3D Object Detection using Artificially Generated LiDAR Data

Md Ishat-E-Rabban, Sumedh Koppula, Sparsh Bhogavilli, Venkata Sairam Polina
Maryland Robotics Center, University of Maryland College Park
{ier, sumedhrk, sbhogavi, sairamp}@umd.edu

1 Introduction

3D object detection is a crucial part of autonomous driving. Recent methods excel with high detection rates, but only if the 3D input data is derived from precise and expensive LiDAR sensors. Methods based on less expensive monocular or stereo image data have resulted in significantly lower accuracy. It is believed that the discrepancy is due to subpar models for image-based depth estimation. But, according to Wang et al. [1], the difference in performance is not caused by the quality of the data or ineffectiveness of image-based object detection models, but rather by the representation of the data. In [1], a new framework called *Pseudo-LiDAR* was proposed, which converts image-based depth maps to 3D point cloud representations in order to effectively simulate the LiDAR signal.

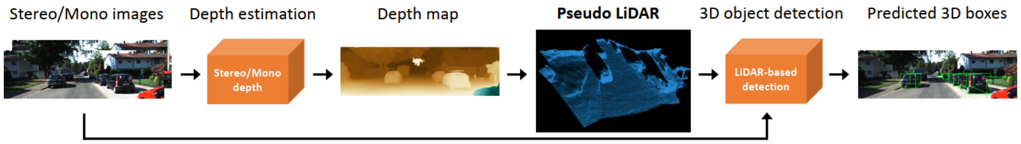

Figure 1: Basic Pseudo-LiDAR pipeline. Image source: [1]

The basic Pseudo-LiDAR pipeline is shown in Figure 1. It takes as input a pair of images and uses a stereo depth estimation network to compute the depth map. Then the depth of the pixels are back projected in the 3D space to create a point cloud, which is called Pseudo-LiDAR. Finally, a 3D object detection model is employed to identify the 3D objects in the Pseudo-LiDAR point cloud.

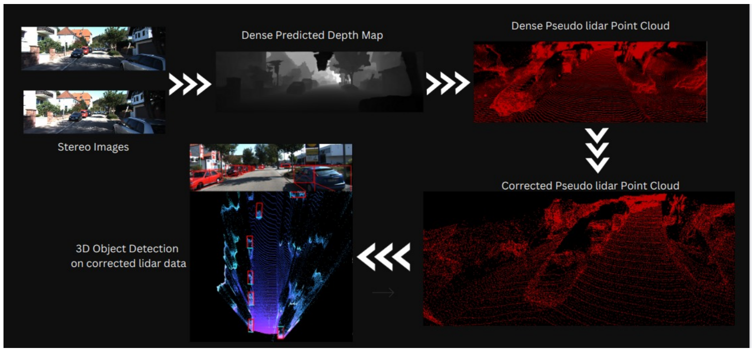
Dense Predicted Depth Map
Dense Pseudo lidar Point Cloud
Stereo Images
Corrected Pseudo lidar Point Cloud
3D Object Detection
on corrected lidar data

Figure 2: Pesudo-LiDAR++ pipeline with depth correction.

You et al. pointed out some limitations of Pseudo-LiDAR in [2]. For example, Pseudo-LiDAR incurs higher error in depth estimation of far-away objects compared to nearby ones. To improve the performance of Pseudo-LiDAR, You et al. [2] used cheap sparse LiDAR (4 beams) sensor data. This

CMSC 733: Computer Processing of Pictorial Information (Fall 2022), .

model, called *Pseudo-LiDAR++*, uses sparse LiDAR data in conjunction with stereo images. The data obtained from sparse LiDAR sensors acts as ground truths for depth estimation and is used to correct the errors of the dense point cloud generated from the stereo images. The process of rectifying the depth estimation using sparse LiDAR data is called depth correction. The Pseudo-LiDAR++ pipeline is shown in Figure 2.

A key characteristics of both Pseudo-LiDAR and Pseudo-LiDAR++ is that any existing stereo depth estimation model and object detection model can be incorporated into the pipeline. For example, Pseudo-LiDAR [1] used the state-of-the-art depth estimator, PSM-Net [3], and the then best performing object detectors, AVOD [4] and Frustum point-net [5], for depth estimation and object detection, respectively.

1.1 Project Tasks

In this project, we have explored the Pseudo-LiDAR and Pseudo-LiDAR++ models. We list the tasks we have completed as part of this project below.

