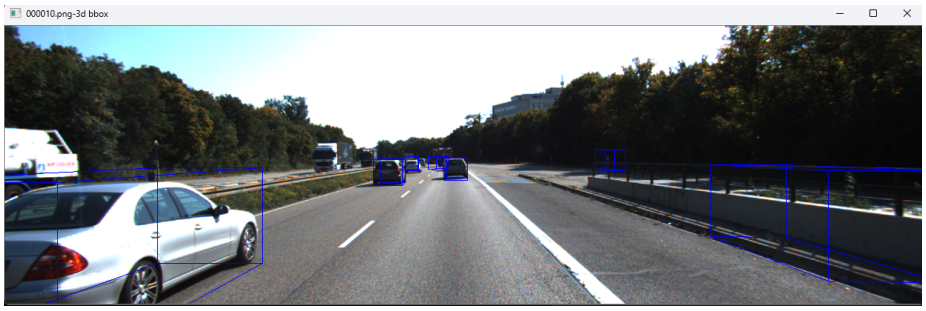
4. *3D AP* - Full 3D bounding-box AP.

Finally, we compare model performance across three classes of objects–pedestrians, cyclists, and cars. For all of these comparisons, we specify the threshold of the intersection over union (IoU) metric which is determines whether an object is detected or not.

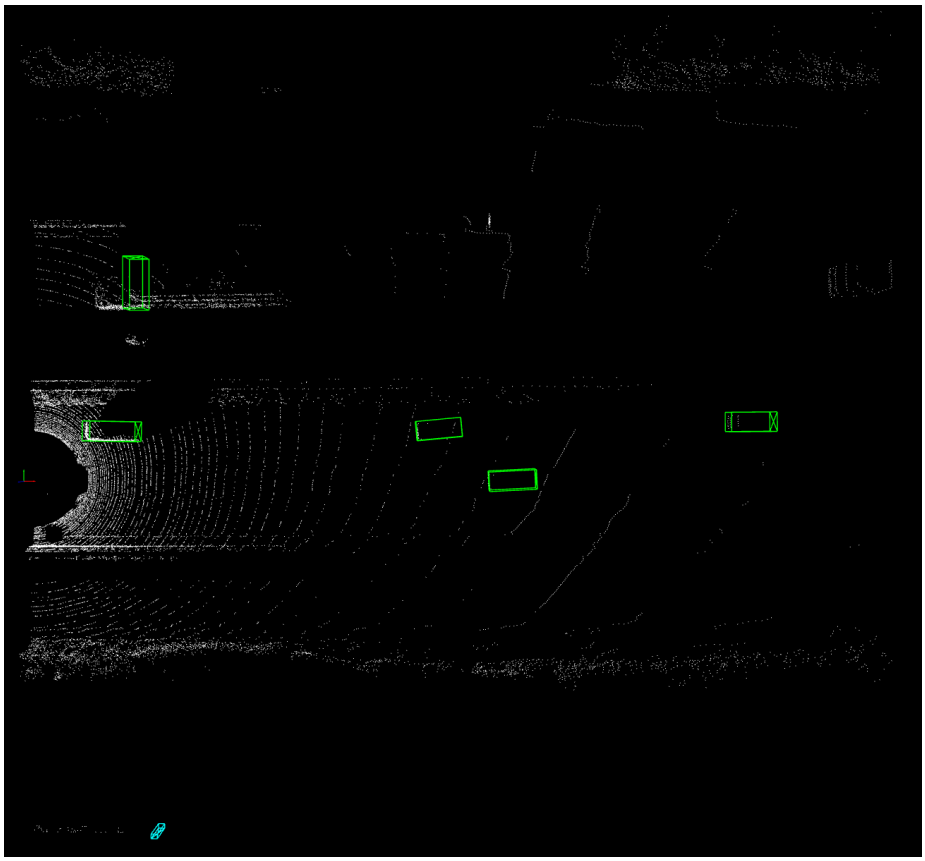
PointRCNN's 3D rendering of KITTI scene #10.



