Enabling 3-D Object Detection With a Low-Resolution LiDAR

Lin Bai, *Graduate Student Member, IEEE*, Yiming Zhao, *Member, IEEE*, and Xinming Huang, *Senior Member, IEEE*

***Abstract*—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 is an economically viable solution, but the point cloud sparsity makes it extremely challenging. In this letter, we propose a two-stage neural network framework that enables 3-D object detection using a low-resolution LiDAR. Taking input from a low-resolution LiDAR point cloud and a monocular camera image, a depth completion network is employed to produce dense point cloud that is subsequently processed by a voxel-based network for 3-D object detection. Evaluated with KITTI dataset for 3-D object detection in bird-eye view (BEV), the experimental result 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 3-D vehicle detection results are close to those using 64-line high-resolution LiDARs.**

***Index Terms*—3-D vehicle detection, camera, low-resolution light detection and ranging (LiDAR).**

I. Introduction

In RECENT years, much research has been focused on the autonomous driving technology. Light detection and ranging (LiDAR) is one of the most important sensors for perception tasks, such as drivable region segmentation, object detection, and vehicle tracking. Different from images captured by cameras, point cloud generated by LiDARs supplies 3-D spatial information of the objects in the form of (*x, y, z*) coordinates and intensity. This alleviates the barrier of distance estimation and makes 3-D object detection or tracking much more accurate. However, the price of high-resolution LiDARs is much higher than their low-resolution counterparts. The specifications of the most popular Velodyne 64-line LiDAR HDL-64E and 16-line LiDAR VLP-16 are compared in Table I. As we can see, the cost of a low-resolution LiDAR is only about 1/18 of the high-resolution ones. Therefore, it is more economical to consider 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. 1, we can barely find objects from the depth map captured from a 16-line LiDAR, while in the 64-line LiDAR objects are more visible.

TABLE I
COMPARISON OF VLP-16 AND HDL-64E LIDARS

LiDAR type	VLP-16	HDL-64E
Channel number	16	64
Res.(vertical / horizontal)	2°/ 0.2°	0.4°/ 0.1728°
Power (Watts)	8	60
Price (USD)	$4'000	$75'000

Manuscript received 27 January 2022; revised 18 April 2022; accepted 19 April 2022. Date of publication 26 April 2022; date of current version 24 November 2022. This work was supported in part by the U.S. National Science Foundation under Grant 2006738, and in part by The MathWorks. This manuscript was recommended for publication by P. R. Panda. *(Corresponding author: Lin Bai.)*

The authors are with the Department of Electrical and Computer Engineering, Worcester Polytechnic Institute, Worcester, MA 01609 USA (e-mail: lbai2@wpi.edu; yzhao7@wpi.edu; xhuang@wpi.edu).

Digital Object Identifier 10.1109/LES.2022.3170298

II. Related Works

A. Low-Resolution LiDAR and Depth Completion

Some research works focused on segmentation using low-resolution LiDARs. Gigli *et al.* [1] 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 close to that from high-resolution LiDAR within a reasonable degradation. A supervised domain adaptation was utilized by [3] to predict the low-resolution point cloud into high-resolution point cloud in spherical coordinate and further evaluated the results in 3-D semantic segmentation task. Low-resolution LiDARs had been also employed for object tracking tasks. In [4], an LiDAR-based system was proposed for estimation of actual positions and velocities of the detected vehicles. Some other works utilized depth completion for 2-D object detection, such as [5] and [6]. In [5], 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 [7] network for 2-D object detection. Similarly, Farahnakian and Heikkonen [6] introduced a self-supervised depth completion network to fill the high-resolution depth map before detection 2-D objects.

B. High Resolution LiDAR for BEV Object Detection

Nearly all state-of-the-art object detectors utilize high-resolution LiDAR. In [8], it first transformed the point cloud into bird-eye view (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. Barrera *et al.* [9] further refined the previous version into an end-to-end model and achieved better performance.

