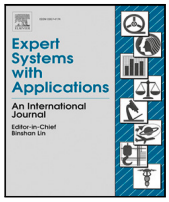
Expert Systems With Applications 242 (2024) 122716

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



MMAF-Net: Multi-view multi-stage adaptive fusion for multi-sensor 3D object detection

Wensheng Zhang [a], Hongli Shi [a], Yunche Zhao [a], Zhenan Feng [b,*], Ruggiero Lovreglio [b]

[a] School of Traffic and Transportation, Shijiazhuang Tiedao University, Shijiazhuang 050043, China
[b] School of Built Environment, Massey University, Auckland 0632, New Zealand

ARTICLE INFO

Keywords:
3D object detection
Multi-sensor fusion
Attention mechanism
Joint regression loss
Autonomous driving

ABSTRACT

In this paper, we propose a 3D object detection method called MMAF-Net that is based on the multi-view and multi-stage adaptive fusion of RGB images and LiDAR point cloud data. This is an end-to-end architecture, which combines the characteristics of RGB images, the front view of point clouds based on reflection intensity, and the bird's eye view of point clouds. It also adopts a multi-stage fusion approach of "data-level fusion + feature-level fusion" to fully exploit the strength of multimodal information. Our proposed method addresses key challenges found in current 3D object detection methods for autonomous driving, including insufficient feature extraction from multimodal data, rudimentary fusion techniques, and sensitivity to distance and occlusion. To ensure the comprehensive integration of multimodal information, we present a series of targeted fusion methods. Firstly, we propose a novel input form that encodes dense point cloud reflectivity information into the image to enhance its representational power. Secondly, we design the Region Attention Adaptive Fusion module utilizing an attention mechanism to guide the network in adaptively adjusting the importance of different features. Finally, we extend the 2D DIOU (Distance Intersection over Union) loss function to 3D and develop a joint regression loss based on 3D_DIOU and SmoothL1 to optimize the similarity between detected and ground truth boxes. The experimental results on the KITTI dataset demonstrate that MMAF-Net effectively addresses the challenges posed by highly obscured or crowded scenes while maintaining real-time performance and improving the detection accuracy of smaller and more difficult objects that are occluded at far distances.

1. Introduction

The rapid development of computer vision technology has led to significant progress in autonomous driving applications (Qian et al., 2022). Object detection on the roadway is an essential part of the visual awareness system for autonomous driving (Ranft & Stiller, 2016). Deep learning-based 2D object detection can only classify and localize objects in two dimensions, which does not fully meet the needs of autonomous driving (Mukhtar et al., 2015). In contrast, 3D object detection provides geometric information such as the size and directional angle in 3D space. This information can be used directly to measure the distance between the autonomous vehicle and key objects, thus improving the safety of autonomous vehicles (Ghasemieh & Kashef, 2022).

LiDAR and camera are two of the most widely used sensors in autonomous driving (Alaba & Ball, 2023). The point clouds scanned by LiDAR can provide precise depth information and geometric structure (Arnold et al., 2019). However, the inherent characteristics of LiDAR often result in point cloud data that is discrete and disordered (Mao et al., 2021). Sparse data distribution is particularly challenging for recognizing and locating distant, small, or obscured objects. Cameras are also susceptible to extreme weather conditions, but the image data they produce can offer rich semantic clues and high-resolution appearance information (Chang & Chen, 2018). Fusing these two kinds of information using a multimodal method can effectively combine their advantages and improve the performance of 3D object detection tasks (Chen et al., 2023).