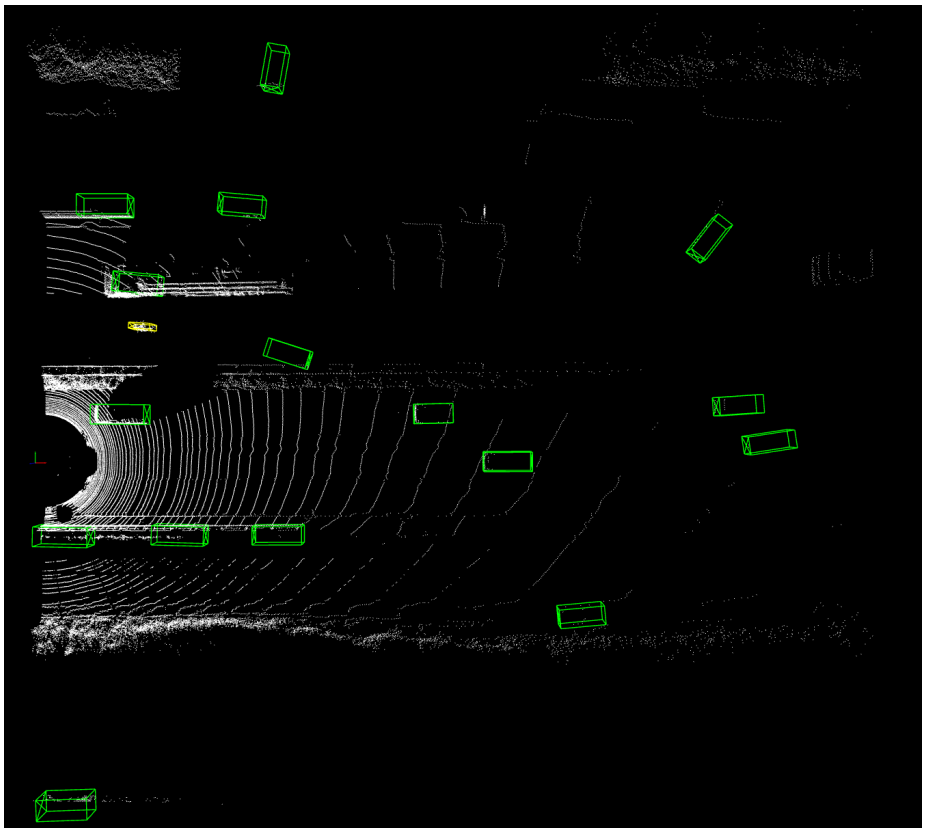
SECOND's 3D rendering of KITTI scene #10.



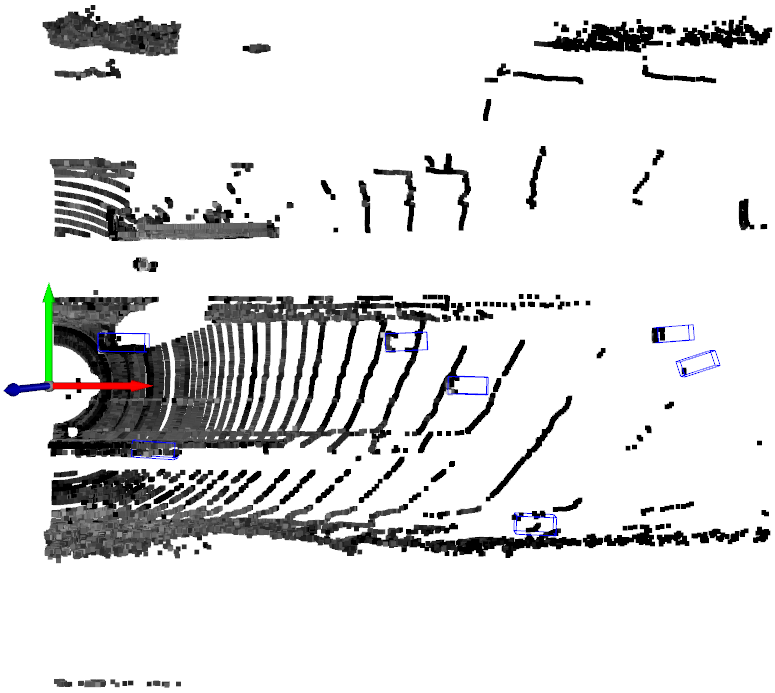
PointPillar's 3D rendering of KITTI scene #10.



