3D Vehicle Detection Using Camera and Low-Resolution LiDAR

Lin Bai, Yiming Zhao and Xinming Huang

Abstract— Nowadays, Light Detection And Ranging (LiDAR) has been widely used in autonomous vehicles for perception and localization. However, the cost of a high-resolution LiDAR is still prohibitively expensive, while its low-resolution counterpart is much more affordable. Therefore, using low-resolution LiDAR for autonomous driving perception tasks instead of high-resolution LiDAR is an economically feasible solution. In this paper, we propose a novel framework for 3D object detection in Bird-Eye View (BEV) using a low-resolution LiDAR and a monocular camera. Taking the low-resolution LiDAR point cloud and the monocular image as input, our depth completion network is able to produce dense point cloud that is subsequently processed by a voxel-based network for 3D object detection. Evaluated with KITTI dataset, the experimental results shows that the proposed approach performs significantly better than directly applying the 16-line LiDAR point cloud for object detection. For both easy and moderate cases, our detection results are comparable to those from 64-line high-resolution LiDAR. The network architecture and performance evaluations are analyzed in detail.

I. INTRODUCTION

In recent years, many research have been focused on autonomous driving technology. LiDAR is one of the most important sensors for perception tasks such as drivable region segmentation [1][2], object detection [3] and vehicle tracking [4]. Different from images captured by cameras, point cloud generated by LiDARs supplies 3D spatial information of the objects in the form of (X, Y, Z) coordinates and intensity. This alleviates the barrier of distance estimation and makes 3D object detection or tracking much more accurate. However, the price of high-resolution LiDARs is much higher than their low-resolution counterparts. The specification of the most popular Velodyne 64-line LiDAR HDL-64E and 16-line LiDAR VLP-16 is compared in Table I. As we can see, the cost of a low-resolution LiDAR is only about 1/10 of the high-resolution ones.

| LiDAR type | VLP-16 | HDL-64E |
|------------|--------|---------|
| Range (meters) | 100 | 120 |
| Channel number | 16 | 64 |
| FOV | -15°~ +15° | -2°~ +24.9° |
| Res. (vertical) | 2° | 0.4° |
| Res. (horizontal) | 0.2° | 0.1728° |
| Power (Watts) | 8 | 60 |
| Price (USD) | $8'000 | $75'000 |

*This work was supported by US National Science Foundation
Lin Bai, Yiming Zhao and Xinming Huang are with the Department of Electrical and Computer Engineering, Worcester Polytechnic Institute, Worcester, MA 01609, USA lbai2, yzhao7, xhuang@wpi.edu

Therefore, it is necessary to pay attention to the low-resolution LiDARs in order to build low-cost autonomous driving systems. However, it is a major challenge to perform object detection from the point cloud produced by a low-resolution LiDAR since it is too sparse to even show the shapes of objects. As illustrated in Fig. 2, we can barely find objects from the depth map captured from 16-line LiDAR, while the 64-line LiDAR data is much more visible.

II. RELATED WORKS

A. Low-Resolution LiDAR for Perception

Some research works were focused on segmentation using low-resolution LiDARs. [5] introduced the local normal vector for the LiDAR’s spherical coordinates as an input channel. Based on the existing LoDNN architectures [2], its road segmentation performance using low-resolution LiDAR was comparable to that from high-resolution LiDAR within a reasonable degradation. A supervised domain adaption was utilized by [6] to predict the low-resolution point cloud into high-resolution point cloud in spherical coordinate and further evaluated the results in 3D semantic segmentation task. Low-resolution LiDARs had been also employed for object tracking tasks. In [7], a LiDAR-based system was proposed for the actual position and velocities estimation of the detected vehicles. The tracking results was evaluated in terms of distance between LiDAR and target vehicles. The tracking performance decreases sharply with the increasing of distance. Some other works utilized depth completion for 2D object detection, such as [12] and [11]. In [12], a weighted depth filling algorithm was proposed to make the high-resolution (HDL-64E) LiDAR depth map even denser. Subsequently, this dense depth map was concatenated with the corresponding RGB image as the input of YOLOv3 [13] network for 2D object detection. Similarly, the authors of [11] introduced a self-supervised depth completion network to fill the high-resolution (HDL-64E) LiDAR depth map. 2D object detection networks such as Faster R-CNN [14] and SSD [15] were trained using dense depth map and image as inputs.
B. High Resolution LiDAR for BEV Object Detection

Nearly all state-of-the-art object detectors utilize high-resolution LiDAR. In [16], it first transformed the point cloud into BEV map, and then extracted the ground and proposed the objects in two branches separately. Finally the objects were predicted by a post-processing block. [17] further refined the previous version into an end-to-end model and achieved better performance. Single-stage detector, PIXOR, was proposed in [9] by using 2D convolution on the voxelized BEV map. Without any anchor, it achieved real-time processing speed.