Effective fusion of heterogeneous data is essential for multi-sensor fusion in 3D object detection, owing to the differences in feature representation across modalities. Depending on the stage of fusion, fusion strategies can be classified into three types: data-level fusion, feature-level fusion, and decision-level fusion (Zhang et al., 2020). Data-level fusion (Dou et al., 2019) combines the raw data or preprocessed data features, but this approach produces redundant information and requires heavy computation during data storage and processing due to variations in raw data quality. Feature-level fusion (Liang et al.,

* Corresponding author.
E-mail addresses: zws@stdu.edu.cn (W. Zhang), shihongli0110@163.com (H. Shi), zhaoyunche1998@163.com (Y. Zhao), z.feng1@massey.ac.nz (Z. Feng), r.lovreglio@massey.ac.nz (R. Lovreglio).

https://doi.org/10.1016/j.eswa.2023.122716
Received 13 August 2023; Received in revised form 22 September 2023; Accepted 23 November 2023
Available online 30 November 2023

2019) tends to conduct a comprehensive analysis and fusion process after extracting the original data features. Although this design plays an important role in ensuring information consistency and reducing data redundancy, it often lacks sensitivity to correlations between multi-sensor features and is prone to coarse fusion. Decision-level fusion (Pang et al., 2020) performs joint analysis and inference of independent sensors, providing good interference immunity and fault tolerance; however, it overlooks potential information generated from sensor fusion. Recent approaches (Deng & Czarnecki, 2019; Ku et al., 2018; Wang et al., 2018) project point clouds into the bird's eye view (BEV) plane and later fuse features with RGB images. However, some of the existing approaches, such as MV3D (Chen et al., 2017) and AVOD (Ku et al., 2018), concatenate or average different view features without considering the varying importance of different modal information in 3D object detection. Moreover, the most commonly used metric for bounding box generation, Intersection over Union (IoU), has limitations. Reference (Zheng et al., 2020) proposes the Distance Intersection over Union (DIOU) loss function to optimize IoU and improve convergence, but it is only applicable to 2D cases. The 3D bounding box contains more parameter information, which makes it more difficult to regress the bounding box. Existing 3D object detection methods (Chen, Li, & Zhao, 2022; Guo et al., 2021; Wang et al., 2019) typically use the SmoothL1 loss function for bounding box regression, but the SmoothL1 loss function regresses bounding points independently without accounting for point-to-point constraints, potentially misaligning with the object detection evaluation criteria (Girshick, 2015; Liu et al., 2016).

In this paper, we propose MMAF-Net, a multi-sensor 3D object detection method that addresses the challenges mentioned above. MMAF-Net is based on multi-view multi-stage adaptive fusion, which combines the characteristics of RGB images, the front view of point clouds based on reflection intensity, and the BEV of point clouds. Our approach utilizes a multi-stage fusion strategy that combines data-level fusion and feature-level fusion to maximize the characterization capability of multimodal data. Additionally, we introduce a set of guided optimization methods to further enhance the performance of 3D object detection. First, in the data preprocessing phase, we match and fuse images and point clouds based on their spatial coordinate transformation relationships. This fusion produces a new data input form called RGB-Dense-Reflectivity (RGB-DR). Second, we construct the Region Attention Adaptive Fusion (RAAF) module, which applies the attention mechanism in the object detection stage and adaptively readjusts the feature weights. In terms of the loss function, a joint regression loss function based on 3D_DIOU and SmoothL1 is proposed in this paper. The joint regression loss, together with classification loss and angle loss, forms the multi-task loss function for MMAF-Net. In summary, the key contributions of our work are as follows:

(1) We design a point cloud densifying method that combines Delaunay linear interpolation and bootstrap filtering in the data preprocessing stage, which encodes the dense point cloud reflection intensity into the image channel to generate a new data input form called RGB-DR. This enables the network to learn more feature information and enhances its robustness.

(2) We incorporate the attention mechanism in the feature fusion and construct the RAAF module to adaptively adjust the weights of different modal features, thus mitigating the problem of coarse feature fusion.

(3) We extend the DIOU loss function to 3D space and design a joint regression loss that combines 3D_DIOU and SmoothL1 to optimize the original loss function. This unifies the evaluation criteria and further optimizes the similarity between the detected box and the ground truth.

(4) We demonstrate, based on evaluations on the KITTI benchmark, that our MMAF-Net improves the detection performance of small and distant objects as well as obscured objects while ensuring real-time detection.