PointRCNN's bird's eye view (BEV) rendering of KITTI scene #10.



SECOND's bird's eye view (BEV) rendering of KITTI scene #10.



PointPillar's bird's eye view (BEV) rendering of KITTI scene #10.

Results and Discussion

In this section we compare the performance of all three models by the specificity of the scene, difficulty of detection, and object of detection. The best model for each specificity-difficulty pair is bolded."

Quantitative Analysis

Pedestrians *IoU Threshold = 0.5*

PointRCNN

Specificity	Easy	Moderate	Hard
2D BBox AP	**73.54%**	65.83%	62.27%
AOS	**71.29%**	**63.17%**	**59.48%**
BEV AP	**67.51%**	**60.27%**	**54.09%**
3D AP	**61.84%**	**57.02%**	**51.15%**

SECOND

Specificity	Easy	Moderate	Hard
2D BBox AP	69.21%	**66.12%**	**63.43%**
AOS	65.40%	61.91%	58.93%
BEV AP	63.11%	56.77%	53.83%
3D AP	58.68%	53.90%	49.75%

PointPillar

Specificity	Easy	Moderate	Hard
2D BBox AP	64.40%	61.43%	57.62%
AOS	49.35%	46.73%	43.84%
BEV AP	59.11%	54.32%	50.50%
3D AP	51.35%	47.98%	43.80%

When it comes to performance on pedestrians, PointRCNN performed the best across 10 of 12 specificity-difficulty pairs. The two pairs in which is was not the best was the IoU on 2D bounding boxes for medium and hard samples–in these cases, the SECOND model performed better. This highlights a key trend in these datapoints–while PointRCNN generally performs best, the SECOND model is more robust to difficult samples in that its performance degrades less. Additionally, we see consistent degradation from all models as we increase the specificity of the scene (from 2D -> AOS -> BEV -> 3D). This suggests that handling the added localization, height, and orientation considerations degrade performance.

Cyclists *IoU Threshold = 0.5*

PointRCNN

Specificity	Easy	Moderate	Hard
2D BBox AP	**89.75%**	**77.67%**	**75.27%**
AOS	**89.67%**	**77.20%**	**74.70%**
BEV AP	**88.36%**	**74.44%**	**71.01%**
3D AP	**87.72%**	**72.57%**	**69.94%**

SECOND

Specificity	Easy	Moderate	Hard
2D BBox AP	86.66%	76.53%	72.67%

Specificity	Easy	Moderate	Hard
AOS	86.33%	75.98%	72.12%
BEV AP	83.95%	69.92%	66.34%
3D AP	80.72%	66.56%	62.22%

PointPillar

Specificity	Easy	Moderate	Hard
2D BBox AP	86.24%	73.06%	70.17%
AOS	85.02%	69.08%	66.28%
BEV AP	84.41%	67.13%	63.74%
3D AP	81.76%	63.66%	60.90%

For detecting cyclists, PointRCNN dominates SECOND and PointPillar. It has the best performance in all 12 performance-difficulty pairs, often by a decent margin. It's interesting to note that all three models have fairly degraded performance on moderate and hard difficulty samples–this suggests that detecting cyclists can be represent a challenging edge case in low-visibility scenarios. Generally, detecting cyclists is harder than detecting pedestrians. Finally, AOS tracks AP closely, showing that once cyclists are detected, orientation is estimated reasonably well in easy cases.

