HDNET: Exploiting HD Maps for 3D Object Detection

Bin Yang[1,2] **Ming Liang**[1] **Raquel Urtasun**[1,2]
[1]Uber Advanced Technologies Group [2]University of Toronto
{byang10, ming.liang, urtasun}@uber.com

Abstract: In this paper we show that High-Definition (HD) maps provide strong priors that can boost the performance and robustness of modern 3D object detectors. Towards this goal, we design a single stage detector that extracts geometric and semantic features from the HD maps. As maps might not be available everywhere, we also propose a map prediction module that estimates the map on the fly from raw LiDAR data. We conduct extensive experiments on KITTI [1] as well as a large-scale 3D detection benchmark containing 1 million frames, and show that the proposed map-aware detector consistently outperforms the state-of-the-art in both mapped and un-mapped scenarios. Importantly the whole framework runs at 20 frames per second.

Keywords: 3D Object Detection, HD Maps, Autonomous Driving

1 Introduction

Autonomous vehicles have the potential of providing cheaper and safer transportation. A typical autonomous system is composed of the following functional modules: perception, prediction, planning and control [2]. Perception is concerned with detecting the objects of interest (*e.g.* vehicles) in the scene and track them over time. The prediction module estimates the intentions and trajectories of all actors into the future. Motion planning is responsible for producing a trajectory that is safe, while control outputs the commands necessary for the self-driving vehicle to execute such trajectory.

3D object detection is a fundamental task in perception systems. Modern 3D object detectors [3, 4] exploit LiDAR as input as it provides good geometric cues and eases 3D localization when compared to camera-only approaches. In the context of real-time applications, single-shot detectors [5, 6, 7] have been shown to be more promising than proposal-based methods [8, 4] as they are very efficient and can produce very accurate estimates. However, object detection is far from solved as many challenges remain, such as dealing with occlusion and the sparsity of the LiDAR at long range.

Most self-driving systems have access to High-Definition (HD) maps that contain geometric and semantic information about the environment. While HD maps are widely used by motion planning systems [9, 10], they are vastly ignored by perception systems [11]. In this paper we argue that HD maps provide strong priors that can boost the performance and robustness of modern object detectors. Towards this goal, we derive an efficient and effective single-stage detector that operates in Bird's Eye View (BEV) and fuses LiDAR information with rasterized maps. Bird's eye view is a good representation for 3D LiDAR as it is amenable to efficient inference and retains the metric space. Since HD maps might not be available everywhere, we also propose a map prediction module that estimates the map geometry and semantics from a single online LiDAR sweep.

Our experiments on the public KITTI BEV object detection benchmark [1] and a large-scale 3D object detection benchmark TOR4D [3, 12] show that we can achieve significant Average Precision (AP) gain on top of a state-of-the-art detector by exploiting HD maps. On TOR4D when HD maps are available, we achieve 2.42%, 3.43% and 5.49% AP gains for ranges over 0-70 m, 30-50 m and 50-70 m respectively. On KITTI, where HD maps are unavailable, we show that when using a pre-trained map prediction module (trained on a different continent) we can still get 2.87% AP gain, surpassing all competing methods including those which also exploit cameras. Importantly, the proposed map-aware detector runs at 20 frames per second.

2nd Conference on Robot Learning (CoRL 2018), Zürich, Switzerland.

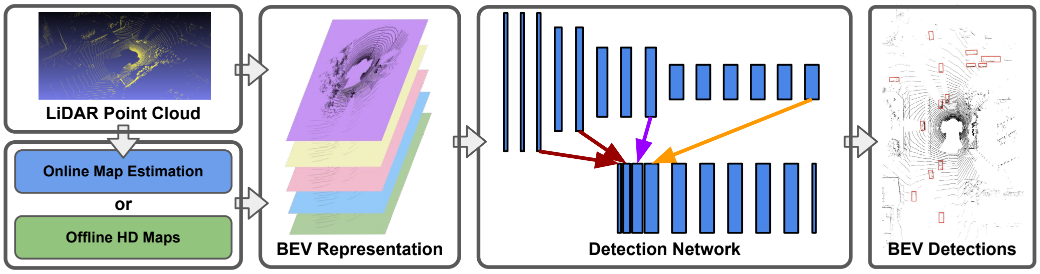

Figure 1: The overall architecture of the proposed map-aware single-stage 3D detector that operates on bird's eye view LiDAR representation.