- We have set up the environment of Pseudo-LiDAR++ [2] in our PC, trained and tested the Pseudo-LiDAR++ model using the publicly available code provided by the authors.
- We have modified the depth estimation network by adding new 3D convolutional layers.
- We have implemented the depth correction module of the Pseudo-LiDAR model by ourselves.
- We have successfully integrated a new 3D object detector with Pseudo-LiDAR++ , called SFA3D [6], which has not been previously used in any Pseudo-LiDAR variant.
- We have tested the our model on the KITTI dataset and visualized the test cases which shows a lower number of false positives.

2 Related Works

2.1 Image-based 3D Object Detection

A key component of any image-based 3D object detection methods is an effective depth estimation approach. Image-based 3D object detection methods take as input image(s) and use monocular (for single image) and stereo (for multiple images) depth estimation methods to compute a depth map of the scene. Examples of efficient monocular depth estimation methods include Deep Ordinal Regression Network (DORN) [7] and the work of Godard et al. [8] and that of stereo vision methods include Pyramid Stereo Network (PSM-Net) [3] and the work of Mayer et al. [9].

The accuracy of image-based object detection systems has increased dramatically over the past few years. DORN [7] combines multi-scale features with ordinal regression to predict pixel depth with remarkably low discrepancies. PSM-Net [3] applies Siamese networks for disparity estimation, followed by 3D convolutions for refinement, resulting in extremely low outlier rates. However, although these image-based object detection methods have made significant progress, the performance of image-based 3D object detection lags behind point cloud or LiDAR-based methods as discussed in the next section.

2.2 Point Cloud or LiDAR-based 3D Object Detection

Recent methods leverage point cloud or LiDAR data to achieve better performance than image-based methods in 3D object detection task. PointNet [10] provides a deep neural network for classification and segmentation of 3D point sets, which can be used in each frustum as a 2D object detection framework. VoxelNet [11] converts 3D points into voxels and uses 3D convolutions to extract features. Recent works on object detection includes Bird-net [12], which uses Birds Eye View (BEV) of the point cloud to detect objects. MV3D [13] projects LiDAR points into both bird-eye view (BEV) and frontal view to obtain multi-view features. Lang et al. proposes a fast encoding-decoding based method called PointPillars [14] to accelerate object detection.

Works on point cloud or LiDAR-based 3D object detection can be broadly categorized into two classes. The first class of methods directly operate on the unordered point clouds in 3D by mostly applying some variant of PointNet [10] or applying 3D convolution over the neighbors [11]. The second class operate on quantized 3D/4D tensor data, which are generated by discretizing the location of the 3D points into some fixed grid [13].

2.3 Pseudo-LiDAR

Point cloud based 3D object detection methods outperform the image-based methods, but it is more expensive to capture the 3D point cloud instead of 2D images as LiDARs are more expensive than cameras. To this end, the Pseudo-LiDAR framework [1, 2, 15] has been recently proposed by Wang et al. Like image-based 3D object detection models, Pseudo-LiDAR first uses an image-based depth estimation model to obtain predicted depth of each image pixel. The resulting depths are then projected back to the 3D space to create a point cloud which is called *Pseudo-LiDAR*. The Pseudo-LiDAR points are then treated as a point cloud, over which any LiDAR-based 3D object detector can be applied. The depth estimation network and the object detection network can be trained separately to make use of the state-of-the-art models for both tasks.

The Pseudo-LiDAR framework was first proposed in [1], where the authors used PSM-net [3] as the depth estimation network and AVOD [4] and Frustum Point-net [5] as the object detector. In the next version of the pseudo-LiDAR framework, which is called *Pseudo-LiDAR++* [2], the authors pointed out some limitations of the original Pseudo-LiDAR paper and proposed some modifications to overcome those limitations. In this version, they used a custom depth estimation module which was an extension to PSM-net. They also introduced cheap sparse LiDAR sensors and formulated a graph diffusion based solution to fuse image and sparse LiDAR data. In the next version of the framework [15], the authors showed how to train the network in an end to end manner, as opposed to separate training for depth estimation and object detection modules. The evaluation of all three Pseudo-LiDAR models were based on the KITTI dataset [16].

