Multi-modal 3D Object Detection by 2D-guided Precision Anchor Proposal and Multi-layer Fusion

Yi Wu[a], Xiaoyan Jiang[a,*], Zhijun Fang[a], Yongbin Gao[a], Hamido Fujita[b]

[a]*School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, No.333 of Longteng Road, Shanghai, China*
[b]*Faculty of Sofeware and Information Science, Iwate Prefectural University (IPU), Iwate 020-0693, Japan*

Abstract

3D object detection, of which the goal is to obtain the 3D spatial structure information of the object, is a challenging topic in many visual perception systems, e.g., autonomous driving, augmented reality, and robot navigation. Most existing region proposal network (RPN) based 3D object detection methods generate anchors in the whole 3D searching space without using semantic information, which leads to the problem of inappropriate anchor size generation. To tackle the issue, we propose a 2D-guided precision anchor generation network (PAG-Net). Specifically speaking, we utilize a mature 2D detector to get 2D bounding boxes and category labels of objects as prior information. Then the 2D bounding boxes are projected into 3D frustum space for more precise and category-adaptive 3D anchors. Furthermore, current feature combination methods are early fusion, late fusion, and deep fusion, which only fuse features from high convolutional layers and ignore the data missing problem of point clouds. To obtain more efficient fusion of RGB images and point clouds features, we propose a multi-layer fusion model, which conducts nonlinear and iterative combinations of features from multiple convolutional layers and merges the global and local features effectively. We encode point cloud with the bird's eye view (BEV) representation to solve the irregularity of point cloud. Ex-

*Corresponding author
Email address: xiaoyan.jiang@sues.edu.cn (Xiaoyan Jiang)

perimental results show that our proposed approach improves the baseline by a large margin and outperforms most of the state-of-the-art methods on the KITTI object detection benchmark.

Keywords: 3D object detection, Multi-modal, Autonomous driving, Feature fusion, Point cloud

1. Introduction

The target of 3D object detection is to classify the category and estimate oriented 3D bounding boxes of physical objects from sensor data. It plays an important role in many applications, of which autonomous driving is the typical scenario of adopting the 3D object detection technique. For safe motion planning for self-driving cars, the surrounding environment should be recognized in a 3D manner providing spatial location, size, and orientation of the objects. By deploying sensors, i.e., LIDAR, RGB-D, and stereo cameras, on autonomous vehicles and mobile devices, a large amount of raw 2.5D/3D data can be obtained easily. Afterwards, deep learning methods are the commonly used technique to deal with these data for accurate 3D object detection[1, 2, 3, 4, 5]. Compared with 2D object detection, 3D object detection also needs to estimate the oriented 3D regression boxes from unstructured or heterogeneous data, which makes it more difficult.

In autonomous driving scenarios, single-modal-based 3D object detection is challenging due to: 1) *objects with close location.* As shown in Fig 1(a), bicycles are spatially together and their corresponding point clouds are also closely clustered, which makes it is difficult to detect each bicycle accurately. 2) *structural ambiguity of point clouds.* There are a lot of different objects with similar geometric structures. As shown in Fig 1(b), pedestrians and road columns are clearly visible in the image, yet have similar structures in the LIDAR modality. 3) *occlusion and truncation between objects.* As shown in Fig 1(c), in the left region of the image, the pedestrian is occluded by another person. Since obscured 3D objects show incomplete appearance, they are difficult to be detected

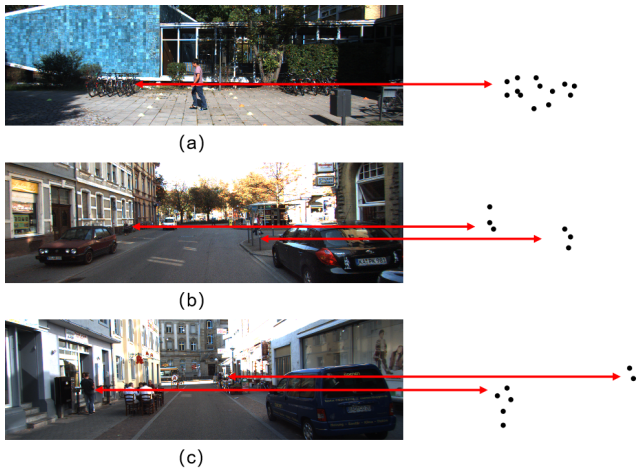

Figure 1: Challenges of single-modal-based 3D object detection: (a) objects with close location, (b) structural ambiguity of point clouds, and (c) occlusion and truncation between objects. The black points are the corresponding sparse point clouds clustered of objects. Images are taken from the KITTI dataset [6].

by only using raw point clouds or RGB images.

How to represent point clouds and what deep network architectures to adopt for efficient 3D object detection remain open problems. Most existing works transform point cloud data to multiple planes by projection or to regular volumetric voxel grids by quantization, then apply convolutional networks. However, data representation transformation in this way not only increases computational complexity, but also loses important original 3D patterns of the data. Recently, several new deep neural network architectures have been proposed to process point clouds directly [7, 8, 9], i.e., pointnet and pointnet++. However, their computational complexity is too expensive. Since autonomous driving has high precision and real-time requirement on object spatial estimation, modern self-driving cars are commonly equipped with multiple sensors to provide rich data for perceiving the surroundings [10]. LIDAR point clouds have the advantage of providing accurate depth information, while RGB images preserve much more detailed semantic information [11]. Therefore, the multi-modal fusion of LIDAR point clouds and RGB images should be able to improve the

performance and ensure safety of self-driving cars. Currently, more and more researches focus on multi-modal fusion for 3D object detection [12, 13, 14].

Current multi-modal-based 3D object detection methods have two crucial problems: inappropriate anchor sizes and fusion without considering interaction from low-layer features. Firstly, similar to 2D object detectors, most exiting networks for 3D object detection rely on a 3D region proposal generation step to reduce 3D searching space. However, missed instances at the proposal generation stage have negative impact on the detection stages afterwards. In particular, small and obscured objects are easily missed. Therefore, obtaining precise anchor proposals is crucial for the final detection performance. Currently, the anchor size of most 3D detectors is determined by the clustering results of the trained samples for each class [11, 15]. Since these anchors are proposed in the whole 3D searching space without considering semantic information, 3D object detectors have poor performance in proposing appropriate anchor size that leads to poor performance in the final detection. Secondly, current multi-modal-based 3D object detectors learn multi-view representation and extract these features, then fuse features from higher layers [11, 16]. Typical feature combination methods adopted by prior works are early fusion, late fusion, and deep fusion. Early fusion [14] directly combines features from multi-views in the input stage, while late fusion [17] uses separate subnetworks to learn feature transformation independently and combines their outputs in the prediction stage. Deep fusion [11] fuses multi-view features hierarchically. However, the above methods ignore the interaction of different convolutional layers and lose useful information from lower convolutional layers.