2 Related Work

In this paper we show how modern 3D object detectors can benefit from HD maps. Here we revisit literatures on 3D object detection from point clouds, as well as works that exploit priors from maps.

2.1 3D Object Detection in Point Clouds

Some detectors [13, 14, 15] search objects in 3D space densely via sliding window. Due to the sparsity of LiDAR point cloud, these approaches suffer from expensive computation, as many proposals are evaluated where there is no evidence. A more efficient representation of point clouds is to exploit 2D projections. MV3D [4] uses a combination of range view and bird's eye view projections to fuse multi-view information. PIXOR [3] and FAF [12] show superior performance in both speed and accuracy by exploiting the bird's eye view representation alone. Apart from grid-like representations, some works [16, 17] learn feature representations from un-ordered point sets for object detection. In addition to point cloud based detection, many works [4, 17, 18] try to fuse data from multiple sensors to improve performance.

2.2 Exploiting Priors from Maps

Maps contain geographic, geometric and semantic priors that are useful for many tasks. [11] leverages dense priors from large-scale crowd-sourced maps to build a holistic model that does joint 3D object detection, semantic segmentation and depth reconstruction. However, the priors are extracted by rendering a 3D world from the map, which is very time-consuming. In [19, 20], the crowd-sourced maps are also used for fine-grained road segmentation. 3D object detectors often use the ground prior [21, 22]. However, they treat ground as a plane which is often inaccurate for curved roads. In this paper, we exploit the point-wise geometric ground prior as well as the semantic road prior for 3D object detection. In [23, 24], online estimated maps are used to reason the location of objects in 3D. However, the proposed generative model is very slow and thus not amenable for robotics applications. [25] uses HD maps to predict the intention of different traffic actors like vehicles, pedestrians and bicyclists, which belongs to the more high-level prediction system.

3 Exploiting HD Maps for 3D Object Detection

HD maps are typically employed for motion planning but they are vastly ignored for perception systems. In this paper we bridge the gap and study how HD maps can be used as priors for modern 3D object detectors. Towards this goal, we derive a single-stage detector that can exploit both semantic and geometric information extracted from the maps (whether built offline or estimated online). We refer the reader to Figure 1 for the overall architecture of our proposed map-aware detector.

3.1 Input Representation

We project the LiDAR data to bird's eye view (BEV) as this provides a compact representation that enables efficient inference. Note that this is a good representation for our application as vehicles

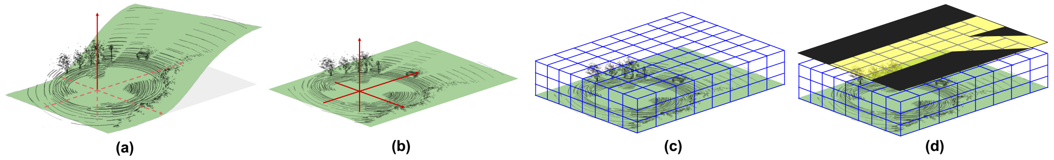

Figure 2: BEV LiDAR representation that exploits geometric and semantic HD map information. (a) The raw LiDAR point cloud. (b) Incorporating geometric ground prior. (c) Discretization of the LiDAR point cloud. (d) Incorporating semantic road prior.

drive on the ground. Figure 2 illustrates how we incorporate priors from HD maps into our BEV LiDAR representation.

We treat the Z axis as feature channel and exploit 2D convolutions. This is beneficial as they are much more efficient than 3D convolutions. It is therefore important to have discriminant features along the Z axis. However, LiDAR point clouds often suffer from translation variance along the Z axis due to the slope of the road, particularly at range. For example, 1 degree of road slope would lead to 1.22 meters offset in Z axis at 70 meters distance. To make things worse, standard LiDAR sensors have very sparse returns after 50 meters.