3 Methods

The Pseudo-LiDAR pipeline we implement in this project closely follows Pseudo-LiDAR++ [2]. Our pipeline contains three major modules as listed below:

- **Depth Estimation**: For stereo depth estimation, we use Stereo Depth Network (SDN) which is also used in Pseudo-LiDAR++. SDN is an extension of Pyramid Stereo Network [3] (PSM-Net) which is fine tuned to render better depth disparity. Our depth estimation module is described in Section 3.1.
- **Depth Correction**: The estimated depth map is projected into the 3D space to create a point cloud. Following Pseudo-LiDAR++, we use an additional step to rectify the point cloud generated so as to minimize the depth error in faraway objects. This step is called depth correction. We implement the depth correction module by ourselves similar to the one used in Pseudo-LiDAR++. Our depth correction module is described in Section 3.2.
- **Object Detection**: Unlike Pseudo-LiDAR++ or any other existing Pseudo-LiDAR variant, we use a new 3D object detection network. It is called *Super Fast and Accurate 3D Object Detection* or SFA3D [6]. SFA3D is a variant of Feature Pyramid Network [17] (FPN) which is better suited for detecting keypoints of objects. Our object detection module is described in Section 3.3.

3.1 Depth Estimation: SDN

The depth estimation module takes as input a pair of left-right images I_l and I_r captured using two cameras with a horizontal offset, and applies a stereo disparity estimation algorithm to produce a disparity map Y of the same size as either of the two input images. We assume that the depth estimation algorithm uses the left image, I_l, as a reference and logs in Y the horizontal disparity to I_r for each pixel. Thus, we can determine the 3D location of each pixel in the left camera's coordinate. We create a 3D point cloud with the coordinates of all the pixels back-projected into 3D space. The 3D coordinate frame is created from such a point cloud by applying a reference vantage point and viewing angle.

Now we describe the depth estimation method used for producing the disparity map. We use Stereo Depth Network (SDN) introduced in Pseudo-LiDAR++ to produce the disparity map. SDN is an upgraded version of Pyramid Stereo Network (PSM-Net) [3]. First we describe how PSM-Net works (Section 3.1.1). Then we describe how PSM-Net is modified to formulate SDN (Section 3.1.2).

3.1.1 PSM-Net

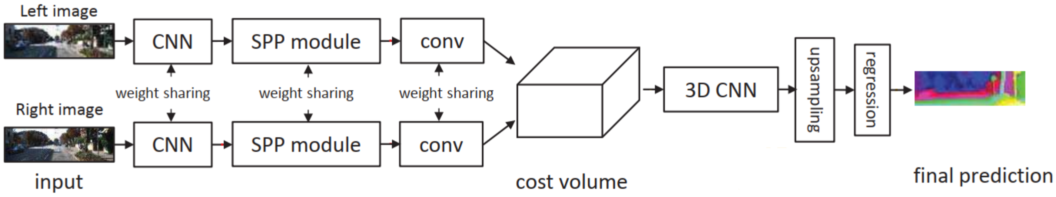

Figure 3: Depth Estimation using Pyramid Stereo Network (PSM-Net). Image source: [3]

PSM-Net is a pyramid stereo matching network (PSMNet) that exploits global context. It enlarges the receptive field by by applying spatial pyramid pooling and dilated convolution which helps to extend pixel-level features to region-level features with different scales of receptive fields. Thus global and local feature clues are combined to form the cost volume for reliable disparity estimation. PSM-Net also uses a stacked hourglass 3D CNN with intermediate supervision which further improves the utilization of global context information. The pipeline of PSM-Net is shown in Figure 3.

3.1.2 SDN

Two changes are made to PSM-Net to facilitate depth estimation. The resultant network is called SDN. First, the loss function is modified to directly optimize the depth loss. Pseudo-LiDAR puts a strong emphasis on tiny depth errors of nearby objects. The change from the disparity loss to the depth loss corrects this disproportionally strong emphasis. The second change is made to make sure that all neighborhoods are operated upon in an identical manner. To ensure this, instead of a disparity map, the depth cost volume is constructed, which encodes features describing the probability of the depth map. As a result, the 3D convolutions operate on the grid of depth, instead of disparity, which ensures that depths throughout the image are treated identically, independent of their location.