In this paper, we propose an effective multi-modal-based 3D object detection approach. To generate accurate 3D anchor sizes, we propose a 2D-guided PAG-Net to encode the 2D location and category information of objects provided by reliable 2D detectors. When considering sparsity and irregularity of point cloud, we project LIDAR point cloud to BEV for using 2D Convolutional Neural Network (2D CNN) to extract features. In order to obtain robust and discriminative fused features for 3D RPN, we propose a multi-layer fusion model based

on deep fusion to fuse RGB image with BEV map feature. Nonlinear combination of multi-layer features and iterative operation are adopted to ensure the discriminability of object features. The proposed approach jointly optimizes 3D detection results by the 2D-guided PAG-Net and the multi-layer fusion model to improve the object classification and 3D pose regression performance. In particular, we update the encoding method of 3D bounding box by encoding with four corners and one height to reduce the regression redundancy with physical constraints and semantic information. Main contributions are summarized as follows:

- We propose the PAG-Net which is jointly trained with 3D RPN to generate accurate and category-adaptive anchors. The PAG-Net utilizes prior information provided by the 2D detector to generate anchors that well fit different objects, and further significantly improves the detection performance of small objects.
- We propose a novel multi-layer feature fusion approach that executes nonlinear combination of features from multiple convolutional layers iteratively to merge both global and local features of RGB images and point clouds.
- Experimental evaluation validates the state-of-the-art performance of the proposed approach on the challenging KITTI dataset [6].

The remaining of the paper is organized as follows. We first review the related work on 3D object detection in Section 2. In Section 3, we present our approach, including the network architecture, details of PAG Net, multi-layer fusion model, and the proposed 3D bounding box encoding. Experimental results and analysis are shown in Section 4. Finally, Section 5 summarizes our work.

2. Related work

Objects in the 3D world generally do not follow any particular orientation. Box-based detectors have difficulties in enumerating all orientations or fitting an axis-aligned bounding box to rotated objects [18]. To overcome these difficulties, there exist the following techniques: image-based, LIDAR-based, and multi-modal-based 3D object detection.

Image-based 3D object detection. Current image-based 3D object detection can be divided into monocular-based and stereo-based depending on the input data. Authors in [1, 19, 2, 20, 21, 22] focus on 2D object proposal generation of monocular images using 2D object detectors. Then 3D bounding boxes are estimated by the geometry relations between 2D box edges and 3D box corners. Ma et. al. [23] generate depth map using monocular depth estimation, then estimate 3D bounding box by concatenating RGB images and depth maps.

Some works [24, 3, 25] utilize stereo vision, which provides a fixed parallax of scenes for depth estimation. However, monocular and stereo data cannot provide raw depth information. As we know, precision depth information is particularly important for 3D object detection. The performance of image-based 3D detection approaches is bounded by the accuracy of the depth estimation. Unfortunately, the inferred depth map from the image-based methods cannot guarantee the accuracy. Our work shows how to incorporate LIDAR and image data to improve 3D object detection.

LIDAR-based 3D object detection. The input of standard convolutional operations is required to be regular, while the point cloud data is an unstructured collection of points in a 3D space. Hence, a large number of studies pay attention to representation learning on point clouds. Most existing works transform point cloud data to regular 3D voxel grid representation [26, 27, 28, 29, 30] before feature extracting . Sliding Shapes [31] and Vote3D [32] apply SVM classifiers on 3D grids encoded with geometry features. [28] encodes each nonempty voxel with 6 statistical quantities that are derived from

all the points contained within the voxel. VoxelNet [27] designs a novel voxel feature encoding (VFE) layer, which enables inter-point interaction within a voxel, by combining point-wise features with a locally aggregated feature. Some recently proposed methods [33, 4, 34, 35] improve feature representation with 3D Convolutional Neural Network (3D CNN), but require expensive computations. In addition to the 3D voxel representation, some works transform point cloud to collections of images [11, 15, 16, 36].

Recently, few works [7, 8] design a new type of neural network architecture that directly consumes LIDAR point clouds without converting them to other formats. PointNet [7] provides a unified architecture to learn point-wise feature. However, PointNet does not capture local feature, which limits it to complex scenes. To solve this problem, PointNet++ [8] introduces a hierarchical neural network using PointNet on a nested partition to learn local features with increasing contextual scale.

There are also several studies that combine the above methods of processing point clouds. [28] combines both voxel-based CNN and point-based shared multilayer perceptron (SharedMLP) for efficient point cloud feature learning. In comparison, PV-RCNN [29] takes advantage of both the voxel-based and point-based feature learning to enable both high-quality 3D proposal generation and flexible receptive fields. Some studies propose lightweight or low-precision network [37, 32, 4, 38, 13, 39, 40, 41]. 3DSSD [42] firstly presents a lightweight and effective point-cloud-based 3D single stage object detector. However, these methods lag behind in complex computation for converting point cloud to other formats or directly consuming them.

Multi-modal-based 3D object detection. A large number of researches focus on multi-modal-based 3D object detection [16, 12, 13]. Multi-modal-based 3D object detection is a non-trivial task, because it relies on the data derived from different sources. As we know, RGB image provides dense appearance and texture information, while LIDAR point cloud provides accurate depth information. Multi-modal-based methods can improve the accuracy of detection compared to LIDAR-based 3D detection, particularly for small objects. There-

fore, we leverage multi-modal feature fusion methods that combine RGB image and LIDAR point cloud to enhance the information of objects and to reduce the 3D search space.

3D object proposal: 3D object proposal methods [24, 15, 25] generate a small set of 3D candidate boxes in order to cover most of the objects in 3D space. 3DOP [24] exploits stereo imagery to place proposals in the form of 3D bounding boxes. Mono3D [1] exploits the ground plane prior and utilizes some segmentation features to generate 3D proposals from a single image. Both 3DOP and Mono3D use hand-crated features. 3D RPNs have previously been proposed in [17] for 3D object detection from RGB-D images. MV3D [11] extends the 2D RPN of Faster R-CNN [43] to 3D by corresponding every pixel in the BEV feature map to multiple prior 3D anchors. These anchors are crucial for final detection performance, then are fed to the RPN to generate 3D proposals. However, this RPN architecture does not work well for small object instances in BEV. To this end, an Aggregate View Object Detection architecture for autonomous driving (AVOD) [15] proposes an RPN with a novel architecture which aims to fuse full resolution feature crops from RGB image and BEV feature maps, allowing the generation of high recall proposals for small objects. However, this anchor size of AVOD is determined by the clustering results of the trained samples for each class, which may not be suitable for complicated scenarios. In contrast, we propose a more precise and effective solution with PAG-Net.