To remedy this translation variance, we exploit detailed ground information from the HD maps. Specifically, given a LiDAR point cloud $\{(x_i, y_i, z_i)\}$, we query the ground point at each location (x_i, y_i) from the map, denoted as (x_i, y_i, z_i^0) and replace the absolute distance z_i with the distance relative to the ground $z_i - z_i^0$. We then discretize the resulting representation into a 3D occupancy grid enhanced to also contain LiDAR intensity features following [3]. Specifically, we first define the 3D dimension $L \times W \times H$ of the scene that we are interested in. We then compute binary occupancy maps at a resolution of $d_L \times d_W \times d_H$, and compute the intensity feature map at a resolution of $d_L \times d_W \times H$. The discretized LiDAR BEV representation is of size $\frac{L}{d_L} \times \frac{W}{d_W}$, with $\frac{H}{d_H} + 3$ channels[1].

The LiDAR provides a full scan of the surrounding environment, containing moving objects and static background (*e.g.* roads and buildings). However, in the context of self driving cars we mainly care about moving objects on the road. Motivated by this, we exploit the semantic road mask available on HD maps as prior knowledge about the scene. Specifically, we extract the road layout information from the HD maps and rasterize it onto the bird's eye view as a binary channel at the same resolution as the discretized LiDAR representation. We then concatenate the road mask together with the LiDAR representation along the channels to create our input representation.

3.2 Network Structure

We adopt a fully convolutional network for single-stage dense object detection. Figure 3 (left) shows the detection network, which is composed of two parts: a *backbone network* that extracts multi-scale features and a *header network* that outputs pixel-wise dense detection estimates.

Backbone network: consists of four convolutional blocks, each having {2, 2, 3, 6} `conv2D` layers with filter number {32, 64, 128, 256}, filter size 3, and stride 1. We apply batch normalization [26] and ReLU [27] after each `conv2D` layer. After each of the first three convolutional blocks there's a `MaxPool` layer with filter size 3 and stride 2. Multi-scale features are generated by resizing and concatenating feature maps from different blocks. The total downsampling rate of the network is 4.

Header network: consists of five `conv2D` layers with filter number 256, filter size 3 and stride 1, followed by the last `conv2D` layer that outputs pixel-wise dense detection estimates $(p, cos(2\theta), sin(2\theta), dx, dy, \log w, \log l)$. p denotes the confidence score for object classification, where the rest denotes the geometry of the detection. Specifically, we parameterize the object orientation (in XY plane) by $(cos(2\theta), sin(2\theta))$ with a period of π so that we do not distinguish between heading forward and backward[2]. $(dx, dy, \log w, \log l)$ are commonly used parameterization [28]

[1] Two additional occupancy channels are added to cover points outside the height range.

[2] Because they are the same in terms of IoU computation.

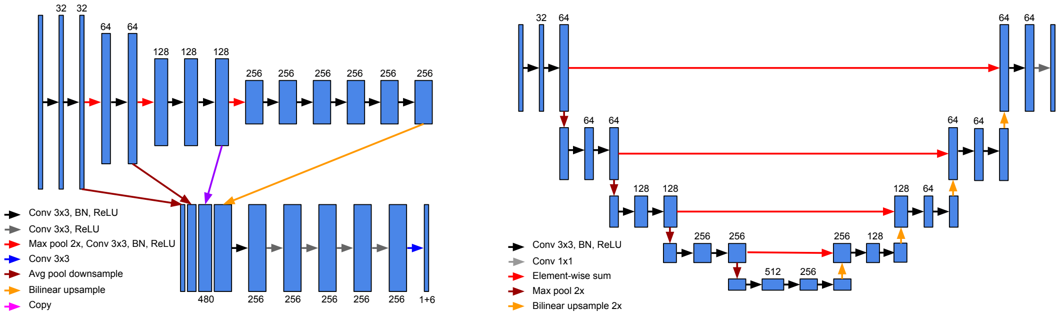

Figure 3: Network structures for object detection (left) and online map estimation (right).

for object center offset and size. The advantages of producing dense detections are two-fold: it is efficient to compute via convolutions and the maximum recall rate of objects is 100%.

