Three-Attention Mechanisms for One-Stage 3D Object Detection Based on LiDAR and Camera

Lihua Wen, *Member, IEEE,* Kang-Hyun Jo, *Senior Member, IEEE*

***Abstract*—This paper studies one-stage 3D object detection based on LiDAR point clouds and RGB images that aims to boost 3D object detection accuracy based on three attention mechanisms. Currently, most of the previous works converted LiDAR point clouds into bird's-eye-view (BEV) images, achieving significant performance. However, they still have a problem due to partial height information (z-axis value) loss during the conversion. To eliminate this problem, the height information of the LiDAR point clouds is projected onto an RGB image and embedded into the original RGB image to generate a new image, named RGB$^{\mathbf{D}}$. This is the first attention mechanism to improve 3D detection accuracy. Moreover, two other attention mechanisms extract more discriminative global and local features, respectively. Specifically, the global attention network is appended to a feature encoder, and the local attention network is used for the view-specific region of interest (ROI) fusion. Massive experiments evaluated on the KITTI benchmark suite show that the proposed approach outperforms state-of-the-art LiDAR-Camera-based methods on the car class (Easy, Moderate, Hard): 2D (90.35%, 88.47%, 86.98%), 3D (85.12%, 76.23%, 74.46%), and BEV (89.64%, 86.23%, 85.60%).**

***Index Terms*—One-stage, 3D object detection, three attention mechanisms, LiDAR, camera.**

I. Introduction

WITH the rapid development of autonomous vehicles, three-dimensional (3D) object detection has become more important. 2D object detection only analyzes what the objects are and where they are in the 2D image. However, autonomous vehicles have to analyze road targets and their accurate locations in real time. Achieving accurate and real-time 3D object detection is a huge challenge. Currently, 3D object detection mainly get 3D data from either LiDAR or camera.

Recently, 2D object detection [1]–[3] with deep learning has drawn much attention. Most researchers study 3D object detection based on LiDAR point clouds using 2D detection methods. Point clouds generated by LiDAR are sparse and irregular. Hence, representative studies either convert point clouds into 2D front view images [4], 2D bird's-eye-view (BEV) images [5], or structured voxel-grid representations [6], [7]. Then, 2D convolutional layers are used to extract features from the converted images. Some point-based methods [8], [9] directly utilize multi-layer perceptron (MLP) to aggregate features from point clouds. However, LiDAR-based approaches suffer 3D information loss in distant regions due to the sparsity of the point clouds. Compared with point-based methods [8], [9], the BEV-based method [5] is a little faster, however, it suffers partial information loss during the conversion. This work employs the RGB$^{\mathbf{D}}$ image to reduce the information loss.

2D RGB images possess dense texture, and high-resolution images have enough cues for small objects. However, it is difficult to extract accurate 3D localization features when using monocular images due to the lack of depth information [10]–[12]. Currently, even if stereo images are used [13], the accuracy of the estimated depth is not guaranteed. Therefore, some studies [4], [14]–[16] take mutual advantage of 2D images and point clouds to achieve accurate 3D object detection. These methods directly fuse the view-specific features by a common concatenation [4] or an element-wise mean operation [14], resulting in poor accuracy of 3D object detection. This paper adopts the region of interest attention (RA) fusion mechanism to deeply merge the view-specific features.

MV3D [4] and AVOD [14] adopt the two-stage frameworks to detect 3D objects based on point clouds and RGB images. The first stage generates 3D proposals, and the following stage refines the proposals to predict 3D objects. Compared with the one-stage method, the two-stage 3D detection model is relatively time-consuming. Therefore, some works [15] [16] utilize the one-stage framework to detect 3D objects. Without the second stage to refine 3D proposals, the above one-stage works yield a worse detection accuracy than the two-stage methods. After analysis, enhancing the feature representation of one-stage methods is the most effective way to improve 3D object detection. Hence, a global feature attention (GFA) mechanism is used for boosting global feature representation.

To overcome the above drawbacks, this paper presents a novel one-stage 3D object detection framework, as shown in Fig. 1, based on three-attention mechanisms, called TAO3D, which takes raw point cloud and RGB image as inputs. Three attention mechanisms are used to obtain discriminative features. First, the height attention (HA) mechanism is introduced as an auxiliary attention module before the RGB image is fed into a network. Second, a global feature attention (GFA) mechanism models the long-range dependencies in the channel and spatial dimensions simultaneously at the feature extraction phase. Finally, a region of interest attention (RA) mechanism weights RGB image ROIs and BEV ROIs using two learnable parameters.

The main contributions of this framework are summarized as follows:

Manuscript received Month xx, 20xx; accepted Month xx, 20xx. Date of publication Month, xx, 20xx; date of current version Month, xx, 20xx. This work has been supported by University of Ulsan).(Corresponding author: Kang-Hyun Jo.)

Lihua Wen is with the Graduate School of Electrical Engineering, University of Ulsan, 44610, South Korea, e-mail: wenlihuawlh@gmail.com.

Prof. Kang-Hyun Jo is with the Graduate School of Electrical Engineering, University of Ulsan, 44610, South Korea, e-mail: acejo@ulsan.ac.kr.