Multi-modal fusion: Normally, distribution and density of different modal features are different because feature representations are distinct. How to efficiently fuse features from unstructured or heterogeneous data is a core challenge in multi-modal-based 3D object detection. Qin et.al. [44] explore the fusion of RGB, depth and optical flow using a mixture-of-experts framework for 2D pedestrian classification. PointPainting [45] fuses LIDAR with RGB images by projecting image-based semantic segmentation results [46] to point cloud and appending the class scores to each point. To combine multi-view features from different data, prior works usually use early fusion [14], late fusion [17], and

deep fusion [11], which ignore the interaction of features from low and high layers. Unlike these methods, we propose a multi-layer fusion that lavages the combination of low and high layer features to fuse both global and local features of multiple views.

3. The proposed approach

As shown in Fig 2, we propose a deep learning framework to deal with multi-modal-based 3D object detection by 2D-guided PAG-Net and multi-layer fusion. Inspired by [27, 15, 45], we take BEV maps projected by point clouds and RGB images as input. The outputs of the model are oriented 3D bounding box and classification of the object.

The proposed framework is mainly composed of the following models: 1) *feature and prior information extraction network.* Feature Pyramid Network (FPN) is used to extract features from both RGB images and BEV maps. Meanwhile, a pretrained Faster RCNN [43] is used to extract 2D objects along with category labels in RGB images. 2) *PAG-Net.* With this prior information and a known camera projection matrix, we get the 3D frustum region and the category of the object. Using the object class and the corresponding object 3D frustum region as prior knowledge, we generate precise and category-adaptive anchors (section 3.1). 3) *multi-view feature fusion network (section 3.2).* We apply a 1×1 convolutional kernel on the FPN output feature maps of RGB image and BEV map. Afterwards, we use the crop and resize operation to extract feature crops. Finally, we merge these feature crops with multi-layer fusion. 4) *final detection and pose estimation network.* We estimate the oriented bounding boxes from the 3D proposals and categorize the classification label for selected regions.

3.1. Precision anchor generation network

Inspired by F-PointNet [9] and AVOD [15], we propose a 2D-guied PAG-Net. Category labels and 2D bounding boxes of objects estimated by mature 2D object detectors are used as prior information for precision and class-adaptive 3D

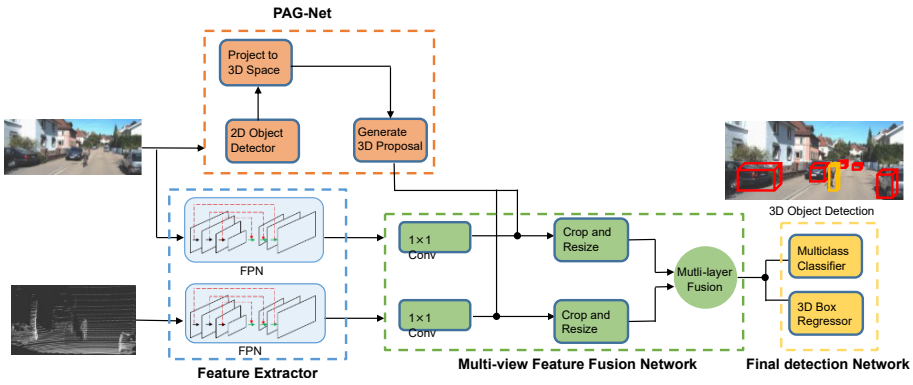

Figure 2: The proposed network architecture. The feature and prior information extraction network is shown in blue, PAG-Net in orange, multi-view feature fusion network in green, and the final detection and pose estimation network in yellow.

anchors generation.

Precision anchor generation. In the AVOD model, the dimensions of the anchors are determined by clustering the training samples for each class. Anchors in RPN are selected to obtain region of interests (RoIs), which are used by the detection and pose estimation network to detect and locate objects. Specifically speaking, the accuracy of anchor selection has heavy impact on the final detection. An appropriate prior information such as location and class label of the objects can improve the performance of the model. In multi-stage object detectors [47], the anchors' sizes are considered as the prior information about the size of the objects to be detected [48].

Current 2D object detectors have high detection accuracy in many scenes. With the proposal of FPN [49], small objects can also be well detected. Given an RGB image, we firstly use a pretrained Faster RCNN [43] to generate 2D bounding boxes and class labels for objects. We use the 2D bounding box and class label of the object as the prior 3D space information. By this way, invalid computation in 3D searching space is reduced.

As shown in Fig 3, with the known projection matrix as a priori, we can get a 3D frustum by projecting a 2D bounding box region to a 3D viewing frustum.

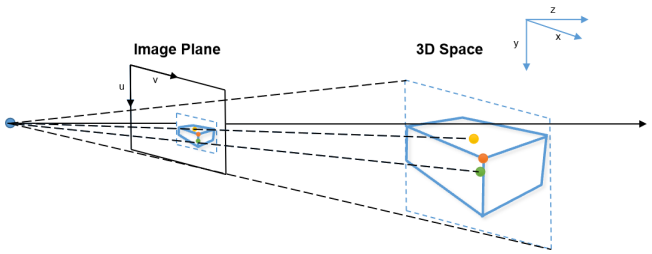

Figure 3: Projecting a 2D bounding box to the 3D frustum space with the knowm camera projection matrix.

The frustum determines the 3D search space for the object. In these 3D search space, PAG-Net generates precision and category-adaptive 3D anchors, which can improve the detection performance for small objects and reduce 3D searching space. Afterwards, 3D proposals are generated by RPN from these anchors and projected to both the RGB image plane and the BEV in the 3D space.

3D proposal generation. We generate 3D proposals from these precision and category-adaptive 3D anchors. Similar to RPN [43], we regress 3D anchor box by computing $\mathrm{t} = (\Delta x, \Delta y, \Delta z, \Delta l, \Delta w, \Delta h)$, which are the normalized offsets in centroid and dimensions between anchors and ground truth bounding boxes. We use a multi-task loss to simultaneously classify object/background and do 3D box regression [11]. Smooth L1 loss [26] is used for 3D box regression, and cross-entropy loss for "objectness". During training, we compute the 2D intersection over union(IoU) overlap in BEV between anchors and ground truth. An anchor is considered to be positive if its overlap is above 0.7, and negative if the overlap is below 0.5. Anchors with overlap in between 0.5 and 0.7 are ignored. During both training and testing, empty anchors are ignored to reduce computation. Meanwhile, in order to remove redundant proposals, 2D non-maximum suppression (NMS) at an IoU threshold of 0.8 in BEV is used to keep the top 1024 proposals during training.

3.2. Multi-view feature fusion network

We design a multi-layer feature fusion network to combine BEV maps and RGB images for 3D object detection. The fused feature maps are directly fed

into the detection network for classification and regression.