3.3 Learning and Inference

Adding priors (semantic prior in particular) at the input level has both advantages as well as challenges. On one hand, the network can fully exploit the interactions between priors and raw data; on the other hand, this may lead to overfitting issues, and potentially poor results when HD maps are unavailable or noisy. In practice, having a detector that works regardless of map availability is important. Towards this goal, we apply *data dropout* on the semantic prior, which randomly feeds an empty road mask to the network during training. Our experiments show that data dropout largely improves the model's robustness to map availability.

We employ the commonly used multi-task loss [28] to train the detection network. Specifically, we use focal loss [5] on the classification task and smooth ℓ_1 loss on the regression task. The total loss is simply the combination of classification loss summed over all pixels and regression loss summed over all positive pixels (no sampling strategy is required). We determine positive and negative samples according to pixel distance to the nearest ground-truth box center. We also normalize the regression targets to have zero mean and unit variance before training.

During inference, we pick all pixels on the output feature map whose confidence score is above a certain threshold, and decode them into oriented bounding boxes. Non-Maximum-Suppression (NMS) with 0.1 Intersection-Over-Union (IoU) threshold is then applied to get the final detections.

4 Online Estimation of HD Maps

So far we have shown how to exploit geometric and semantic priors from the already built HD maps. However, such maps might not be available everywhere. To tackle this problem, we create the map priors online from a single LiDAR sweep. In our scenario, there is no need to estimate dense 3D HD maps. Instead, we only have to predict the BEV representation of the geometric and semantic priors (shown in Figure 2). In this way, the estimated map features can be seamlessly integrated into the current framework.

Network structure: We estimate online maps from LiDAR point clouds via two separate neural networks that tackle ground estimation and road segmentation respectively. The network structure for these two tasks are the same U-Net structure [29] as shown in Figure 3 (right). This has the advantages of retaining low-level details and producing high-resolution predictions.

Ground estimation: Existing approaches [22, 21] assume that the ground is a plane. This is, however, inaccurate, particularly at long range. Instead, in this paper we predict the ground height value for every location on the BEV space. This results in a much more precise estimate. During training, we extract ground-truth ground from existing HD maps, and compute ℓ_2 loss only on locations where there exist LiDAR points. Empirically we find that ignoring empty locations during training leads to better performance.

Method	HD Maps	Ground	Road	Time (ms)	AP@0.7IoU (%)		
					0-70 m	30-50 m	50-70 m
FAF [12]	no	N/A	N/A	30	77.48	N/A	N/A
PIXOR [3]	no	N/A	N/A	17	78.82	N/A	N/A
PIXOR++	no	N/A	N/A	17	81.78	78.34	61.04
HDNET (online)	no	√		32	+0.91	+1.69	+0.92
			√	32	+0.88	+1.54	+1.87
		√	√	47	**+1.33**	**+2.34**	**+1.88**
HDNET (offline)	yes	√		17	+1.58	+2.29	+3.24
			√	17	+1.05	+1.53	+2.52
		√	√	17	**+2.42**	**+3.43**	**+5.49**

Table 1: Evaluation results on TOR4D benchmark at different ranges. *PIXOR++* is the baseline detector without exploiting HD maps. *HDNET (online)* exploits online estimated map priors on top of the baseline, while *HDNET (offline)* extracts priors from offline built HD maps.

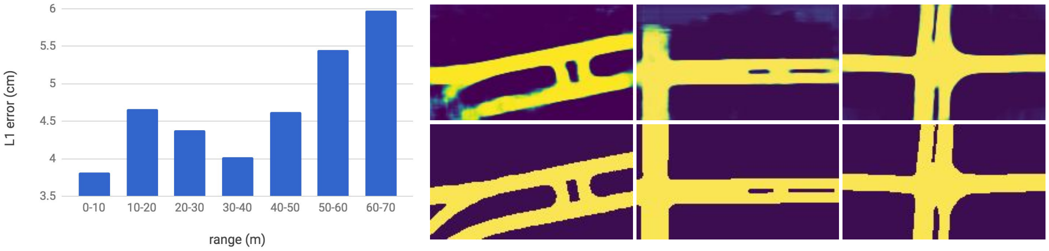

Figure 4: Evaluation of online map estimation. Left: range-wise error of ground height estimation. Right: the predicted road mask (1st row) versus the ground-truth road mask (2nd row).