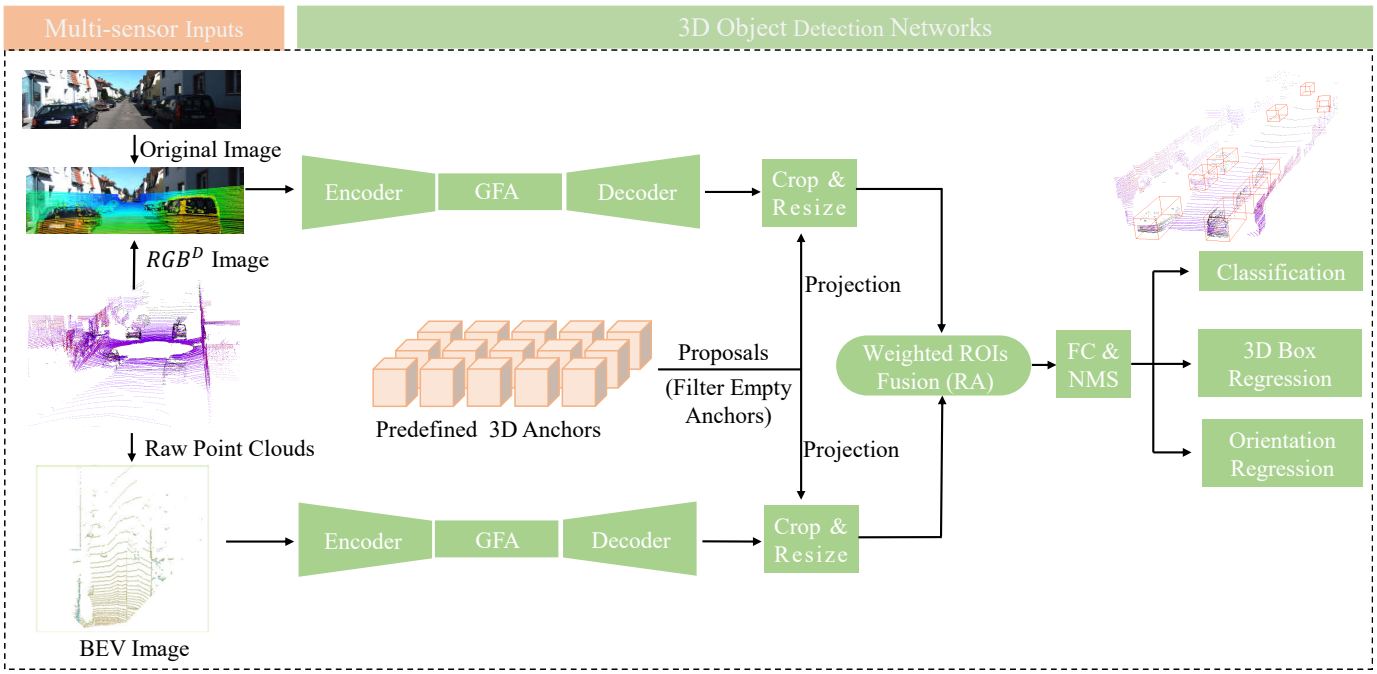

Figure 1: The architecture of a one-stage 3D object detection network based on LiDAR and camera. The model first employs two sibling branches to extract the features from RGB^D images and BEV images, respectively. Second, the prior anchors have filtered the empty anchors, and then projected onto RGB^D feature maps and BEV feature maps to crop equal length view-specific ROIs. Finally, the fused ROIs are utilized for classification and regression. Best viewed with color.

1) It takes the raw LiDAR point clouds and RGB^D images as inputs instead of the RGB image. RGB^D images contain the height information from point clouds.
2) The GFA mechanism captures the feature dependencies in both the channels and spatial dimensions at the feature extraction stage and makes the features much more discriminative.
3) The RA mechanism weights the paired BEV ROIs and RGB image ROIs firstly and then fuses them using the addition operation. This gives more weight to important features.

The proposed one-stage 3D object detection framework outperforms state-of-the-art LiDAR-Camera-based methods on the KITTI benchmark [17].

II. Related Work

This section mainly reviews the related works for 3D object detection based on LiDAR and camera, and the attention networks for computer vision tasks.

A. LiDAR-Camera-Based 3D Object Detection

MV3D [4] first introduced multi-modality (RGB image, front image, BEV image) 3D object detection with three backbones to extract view-specific features. Compared with MV3D [4], AVOD [14] only takes RGB images and BEV images as inputs to reduce model runtime. Both MV3D and AVOD make use of a two-stage 3D object detection framework. To speed up training and inference, AVOD-SSD [15] and [16] adopt a one-stage 3D object detection framework. The models run a little faster than AVOD [14], but both the performances greatly drop. In the first step, the proposed work enhances the one-stage detection framework with the RGB^D images at the input phase of the RGB image.

B. Attention Networks

Attention modules model long-range dependencies and have been widely applied in segmentation tasks [18], [19]. DANet [18] introduces a self-attention mechanism to capture rich contextual dependencies for scene segmentation, which models the semantic interdependencies in the channels and spatial dimensions, respectively. CBAM [20] sequentially infers attention maps along two separate dimensions, channel and spatial, then the attention maps are multiplied to the input feature map for adaptive feature refinement. MCF3D [21] introduces a self-attention mechanism for 3D object detection. Different from the above methods, the proposed work employs two attention mechanisms from the global to the local to boost 3D object detection. In the second step, the proposed method utilizes the GFA to enhance the global feature representations. In the final step, the RA is used to enhance the local feature representations.

III. The Proposed Architecture

The main innovation of the proposed framework, as depicted in Fig. 1, employs three-attention mechanisms to make extracted features discriminative. The proposed 3D detection framework mainly includes two parts: the first one is the multi-sensor inputs, and the other one is the 3D object detection networks.

A. Multi-sensor Inputs

This paper directly takes the raw LiDAR point clouds and RGB images as inputs. In the preprocessing, the BEV images and RGB^D images are generated simultaneously by fixed means. Note that LiDAR and camera use two different coordinate systems. In the coordinate system of LiDAR, the x-axis points forward, the y-axis points to the left of the vehicle, and the z-axis points upward. However, in the camera's coordinate system, the x-axis points to the right of the car, the y-axis points downward, and the z-axis points forward. That is why after the height information in LiDAR is projected onto the RGB image plane, it is referred to as depth.

1) Bird's-Eye-View Representation: Point clouds are generated by LiDAR, which encodes the 3D (x, y, z) coordinates and intensity information (I) of surrounding objects. Like AVOD [14], a six-channel BEV image encodes the density and height information in each voxel of a LiDAR frame. Different from MV3D [4], the BEV map does not encode the intensity of point clouds. Specifically, the area to encode BEV image is $\{x, y, z \mid x \in [0, 70], y \in [-40, 40], z \in [-2.3, 0.2]\}$. The voxel grid size is 0.1 meter on both of the x-axis and the y-axis. To keep as much height information as possible, the point clouds are equally sliced into five slices along the z-axis, and the height value is the absolute height relative to the ground. The density map is encoded as $min\left(1.0, \frac{log(N+1)}{log(64)}\right)$ in each pillar, where N is the number of points in one pillar. Note that the density features are computed for the whole point clouds while the height feature is computed for five slices.

2) RGB^D Representation: MCF3D [21] encoded the intensity of point clouds as an additional channel of the original RGB image and named it RGB-I. Different from MCF3D, this paper embeds the projected height information of point clouds into the original RGB image, and calls it RGB^D with 3 channels. The whole process is divided into three steps. First, point clouds (X, Y, Z) are mapped onto the original image $(W \times H)$ plane as follows:

$$\begin{pmatrix} u & v & 1 \end{pmatrix}^T = \mathbf{M} \cdot \begin{pmatrix} X & Y & Z & 1 \end{pmatrix}^T, \tag{1}$$

$$\mathbf{M} = \mathbf{P}_{rect} \cdot \begin{pmatrix} \mathbf{R}_{velo}^{cam} & \mathbf{t}_{velo}^{cam} \\ 0 & 1 \end{pmatrix}, \tag{2}$$

where (u, v) is the image coordinate, $\mathbf{P}_{rect}$ is a project matrix, $\mathbf{R}_{velo}^{cam}$ is the rotation matrix from LiDAR to the camera, $\mathbf{t}_{velo}^{cam}$ is a translation vector, and $\mathbf{M}$ is the homogeneous transformation matrix from LiDAR to the camera.