3.2 Depth Correction

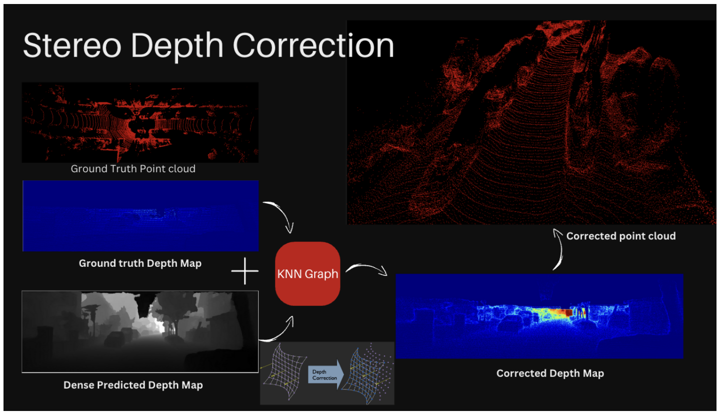

Figure 4: Depth Correction pipeline

The discrete nature of pixels incurs a fundamental limitation in stereo depth estimation methods, because the disparity needs to be quantized at the level of individual pixels although the depth is continuous. One way to reduce the quantization error is to use higher resolution images, but this is a computationally expensive. To this end, Pseudo-LiDAR++ employs a hybrid approach to correct this bias. A cheap LiDAR is used to obtain extremely sparse (e.g., 4 beams) but accurate depth measurements. These sensor measurements are too sparse to capture object shapes and cannot be used alone for object detection. But, by projecting the LiDAR points into the image plane, we can obtain exact depths of a small portion of pixels, which are called *landmark* pixels. The depth correction module uses depths of these landmark pixels to correct the depth estimation of the dense pseudo-LiDAR points.

Depth correction is performed using a graph-based depth correction (GDC) algorithm that combines the dense stereo depths obtained from the left-right images and the sparse accurate LiDAR measurements. In the GDC algorithm, first landmark pixels are matched with pseudo-LiDAR points using kd-trees [18], which makes sure those points possess the exact depths. Next, object shapes captured by forming a weighted k-nearest neighbor (KNN) graph. Finally the depths of the landmark pixels are diffused throughout the whole pseudo-LiDAR point cloud to construct the corrected depth estimation. A schematic diagram of depth correction is provided in Figure 4. A more detailed description of the depth correction method can be found in Section 4 of Pseudo-LiDAR++ [2].

3.3 Object Detection: SFA3D

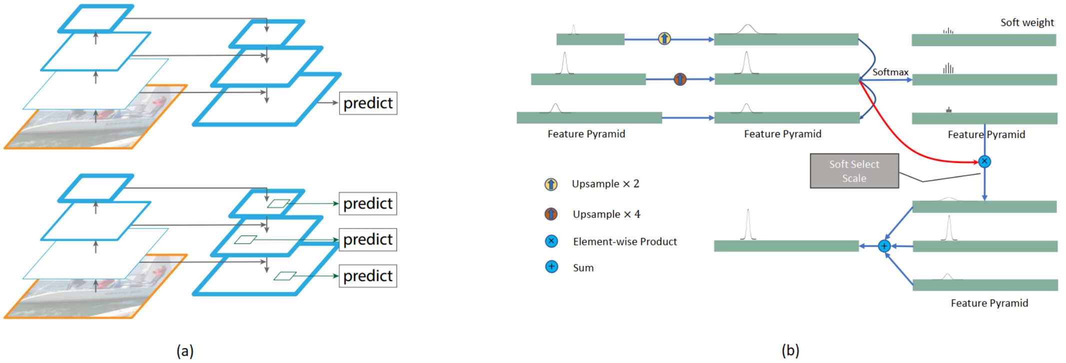

Figure 5: (a) FPN Model. Image source: [17]. (b) KFPN Model. Image source: [19].

In this project, we use a new 3D object detection network which has not been used in any previous Pseudo-LiDAR variants. This network is called SFA3D [6] which stands for Super Fast and Accurate 3D Object Detection. The SFA3D model is described below.

SFA3D takes as input the depth corrected point cloud and passes the Birds Eye View (BEV) image of the point cloud to be processed by a Keypoint Feature Pyramid Network [19] (KFPN). The KFPN is based on Feature Pyramid Networks [17] (FPN). FPN leverages the pyramidal shape of a convolutional block's feature hierarchy and creates a feature pyramid that has strong semantic cues at all scales. To this end, the architecture of FPN combines low-resolution, semantically strong features with high-resolution, semantically weak features using a top-down pathway and lateral connections. The architecture of FPN is shown in Figure 5(a). The central concept of a KFPN is a keypoint, which appears irrespective of the scale. In KFPN, the features at the lowest scale of the pyramid are resized to the largest scale and are linearly weighted using softmax. The architecture of KFPN is shown in Figure 5(b).