Road segmentation: We predict pixel-wise BEV road segmentation as the estimated road prior. During training we use the rasterized road mask as ground-truth label and train the network with cross-entropy loss summed over all locations. We refer the reader to Figure 2 for an illustration.

5 Experiments

We validate the proposed method on two benchmarks: a large-scale 3D object detection benchmark TOR4D [3, 12], which has over one million frames as well as corresponding HD maps; and the public KITTI BEV object detection benchmark [1]. We denote our baseline detector without exploiting map priors as PIXOR++ since it follows the overall architecture of PIXOR [3], and denote the map-aware detector which exploits map priors as HDNET.

5.1 Evaluation on TOR4D Benchmark

Implementation details: We discretize the LiDAR point cloud within the following 3D region $x \in [-70.4, 70.4]$, $y \in [-40, 40]$, $z \in [-2, 3.4]$ in vehicle coordinate system (with the ego-car located at the origin). We use a discretization resolution of 0.2 meter in all three axes. The detection network is trained with stochastic gradient descent with momentum for 1.5 epochs with a batch size of 32. The initial learning rate is 0.02 and is multiplied by 0.1 after 1 and 1.4 epochs.

Baseline PIXOR++ detector: The baseline detector achieves 81.78% AP within 0-70 meters, outperforming FAF [12] and PIXOR [3] by 4.3% and 2.96% respectively. What's more, the baseline detector runs at 17 ms per frame. In terms of performance at different ranges, we observe an AP drop of 3.44% and 20.74% when evaluated at 30-50 m range and 50-70 m range. This suggests that long range detection is significantly difficult.

Online map estimation: Figure 4 shows the evaluation results of the online map prediction module. In ground height estimation task, we are able to achieve < 5 cm L1 error pixel-wise within 50 m range. When the range becomes longer, the error increases since we have more sparse LiDAR

Method	Input	Time (ms)	AP@0.7IoU (%)		
			easy	moderate	hard
BirdNet [22]	LiDAR	110	75.52	50.81	50.00
MV3D [4]	LiDAR	240	85.82	77.00	68.94
PIXOR [3]	LiDAR	**35**	81.70	77.05	72.95
VoxelNet [16]	LiDAR	225	89.35	79.26	77.39
MV3D [4]	LiDAR + Image	360	86.02	76.90	68.49
F-PointNet [17]	LiDAR + Image	170	88.70	84.00	75.33
AVOD [18]	LiDAR + Image	80	86.80	85.44	77.73
ContFuse [30]	LiDAR + Image	60	88.81	85.83	77.33
PIXOR++	LiDAR	**35**	**89.38**	83.70	77.97
HDNET	LiDAR + Map	50	89.14	**86.57**	**78.32**

Table 2: Evaluation results on KITTI BEV object detection benchmark (car). *PIXOR++* is the baseline detector, while *HDNET* exploits map priors from an online map estimation module pre-trained on TOR4D dataset.

observations there. In road segmentation task, we achieve 97.70% pixel-wise accuracy and 92.86% IoU on the validation set. Note that these results are from one single LiDAR sweep only.

Map-aware HDNET detector: As seen from Table 1, HDNET outperforms the baseline PIXOR++ by 1.33%/2.42% in settings with online/offline maps. When it comes to longer range, the gain in both settings increases, especially for the offline one (with 5.49% gain in 50-70 m). This shows that map priors are especially helpful for the more difficult long range detection. Comparing geometric and semantic priors in both settings, we have three main observations: (1) geometric prior generally helps more than semantic prior, while in online setting the advantage diminishes (especially in long range) due to larger errors in online ground height estimation; (2) geometric prior and semantic prior are very complementary to each other; (3) in all settings and all ranges, we show that exploiting map priors could bring consistent gain over a state-of-the-art detector.