Second, the points $\{(x, y, z) \mid x \in X, y \in Y, z \in Z\}$ located into the image size $W \times H$ are kept. Meanwhile, the LiDAR points are projected to the camera coordinates and denoted as (x_c, y_c, z_c):

$$\begin{pmatrix} x_c & y_c & z_c \end{pmatrix}^T = \mathbf{M} \cdot \begin{pmatrix} x & y & z & 1 \end{pmatrix}^T. \tag{3}$$

Finally, z_c is mapped between 0 and 255 and then assigned to the corresponding image coordinate (u, v). Fig. 2 shows the difference between the original RGB image and the RGB^D image.

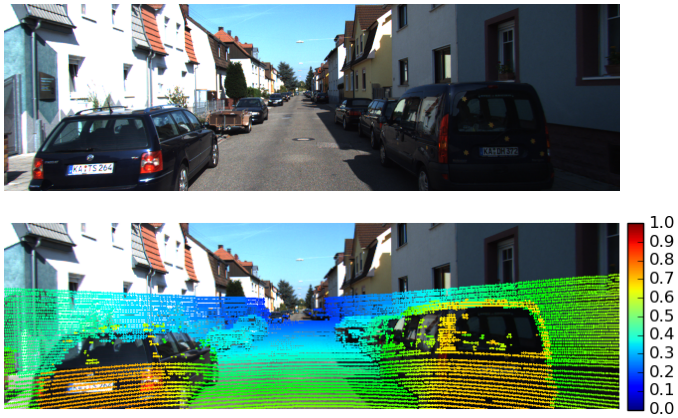

Figure 2: The top image is the original RGB image and the bottom one is the RGB^D image. Red color means the depth is shallow, and blue color means the depth is deep. Best viewed with color.

3) Proposals Generation: Based on the area of LiDAR point cloud $\{x, y \mid x \in [0, 70], y \in [-40, 40]\}$, a set of 3D prior anchor boxes are placed onto it. Each 3D prior anchor box is parameterized by the center (c_x, c_y, c_z) and the size (w, h, l) in meters. To generate the 3D prior anchor box grid, (c_x, c_y) pairs are sampled at an interval of 0.5 meters in the above area, and c_z is computed based on the LiDAR's height above the ground plane [14]. In this way, 89,600 anchors are generated in total. The size (w, h, l) is clustered from the ground truth of KITTI's training dataset [17]. For the car class, (w, l, h) takes the values of $(1.58, 3.51, 1.51)$ and $(1.65, 4.23, 1.55)$. For the pedestrian and cyclist class, (w, l, h) takes the values of $(0.63, 0.82, 1.77)$ and $(0.57, 1.77, 1.72)$, respectively. Specifically, each location has four anchors with two sizes and two orientations $\{0°, 90°\}$ for the car class.

Since the LiDAR point cloud is sparse, this causes a large number of empty anchors. After our statistics, there are about 5K to 25K anchors that contain LiDAR points. To speed up computation, the empty anchors are removed by computing an integral image over the point occupancy map [14] in both the training and testing stages. Based on the non-empty 3D anchors, the sampling method, as introduced in section III-B5, is employed to generate the 3D proposals. The 3D proposals are projected onto the BEV and RGB image plane to get the paired view-specific ROI crops, and the ROI crops are resized to a fixed size $N \times N \times C_r$. Note that the 3D proposal generation is completed in the preprocessing.

B. 3D Object Detection Networks

This section will introduce the 3D detection network in the order of use, as shown in Fig. 1.

1) Feature Encoder and Decoder: The feature detector comprises two sibling branches, one is for RGB image feature extraction, and the other one is for BEV feature extraction. Each branch consists of a feature encoder and feature decoder. VGG-16 [22] is chosen as the feature encoder. Our encoder, as shown in Fig. 3, differs from the VGG-16 encoder as follows:

- The first four convolution blocks are kept, and the fifth convolution block and the fully connected layers are removed.
- All convolution channel numbers are reduced to half of the original VGG-16.

For the decoder, three deconvolutional layers are used to obtain a high-resolution feature map. The high-resolution map offers more information for small objects. Same as FPN [23], lateral connections link the encoder and the decoder to build high-level semantic feature maps at all scales. Fig. 3 shows the details.

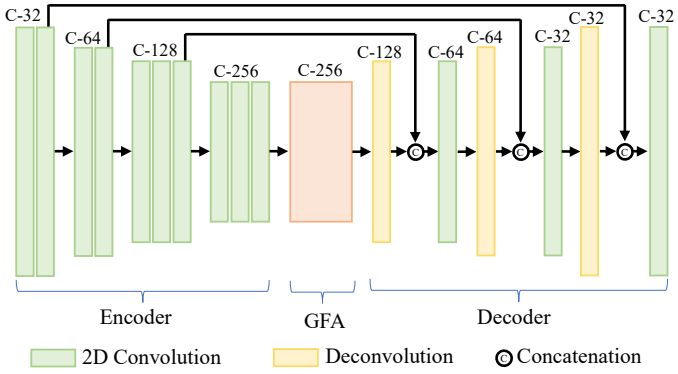

Figure 3: The feature extraction network, which includes three parts: encoder, GFA, and decoder. 'C-#' means the number of feature map channels. Best viewed with color.

2) Global Feature Attention: Inspired by DANet [18], the global feature attention (GFA) mechanism is proposed to integrate local features with their global dependencies adaptively. The GFA mechanism is much more efficient and also requires less computation as compared to DANet. The GFA mechanism includes two attention networks, as shown in Fig. 4. One is the position attention network (PAN), and the other one is the channel attention network (CAN).

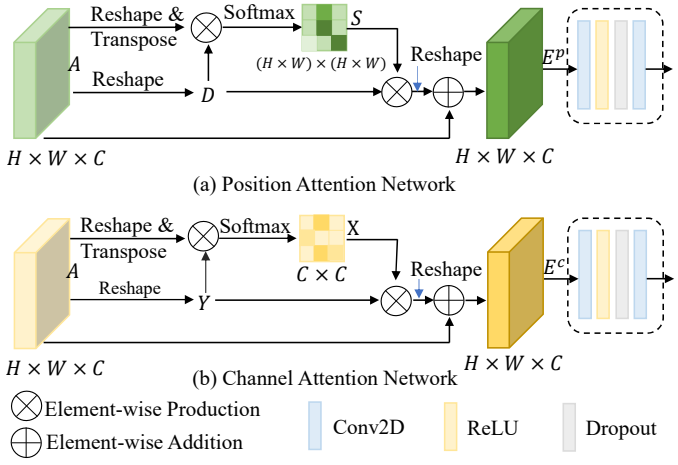

Figure 4: The global feature attention mechanism includes the position attention network, the channel attention network, and an auxiliary convolution block. Best viewed with color.