Cars *IoU Threshold = 0.7*

PointRCNN

Specificity	Easy	Moderate	Hard
2D BBox AP	90.52%	89.23%	**88.94%**
AOS	90.52%	89.13%	88.78%
BEV AP	89.81%	87.12%	85.84%
3D AP	88.38%	78.19%	**77.72%**

SECOND

Specificity	Easy	Moderate	Hard
2D BBox AP	**90.81%**	**89.98%**	89.18%
AOS	**90.80%**	**89.90%**	**89.01%**
BEV AP	89.88%	87.83%	**86.47%**
3D AP	**88.52%**	**78.61%**	77.33%

PointPillar

Specificity	Easy	Moderate	Hard
2D BBox AP	90.65%	89.33%	86.66%
AOS	90.48%	88.68%	85.73%
BEV AP	**89.96%**	**87.88%**	85.77%
3D AP	86.63%	76.74%	74.17%

Unlike the case of pedestrian and cyclist data where one model was dominant, all models perform nearly equally for car classification, with only 1-2% difference for all specificity-difficulty pairs. This may be occurring because detecting cars is much easier than detecting pedestrian or cyclists, especially for harder samples. The most difficult challenge in car detection appears to be rendering with more specificity in low visibility, as there is a large dropoff in all models when comparing 3D and 2D specificities for moderate and hard samples.

All Categories *Average Results for Pedestrians, Cyclists, and Cars*

PointRCNN

Specificity	Easy	Moderate	Hard
2D BBox AP	**84.60%**	**77.58%**	**75.49%**
AOS	**83.83%**	**76.50%**	**74.32%**
BEV AP	**81.89%**	**73.94%**	**70.31%**
3D AP	**79.31%**	**69.26%**	**66.27%**

SECOND

Specificity	Easy	Moderate	Hard
2D BBox AP	82.23%	77.54%	75.09%
AOS	80.84%	75.93%	73.35%
BEV AP	78.98%	71.51%	68.88%
3D AP	75.97%	66.36%	63.10%

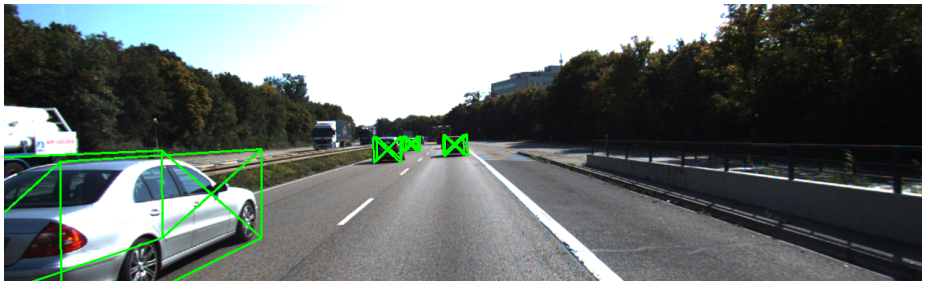
PointPillar

Specificity	Easy	Moderate	Hard
2D BBox AP	80.43%	74.61%	71.48%
AOS	74.95%	68.16%	65.29%
BEV AP	77.83%	69.78%	66.67%
3D AP	73.25%	62.79%	59.62%

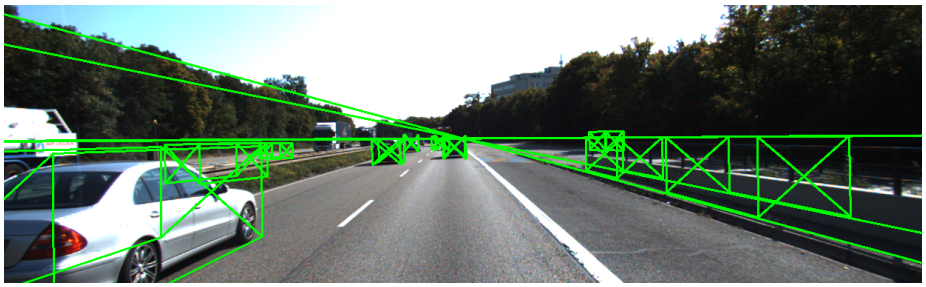
When considering overall performance, PointRCNN is the best model across all 12 specificity-difficulty pairs. This follows logically since it was the best

pedestrian and cyclist detection model, and tied with the other models for car detection. SECOND ironically performs almost as well, and PointPillar is a clearly inferior model–especially for pedestrian and cyclist detection.