5.2 Evaluation on KITTI BEV Benchmark

Implementation details: We set the region of interest for the point cloud to $[0, 70.4] \times [-40, 40] \times [-3, 1]$ meter and use a discretization resolution of $0.1 \times 0.1 \times 0.2$ meter. Following [3], we use the same ResNet [31] based network structure that's more robust to overfitting. Since KITTI has only thousands of training frames and we do not use pre-trained weights for the network, we apply the following data augmentation during training. For each frame of LIDAR point cloud, we apply random scaling ($0.9 \sim 1.1$ for all 3 axes), translation ($-5 \sim 5$ meter for X and Y axes) and rotation ($-5 \sim 5$ degrees along Z axis). KITTI only annotates objects within the camera field of view (FOV), therefore we ignore all pixels outside the camera FOV during training. We use $\alpha = 0.75$ and $\gamma = 0.5$ in the focal loss. We train the network with stochastic gradient descent with momentum for 50 epochs with a batch size of 16 frames. The initial learning rate is 0.01 and we decay it by 0.1 after 30 and 45 epochs respectively.

BEV object detection: We compare PIXOR++ and HDNET with other state-of-the-art detectors on KITTI BEV object detection benchmark and show the evaluation results in Table 2. Our baseline detector PIXOR++ has already surpassed all the competing LiDAR based detectors, outperforming the second best VoxelNet [16] by 4.44% AP in moderate setting while being the fastest in runtime (35 ms). Because KITTI doesn't have HD maps information, we transfer the map prediction module trained on TOR4D to KITTI without any fine-tuning. Note that we do not exploit any object labels from the TOR4D dataset. With map priors added, HDNET achieves an absolute 2.87% AP gain in moderate setting, setting the new record on the benchmark. Note that HDNET even outperforms other competitors that exploit camera images (*e.g.* ContFuse [30]) or additional object labels from outside datasets (*e.g.* F-PointNet [17]), which highlights the long-ignored value of HD maps in 3D object detection.

Method	AP@easy	AP@mod.	AP@hard
baseline	87.31	79.29	76.02
+ground plane (3DOP [21])	86.67	79.23	79.17
+ground surface (ours)	**88.45**	**82.60**	**80.43**

(a) Ground prior

Method	AP
multi-task learning	+0.25
output masking	-1.14
input fusion (ours)	**+2.42**

(b) Road prior

Map Avail.	0%	50%	100%
w/o dropout	-11.57	-6.37	+0.77
w/ dropout	**-0.20**	**+0.34**	**+0.84**

(c) Data dropout on road prior

Range	0-10	10-20	20-30	30-40	40-50	50-60	60-70	0-70
online map	-0.15	+0.44	+1.17	+2.45	+1.88	+0.97	+1.03	+1.33
offline map	-0.09	+0.65	+1.45	+2.93	+3.71	+4.25	+6.38	**+2.42**

(d) Detection performance at different ranges

Table 3: Ablation studies on different ways to exploit ground prior and road prior, robustness to unavailable maps, and range-wise detection performances. All numbers are AP (%) values.

5.3 Ablation Studies

We conduct four ablation studies investigating different ways to exploit the geometry and semantic priors, the effect of data dropout on robustness to unavailable maps, and the detection performance with regard to different ranges using priors from online estimated maps or offline built maps.

5.3.1 Ground prior

We model the geometric prior as a point-wise ground surface, which is more flexible and accurate than commonly used ground plane assumption [21, 22]. We compare these two design choices. For the ground plane counterpart, we use the publicly available ground plane results from 3DOP [21], which were generated by road segmentation and RANSAC fitting. In contrast, we directly predict the point-wise ground. The way how the ground prior is used in detection is the same for these two methods. We evaluate these two priors using HDNET on KITTI validation set (without data augmentation) and show the results in Table 3a. Both methods can boost the detection performance of the baseline, while our point-wise ground parameterization achieves significantly more gain.

5.3.2 Road prior

We incorporate semantic road prior into the input representation of the detector, while there are methods that use such prior at the output level. Two variants are created for output level fusion: multi-task learning and output masking. For multi-task learning, we add another output branch to the detection network to predict the road mask during training. For output masking, we mask the detection output with the road mask during both training and testing stages. We show the comparative results using the road prior extracted from offline HD maps in Table 3b, where we see that our input fusion performs the best among all three methods. For multi-task learning, we observe limited performance gain from jointly learning road mask and object detection. Output masking hurts the detection performance as errors in road prior is unrecoverable.