The PAN, as shown in Fig. 4a, focuses on modeling rich contextual relationships over local features. Given the local feature $\mathbf{A} \in \mathbb{R}^{H\times W\times C}$, first $\mathbf{A}$ is reshaped to $\mathbf{D} \in \mathbb{R}^{N\times C}$, where $N = W \times H$ is the number of pixels. Meanwhile, $\mathbf{A}$ is reshaped and transposed to $\mathbf{A}^{RT} \in \mathbb{R}^{C\times N}$. After that, matrix multiplication is employed between $\mathbf{D}$ and $\mathbf{A}^{RT}$ and also the softmax function is utilized to compute the spatial attention map $\mathbf{S} \in \mathbb{R}^{N\times N}$:

$$s_{ji} = \frac{exp(\mathbf{D}_i \cdot \mathbf{A}_j^{RT})}{\sum_{i=1}^{N} exp(\mathbf{D}_i \cdot \mathbf{A}_j^{RT})}, \tag{4}$$

where s_{ji} measures the i-th position's influence on the j-th position, and $i, j \in [1, W \times H]$. The more similar the feature representations of the two locations, the higher the correlation between them.

Second, matrix multiplication is utilized between $\mathbf{D}$ and $\mathbf{S}$, and the product is reshaped to $\mathbb{R}^{H\times W\times C}$. Finally, an element-wise sum operation is performed as follows:

$$\mathbf{E}_j^p = \alpha \sum_{i=1}^{N} (s_{ji}\mathbf{D}_i) + \mathbf{A}_j, \tag{5}$$

where α is a learnable parameter to re-weight the new generative feature map. From Equation 5, it can be concluded that the resulting feature $\mathbf{E}_j^p$ at each location j is a weighted sum of the feature at all locations and the original feature.

The CAN, as shown in Fig. 4b, is designed to exploit the interdependencies between channel maps, since each high-level channel map possesses different semantic responses. The CAN emphasizes the interdependence of the feature maps and boosts the feature representation of specific semantics. The whole reasoning process is the same as that of the PAN, and the difference is that $\mathbf{A}$ is reshaped to $\mathbf{Y} \in \mathbb{R}^{N\times C}$ firstly. The details are as follows:

$$x_{ji} = \frac{exp(\mathbf{A}_j^{RT} \cdot \mathbf{Y}_i)}{\sum_{i=1}^{C} exp(\mathbf{A}_j^{RT} \cdot \mathbf{Y}_i)}, \tag{6}$$

where x_{ji} measures the i-th channel's influence on the j-th channel, and $i, j \in [1, C]$.

$$\mathbf{E}_j^c = \beta \sum_{i=1}^{C} (\mathbf{Y}_i x_{ji}) + \mathbf{A}_j, \tag{7}$$

where β is a learnable parameter to re-weight the new generative feature map. The feature $\mathbf{E}_j^c$ at each channel j is a weighted sum of the feature at all channels and the original feature. This adds the benefit of enhancing the distinguishing ability of each channel.

Also, an auxiliary convolutional block is appended to each PAN and CAN, which includes a 2D convolutional layer, a ReLU activation function, a dropout layer (rate=0.5), and a 2D convolutional layer. The filter size of the two convolutional layers is 3×3. Note that the two outputs' shapes of 2D convolutions are the same as those of the PAN and the CAN. The auxiliary convolution block achieves 1.42% gains in 3D object detection.

Finally, an element-wise sum operation is performed for the outputs of the PAN and the CAN. Then the result is fed into the next stage.

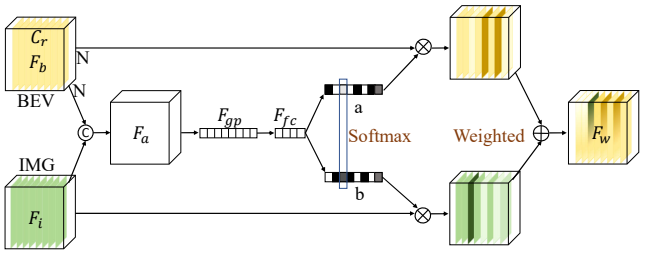

Figure 5: The region of interest attention (RA). It introduces a soft self-attention mechanism to weight each channel of the BEV and RGB image as pair and is a new fusion method besides the element-wise addition and concatenation operation.

3) Region of Interest Attention (RA): MV3D [4] and AVOD [14] only simply combine the ROI crops from both the RGB image proposals and the BEV proposals by an addition operation or a concatenate operation. This paper introduces the RA, as shown in Fig. 5, to weight RGB image ROIs and BEV ROIs by channel. Specifically, for any given BEV ROI $\mathbf{F}_b \in \mathcal{R}^{N\times N\times C_r}$ and an image ROI $\mathbf{F}_i \in \mathbb{R}^{N\times N\times C_r}$, first $\mathbf{F}_b$ and $\mathbf{F}_i$ are fused as $\mathbf{F}_a \in \mathbb{R}^{N\times N\times 2C_r}$ by a channel concatenation operation. Second, the $\mathbf{F}_a$ is fed into a global mean-pooling to output $\mathbf{F}_{gp} \in \mathbb{R}^{2C_r}$. Third, the $\mathbf{F}_{gp}$ is fed into a two-layer fully connected layers (FC). The first layer of the FC outputs $\mathbf{F}_1 \in \mathbb{R}^{d\times 1}$, where $d\ \{d \mid d = max(2C_r/r, 32)\}$ is a parameter based on a reduction ratio r to optimize the efficiency of this model. The second layer of the FC outputs $\mathbf{F}_{fc} \in \mathbb{R}^{2C_r\times 1}$. The $\mathbf{F}_{fc}$ is reshaped as $\mathbf{F}_2 \in \mathbb{R}^{2\times C_r}$. Then a softmax function is used for the $\mathbf{F}_2$ by channel.

$$\begin{aligned} a_c &= \frac{e^{A_c}}{e^{A_c} + e^{B_c}}, \\ b_c &= \frac{e^{B_c}}{e^{A_c} + e^{B_c}}, \end{aligned} \tag{8}$$

where $\mathbf{A}$, $\mathbf{B}$ are the first-row vector and the second-row vector of $\mathbf{F}_2$, $\mathbf{a}$ and $\mathbf{b}$ are the attention vector for $\mathbf{F}_b$ and $\mathbf{F}_i$, respectively, and $c \in [1, C_r]$ is the channel number of each ROI.

Finally, the fusion of the paired ROIs with the RA mechanism is as follows:

$$\begin{aligned} \mathbf{F}_w^c &= a_c \cdot \mathbf{F}_b^c + b_c \cdot \mathbf{F}_i^c, \\ w.r.t &\quad a_c + b_c = 1, \end{aligned} \tag{9}$$

where $\mathbf{F}_w$ is the weighted sum between $\mathbf{F}_b$ and $\mathbf{F}_i$. Compared with the previous fusion methods, the RA is an attention mechanism for the fusion of view-specific ROIs.