2. Related works

From the perspective of input sensors and information sources, existing methods can be divided into three categories: camera-based methods (Chen et al., 2016; Li et al., 2019), LiDAR-based methods (Charles et al., 2017; Shi et al., 2019; Yan et al., 2018; Yang et al., 2020), and LiDAR-camera methods (Chen et al., 2017; Guo et al., 2021; Ku et al., 2018; Yoo et al., 2020; Zhu et al., 2021). In this section, we will briefly review the development of 3D object detection and the challenging problems of inadequate multimodal fusion and inefficient feature fusion in object detection.

2.1. Camera-based methods

The camera-based methods aim to perform 3D object detection tasks using a monocular or stereo camera. While monocular images can provide rich semantic information at a low cost, they lack depth information. To overcome this limitation, many works have combined geometric features and depth estimation networks to achieve 3D object detection (Chen et al., 2016; Chen, Liu, & Zhao, 2022; Mousavian et al., 2017). For instance, Mono3D (Chen et al., 2016) utilizes priori information to generate several proposals, which are scored using semantics and context, resulting in more accurate 3D detection results. Deep3DBox (Mousavian et al., 2017) combines 2D object detection results with geometric constraints and employs a deep neural network to estimate the orientation and size of 3D detection boxes. M3DGAF (Chen, Liu, & Zhao, 2022) designs a geometry-appearance perception module for obtaining appearance features, which can be used to estimate reliable orientations. To further improve the detection performance of camera-based methods, researchers use stereo cameras that provide more structured geometric information for effective 3D object detection (Chen et al., 2020; Li et al., 2019). However, due to their complex configuration and calibration processes, as well as the impact of multiple factors in real-world traffic environments, camera-based methods have not yet achieved satisfactory performance in dealing with traffic situations.

2.2. LiDAR-based methods

The LiDAR-based methods can be broadly categorized as projection-based, point-based, and voxel-based methods. The projection-based methods (Mohapatra et al., 2021; Yang et al., 2018) estimate 3D bounding box parameters by projecting the original point cloud data onto a 2D viewpoint. This approach not only neglects the size and position of the object but also misses object geometry information.

PointNet (Charles et al., 2017) constructs spherical areas to extract point-based features. However, it cannot extract local features effectively. PointNet++ (Qi et al., 2017) overcomes this shortcoming by iteratively extracting features from local areas of the point set to improve detection accuracy. As the first two-stage 3D object detection method using only the raw point cloud as input, PointRCNN (Shi et al., 2019) directly segments point clouds to obtain foreground points, and then fuses semantic features and local spatial features. 3DSSD (Yang et al., 2020) proposes a one-stage pipeline and uses fusion sampling methods to speed up the inference process.

For the voxel-based methods, the most representative work is VoxelNet (Zhou & Tuzel, 2018) which divides the irregular point clouds into regular voxel grids via a 3D convolutional neural network and uses a voxel feature encoding layer to extract local features from non-empty voxels. SECOND (Yan et al., 2018) introduces a sparse 3D convolution neural network to improve detection speed and memory usage. Compared with RGB images, point clouds can only provide low-resolution appearance information, resulting in LiDAR-based methods having poor performance and robustness when detecting small objects at a distance.