On top of KFPN, SFA3D employs a custom loss function and learning rate scheduling techniques to perform object detection. The SFA3D module outputs the classes of the objects in the image, the distance of the center of the objects from the ego vehicle, and the angle, dimensions, and the z coordinate of the objects.

4 Results

4.1 Dataset

We use the KITTI benchmark dataset [16] to train and test the Pseudo-LiDAR framework. The KITTI dataset contains left-right color images captured from a vehicle which can be used to train and test the stereo depth estimation models. The dataset also contains the dense 3D point cloud captured using LiDAR sensors which can be used for training and testing 3D object detection models. The KITTI dataset is used in all the Pseudo-LiDAR variants [1, 2, 15] to evaluate model performance.

4.2 Platform

The experiments are conducted using 11th Gen Intel® Core™ i7-11700 @ 2.50GHz 16 core CPU and Nvidia GeForce RTX 3060 12GB of memory, respectively.

4.3 Outputs

In this section, we report the output of our custom Pseudo-LiDAR pipeline. In our pipeline, the depth estimation model is SDN [2] and the object detection model is SFA3d [6]. We report the output of our model both with and without depth correction. We present three qualitative examples in Figure 6, 7, and 8. The point clouds with and without depth correction are shown in the right and left columns, respectively, with the ground truth in the middle. A visual inspection of the outputs reveals that depth correction leads to better performance as it reduces the number of false positives.

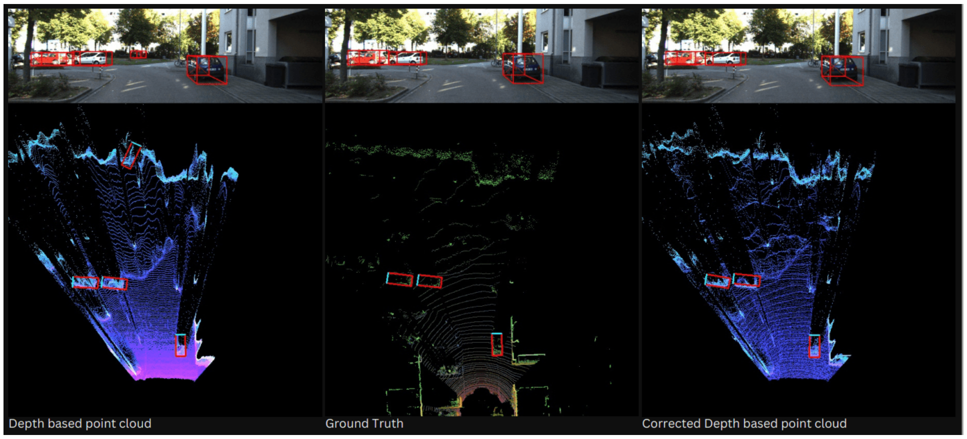

Figure 6: Test set sample 1

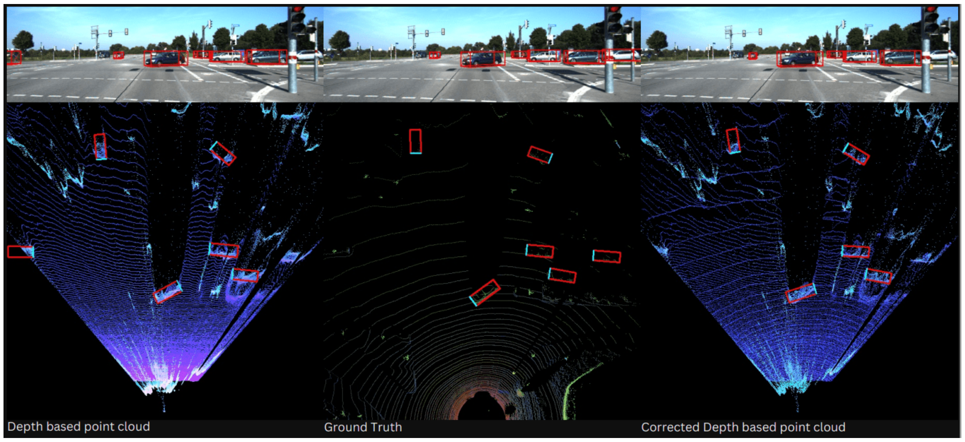

Figure 7: Test set sample 2

5 Conclusion

In this project, we have explored several Pseudo-LiDAR variants [1, 2, 15]. We have redesigned the depth estimator by adding 3D convolutional layers in the stacked hourglass model. We have