As mentioned earlier, due to the extreme sparsity, low-resolution LiDAR depth map does not supply enough shape information of the objects, but some sub-samples of the precise depth information. Meanwhile, the RGB image supplies rich context information. Thus, we argue that when fusing sparse depth map and RGB image together, object detection becomes possible.

III. PROPOSED CNN-BASED FRAMEWORK

In this paper, we investigate the possibility of low-resolution LiDAR usage in BEV object detection task. In Fig. 2, red box, orange box and blue box represents the vehicle in short range, medium range and long range respectively. For short range vehicles, their shapes are clearly visible from dense depth maps. In sparse depth maps, the shapes are very blurry but still recognizable since the number of points hitting on the vehicles is still large enough. Concerning to the medium and long range vehicles (in orange and blue boxes), we can only get a small number of points even using 64-line LiDAR. While in the sparse depth map from 16-line LiDAR, the number of hit points is few to none. Taking the medium range vehicles in orange boxes in Fig. 2 (h) for instance, it is easy to recognize them as obstacles due to sharp distance distinction but difficult to recognize them as vehicles. This also applies to vehicles with occlusion (green boxes in Fig. 2 (c), (f) and (i)). The long range vehicles in blue box (in both Fig. 2(e) and (h)) get too few point to be correctly localized and classified. According to the analysis above, we found that unlike the depth map from 64-line LiDAR, 16-line LiDAR depth map does not show reliable context information but accurate distance information. This implies that 16-line LiDAR depth map is more useful for depth estimation rather than context information extraction. Therefore, to better use the information from 16-line depth map, we put a depth completion network prior to the object detector to generate a dense depth map with context information (Fig. 3). After the dense depth map is generated, it is sent into 3D object detector, as demonstrated in Fig. 3.
adapted here with some modifications. It requires two inputs, RGB image and low-resolution sparse depth map. The RGB image supplies the context information in detail, while the sparse depth map supplies the precise depth information for some pixels on the image. The sensor fusion strategy adopted here is also referred as early fusion. To make the network more compact, we first replace the ResNet-34 backbone with ResNet-18. For performance improvement, global attention modules and an Atrous Spatial Pyramid Pooling (ASPP) [18] module are placed to bridge the encoder and decoder.

As shown in Fig. 5, the global attention module is used to extract global context information of the feature map by global pooling layer, and then fuse the global information back to guide the feature learning. Through adding this module, the global information is merged into features without up-sampling layer. This helps the decoder part to achieve better performance. Besides, an ASPP module (Fig. 6) is placed between encoder and decoder, with each convolution dilated rate 2, 4, 8 and 16. The ASPP module concatenates feature maps with different field of perception, so that decoder has a better understanding of the context information.

The loss function of depth completion network is shown in Eq. (1) which calculates the Mean Square Error (MSE) between the predicted depth map and the ground truth.

$$L_{\text{depth}} = \frac{1}{N} \sum_{x} \sum_{y} (depth_{\text{pred}} - depth_{\text{gt}})^2$$  \hspace{1cm} (1)

B. Object Detection Network

The object detection network adopted in this framework is PIXOR [9]. Its main idea is to take the advantage of 2D convolution and anchor-free network to realize super-fast point cloud object detection in BEV. PIXOR consists of two steps. The first step is to reform the representation of input point cloud. It reduces 3 degrees of freedom to 2 in BEV, and extracts the 3rd freedom (z or height) as another input feature map channel. So that 2D convolution instead of 3D convolution can be used to greatly decrease the computation complexity. The second step is to feed reformed input feature map into an anchor-free one-stage object detector network (Fig. 7). For the highly efficient computation on dense predictions, a fully convolutional architecture is utilized to build the backbone and header of PIXOR. Without any predefined anchors and proposals, PIXOR outputs the predicted class and orientation from header in a single network.

Concerning to the loss function, the total loss of object detection consists of the classification loss and the regression
loss (Eq. 2), where \( \lambda_{\text{cls}} \) and \( \lambda_{\text{reg}} \) are the corresponding coefficients. The classification loss \( L_{\text{cls}} \) targets to correctly predict the object (cars in our case) and the regression loss \( L_{\text{reg}} \) aims to refine the size, center and the orientation of the predicted bounding boxes.

\[
L_{\text{detect}} = \lambda_{\text{cls}} L_{\text{cls}} + \lambda_{\text{reg}} L_{\text{reg}}
\] (2)

C. Implementation Details

The depth completion network is firstly trained on KITTI Depth Completion dataset. The depth completion network is trained with batch size of 4, and learning rate starts at 1e-4 which decreases every 5 epochs. The total number of training epoch is 10. After training the depth completion network and keeping it as is, we move on to train the object detector from scratch. The KITTI Object Detection dataset has been split into training and validation parts according to [10]. The optimizer is Adam, with batch size 8. The learning rate starts at 1e-3 and reduces by a factor of 2 when the validation loss does not decrease. Finally, we fine-tune the entire framework with both depth completion network and object detection network together, with 16-line point cloud and images as input and vehicles in BEV as output.