5.3.3 Data dropout

Data dropout is applied to the map prior during training stage to improve the model's robustness to missing maps. To validate the effectiveness of data dropout, we train two models with 100% map availability. The only difference is that one model is trained with data dropout applied to the road prior, while the other model doesn't use data dropout. During testing, we manually control the availability of road prior at $\{0\%, 50\%, 100\%\}$ and compare the performance.

The evaluation results are shown in Table 3c, where we see that without data dropout, AP drops by 11.57%/6.37% at 0%/50% map availability. However, when data dropout is introduced, the map-aware detector is able to surpass the baseline when map's available; and when map's unavailable, the performance is almost as good as the baseline.

5.3.4 Performance at various ranges

We define the range as the distance from the object to the ego-car on the XY plane. We divide the full 70 meters range into 7 bins, and when evaluating on each range bin we ignore the detections

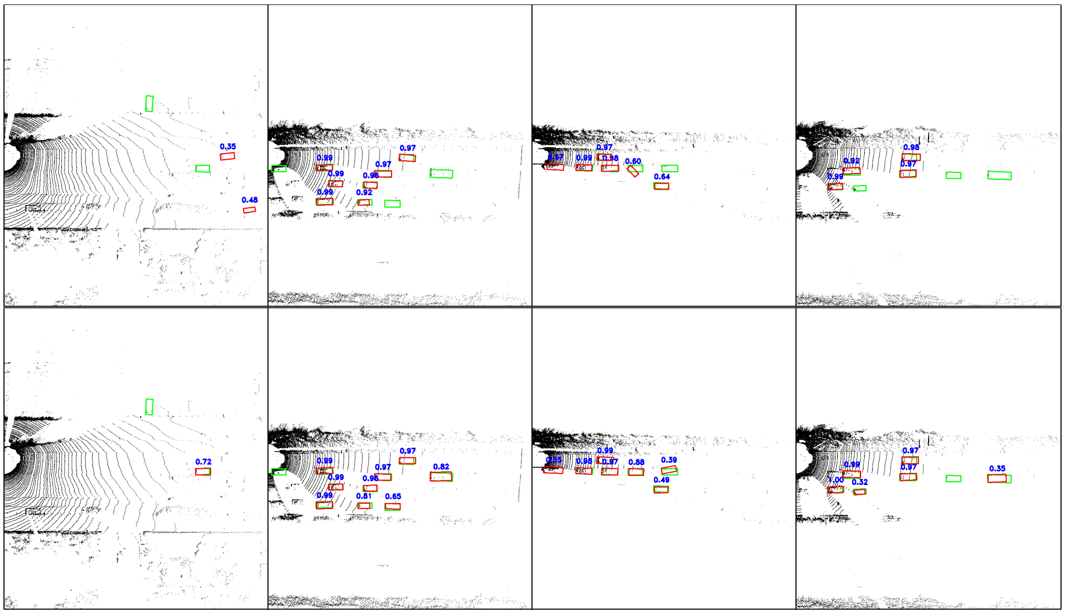

Figure 5: Qualitative comparison of baseline PIXOR++ detector without map (1st row) and HDNET with online map (2nd row) on KITTI BEV object detection benchmark validation set. We show ground-truth labels in green, detection results in red, and detection scores in blue.

and labels outside this range bin. We conduct the fine-grained evaluation in both online and offline settings and show results in Table 3d.

For offline setting, we see that the gain increases as the object range becomes larger. At > 40 meters range, the gain is larger than 3%. This makes sense as the map priors help more when we have fewer LiDAR observations. For online setting, we have the similar trending in near range (< 40 meters), but the gain starts to decrease at long ranges (> 40 meters). The reason is that map priors are estimated from single LiDAR sweep, which is very sparse in long range.

5.4 Qualitative Results

We compare some qualitative results from two detector variants: the baseline PIXOR++ detector without map, and the HDNET detector with online map estimation on the validation set of KITTI BEV object detection benchmark in Figure 5.