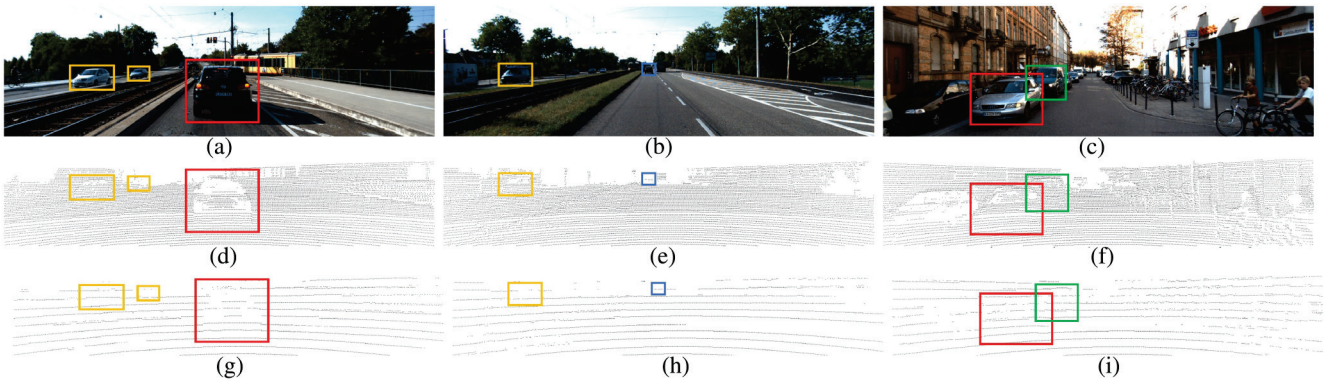

Fig. 1. Comparison of depth map from 16-line LiDAR (bottom) and 64-line LiDAR (middle) to their RGB image (top), on which red boxes represent the short-range vehicles, orange boxes show the medium range vehicles, and the far vehicles are marked by blue boxes. Green boxes illustrate the occluded vehicles. (a) Image scenario 1. (b) Image scenario 2. (c) Image scenario 3. (d) Dense depth map scenario 1. (e) Dense depth map scenario 2. (f) Dense depth map scenario 3. (g) Sparse depth map scenario 1. (h) Sparse depth map scenario 2. (i) Sparse depth map scenario 3.

Single-stage detector, PIXOR, was proposed in [10] by using 2-D 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 subsamples 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, 3-D object detection becomes possible.

III. Proposed 3-D Object Detection Framework

In this letter, we investigate the possibility of low-resolution LiDAR usage in BEV object detection task. In Fig. 1, red box, orange box, and blue box represent 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 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. 1(h) for example, 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. 1(c), (f), and (i)]. The long range vehicles in blue box [in both Fig. 1(e) and (h)] get too few points 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. After the dense depth map is generated, it is sent to 3-D object detector, as demonstrated in Fig. 2.

A. Depth Completion Network

The depth completion network aims to fill the sparse depth map from 16-line LiDAR point cloud with the help of RGB image. The state-of-the-art depth completion network [11] is adopted 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) [13] module are placed to bridge the encoder and decoder.

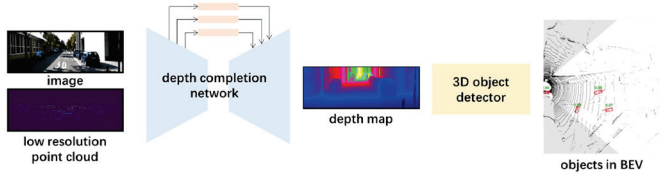

Fig. 2. Proposed framework for 3-D object detection using low-resolution point cloud and RGB image.

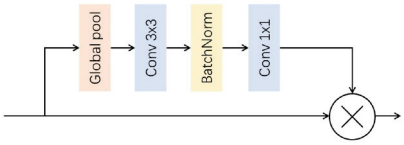

Fig. 3. Structure of global attention module.

As shown in Fig. 3, 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 is placed between encoder and decoder, with each convolution dilated rate 2, 4, 8, and 16. The ASPP module concatenates feature maps with different fields of perception, so that decoder has a better understanding of the context information.

The loss function of depth completion network is the mean square error (MSE) between the predicted depth map and the ground truth.

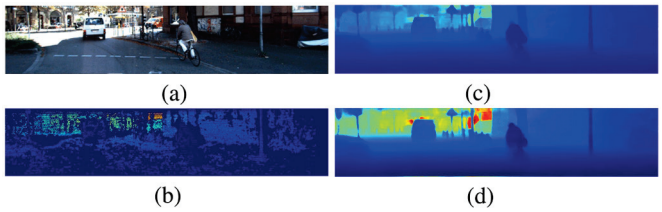

Fig. 4. Comparison of depth map from 16-line LiDAR with and without GAM module to their RGB image and ground truth. (a) Image. (b) Depth ground truth. (c) Depth map without GAM. (d) Depth map with GAM.