Equal-sized feature crops generation. The crop and resize operation [15] is used to extract crops for every anchor from images and BEV maps. We obtain RoIs by projecting the 3D anchors onto the BEV and image feature maps. Afterwards, we extract multi-view feature map crops from RoIs. These feature map crops are then bilinearly resized to 3×3 to obtain equal-length feature vectors. Due to the expensive computation of processing such high-dimensional feature map crops with the RPN, we apply a 1×1 convolutional kernel [12] on these output feature map crops. It is an efficient dimensionality reduction mechanism. The 1×1 convolution acts on every pixel position in each feature map according to the following format:

$$f_{out} = \sigma(\sum_{i=1}^{\widetilde{D}} w_i f_i + b), \tag{1}$$

where f_i is the pixel value at each of the input feature map channels, w_i is a learned weight, b is a learned bias term, $\widetilde{D}$ is channels of the input feature map, f_{out} is the output pixel value of feature map by 1×1 convolution, and σ is a non-linear activation function.

The 1×1 convolutions can be considered as linear coordinate-based transformations in the filter space, followed by a non-linear activation function [15]. By learning, RPN not only retains useful information for proposal generation tasks, but also reduces the input feature map dimensionality.

Multi-layer feature fusion. To combine the equal-sized feature crops, deep fusion [15] first connects multi-view features, then performs feature extraction. The above process is iteratively operated to enable more interactive feature fusion of intermediate layers of CNN from different views. For a network including L convolutional layers, deep fusion combines f_v, $v \in \{BEV, FV, RGB\}$, which are the feature maps in the input stages from bird's eye view, the front view, and the image plane, respectively. Deep fusion process is designed as

follows:

$$
\begin{aligned}
&f_0 = f_{BEV} \oplus f_{FV} \oplus f_{RGB}, \\
&f_l = {H_l}^{BEV}(f_{l-1}) \oplus {H_l}^{FV}(f_{l-1}) \oplus {H_l}^{RGB}(f_{l-1}), \forall l = 1, ..., L,
\end{aligned}
\tag{2}
$$

where $\{{H_l}^{BEV}$, ${H_l}^{FV}$, ${H_l}^{RGB}$, l=1,...,L$\}$ denote the operation of feature extraction from different views, that is, BEV, FV, and RGB image, $\oplus$ is a join operation of element-wise mean to fuse features of the intermediate layers from different views.

Deep fusion uses element-wise mean which is an operation between two tensors that calculates the mean on corresponding elements within the respective tensors. Meanwhile, the same feature f_{l-1} is used for feature transformation on different modalities. As a result, the quality of feature fusion depends largely on the features f_0 after the first fusion, which makes it is difficult to extract the most representative features.

Inspired by [9], we propose a novel multi-layer feature fusion approach to encode more representative features from multiple layers for effective 3D object detection. The architecture of the proposed multi-layer fusion is shown in Fig 4. The multi-layer feature fusion is calculated as follows:

$$
\begin{aligned}
&f_0 = f_{BEV} \otimes f_{FV} \otimes f_{RGB}, \\
&M_0^{BEV} = f_{BEV}, M_0^{FV} = f_{FV}, M_0^{RGB} = f_{RGB} \\
&f_l = {M_l}^{BEV} \otimes {M_l}^{FV} \otimes {M_l}^{BEV}, \forall l = 1, ..., L,
\end{aligned}
\tag{3}
$$

where

$$
\begin{aligned}
{M_l}^{BEV} &= {H_l}^{BEV}(M_{l-1}^{BEV}) \otimes f_{l-1}, \\
{M_l}^{FV} &= {H_l}^{FV}(M_{l-1}^{FV}) \otimes f_{l-1}, \\
{M_l}^{RGB} &= {H_l}^{RGB}(M_{l-1}^{RGB}) \otimes f_{l-1},
\end{aligned}
\tag{4}
$$

where f_{BEV}, f_{FV}, f_{RGB} are the feature maps from the bird's eye view, the front view, and the image plane, respectively, $\{{H_l}^{BEV}, {H_l}^{FV}, {H_l}^{RGB}, l = 1, ..., L\}$ denote the operation of feature extraction from different views, that is, BEV, FV, and RGB image, and $\otimes$ is concatenation operation which consolidates dimension to implement feature fusion.

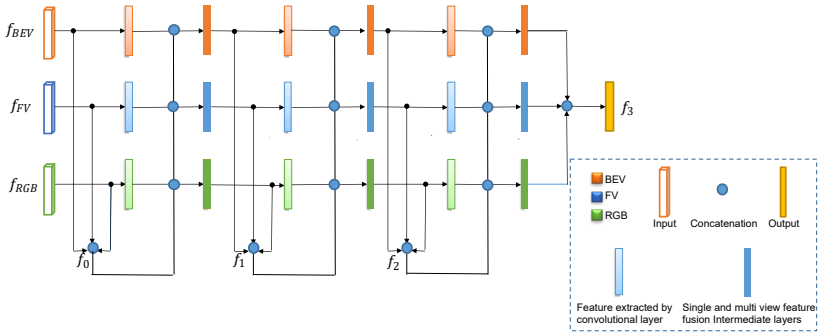

Figure 4: Architecture of the proposed multi-layer feature fusion scheme. We instantiate the join nodes in multi-layer fusion with concatenation operation.

Unlike element-wise operation $\oplus$, the concatenation operation $\otimes$ makes the number of fusion channels equal to the sum of channels in each feature map. The proposed multi-layer feature fusion model uses concatenation for the join operation, which can adjust parameters automatically by assigning weights to different modal features. In addition, we integrate local and global features to obtain features of the receptive field with semantic information. We achieve comprehensive interaction of multi-layer feature by iterative nonlinear combination. Compared with the previous fusion methods, multi-layer fusion improves the effectiveness of the feature fusion and makes the fused features contain richer information, which can improve the detection accuracy. The proposed RPN uses multi-layer feature fusion to perform multi-modal feature fusion on high resolution feature maps for reliable 3D object proposals. Finally, detection and pose estimation network predicts oriented 3D bounding boxes for objects.