4) Loss Function: The equal-length feature $\mathbf{F}_w$ is fed into a three-layer FCs (2048, 2048, 2048) to deeply merge and then the fused tensor is fed into a 3D detection head with three parallel branches: classification (class no.), box regression (10), and angle regression (1). Note that each branch only has one FC layer and the number in each bracket means the dimension of FC. MV3D [4] encodes a 3D box as eight corners and regresses them. However, it does not consider the physical constraints of a 3D box. To reduce the redundancy and keep the physical constraints, AVOD [14] encodes a 3D box with four corners and two heights. Different from AVOD, our method proposes a plane-based 3D bounding box with an 11-dimensional vector $(x_1 \cdots x_4, y_1 \cdots y_4, h_1, h_2, \theta)$. The corresponding regression residuals between the 3D anchors and ground truth are defined as follows:

$$\begin{aligned} \Delta x &= \frac{x_c^g - x_c^a}{d^a}, \quad \Delta y = \frac{y_c^g - y_c^a}{d^a}, \\ \Delta h &= log(\frac{h^g}{h^a}), \quad \Delta\theta = sin(\theta^g - \theta^a), \end{aligned} \tag{10}$$

where $d^a = \sqrt{(x_2 - x_1)^2 + (y_4 - y_1)^2}$ is the diagonal of the base of the anchor box. The localization loss function and the angle loss function are as follows:

$$L_{box} = \sum_b Smooth_{L1}(\Delta b), \tag{11}$$

$$L_{angle} = \sum_\theta Smooth_{L1}(\Delta\theta), \tag{12}$$

where the $b \in (x_1 \cdots x_4, y_1 \cdots y_4, h_1, h_2)$ and $Smooth_{L1}$ is the smooth L1 loss function in the Fast R-CNN [24].

For the object classification loss, the focal loss [25] is used:

$$L_{cls} = -\alpha_a(1 - p^a)^\gamma log(p^a), \tag{13}$$

where p^a is the class probability of an anchor, $\alpha = 0.25$, and $\gamma = 2$. The total loss can be formulated as follows:

$$Loss = \frac{1}{N_{pos}}(\beta_1 L_{box} + \beta_2 L_{cls} + \beta_3 L_{angle}), \tag{14}$$

where N_{pos} is the number of positive anchors and $\beta_1 = 7.0$, $\beta_2 = 5.0$, and $\beta_3 = 1.0$. For the car class, an anchor is defined as positive if it has a 2D IoU greater than 0.60 (pedestrian/cyclist is 0.3) with its paired ground truth. If it has a 2D IoU less than 0.55 (pedestrian/cyclist is 0.3), the anchor is labeled as negative. The other anchors are ignored when computing the loss.

5) Training and Inferring: In training, the proposed model is trained using mini-batches containing 16,384 proposals (positive and negative ratio 1:1) for one frame.

In inferring (validation and testing), the non-empty anchors will be directly used as the proposals to crop and resize the view-specific ROIs. 2D non-maximum suppression (NMS) at an IoU threshold of 0.01 on the BEV boxes is utilized to remove the redundant 3D proposals, and the top 15 3D predictions are kept.

IV. Experiments

A. Dataset and Metric

The proposed model is trained and evaluated on the KITTI dataset [17]. The KITTI object dataset possesses 7,481 training frames and 7,518 testing frames. Each frame is comprised of a point cloud, stereo RGB images, and calibration data. In this research, only a point cloud and the left image with their calibration data are used. The KITTI includes seven classes: car, van, truck, pedestrian, person (sitting), cyclist, and tram. Because the number of other categories is small, the car, pedestrian, cyclist classes are used for comparison. MV3D [4] is the

Table I: Comparison with state-of-the-art methods. All methods are compared using the 3 difficulties: easy (E), moderate (M), and hard (H). For easy understanding, the top two numbers are highlighted in bold and italic for each column and the second best is shown in blue. All methods accept RGB images and point clouds as input. ”-” means that the data can not be found.

Method	Pub.Year	Stage (s)	Number of Parameters	Runtime (ms)	3D (%)				BEV (%)			
					E	M	H	mAP	E	M	H	mAP
MV3D [4]	2017	Two	-	360	71.29	62.68	56.56	63.51	86.55	78.10	76.67	80.44
F-PointNet [26]	2017		-	170	83.76	70.92	63.65	72.78	88.16	84.02	76.44	82.87
PC-CNN [27]	2018		-	500	57.63	51.74	51.39	53.59	83.61	77.36	69.61	76.86
AVOD [14]	2018		38,073,528	80	83.11	74.02	67.84	74.99	-	-	-	-
AVOD-FPN [14]	2018		-	100	84.41	74.44	68.65	75.83	89.37	86.09	79.13	84.86
MVX-Net [28]	2019		-	150	85.50	73.30	67.40	75.40	89.50	84.90	79.00	84.47
MCF3D [21]	2019	Three	-	160	84.11	75.19	74.23	77.84	88.82	86.11	79.31	84.75
AVOD-SSD [15]	2018	One	13,399,918	90	82.36	72.92	67.07	74.12	89.00	85.08	78.91	84.33
Cont-Fuse [29]	2019		-	60	**86.32**	73.25	67.81	75.79	**95.44**	**87.34**	82.43	***88.40***
Complex-Retina [16]	2019		-	90	78.62	72.77	67.21	72.87	89.01	84.69	78.71	84.14
Proposed	-		20,575,616	110	85.12	**76.23**	**74.46**	***78.60***	89.64	86.23	***85.60***	87.16

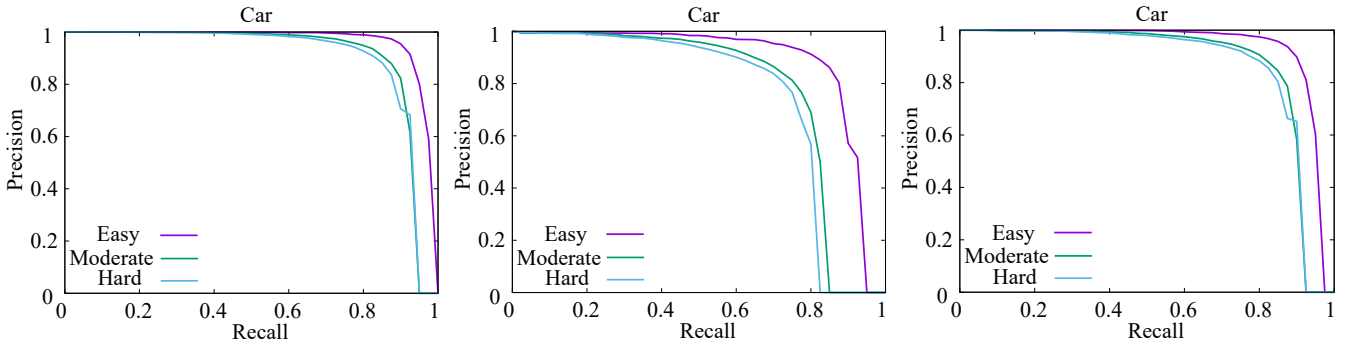

Figure 6: Visualization of the Precision-Recall curve for the car class. From left to right are the curves of 2D, 3D, and BEV. In each curve, each color line denotes one difficulty of the car class. The curves are drawn according to the best-proposed model.

pioneer in the multi-modal 3D object detection and divides the training dataset into two subsets (ratio=3,712:3,769): the training subset and the validation subset. To compare fairly with its results, among the subsequent articles employ the same criteria as MV3D. Two models are trained for the car class and pedestrian/cyclist classes, respectively, since the training dataset has an unbalanced amount of training data for the car class and the pedestrian/cyclist classes.