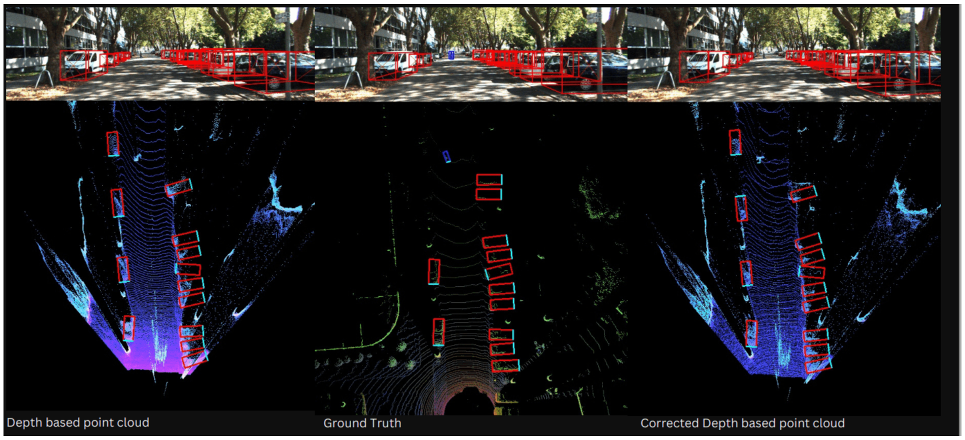

Figure 8: Test set sample 3

incorporated a new object detector, SFA3D, in the Pseudo-LiDAR++ pipeline. We have examined how the performance of the basic Pseudo-LiDAR framework [1] improves with the introduction of depth correction as demonstrated in Pseudo-LiDAR++ [2]. We have implemented the depth correction algorithm ourselves, tested our model on the KITTI dataset, and visualized the outputs. In future, we intend to study the end-to-end Pseudo-LiDAR variant [15] which trains the network as a whole instead of training the depth estimator and object detector separately.

References

[1] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger, "Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8445–8453.

[2] Y. You, Y. Wang, W.-L. Chao, D. Garg, G. Pleiss, B. Hariharan, M. Campbell, and K. Q. Weinberger, "Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving," *arXiv preprint arXiv:1906.06310*, 2019.

[3] J.-R. Chang and Y.-S. Chen, "Pyramid stereo matching network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5410–5418.

[4] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, "Joint 3d proposal generation and object detection from view aggregation," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–8.

[5] 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, pp. 918–927.

[6] N. M. Dung, "Super-Fast-Accurate-3D-Object-Detection-PyTorch," https://github.com/maudzung/Super-Fast-Accurate-3D-Object-Detection, 2020.

[7] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2002–2011.

[8] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 270–279.

[9] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4040–4048.

[10] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[11] 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.

[12] J. Beltrán, C. Guindel, F. M. Moreno, D. Cruzado, F. Garcia, and A. De La Escalera, "Birdnet: a 3d object detection framework from lidar information," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 3517–3523.

[13] 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, pp. 1907–1915.

[14] 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/CVF conference on computer vision and pattern recognition*, 2019, pp. 12 697–12 705.

[15] R. Qian, D. Garg, Y. Wang, Y. You, S. Belongie, B. Hariharan, M. Campbell, K. Q. Weinberger, and W.-L. Chao, "End-to-end pseudo-lidar for image-based 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5881–5890.

[16] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3354–3361.

[17] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[18] M. Shevtsov, A. Soupikov, and A. Kapustin, "Highly parallel fast kd-tree construction for interactive ray tracing of dynamic scenes," in *Computer Graphics Forum*, vol. 26, no. 3. Wiley Online Library, 2007, pp. 395–404.

[19] P. Li, H. Zhao, P. Liu, and F. Cao, "Rtm3d: Real-time monocular 3d detection from object keypoints for autonomous driving," in *European Conference on Computer Vision*. Springer, 2020, pp. 644–660.