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

B. Object Detection Network

The object detection network used in this framework is PIXOR [10]. Its main idea is to take advantage of 2-D 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 third freedom (z or height) as another input feature map channel. So that 2-D convolution instead of 3-D convolution is necessary to greatly decrease the computation complexity. The second step is to feed the reformed input feature map into an anchor-free one-stage object detector network. 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 the loss function, the total loss of object detection consists of the classification loss and the regression loss (1), where λ_{cls} and λ_{reg} are the corresponding coefficients. The classification loss $\mathcal{L}_{cls}$ targets to correctly predict the object (cars in our case) and the regression loss $\mathcal{L}_{reg}$ aims to refine the size, center, and the orientation of the predicted bounding boxes

$$\mathcal{L}_{\text{detect}} = \lambda_{\text{cls}}\mathcal{L}_{\text{cls}} + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}}. \tag{1}$$

C. Implementation Details

The depth completion network is first 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 five epochs. The total number of training epoch is 10. After training the depth completion network and keeping as it 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 [12]. 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.

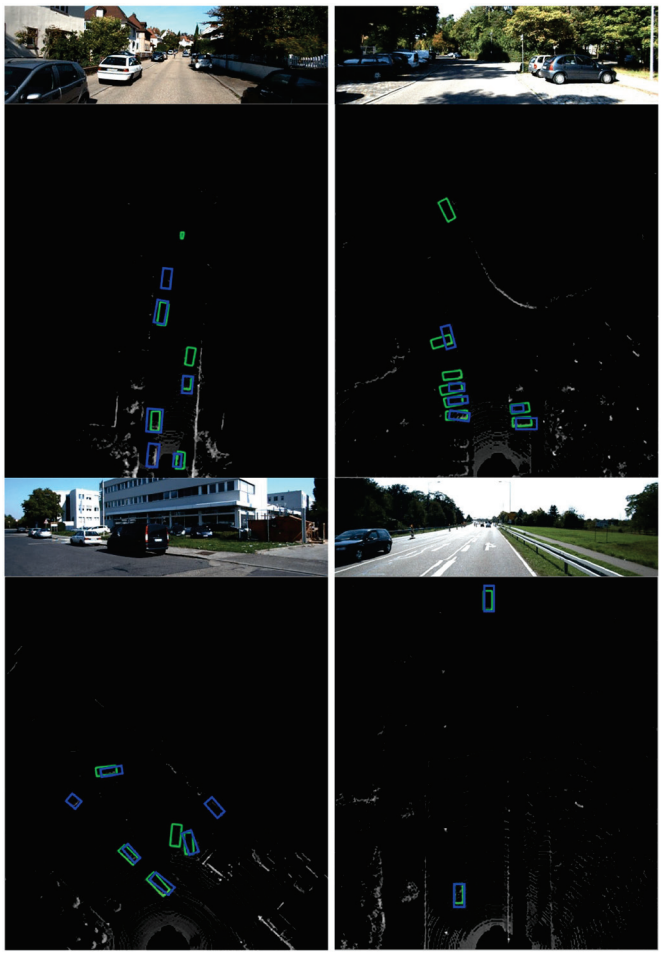

Fig. 5. Visualization of object detection from the proposed framework, where the green boxes are ground truth and the blue boxes represent the predicted results.

IV. EXPERIMENT RESULTS

A. Evaluation Dataset

Training and evaluation of the whole framework both employ KITTI dataset (both depth completion and object detection). Before feeding into the framework mentioned above, the point clouds are down-sampled to emulate the VLP-16 low-resolution LiDAR. KITTI depth completion dataset contains 85 898 training data and 1000 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 7481 training data and 7518 testing data. Evaluation is categorized into three regimes: 1) easy; 2) moderate; and 3) hard, representing objects at different occlusion and truncation levels.

B. Depth Completion Performance Evaluation

As described in Section 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 Table II. Adding GAM modules results in the performance improvement of about 3.6% and 7.0% measured by root mean square root (RMSE) and mean average error (MAE), respectively.

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 BEV

Detection Networks	Input	IoU=0.5			IoU=0.7		
		Easy	Moderate	Hard	Easy	Moderate	Hard
PIXOR[10]	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**

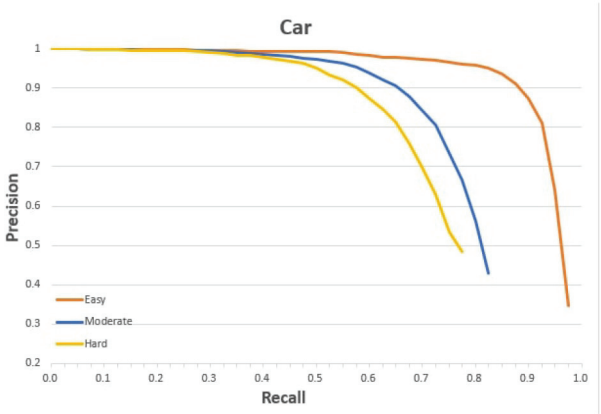

Fig. 6. Precision-Recall curve of the proposed framework on KITTI val dataset.