KITTI's object detection metric is defined as 11-point Average Precision (AP). Intersection-over-Union (IoU) is the generic evaluation criterion for object detection. In the evaluation of 2D, 3D, and BEV detection, IoU is at the threshold of 0.7 for the car class and 0.5 for the pedestrian/cyclist classes. According to the bounding box height, truncation levels, and occlusion classes of objects, KITTI groups all objects into three difficulty classes: easy (E), moderate (M), and hard (H). In the evaluation, prediction results are evaluated by the program that comes with the KITTI dataset, and the program outputs the three results for the easy, moderate, and hard class, respectively.

B. Implementation Details

Since the 2D RGB camera images are of different sizes, the images are center-cropped into a uniform size of 1200×360. Each point cloud is voxelized as a 700×800×6 BEV pseudo image. The proposed model is implemented using TensorFlow on one NVIDIA 1080 Ti GPU with a batch size of 1. Adam is the optimizer. Our model is trained for a total of 120K iterations with the initial learning rate of 0.0001, and decayed by 0.1 at 60K iterations and 90K iterations. The whole training process takes only 12 hours, and the proposed model is evaluated from 100K iterations to 120K iterations every 5K iterations.

C. Comparison with State-of-the-Art Methods

All experiments are evaluated on the KITTI validation subset. It should be noted that no currently published one-stage method publicly provides results on the pedestrian/cyclist classes for 3D object detection based on LiDAR and RGB images. Hence, the comparison is only for the car class in Table I. All methods are grouped into three sets: one-stage methods, two-stage methods, and three-stage methods based on LiDAR and image. For a fair comparison, this article only compares with the state-of-the-art methods in the past five years that use LiDAR and images as input. Most of the methods only disclose 3D and BEV performance. Thus, 2D detection performance is not listed in Table I.

In the 3D object detection, our proposed method outperforms all state-of-the-art methods with noticeable margins except for a slightly lower score than Cont-Fuse [29] in the 'E' column. Specifically, our proposed method achieves 1.04% gains on the most important 'M' column compared

with the second-best performing method, and also outperforms the second-ranked MCF3D [21] by 0.76% on the mean average precision (mAP). In BEV object detection, the overall performance of Cont-Fuse [29] is better than ours, but our method outperforms Cont-Fuse [29] in the 'H' column with a big margin of 3.17%. For the inference time, the proposed method still achieves a comparable speed by taking the high precision into account. The precision-recall curves of the best-proposed model are shown in Fig. 6.

To further understand the proposed model, Table II shows the number of parameters for each component in the proposed framework.

Table II: The number of parameters for each component.

Component	Total	Base Network	GFA	RA
Number of Parameters	20,575,616	15,852,928	4,718,592	4,096

Since the state-of-the-art methods, shown in Table I, do not provide the pedestrian/cyclist class results, the pedestrian/bicycle results cannot be compared with other methods. This paper provides the results of pedestrian/bicycle for reference in Table III.

Table III: The 3D and BEV object detection accuracy for the pedestrian and cyclist. To ensure the best visual effect, the table does not show the 'Easy (E)' results.

Class	2D (%)		3D (%)		BEV (%)	
	M	H	M	H	M	H
Pedestrian	58.44	52.11	65.39	59.29	65.47	59.38
Cyclist	43.58	38.97	43.23	38.31	43.27	38.37

D. Ablation Study

In this section, massive experiments are utilized for analysis and ablation of the proposed model on the KITTI validation subset. To ensure the best visual effect, all tables do not show the 'Easy (E)' results.

Fig. 7 shows the qualitative results of 3D object detection. To more directly compare the prediction results with the ground truth values, the green and red color represent the ground truth and prediction, respectively. It can be seen that our method can detect and localize 3D objects well. Compared with the pedestrian/bicycle results, the performance of the car detection is much better due to the larger size of the cars.

Table IV: The effect of global feature attention. 'Where use it?' means which feature encoder uses the GFA. The third and the fourth rows show the effect of the auxiliary convolutional block. The best performance is highlighted in bold for the 3D column.

Where use it?	2D (%)		3D (%)		BEV (%)	
	M	H	M	H	M	H
RGB Image	87.85	86.82	74.71	68.46	85.61	79.41
BEV	87.87	86.75	73.65	67.94	85.56	79.31
RGB Image+BEV	88.18	86.53	**75.87**	**73.98**	85.35	85.44
w/o Conv.	87.61	86.27	74.45	68.42	85.08	85.25

1) Effect of Global Feature Attention: Table IV shows how does the GFA affects 3D performance. Two feature encoders are used to extract the BEV features and the RGB image features, respectively. This paper explores the impact of the GFA on RGB image features and BEV features, respectively. Relative to the BEV features (the second row), the GFA is more helpful for RGB image feature extraction (the first row) with 1.06% gains in the moderate class. In addition, one more experiment is employed to analyze the effect of the auxiliary convolutional block in section III-B2. Compared with the results in the last row, the auxiliary convolutional block (the third row) achieves a 1.42% and 5.56% gains in the moderate and hard classes of 3D performance, respectively. The reason is that the convolutional layer further merges the fused features.

2) Effect of Diversity Combination: Table V shows the combinations of different proposed methods and their corresponding performances. Among the four approaches, in terms of a single method, FPN contributes a maximum of 0.95% in 3D performance, because the FPN integrates the high-level semantic feature maps at all scales. Besides, RGB^D images contribute to BEV detection with a 0.89% increase. Compared with the RA (the third row), the GFA (the fourth row) is more helpful to boost 3D performance due to it enhances the global feature representations. Each proposed method only slightly improves the detection performance, however, the proposed framework greatly boosts the detection accuracy based on all proposed methods. Compared with the baseline (the first row), the best-proposed combination (the last row) achieves 1.48%, 3.31%, and 1.15% gains in 2D, 3D, and BEV, respectively. As can be seen, the proposed methods are quite useful for boosting 3D performance.