3.3. Encoding 3D bounding box

In AVOD [15], 3D bounding box is encoded by four corners and two heights as shown in Fig 5(a). Compared with eight-corner encoding [11], this encoding method reduces the redundancy of regression with physical constraints and provides better detection results. However, in some practical scenarios like autonomous driving, objects we interest, such as cars, pedestrians, cyclists, are all located on the ground plane. Therefore, as shown in Fig 5(b), we remove

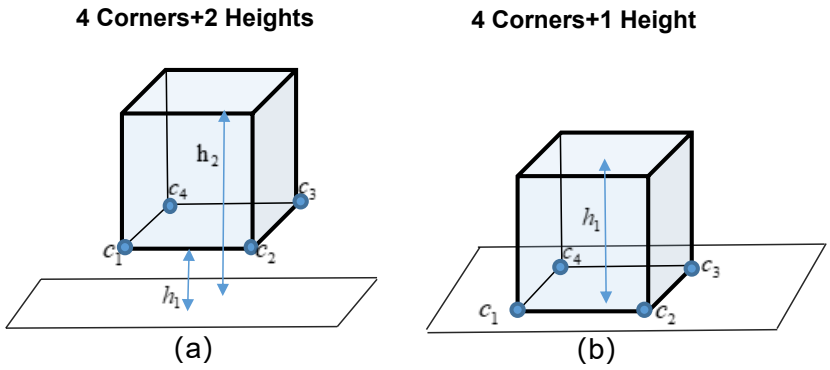

Figure 5: Visual comparison between the 4corners+2heights box encoding proposed in [15], and our 4corners+1height encoding.

one height parameter which denotes the distance of the object from the ground plane, to encode the 3D bounding box. Specifically speaking, we encode the bounding box with four corners and one height which represent the size and the height of the object respectively. Hence, parameters we need to regress are $(\Delta x_1 ... \Delta x_4, \Delta y_1 ... \Delta y_4, \Delta h_1)$, where (x_s, y_s), $s \in (1...4)$ are the coordinates of four vertices at the bottom of the 3D box, and h_1 is the height from the top plane of 3D box to the ground plane. Δ represents the offset between the proposals and ground truth box. The encoding method we propose reduces the box representation from a ten-dimensional vector to a nine-dimensional vector.

4. Experiments

Our approach is evaluated on the KITTI object detection benchmark, focusing on the 3D and BEV detection tasks. We perform ablation studies to evaluate and analyze the performance improvement in 3D object detection for each of the proposed module.

4.1. Dataset

We evaluate our model on the challenging KITTI object detection benchmark [6] and focus on the 3D detection of cars, pedestrians, and cyclists. The

dataset is a novel and challenging real-world computer vision benchmark, including city, rural and highway scenes. It provides 7481 images for training and 7518 images for testing as well as the corresponding point cloud. According to the object size, occlusion state, and truncation level defined by KITTI officially, the dataset is classified into easy, moderate, and hard difficulties for each object category [6].

4.2. Evaluation criteria and parameter configuration

We split the provided 7481 training frames into a training set and a validation set in a ratio of 1:1 following the process mentioned in MV3D [11]. We use the 3D average precision (AP_{3D}) and BEV average precision (AP_{BEV}) to evaluate 3D detection and 3D localization performance respectively. The IoU used in calculating AP_{3D} is the overlapping volume of the 3D object prediction and 3D ground truth, while the IoU used in calculating AP_{BEV} is the overlapping area of the 3D object prediction projected to BEV and the BEV ground truth. The calculate formula for IoU is:

$$IoU = \frac{A(overlap\ of\ ground\ truth\ and\ prediction)}{A(union\ of\ ground\ truth\ and\ prediction)} \tag{5}$$

We also use average heading similarity (AHS) to evaluate global orientation angle. Since the ground truth for the test set is not available and the access to the test server is limited, all detection results are measured using the official KITTI evaluation detection for BEV and 3D on the validation set. In all evaluations, the IoU threshold for car, pedestrian and cyclist are set to 0.7, 0.5, 0.5 respectively. The network is trained by 120K iterations with an ADAM optimizer. The initial learning rate is set to 0.0001 and decays exponentially with a 0.8 decay factor after each 30K iterations.

4.3. Comparison with the state-of-art methods

For each object category, we do evaluation for our proposed methods on three difficulty levels from easy to hard. We compare it with several top-performing algorithms, including RGB-D based approaches: F-PointNet [9]; only LIDAR

based approaches: VoxelNet [27] and BirdNet+ [36]; and a multimodal approach MV3D [11] and AVOD [15]. MV3D uses FV, BEV, and RGB image multi-view, while AVOD and ours use BEV and RGB image. Table 1, Table 2, and Table 3 show the AP_{3D} and AP_{BEV} of the 3D detection results for pedestrian, cyclist and car respectively.

The car class evaluation result is presented in Table 1. Ours methods consistently outperforms all the competing approaches across all three difficulty levels. Compared with baseline AVOD, our model outperforms by 1.55%, 2.2%, 1.45% on easy, moderate, and hard setting of the car category in AP_{3D}.

For pedestrian and cyclist detection tasks in 3D and BEV, we compare our method with our baseline AVOD. From table 1 and table 2, we can see that ours yields substantially higher AP_{3D} and AP_{BEV} than AVOD for every category across all difficult levels. In terms of easy, moderate and hard difficulties of pedestrian, our proposed method was 5.38%, 8.41% and 9.05% higher than AVOD respectively in AP_{3D}. We can see that the increase in the AP_{3D} is more remarkable for the difficulty of moderate and hard compared to easy. This result is attributed to the assist from prior anchor information by 2D detector. Objects in pedestrian class are smaller and have fewer point clouds, so they benefit the most from this prior location and category information. In the experimental

Table 1: Performance comparison of 3D detection and 3D localization for Car class on the KITTI validation set. For evaluation, we show the AP_{3D} and AP_{BEV} (in %) for Cyclist at 3D IoU of 0.7.

		AP(3D)			AP(BEV)		
Method	Class	Easy	Moderate	Hard	Easy	Moderate	Hard
MV3D[11]	Car	71.09%	62.35%	55.12%	86.02%	76.90%	68.49%
VoxelNet[27]		77.47%	65.11%	57.73%	89.35%	79.26%	77.39%
F-PointNet[9]		81.02%	70.39%	62.19%	88.70%	84.00%	75.33%
BirdNet+[36]		70.14%	51.85%	50.03%	84.80%	63.33%	61.23%
*AVOD**[15]		*81.94%*	*71.88%*	*66.38%*	*88.53%*	*83.79%*	*77.73%*
ours		***83.49%***	***74.08%***	***67.83%***	***89.81%***	***87.00%***	***79.71%***

* AVOD is the baseline. Both AVOD and ours use BEV maps and RGB images.

Table 2: Performance comparison of 3D detection and 3D localization for Pedestrian class on the KITTI validation set. For evaluation, we show the AP_{3D} and AP_{BEV} (in %) for Cyclist at 3D IoU of 0.5.