IV. EXPERIMENT RESULTS

A. Dataset

Training and evaluation of the whole framework both employ KITTI dataset (both Depth Completion and Object Detection). Before feed into the framework mentioned above, the point cloud are down-sampled to emulate the VLP-16 low-resolution LiDAR. KITTI depth completion dataset contains 85,898 training data and 1,000 selected validation data. Its ground truth is produced by aggregating consecutive LiDAR scan frames into a semi-dense depth map, about 30% annotated pixels. KITTI object detection dataset has 7,481 training data and 7,518 testing data. Evaluation is categorized into three regimes: easy, moderate and hard, representing objects at different occlusion and truncation levels.

B. Depth Completion Performance Evaluation

As described in Sec. III-A, in order to enhance the depth completion performance, multiple GAM modules have been added to bridge the encoder and the decoder of depth completion network. The performance comparison on validation dataset is illustrated in Tab. II. Adding GAM modules results the performance improvement of about 3.6% and 7.0% measured by Root Mean Square Root (RMSE) and Mean Average Error (MAE) respectively.

TABLE II: Depth completion performance comparison with and without GAM modules

| with GAMs | RMSE (1/mm) | MAE (1/mm) |
|-----------|-------------|------------|
| Yes       | 1592.74     | 537.81     |
| No        | 1651.68     | 578.14     |

Fig. 8(b) and (c) demonstrate the predicted depth maps of depth completion networks with and without GAM modules respectively. And the bottom figure shows the ground truth. In this example, the depth map from depth completion network with GAM module gives objects slightly better shape representation.

Fig. 8: Comparison of depth map from 16-line LiDAR with and without GAM module to their RGB image and ground truth

C. Object Detector Performance Evaluation

The performance of our framework on KITTI object detection validation dataset is illustrated in Tab. III and Fig. 9. The results are shown in two circumstances IoU=0.5 and IoU=0.7 respectively. When IoU=0.5, our frameworks achieves 89.0%, 75.8% and 68.1% detection accuracy for easy, moderate and hard cases respectively. While in case of IoU=0.7, the prediction accuracy is decreased into 75.4%, 61.2% and 55.2% respectively. Comparing to feeding 16-line point cloud directly into PIXOR, our framework pulls up the detection accuracy significantly in all cases. If comparing to PIXOR with 64-line point cloud as input, the performance of our framework is relatively comparable in easy and moderate cases. But in hard case, the prediction accuracy drops around 20% in both IoU criteria. The precision-recall curve is demonstrated in Fig. 10.

D. Analysis and Future Work

1) Location prediction: The depth completion task in KITTI adopts RMSE and MAE as benchmark. However, if training with RMSE or MAE as loss function, the network cannot distinguish object boarder pixels as foreground pixels or background pixels. Instead, to reach higher RMSE and MAE, depth completion network attempts to predict the depth of the car edge as a value between the distance of the vehicle and the distance of the ground in front of the car.
TABLE III: **BEV performance comparison on KITTI object detection validation dataset.** This table shows $AP_{BEV}$ (in %) of the car category, corresponding to average precision of the bird’s-eye view.

| Detection Networks | Input                        | IoU=0.5 Easy | IoU=0.5 Moderate | IoU=0.5 Hard | IoU=0.7 Easy | IoU=0.7 Moderate | IoU=0.7 Hard |
|--------------------|------------------------------|--------------|-----------------|-------------|--------------|-----------------|-------------|
| PIXOR[9]           | LiDAR only (64-line)        | 94.2         | 86.7            | 86.1        | 85.2         | 81.2            | 76.1        |
| PIXOR              | LiDAR only (16-line)        | 60.7         | 51.2            | 46.8        | 53.8         | 47.1            | 39.1        |
| Ours               | LiDAR (16-line) + Camera    | 89.0         | 75.8            | 68.1        | 75.4         | 61.2            | 55.2        |

![Figure 9](image1)

Fig. 9: Visualization of object detection from the proposed framework, where the green boxes are ground truth and the blue boxes represent the predicted results.
Fig. 10: Precision-Recall curve of the proposed framework on KITTI val dataset.