Table V: The effect of different proposed methods. 2D, 3D, and BEV performance are compared on the 'Moderate' difficulty for the car class. *FPN* denotes the feature pyramid network. The best performance is highlighted in bold for each column.

Method Combinations				2D (%)	3D (%)	BEV (%)
FPN	RA	GFA	RGB^D			
				86.99	72.92	85.08
√				87.42	73.87	85.36
	√			87.67	73.22	85.30
		√		87.20	73.34	85.89
√	√			87.43	74.55	85.52
√		√		87.78	73.11	85.20
	√	√		87.55	74.07	85.17
√	√	√		88.18	75.87	85.35
√	√	√	√	**88.47**	**76.23**	**86.23**

3) Effect of RGB^D image: Based on the best combination in Table V, three sets of experiments are used to study the effect of RGB, RGB-I, and RGB^D for the car class, respectively. Table VI shows that RGB^D surpasses the other two images in all aspects. Compared with the RGB image, the RGB-I has very limited performance improvement for the model, and even worse than the RGB image in most performances. In terms of 3D performance, the RGB^D achieves 0.36%, 0.48% gains in the moderate, and hard difficulty, respectively. The experimental results verify that the RGB^D indeed preserves more 3D information of a point cloud.

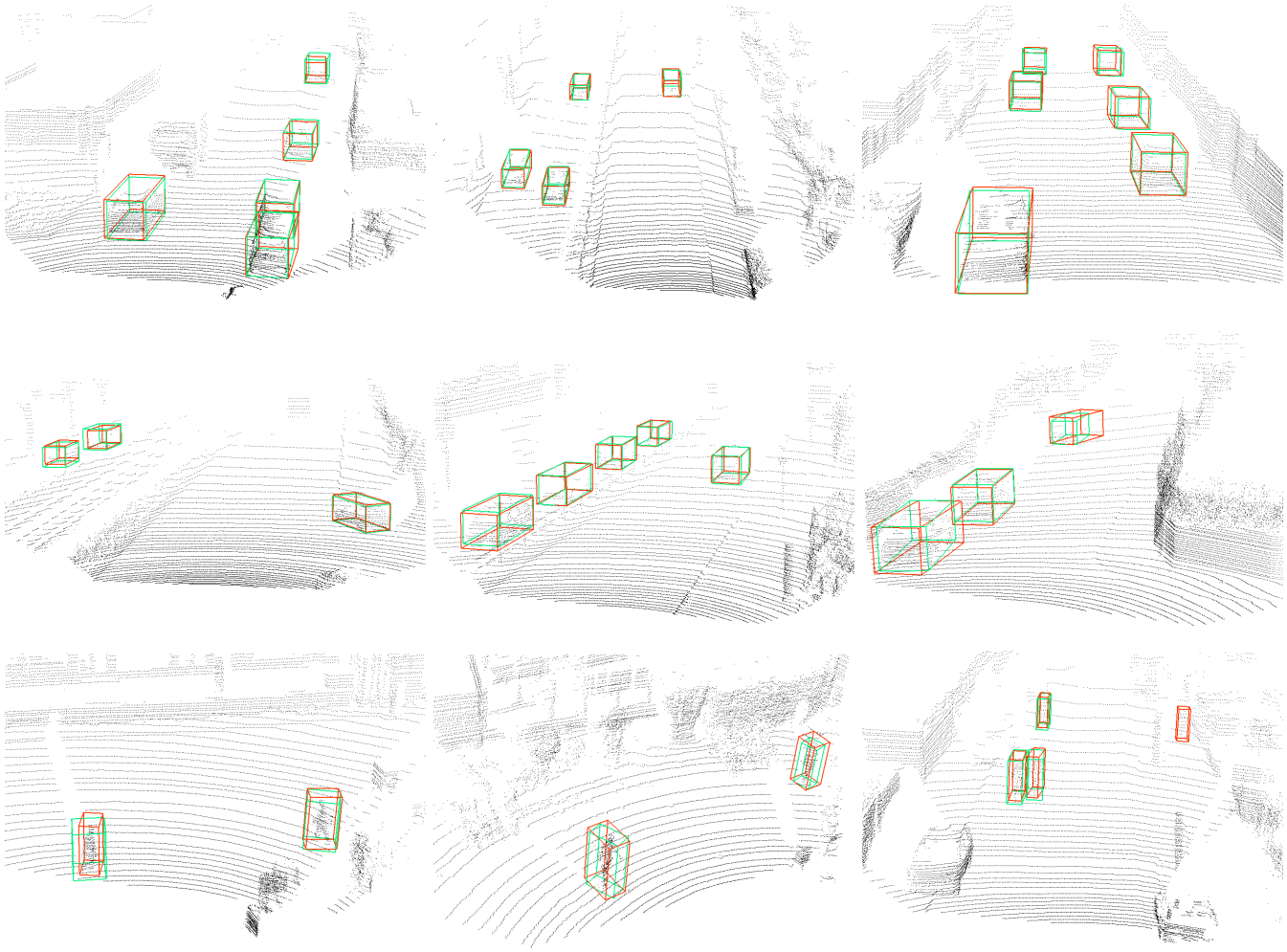

Figure 7: Visualizations of 3D detection results on point clouds. The green color denotes the ground truth and the red color represents the prediction. The first two rows are the results of the car class. The last row is the results of cyclist and pedestrian classes.

Table VI: The comparison of RGB, RGB-I, and RGB^D. The best performance is highlighted in bold for the 3D column.

Class	2D (%)		3D (%)		BEV (%)	
	M	H	M	H	M	H
RGB	88.18	86.53	75.87	73.98	85.35	85.44
RGB-I [22]	88.24	86.94	75.46	68.94	85.48	79.38
RGB^D	**88.47**	**86.98**	**76.23**	**74.46**	**86.23**	**85.60**

Table VII: The effect of input size for the RA. The best performance is highlighted in bold for the 3D column.

RA size	2D (%)		3D (%)		BEV (%)	
	M	H	M	H	M	H
5×5	87.68	86.40	73.10	67.57	85.37	79.30
7×7	87.43	86.84	**74.55**	**68.44**	85.52	79.37
9×9	87.49	86.31	74.28	68.05	85.47	79.44

4) Effect of Region-of-Interest Attention: The RA is a soft attention network for the fusion of paired view-specific ROIs. First, the input size of the RA is analyzed. Taking the efficiency factor into account, three kinds of input sizes are utilized for analysis. The experimental results are shown in Table VII. As can be seen, the input size 7×7 is the best candidate for efficiency and 3D detection accuracy. This may be due to the three reasons: (1) most of the proposals with an approached size, 7×7; (2) if the cropped feature map based on the proposal is resized to a small size, such as 5×5, and the important features may be lost; (3) if the cropped feature map is resized to a large size, the important feature may be diluted.