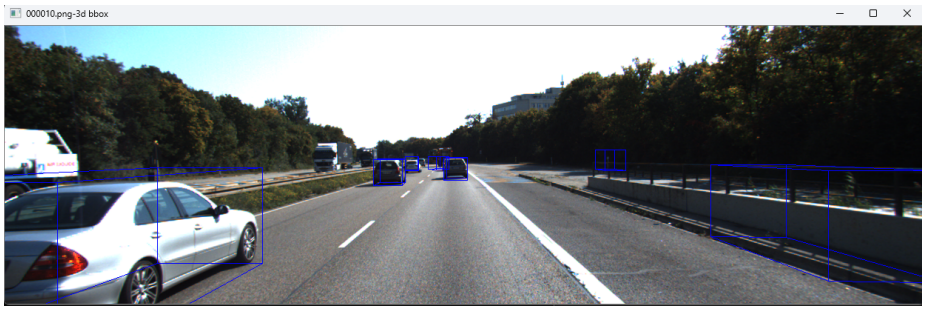
Qualitative Analysis



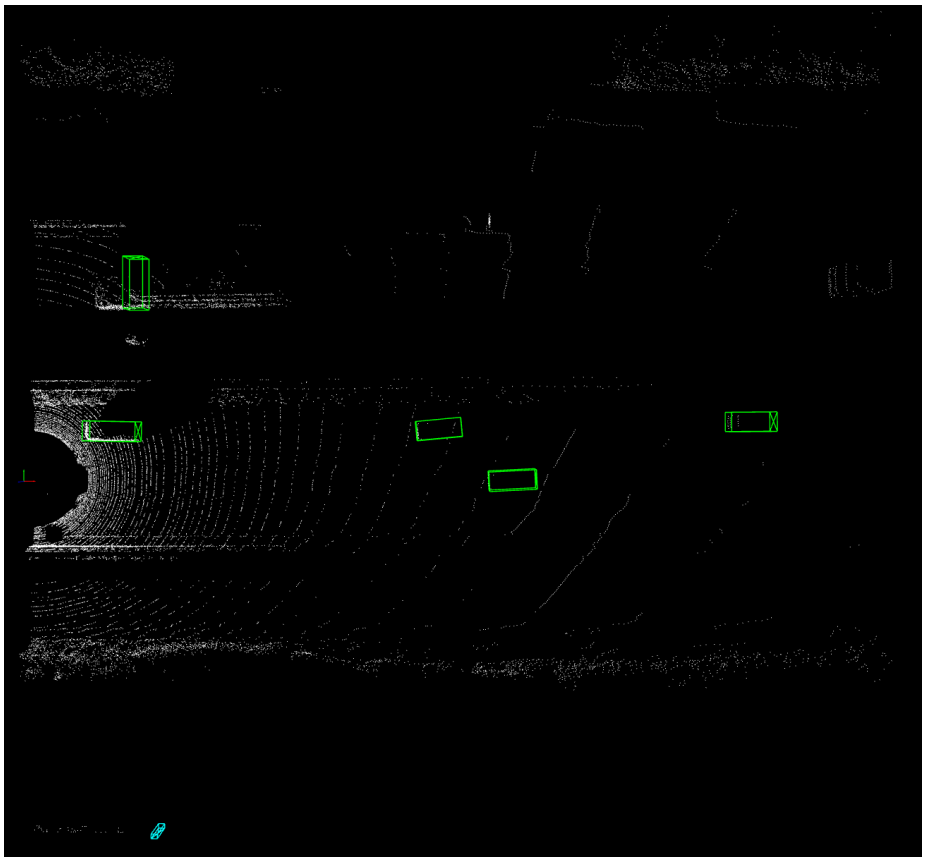
PointRCNN's 3D rendering of KITTI scene #10.