		AP(3D)			AP(BEV)		
Method	Class	Easy	Moderate	Hard	Easy	Moderate	Hard
MV3D[11]		N/A	N/A	N/A	N/A	N/A	N/A
VoxelNet[27]		39.48%	33.69%	31.51%	46.13%	40.74%	38.11%
F-PointNet[9]	Ped.	51.21%	44.89%	40.23%	**58.09%**	50.22%	47.20%
BirdNet+[36]		37.99%	31.46%	29.46%	45.53%	38.28%	35.37%
AVOD[15]*		*46.35%*	*39.00%*	*36.58%*	*50.66%*	*44.75%*	*40.83%*
ours		***51.73%***	***47.41%***	***45.63%***	*55.29%*	***53.84%***	***47.96%***

* AVOD is the baseline. Both AVOD and ours use BEV maps and RGB images.

Table 3: Performance comparison of 3D detection and 3D localization for Cyclist class on the KITTI validation set. For evaluation, we show the AP_{3D} and AP_{BEV} (in %) for Cyclist at 3D IoU of 0.5.

		AP(3D)			AP(BEV)		
Method	Class	Easy	Moderate	Hard	Easy	Moderate	Hard
MV3D[11]		N/A	N/A	N/A	N/A	N/A	N/A
VoxelNet[27]		61.22%	48.36%	44.37%	66.70%	54.76%	50.55%
F-PointNet[9]	Cyc.	**71.96%**	**56.77%**	**50.39%**	**75.38%**	**61.96%**	**54.68%**
BirdNet+[36]		67.38%	47.72%	42.89%	72.45%	52.15%	46.51%
AVOD[15]*		*59.97%*	*44.90%*	*38.80%*	*62.39%*	*52.02%*	*47.87%*
ours		*62.37%*	*47.28%*	*46.56%*	*66.01%*	*57.76%*	*46.98%*

* AVOD is the baseline. Both AVOD and ours use BEV maps and RGB images.

part of AVOD, it mentioned that MV3D does not publicly provides results on the pedestrian and cyclist classes for the 3D object detection task, and hence comparison is done for the car class only. For the same reason, the results for MV3D are absent in Table 2 and Table 3.

Among the "BEV+RGB" methods, our approach outperforms the baseline AVOD by 2.4%, 2.9%, 7.76% AP_{3D} on easy, moderate and hard setting of cyclist class respectively. However, our method does not achieve the best performance on cyclist detection in Table 3, because there are few training samples of the category of cyclist and many scenarios with severe occlusion and cyclists clus-

tered closely in KITTI dataset. Moreover, compared with the raw point cloud data, the BEV data we based on, renders the introducing quantization artifacts. These artifacts can obscure native invariance of the data, and make the cyclist detection fail. F-PointNet [9] uses raw point cloud without projecting to other views. By the benefit of directly utilizing raw point clouds, F-pointnet is able to precisely estimate 3D bounding boxes even under severe occlusion or with very sparse points. Therefore, F-pointnet works better than ours on cyclist detection. For fair comparison, we focus on the multi-modal variant which combines BEV and RGB data.

Compared with the baseline AVOD, table 1, table 2, and table 3 show that our improvement in the AP_{3D} is more remarkable for the classes of pedestrian and cyclist compared to car. This is attributed to 2D-guided PAG-Net, which uses 2D object detector for the location and category of object as the prior information of anchors. This prior 2D information makes our model generate precision and category-adaptive anchors. Furthermore, we can achieve more accurate RoIs through the RPN network to significantly improve the small object detection performance. Moreover, multi-layer fusion uses multiple iterations and high-low level fusion. This fusion method improves the quality of the fused features, making our model achieve significant improvement on moderate and hard difficulties.

Fig 6 shows the P-R curves of the car and pedestrian detection by our pro-

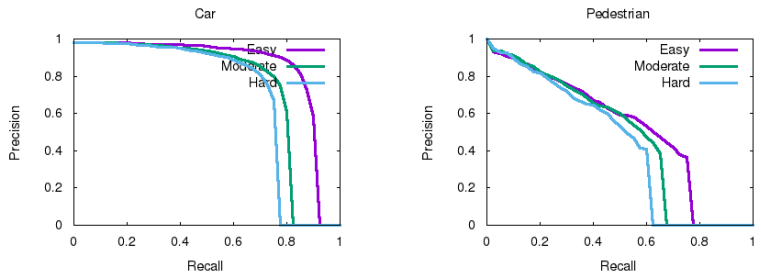

Figure 6: The precision-recall (P-R) curves for car and pedestrian categories. Recall vs hard difficulty is shown in blue, moderate difficulty in green, and the easy difficulty in purple.

posed model. We can see that the model performs better on the car detection than the pedestrian detection. On the car detection, the precision drops significantly when the recall rate is over 0.7. However, the precision on the pedestrian detection drops continuously from the beginning and drops significantly when the recall rate is over 0.6. The reason is that the point clouds of pedestrians and cyclists are very sparse and similar, especially for small objects. Therefore, the performance of the pedestrian and cyclist detection is worse than that of the car.

4.4. Ablation studies

We use the AVOD [15] framework as the baseline in order to evaluate the performance improvement in 3D object detection using 2D-guided PAG-Net and multi-layer fusion which combines RGB and BEV features.

The effect of the added PAG-Net in anchor strategy on the detection performance can be observed in Table 4. The result shows that the PAG-Net improves the detection. The AP_{3D} increases 1.34%, 1.24%, and 0.89% on easy, moderate and hard data of the car class. Baseline+PAG-Net also has enhancement in the detection by 1.05%, 3.02%, and 2.06% increase in AP_{BEV}. The anchors from baseline with no semantic information are discrete in the whole search space, which causes some mismatching between objects and anchors. However, PAG-Net takes the locations and classifications of objects by 2D detector as the prior information to generate precision and category-adaptive anchors, which improves the performance of 3D object detection. The performance gained on smaller classes is much more substantial. Specifically, we achieve an improvement of 5.82%, 6.28%, 6.65% AP_{3D} on the pedestrain class. This shows that PAG-Net is essential to achieve state-of-the-art results. Experiments prove the advantage of our method.

The experimental results of deep fusion and multi-layer fusion on feature fusion are shown in Table 5. The baseline is AVOD with deep fusion, while Baseline++ uses multi-layer fusion to fuse RGB image and BEV map features. As the experimental results show, the method of multi-layer feature fusion achieves

Table 4: Comparison of different anchor generation approaches. Performance are evaluated on the KITTI car class validation set.