From the figure we can easily identify false positives and false negatives in long range for PIXOR++. However, when it comes to HDNET, the detection accuracy and localization precision at long range increases remarkably. This shows that exploiting map priors does help the long range detection a lot.

6 Conclusion

In this paper we address the problem of how to exploit HD maps information to boost the performance of modern 3D object detection systems in the context of autonomous driving. We identify the geometric and semantic priors in HD maps, and incorporate them into the bird's eye view LiDAR representation. Because HD maps are not available everywhere, we also propose a map prediction module that estimates both map priors online from single LiDAR sweep. Experimental results on the public KITTI BEV object detection benchmark and a large-scale 3D object detection benchmark show that the proposed map-aware detector consistently outperforms the baseline detector whether the HD maps are available or not. The whole framework also runs at over 20 frames per second due to the use of BEV representation as well as the single-stage detection framework.

References

[1] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3354–3361. IEEE, 2012.

[2] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, et al. Towards fully autonomous driving: Systems and algorithms. In *IEEE Intelligent Vehicles Symposium (IV)*, 2011.

[3] B. Yang, W. Luo, and R. Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[4] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[5] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[6] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*, 2016.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[8] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.

[9] H. G. Seif and X. Hu. Autonomous driving in the icity?hd maps as a key challenge of the automotive industry. *Engineering*, 2(2):159–162, 2016.

[10] Y. F. Chen, S.-Y. Liu, M. Liu, J. Miller, and J. P. How. Motion planning with diffusion maps. *International Conference on Intelligent Robots and Systems*, 2016.

[11] S. Wang, S. Fidler, and R. Urtasun. Holistic 3d scene understanding from a single geo-tagged image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3964–3972, 2015.

[12] W. Luo, B. Yang, and R. Urtasun. Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[13] D. Z. Wang and I. Posner. Voting for voting in online point cloud object detection. In *Robotics: Science and Systems*, 2015.

[14] M. Engelcke, D. Rao, D. Z. Wang, C. H. Tong, and I. Posner. Vote3deep: Fast object detection in 3d point clouds using efficient convolutional neural networks. *International Conference on Robotics and Automation*, 2017.

[15] B. Li. 3d fully convolutional network for vehicle detection in point cloud. *International Conference on Intelligent Robots and Systems*, 2017.

[16] 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.

[17] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[18] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. Waslander. Joint 3d proposal generation and object detection from view aggregation. *International Conference on Intelligent Robots and Systems*, 2018.

[19] G. Mattyus, S. Wang, S. Fidler, and R. Urtasun. Enhancing road maps by parsing aerial images around the world. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1689–1697, 2015.

[20] G. Máttyus, S. Wang, S. Fidler, and R. Urtasun. Hd maps: Fine-grained road segmentation by parsing ground and aerial images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3611–3619, 2016.

[21] X. Chen, K. Kundu, Y. Zhu, H. Ma, S. Fidler, and R. Urtasun. 3d object proposals using stereo imagery for accurate object class detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[22] J. Beltran, C. Guindel, F. M. Moreno, D. Cruzado, F. Garcia, and A. de la Escalera. Birdnet: a 3d object detection framework from lidar information. *IEEE International Conference on Intelligent Transportation Systems*, 2018.

[23] H. Zhang, A. Geiger, and R. Urtasun. Understanding high-level semantics by modeling traffic patterns. In *Proceedings of the IEEE International Conference on Computer Vision*, 2013.

[24] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun. 3d traffic scene understanding from movable platforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(5): 1012–1025, 2014.

[25] N. Djuric, V. Radosavljevic, H. Cui, T. Nguyen, F.-C. Chou, T.-H. Lin, and J. Schneider. Motion prediction of traffic actors for autonomous driving using deep convolutional networks. *arXiv preprint arXiv:1808.05819*, 2018.

[26] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 448–456, 2015.

[27] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning*, 2010.

[28] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

[29] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015.

[30] M. Liang, B. Yang, S. Wang, and R. Urtasun. Deep continuous fusion for multi-sensor 3d object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[31] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, pages 630–645. Springer, 2016.