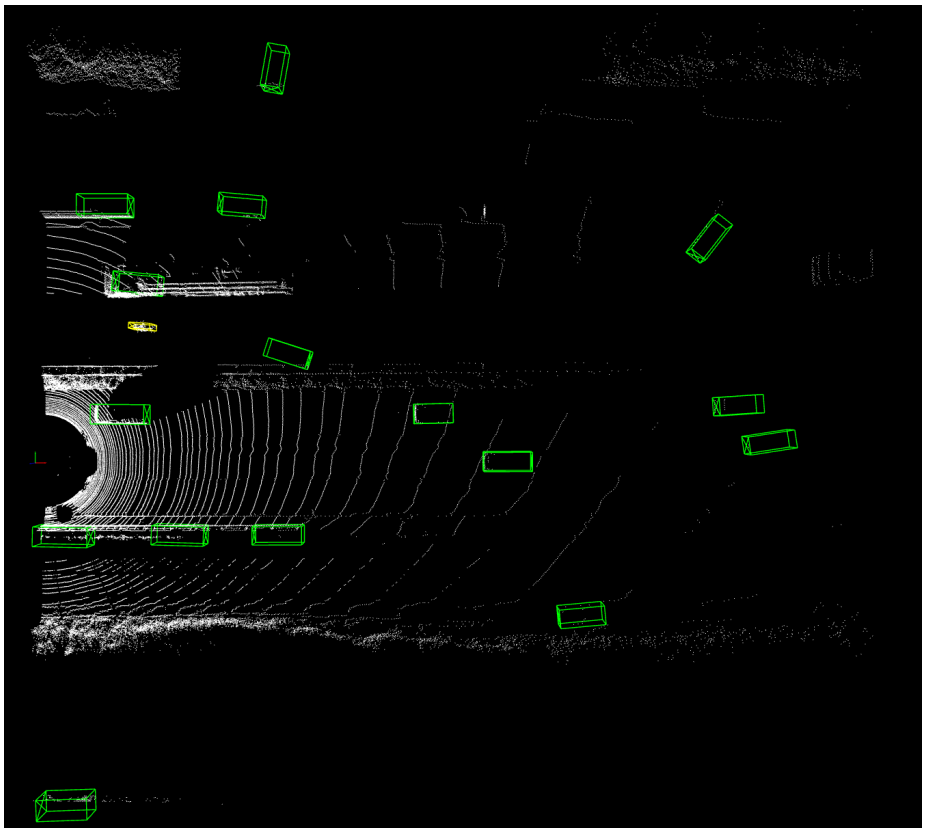
SECOND's 3D rendering of KITTI scene #10.



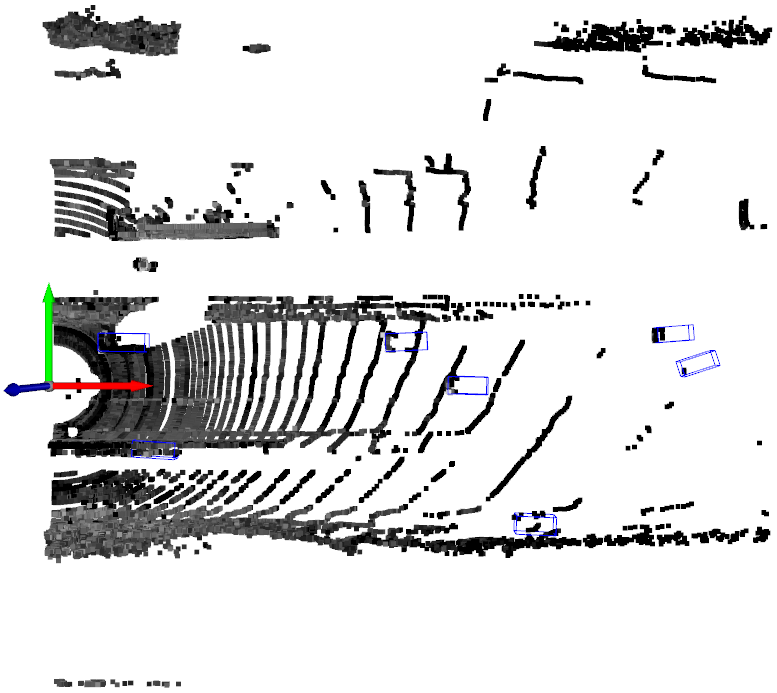
PointPillar's 3D rendering of KITTI scene #10.



PointRCNN's bird's eye view (BEV) rendering of KITTI scene #10.