Method	Class	Easy		Moderate		Hard	
		$AP_{3D}(\%)$	$AP_{BEV}(\%)$	$AP_{3D}(\%)$	$AP_{BEV}(\%)$	$AP_{3D}(\%)$	$AP_{BEV}(\%)$
Baseline	Car	81.94	88.53	71.88	83.79	66.38	77.73
Baseline+PAG-Net	Car	**83.28**	**89.58**	**73.12**	**86.81**	**67.27**	**79.79**
*Delta**	Car	1.34	1.05	1.24	3.02	0.89	2.06
Baseline	Ped.	46.35	50.66	39.00	44.75	36.58	40.83
Baseline+PAG-Net	Ped.	**51.95**	**54.74**	**46.21**	**48.57**	**40.13**	**42.31**
*Delta**	Ped.	5.60	4.08	7.21	3.82	3.55	1.48
Baseline	Cyc.	59.97	62.39	44.90	52.02	38.80	47.87
Baseline+PAG-Net	Cyc.	**61.18**	**64.07**	**46.08**	**54.89**	**39.82**	**48.53**
*Delta**	Cyc.	1.21	1.68	1.18	2.87	1.02	0.66

* Delta is the performance difference between the baseline+PAG-Net and the baseline.

Table 5: Comparison of different fusion approaches. Performance are evaluated on the KITTI car class validation set.

Method	Class	Easy		Moderate		Hard	
		$AP_{3D}(\%)$	AHS(%)	$AP_{3D}(\%)$	AHS(%)	$AP_{3D}(\%)$	AHS(%)
Baseline(Deep fusion)	Car	81.94	81.58	68.47	66.78	67.25	65.28
Baseline++(Multi-layer fusion)	Car	**83.77**	**83.44**	**73.84**	**72.16**	**67.37**	**66.62**
*Delta**	Car	1.83	1.86	5.37	5.38	0.12	1.34
Baseline(Deep fusion)	Ped.	46.35	44.78	36.58	39.31	35.31	36.07
Baseline++(Multi-layer fusion)	Ped.	**52.17**	**46.61**	**42.86**	**44.23**	**41.96**	**40.36**
*Delta**	Ped.	5.82	1.83	6.28	4.92	6.65	4.29
Baseline(Deep fusion)	Cyc.	59.97	61.01	44.90	45.58	38.80	40.43
Baseline++(Multi-layer fusion)	Cyc.	**61.17**	**62.79**	**46.82**	**47.16**	**43.37**	**44.62**
*Delta**	Cyc.	1.20	1.78	1.92	1.58	4.57	4.19

*Delta is the performance difference between the deep fusion and the multi-layer fusion.

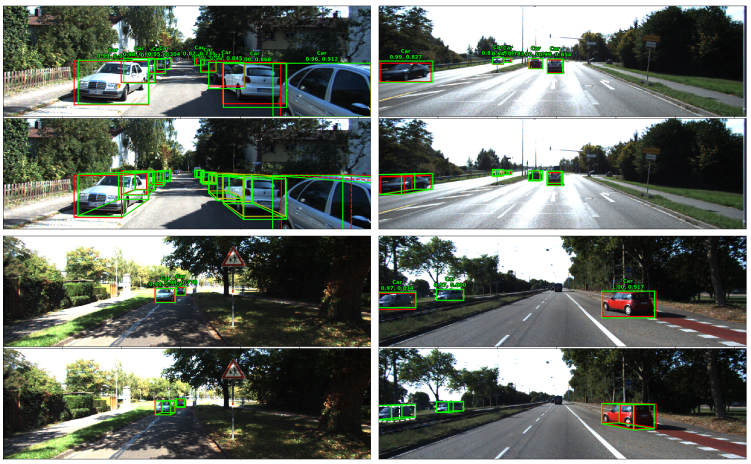

Figure 7: Qualitative results of the proposed framework. 3D boxes are projected to the images for visualization.

1.83%, 5.37%, and 0.12% increase for AP_{3D} of our method comparing to the baseline on easy, moderate, and hard setting of the car class respectively. It also achieves enhancement in the detection by 1.86%, 5.38%, and 1.34% for AHS. For car class, Baseline++ significantly outperforms baseline in AP_{3D} and AHS across moderate difficulty levels. For the pedestrian and cyclist classes, multi-layer fusion also shows to provide significant increasement in AP_{3D} and AHS.

We attribute this performance to the fact that there are partial occlusion and truncation which can be solved by improving the quality of feature fusion. The reason why deep fusion does not work is that it uses element-wise mean of RGB image and BEV map features for the join operation. Because the feature representation, distribution and density in different modal are different, element-wise mean ignores the relationship between multi-modal features and easily generate data redundancy. To deal with these problems, we use concatenation for join operation, which can adjust parameters automatically by learning to allocate different weights to different modal features. In addition, we fuse multi-layer feature including local and global features and use an iterative operation to make the fusion of RGB image and BEV maps more interactive. Meanwhile, the car class has a specific aspect ratio in BEV. We can also easily

judge the direction of pedestrians and cyclists in the RGB image. As a consequence, these more interactive features by multi-layer fusion approach based on RGB image and BEV can also make AHS increase. The output of our model is visualized in Fig 7. We can see that the proposed model achieves accurate 2D and 3D bounding boxes prediction even under particularly challenging cases.

5. Conclusion

In this work, we propose a 2D guided PAG-Net and a multi-layer fusion approach for multi-modal based 3D object detection in autonomous driving scenarios. We use the PAG-Net to generate precision and category-adaptive anchors by 2D object detector to get the category label and 2D bounding box of the object. They are then projected into 3D frustum space where precision anchors are generated to drive 3D object detection. The proposed PAG-Net provides prior information for 3D object detection. Moreover, the proposed multi-layer fusion strategy enables more interaction among RGB images and BEV map features, which makes the fused features contain richer information. Experiments on the KITTI dataset show the improvements of our proposed architecture compared with the state-of-the-art methods on the 3D object detection tasks including 3D localization, orientation estimation, and category classification.

6. Acknowledgments

The work is supported by the following projects: National Natural Science Foundation of China (NSFC) Nr.: 61772328, NSFC Essential project, Nr.: 61831018, U2033218.

References

[1] C. Yan, E. Salman, Mono3D: Open Source Cell Library for Monolithic 3-D Integrated Circuits, IEEE Transactions on Circuits and Systems(TCS) (2018).

[2] G. Brazil, X. Liu, M3D-RPN: Monocular 3D region proposal network for object detection, in: IEEE International Conference on Computer Vision (ICCV), 2019.

[3] P. Li, X. Chen, S. Shen, Stereo R-CNN based 3D object detection for autonomous driving, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[4] M. Engelcke, D. Rao, D. Z. Wang, C. H. Tong, I. Posner, Vote3Deep: Fast object detection in 3D point clouds using efficient convolutional neural networks, in: IEEE International Conference on Robotics and Automation (ICRA), 2017.

[5] A. Mousavian, D. Anguelov, J. Košecká, J. Flynn, 3D bounding box estimation using deep learning and geometry, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[6] A. Geiger, P. Lenz, R. Urtasun, Are we ready for autonomous driving? the kitti vision benchmark suite, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2012.

[7] C. R. Qi, H. Su, K. Mo, L. J. Guibas, PointNet: Deep learning on point sets for 3D classification and segmentation, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[8] C. Qi, L. Yi, H. Su, L. Guibas, PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space, in: International Conference on Neural Information Processing Systems (NIPS), 2017.