(Fig. 11). This "long-tail" boundary effect might make the object localization more difficult. In our future work, a new benchmark will be proposed for LiDAR depth completion in order to remove the boundary effects mentioned above.

2) Vehicles in long range: Due to the extreme sparsity of low-resolution LiDAR depth map, the vehicles in long distances from the LiDAR are only visible in images. Thus for these vehicles, the depth map assisted depth prediction is practically downgraded as depth estimation from image only. This explains a sharp performance drop on hard cases in KITTI dataset.

V. CONCLUSION

In this paper, a 3D object detection framework is proposed for low-resolution LiDAR point cloud. By cascading a depth completion network prior to the object detector, it first converts the sparse point cloud into a much denser depth map that is subsequently processed for 3D object detection. It makes object detection possible from a sparse, low-cost LiDAR by fusing with images captured by a camera. When evaluated on KITTI dataset, the network can achieve comparable object detection accuracy in both easy and moderate cases as that of using high-resolution point cloud.

REFERENCES

[1] Y. Lyu, L. Bai and X. Huang, "Real-Time Road Segmentation Using LiDAR Data Processing on an FPGA," 2018 IEEE International Symposium on Circuits and Systems (ISCAS), Florence, 2018, pp. 1-5
[2] L. Caltagirone, S. Scheidegger, L. Svensson and M. Wahde, "Fast LiDAR-based road detection using fully convolutional neural networks," 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, 2017, pp. 1019-1024
[3] S. Pang, D. Morris, H. Radha, "CLOCs: Camera-LiDAR Object Candidates Fusion for 3D Object Detection," 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, USA, 2020
[4] Y. Cui, H. Xu, J. Wu, Y. Sun and J. Zhao, "Automatic Vehicle Tracking With Roadside LiDAR Data for the Connected-Vehicles System" in IEEE Intelligent Systems, vol. 34, no. 03, pp. 44-51, 2019
[5] L. Gigli et al, "Road Segmentation on low resolution Lidar point clouds for autonomous vehicles," arXiv preprint [arXiv:2005.13102] 2020
[6] A. Elhadidy, M. Afifi, M. Hassounah, Y. Ali and M. ElHelw, "Improved Semantic Segmentation of Low-Resolution 3D Point Clouds Using Supervised Domain Adaptation," 2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES), Giza, Egypt, 2020, pp. 588-593
[7] I. del Pino et al, "Low Resolution Lidar-Based Multi-Object Tracking for Driving Applications," 2017 Iberian Robotics Conference (ROBOT), 2017, pages 287–298
[8] F. Ma, G. V. Cavalheiro and S. Karuman, "Self-Supervised Sparse-to-Dense: Self-Supervised Depth Completion from LiDAR and Monocular Camera," 2019 International Conference on Robotics and Automation (ICRA), Montréal, QC, Canada, 2019, pp. 3288-3295
[9] B. Yang, W. Luo and R. Urtasun, "PIXOR: Real-time 3D Object Detection from Point Clouds," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, 2018, pp. 7652-7660
[10] R. Qian et al., "End-to-End Pseudo-LiDAR for Image-Based 3D Object Detection," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020, pp. 5880-5889
[11] F. Farahnakian and J. Heikkinen, "Fusing LiDAR and Color Imagery for Object Detection using Convolutional Neural Networks," 2020 IEEE 23rd International Conference on Information Fusion (FUSION), Rustenburg, South Africa, 2020, pp. 1-7
[12] M. Seikavandi, K. Nasrollahi, and T. B. Moeslund, "Deep car detection by fusing grayscale image and weighted upsampled LiDAR depth," Proc. SPIE 11605, Thirteenth International Conference on Machine Vision, 1160524, 2020
[13] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," arXiv preprint [arXiv:1804.02767] 2018
[14] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137-1149, 1 June 2017
[15] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu and A. C. Berg, "SSD: Single Shot MultiBox Detector," 2016 European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 2017, pp. 21-37
[16] J. Beltrán, C. Guindel, F. Moreno, D. Cruzado, F. García and A. De La Escalera, "BirdNet+: End-to-End 3D Object Detection in LiDAR Bird’s Eye View," 2020 23rd International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, 2018, pp. 3517-3523
[17] A. Barrera, C. Guindel, J. Beltrán and F. García, "BirdNet+: End-to-End 3D Object Detection in LiDAR Bird’s Eye View," 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), Rhodes, Greece, 2020, pp. 1-6
[18] L.C. Chen, Y. Zhu, G. Papandreou, F. Schroff and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," 2018 15th European Conference on Computer Vision (ECCV), Munich, Germany, 2018, pp. 801-818