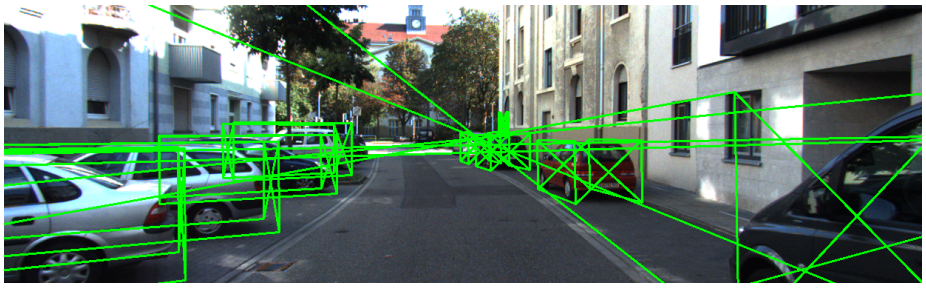


SECOND's bird's eye view (BEV) rendering of KITTI scene #10.

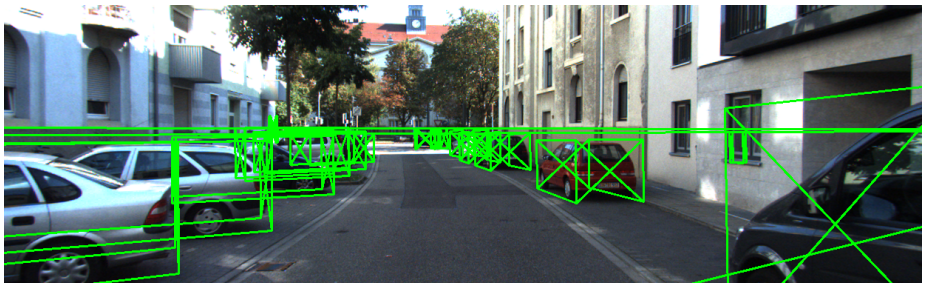


PointPillar's bird's eye view (BEV) rendering of KITTI scene #10.

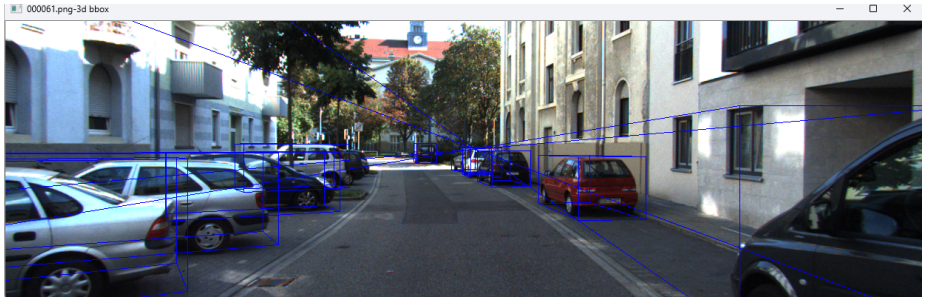
This example captures why PointRCNN is the best model–it captures only the cars in its immediate vicinity, with no false positives. On the other hand, SECOND drastically overpredicts, detecting about 3x as many objects as there actually are. PointPillar also produces false positives, though not nearly as many.



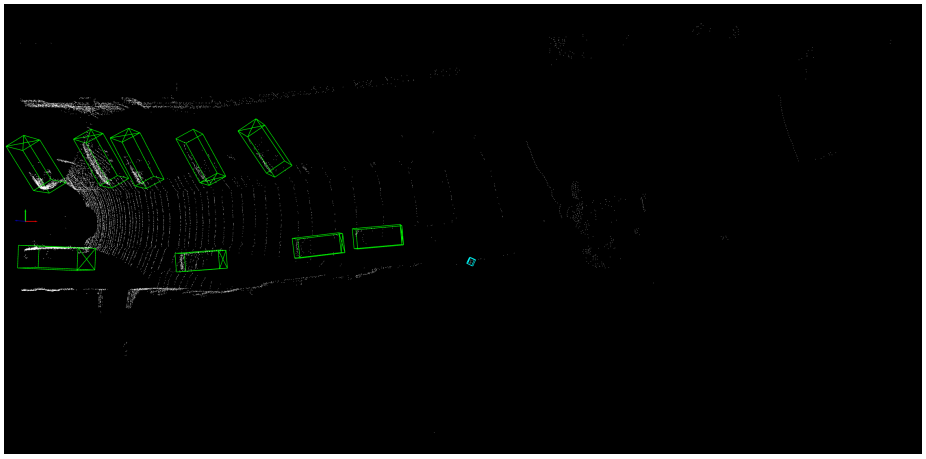
PointRCNN's 3D rendering of KITTI scene #61.