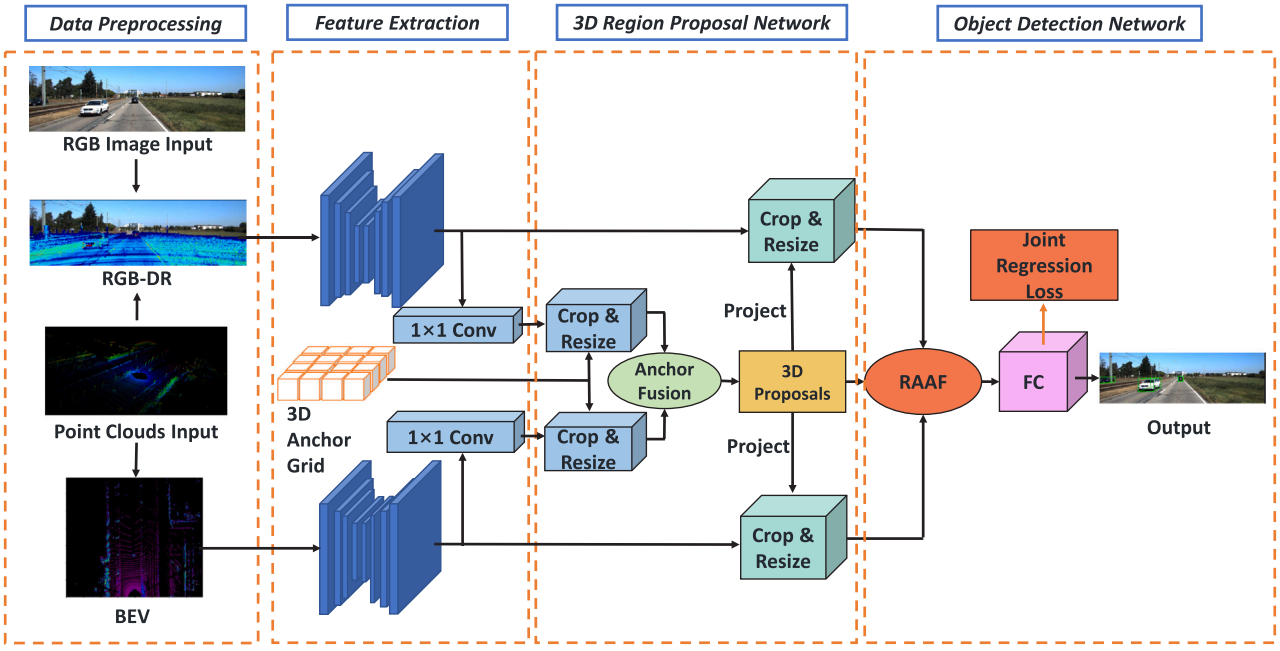


Fig. 1. The architecture of MMAF-Net. The network adopts a two-stage detection framework consisting of four parts. We design a series of fusion methods aimed at different stages to improve the performance of 3D object detection, including the RGB-DR representation, the RAAF module, and the joint regression loss function.

2.3. LiDAR-camera methods

The fusion of camera and LiDAR information is a promising trend for the future of 3D object detection as it allows the fused features to have both semantic and geometric information. Many works have been dedicated to exploring fusion methods of data from cameras and LiDAR (Chen et al., 2017; Guo et al., 2021; Ku et al., 2018; Liang et al., 2018; Qi et al., 2018; Yoo et al., 2020; Zhu et al., 2021). MV3D (Chen et al., 2017) combines the features of the LiDAR bird's eye view, the LiDAR front view, and 2D images, effectively compensating for the lack of depth information in monocular images. AVOD-FPN (Ku et al., 2018) simplifies the feature extraction process by feeding the feature map into a region proposal network for candidate region generation. 3D-CVF (Yoo et al., 2020) adopts an automatically calibrated interpolation projection to convert the 2D image feature map into a smooth, dense BEV feature map. CM3D (Zhu et al., 2021) employs a two-stage fusion strategy that incorporates point-wise fusion and proposal-wise fusion to enhance 3D object detection. DMMF (Guo et al., 2021) designs a deep multi-scale fusion method to leverage complementary information between multimodal data sources. Despite the above research improving the fusion of multi-sensor data features, they disregard the different priorities of camera and LiDAR features in the 3D object detection task, resulting in poor quality of fused features. This highlights the need for further exploration and development of fusion methods that consider the unique characteristics of each sensor's features.

3. Method

The overall structure of MMAF-Net is illustrated in Fig. 1. The method consists of four components: the data preprocessing part, the feature extraction network, the 3D region proposal network (RPN), and the object detection network. The data flow through the architecture is as follows. First, the RGB images and LiDAR point cloud data are preprocessed as input using spatial geometric constraints and a priori knowledge to generate a new image representation (RGB-DR) and convert the 3D point cloud data into a 2D BEV form. Second, feature extraction is performed by two branched streams that connect feature pyramid networks to aggregate multi-scale feature information and generate a high-resolution feature map. Next, the 3D RPN generates anchor boxes of different sizes and aspect ratios evenly over the feature map to produce non-oriented region proposals. Finally, the object detection network is applied to further refine and regress the above non-oriented region proposals and obtain more accurate detection results. In the following subsections (Section 3.1 to Section 3.5), we introduce the multimodal data input, the feature extraction network, the 3D RPN, the object detection network, and the multi-task loss function in the order specified by the framework of MMAF-Net.