[9] C. R. Qi, W. Liu, C. Wu, H. Su, L. J. Guibas, Frustum PointNets for 3D Object Detection from RGB-D Data, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[10] E. Arnold, O. Y. Al-Jarrah, M. Dianati, S. Fallah, D. Oxtoby, A. Mouzakitis, A survey on 3d object detection methods for autonomous driving appli-

cations, IEEE Transactions on Intelligent Transportation Systems (TITS) Vol: 20 (10) (2019) pp: 3782–3795.

[11] X. Chen, H. Ma, J. Wan, B. Li, T. Xia, Multi-view 3D Object Detection Network for Autonomous Driving, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[12] D. Xu, D. Anguelov, A. Jain, PointFusion: Deep Sensor Fusion for 3D Bounding Box Estimation, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[13] Q. Xie, Y.-K. Lai, J. Wu, Z. Wang, Y. Zhang, K. Xu, J. Wang, MLCVNet: Multi-Level Context VoteNet for 3D Object Detection, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) (2020).

[14] Z. Cai, Q. Fan, R. S. Feris, N. Vasconcelos, A unified multi-scale deep convolutional neural network for fast object detection, in: European conference on computer vision (ECCV), 2016.

[15] J. Ku, M. Mozifian, J. Lee, A. Harakeh, S. L. Waslander, Joint 3D Proposal Generation and Object Detection from View Aggregation, in: IEEE International Conference on Intelligent Robots and Systems (IROS), 2018.

[16] J. Deng, K. Czarnecki, MLOD: A multi-view 3D object detection based on robust feature fusion method, in: IEEE Intelligent Transportation Systems Conference (ITSC), 2019.

[17] S. Song, J. Xiao, Deep sliding shapes for amodal 3d object detection in rgb-d images, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[18] M. Z. Zia, M. Stark, K. Schindler, Towards scene understanding with detailed 3d object representations, International Journal of Computer Vision (IJCV) Vol: 112 (2) (2015) pp: 188–203.

[19] A. Simonelli, S. R. Bulo, L. Porzi, M. Lopez-Antequera, P. Kontschieder, Disentangling monocular 3D object detection, in: IEEE International Conference on Computer Vision (ICCV), 2019.

[20] L. Liu, J. Lu, C. Xu, Q. Tian, J. Zhou, Deep fitting degree scoring network for monocular 3D object detection, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[21] A. Naiden, V. Paunescu, G. Kim, B. Jeon, M. Leordeanu, Shift R-CNN: Deep Monocular 3D Object Detection with Closed-Form Geometric Constraints, in: International Conference on Image Processing (ICIP), 2019.

[22] J. Ku, A. D. Pon, S. L. Waslander, Monocular 3D object detection leveraging accurate proposals and shape reconstruction, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[23] X. Ma, Z. Wang, H. Li, P. Zhang, W. Ouyang, X. Fan, Accurate monocular 3D object detection via color-embedded 3D reconstruction for autonomous driving, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[24] X. Chen, K. Kundu, Y. Zhu, H. Ma, S. Fidler, R. Urtasun, 3D Object Proposals Using Stereo Imagery for Accurate Object Class Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2018).

[25] X. Chen, K. Kundu, Y. Zhu, A. G. Berneshawi, H. Ma, S. Fidler, R. Urtasun, 3d object proposals for accurate object class detection, in: Neural Information Processing Systems (NIPS), 2015.

[26] J. Papon, A. Abramov, M. Schoeler, F. Worgotter, Voxel cloud connectivity segmentation - Supervoxels for point clouds, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2013.

[27] Y. Zhou, O. Tuzel, VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[28] Z. Liu, H. Tang, Y. Lin, S. Han, Point-voxel cnn for efficient 3d deep learning, in: Neural Information Processing Systems (NIPS), 2019.

[29] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, H. Li, Pv-rcnn: Point-voxel feature set abstraction for 3d object detection, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[30] Y. Chen, S. Liu, X. Shen, J. Jia, Fast point r-cnn, in: IEEE International Conference on Computer Vision (ICCV), 2019.

[31] S. Song, J. Xiao, Sliding shapes for 3d object detection in depth images, in: European conference on computer vision (ECCV), 2014.

[32] D. Z. Wang, I. Posner, Voting for voting in online point cloud object detection., Robotics: Science and Systems (R: SS) (2015).

[33] B. Li, 3D fully convolutional network for vehicle detection in point cloud, in: IEEE International Conference on Intelligent Robots and Systems (IROS), 2017.

[34] S. Shi, X. Wang, H. Li, PointRCNN: 3D Object Proposal Generation and Detection From Point Cloud, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[35] B. Li, Z. Tianlei, X. Tian, Vehicle detection from 3d lidar using fully convolutional network, Robotics: Science and Systems (R: SS) (2016).

[36] A. Barrera, C. Guindel, J. Beltrn, F. Garca, BirdNet+: End-to-End 3D Object Detection in LiDAR Bird's Eye View, IEEE International Conference on Intelligent Transportation Systems (ITSC) (2020).

[37] K. Klasing, D. Wollherr, M. Buss, A clustering method for efficient segmentation of 3D laser data, in: IEEE International Conference on Robotics and Automation (ICRA), 2008.

[38] B. Yang, W. Luo, R. Urtasun, PIXOR: Real-time 3D Object Detection from Point Clouds, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[39] B. Li, W. Ouyang, L. Sheng, X. Zeng, X. Wang, GS3D: An Efficient 3D Object Detection Framework for Autonomous Driving, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[40] C. R. Qi, X. Chen, O. Litany, L. J. Guibas, Imvotenet: Boosting 3d object detection in point clouds with image votes, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[41] S. Shi, Z. Wang, J. Shi, X. Wang, H. Li, From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2020).

[42] Z. Yang, Y. Sun, S. Liu, J. Jia, 3DSSD: Point-based 3D Single Stage Object Detector, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[43] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, in: Neural Information Processing Systems (NIPS), 2015.

[44] M. Enzweiler, D. M. Gavrila, A multilevel mixture-of-experts framework for pedestrian classification, IEEE Transactions on Image Processing (TIP) Vol: 20 (2011) pp: 2967–2979.

[45] S. Vora, A. H. Lang, B. Helou, O. Beijbom, Pointpainting: Sequential fusion for 3d object detection, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[46] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2020).

[47] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, International journal of computer vision (IJCV) Vol: 115 (3) (2015) pp: 211–252.

[48] A. H. Raffiee, H. Irshad, Class-specific anchoring proposal for 3d object recognition in lidar and rgb images, arXiv preprint arXiv:1907.09081 (2019).

[49] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2017.