MV3D [4] employs the concatenation operation to fuse view-specific ROIs. AVOD [14] exploits the addition operation for fusion. The concatenation as comparison to the addition operation benefits to boost performance, but the performance improvement is a little. Differ from these two operations, the proposed RA pays more attention to the important features through learning. In 3D performance of moderate difficulty class, RA achieves 0.68%, 0.42% gains as compared to the Addition and Concatenation operation, respectively, as shown in Table VIII.

Table VIII: The effect of the different fusion methods. The best performance is highlighted in bold for the 3D column.

Fusion Method	2D (%)		3D (%)		BEV (%)	
	M	H	M	H	M	H
Addition [14]	87.42	86.63	73.87	68.25	85.36	79.20
Concatenation [4]	87.85	86.90	74.13	67.98	85.79	79.22
RA	87.43	86.84	**74.55**	**68.44**	85.52	79.37

Tables III-VIII show that the 2D IoU metric is more helpful for 2D and BEV performance. It ignores the impact of proposals' height during the stage of proposal generation, and it is a drawback for 3D object detection accuracy. Note that the proposed method achieves the best performance in 2D and BEV, hence, the 3D performance is only compared in Tables III-VIII.

V. Conclusion

This paper proposes a one-stage 3D object detection framework based on LiDAR and Camera, which benefits from the three attention mechanisms to boost 3D detection accuracy. First, the height attention mechanism is introduced into the input RGB images to generate the RGB^D images. Second, the global feature attention mechanism is utilized for both the RGB image and the BEV branches at the feature extraction stage. It benefits by capturing the discriminative features from both the channels and spatial dimensions. Finally, the region-of-interest attention mechanism is employed to fuse the paired view-specific ROIs. Our proposed method greatly improves the 3D object detection performance, and it outperforms all state-of-the-art methods based on LiDAR and Camera in 3D object detection.

In the future, the way to generate BEV images can be replaced with a learnable feature generator like SECOND [30]. Additionally, the method of anchor generation can be changed from anchor-based to anchor-free. In this way, with the advantages of RGB images and LiDAR point clouds, 3D object detection based on LiDAR and camera can achieve better performance than LiDAR-based methods.

References

[1] X. Han, H. Liu, F. Sun, and X. Zhang, "Active object detection with multistep action prediction using deep q-network," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3723–3731, June 2019.

[2] N. Lv, C. Chen, T. Qiu, and A. K. Sangaiah, "Deep learning and superpixel feature extraction based on contractive autoencoder for change detection in sar images," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 12, pp. 5530–5538, Dec 2018.

[3] V. Hoang, D. Huang, and K. Jo, "3d facial landmarks detection for intelligent video systems," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2020.

[4] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6526–6534.

[5] B. Li, T. Zhang, and T. Xia, "Vehicle detection from 3d lidar using fully convolutional network," in *Robotics: Science and Systems XII*, 2016.

[6] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4490–4499.

[7] L. Wen and K.-H. Jo, "Fully convolutional neural networks for 3d vehicle detection based on point clouds," in *Intelligent Computing Theories and Application*, 2019, pp. 592–601.

[8] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 77–85.

[9] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 5099–5108.

[10] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, "Monocular 3d object detection for autonomous driving," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2147–2156.

[11] B. Xu and Z. Chen, "Multi-level fusion based 3d object detection from monocular images," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2345–2353.

[12] T. He and S. Soatto, "Mono3d++: Monocular 3d vehicle detection with two-scale 3d hypotheses and task priors," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 8409–8416, 07 2019.

[13] P. Li, X. Chen, and S. Shen, "Stereo r-cnn based 3d object detection for autonomous driving," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 7636–7644.

[14] 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)*, 2018, pp. 1–8.

[15] M. Mozifian. (2018) Real-time 3d object detection for autonomous driving.

[16] L. M, H. Y, Z. N, and Q. Q, "One-stage multi-sensor data fusion convolutional neural network for 3d object detection," *Sensors*, 2019.

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

[18] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3141–3149.

[19] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6450–6458.

[20] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Computer Vision – ECCV 2018*, 2018, pp. 3–19.

[21] J. Wang, M. Zhu, D. Sun, B. Wang, W. Gao, and H. Wei, "Mcf3d: Multi-stage complementary fusion for multi-sensor 3d object detection," *IEEE Access*, vol. 7, pp. 90 801–90 814, 2019.

[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[23] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 936–944.

[24] R. Girshick, "Fast r-cnn," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1440–1448.

[25] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318–327, 2020.

[26] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3d object detection from rgb-d data," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 918–927.

[27] X. Du, M. H. Ang, S. Karaman, and D. Rus, "A general pipeline for 3d detection of vehicles," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 3194–3200.

[28] V. A. Sindagi, Y. Zhou, and O. Tuzel, "Mvx-net: Multimodal voxelnet for 3d object detection," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 7276–7282.

[29] M. Liang, B. Yang, S. Wang, and R. Urtasun, "Deep continuous fusion for multi-sensor 3d object detection," in *The European Conference on Computer Vision (ECCV)*, September 2018.

[30] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, 3337, 2018.

Lihua Wen (S'15) received a bachelor's degree in vehicle engineering from the School of Automotive Engineering, Shanghai University of Engineering Science, Shanghai, in 2015. He is currently working toward the Ph.D. degree in electrical and computer engineering with the Graduate School of Electrical Engineering, University of Ulsan, Ulsan, South Korea. His research interests include image processing, pattern recognition, computer vision, machine learning, and 3D object detection for intelligent vehicles.

Since 2013, he worked as an engineer in Shanghai Automobile Gear Works, Commercial Aircraft Corporation of China, Ltd., and Hyundai Commercial Vehicle (China) Co., Ltd.

Kang-Hyun Jo (Senior Member, IEEE) received a Ph.D. degree in computer controlled machinery from Osaka University, Osaka, Japan, in 1997. After a year of experience with ETRI as a Postdoctoral Research Fellow, he joined the School of Electrical Engineering, University of Ulsan, Ulsan, South Korea. He is currently serving as the Faculty Dean with the School of Electrical Engineering, University of Ulsan, Ulsan, South Korea. His research interests include computer vision, robotics, autonomous vehicle, and ambient intelligence.

Dr. Jo has served as the Director or an AdCom Member with the Institute of Control, Robotics and Systems, The Society of Instrument and Control Engineers, and the IEEE IES Technical Committee on Human Factors Chair, AdCom Member, and the Secretary until 2019. He has also been involved in organizing many international conferences, such as International Workshop on Frontiers of Computer Vision, International Conference on Intelligent Computation, International Conference on Industrial Technology, International Conference on Human System Interactions, and the Annual Conference of the IEEE Industrial Electronics Society. He is currently an Editorial Board Member for international journals, such as the International Journal of Control, Automation, and Systems and Transactions on Computational Collective Intelligence.