3.1. Multimodal representations

In the data preprocessing stage, we utilize spatial coordinate transformation relationships to match and fuse RGB images and LiDAR point clouds to obtain multimodal data inputs while adequately extracting texture, color, and spatial geometry information for subsequent networks.

Since the volume of point cloud data is massive, the direct processing of the original point clouds using 3D convolution is time-consuming and computationally complex. To reduce the computational effort, we obtain a bird's-eye view of the point cloud followed by AVOD. Additionally, we take advantage of the transformation relationship between the LiDAR coordinate system and the pixel coordinate system to generate a front view of the point cloud based on reflection intensity information. We then project this point cloud front view onto a 2D plane and obtain the point cloud reflection intensity map. However, in practical applications, the sparsity of point cloud data generated by LiDAR leads to a much lower resolution of the point cloud reflection intensity map compared to the original image resolution (Ashraf et al., 2017). Therefore, we propose a point cloud densification method combining Delaunay and guided filtering which can generate a high-resolution point cloud reflection intensity map, as shown in Fig. 2. The detailed procedure is as follows. First, an interpolated region of interest is determined on the sparse point cloud reflection intensity map, starting with the coordinates of the first value from top to bottom of each column that is not zero. Second, linear interpolation based on Delaunay triangulation is performed on the points in the region of interest to generate a coarse and dense point cloud reflection intensity map. Finally, the RGB image corresponding to the point cloud reflection intensity map is used as a guide map, and the coarse and dense point

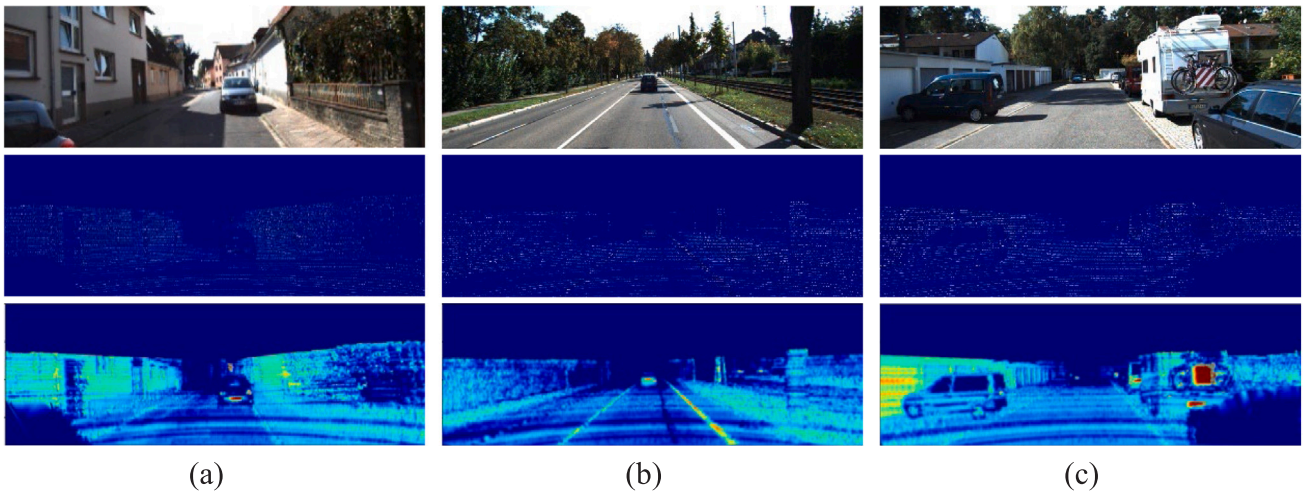


Fig. 2. The illustration of densification of point cloud reflection intensity map. Three scenarios, (a) - (c), were selected to conduct the analysis of point cloud densification.For each group, the top part is the original RGB image, the middle part is the sparse reflection intensity map of the point cloud (best viewed with zoom-in and adjusted contrast ratio), and the bottom part is the dense reflection intensity map of the point cloud processed by densification method.