Fig. 4(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.

C. Object Detection Performance Evaluation

The performance of our framework on KITTI object detection validation dataset is illustrated in Table III and Fig. 5. The results are shown in two circumstances IoU = 0.5 and IoU = 0.7, respectively. When IoU = 0.5, our framework 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 to 75.4%, 61.2%, and 55.2%, respectively. Compared to feeding 16-line point cloud directly into PIXOR, our framework pulls up the detection accuracy significantly in all cases. If compared 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. 6.

In regard to the computations, when running on RTX 2080Ti, the inference time of proposed network is 25.4 ms or 39.8 frames per second (fps). Besides, as the network is aiming for embedded systems, we also tested it on DRIVE PX2 that contains two discrete Pascal GPUs. The inference latency for each point cloud frame is 581.7 ms. If two GPUs run as two threads, the throughput increases to 3.4 fps.

V. CONCLUSION

This letter presents a framework that enables 3-D object detection using a low-resolution LiDAR. By cascading a depth completion network with an object detector, it first converts the sparse point cloud into a denser depth map that is subsequently processed for 3-D object detection. It demonstrates 3-D object detection with only a 16-line LiDAR and a camera. When evaluated on KITTI dataset, the proposed solution achieves high accuracy in object detection for both easy and moderate cases, comparable to the benchmarks using 64-line LiDARs.

REFERENCES

[1] L. Gigli *et al.*, "Road segmentation on low resolution Lidar point clouds for autonomous vehicles," 2020, *arXiv:2005.13102*.
[2] L. Caltagirone, S. Scheidegger, L. Svensson, and M. Wahde, "Fast LIDAR-based road detection using fully convolutional neural networks," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Los Angeles, CA, USA, 2017, pp. 1019–1024.
[3] A. Elhadidy, M. Afifi, M. Hassoubah, Y. Ali, and M. ElHelw, "Improved semantic segmentation of low-resolution 3D point clouds using supervised domain adaptation," in *Proc. 2nd Novel Intell. Lead. Emerg. Sci. Conf. (NILES)*, Giza, Egypt, 2020, pp. 588–593.
[4] I. del Pino *et al.*, "Low resolution lidar-based multi-object tracking for driving applications," in *Proc. Iberian Robot. Conf. (ROBOT)*, 2017, pp. 287–298.
[5] M. Seikavandi, K. Nasrollahi, and T. B. Moeslund, "Deep car detection by fusing grayscale image and weighted upsampled LiDAR depth," in *Proc. SPIE 13th Int. Conf. Mach. Vis.*, 2020, Art. no. 1160524.
[6] F. Farahnakian and J. Heikkonen, "Fusing LiDAR and color imagery for object detection using convolutional neural networks," in *Proc. IEEE 23rd Int. Conf. Inf. Fusion (FUSION)*, Rustenburg, South Africa, 2020, pp. 1–7.
[7] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," 2018, *arXiv:1804.02767*.
[8] J. Beltrán, C. Guindel, F. M. Moreno, D. Cruzado, F. García, and A. De La Escalera, "BirdNet+: End-to-end 3D object detection in LiDAR bird's eye view," in *Proc. 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Maui, HI, USA, 2018, pp. 3517–3523.
[9] A. Barrera, C. Guindel, J. Beltrán, and F. Garcia, "BirdNet+: End-to-end 3D object detection in LiDAR bird's eye view," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Rhodes, Greece, 2020, pp. 1–6.
[10] B. Yang, W. Luo, and R. Urtasun, "PIXOR: Real-time 3D object detection from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, 2018, pp. 7652–7660.
[11] F. Ma, G. V. Cavalheiro, and S. Karaman, "Self-supervised sparse-to-dense: Self-supervised depth completion from LiDAR and monocular camera," in *Proc. Int. Conf. Robot. Automat. (ICRA)*, Montreal, QC, Canada, 2019, pp. 3288–3295.
[12] R. Qian *et al.*, "End-to-end pseudo-LiDAR for image-based 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, 2020, pp. 5880–5889.
[13] L.C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. 15th Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, 2018, pp. 801–818.
[14] H. Liu, H. Yuan, Q. Liu, J. Hou, and J. Liu, "A comprehensive study and comparison of core technologies for MPEG 3-D point cloud compression," *IEEE Trans. Broadcast.*, vol. 66, no. 3, pp. 701–717, Sep. 2020.