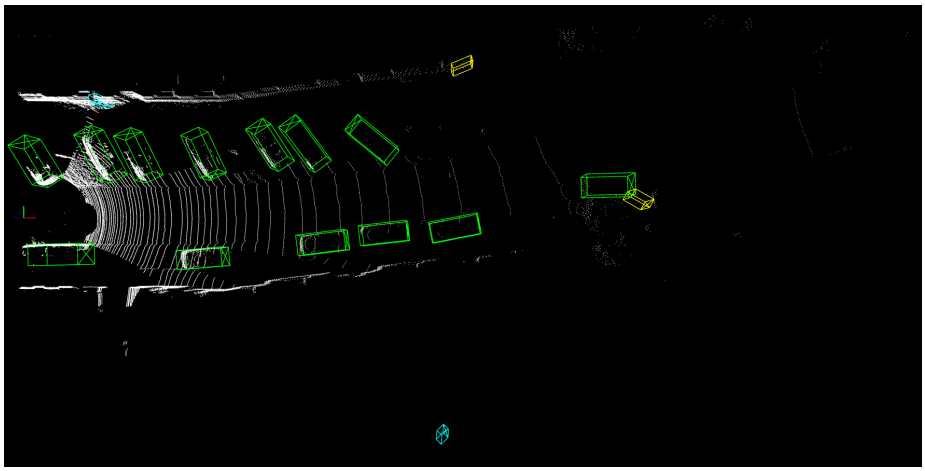
SECOND's 3D rendering of KITTI scene #61.



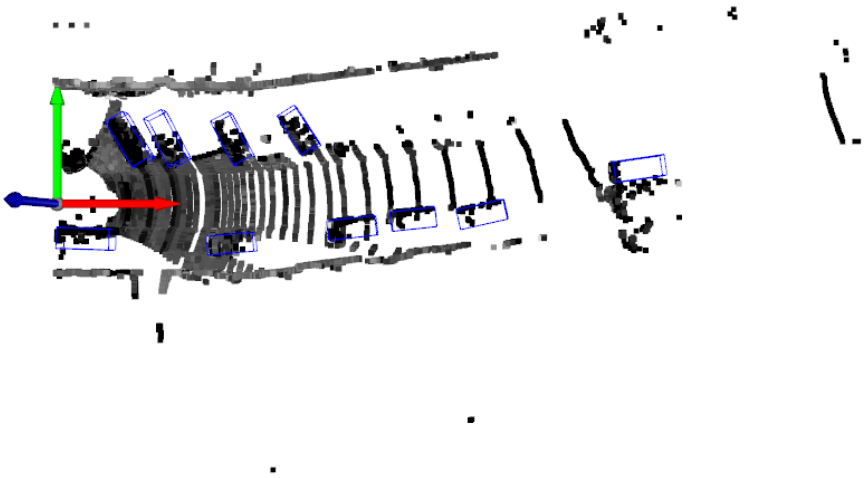
PointPillar's 3D rendering of KITTI scene #61.



PointRCNN's bird's eye view (BEV) rendering of KITTI scene #61.



SECOND's bird's eye view (BEV) rendering of KITTI scene #61.



PointPillar's bird's eye view (BEV) rendering of KITTI scene #61.

Conclusion

@Sanjay

References

1. Shi, Shaoshuai, Xiaogang Wang, and Hongsheng Li. "Pointrcnn: 3d object proposal generation and detection from point cloud." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019.
2. Yan, Yan, Yuxing Mao, and Bo Li. "Second: Sparsely embedded convolu-

tional detection." Sensors 18.10 (2018): 3337.
3. Lang, Alex H., et al. "Pointpillars: Fast encoders for object detection from point clouds." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019.