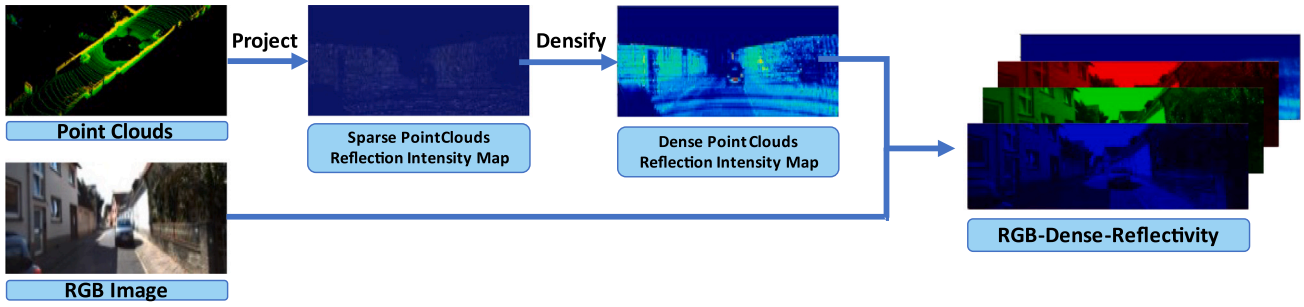


Fig. 3. The illustration of RGB-DR transformation. The 3D point clouds are projected to obtain a sparse reflection intensity map, which is then densified; the resulting dense reflection intensity map is fused with the RGB image to produce a four-channel RGB image-dense reflection intensity input representation. The whole transformation process is based on OpenCV.

cloud reflection intensity map is corrected through guided filtering to produce a more refined dense point cloud reflection intensity map. This method avoids edge distortion above the reflection intensity map by creating a region of interest and introducing a bootstrap filtering algorithm to correct for burrs at the edges of the interpolated image, thus reducing the loss of edge information.

Poor lighting conditions can make it difficult to detect objects, as seen with the black vehicle on the left in Fig. 2(c), where the edges of the vehicle are invisible in the image due to its dark color. However, the dense point cloud reflection intensity map compensates for the lack of information caused by the lighting problem and clearly shows the vehicle's outline. Compared to sparse point cloud reflection intensity maps, dense point cloud reflection intensity maps can significantly enhance roadway object features and thus improve the robustness of detection. To leverage this additional information, we add the dense intensity information as the fourth channel of the image and transform it into an "RGB image + dense reflection intensity" (RGB-DR) representation, thus making two similar features more easily learned by the neural network. Fig. 3 illustrates the formation of the RGB-DR representation, which effectively addresses the issues of information redundancy and excessive computational resource consumption associated with conventional data-level fusion methods.

3.2. Feature extraction network

The feature extraction network is comprised of two parts, the encoder and the decoder. We use a dual-stream structure to extract RGB-DR and BEV features. The encoder is a modified ResNet18 (He et al., 2016), where a 3 × 3 deformable convolution replaces the 3 × 3 normal convolution in the last layer of conv-4 and conv-5. For the RGB-DR stream, the first four residual blocks are retained, and the final fully connected layer is removed. As for the BEV stream, we removed the maximum pooling layer after the first residual block to retain more detailed information about the point cloud features while keeping the rest consistent with the RGB-DR stream. However, increasing the depth of the network could adversely affect the detection accuracy of small objects by generating a feature map with very few pixels occupied by small targets. Therefore, the design of the decoder is inspired by the feature pyramid network (FPN) (Lin et al., 2017) to get a high-resolution feature map by multi-scale feature fusion, the specific structure of which is shown in Fig. 4.

3.3. 3D region proposal network

Inspired by Chen et al. (2017), Ku et al. (2018), we introduce a 3D RPN to generate non-oriented region proposals. To begin, we generate 3D anchor boxes based on the BEV and then project these onto two view feature maps, respectively. The projected corresponding regions are fused via an anchor fusion strategy. These fused features are fed into branches to regress the difference between the anchor boxes and ground truth, ultimately generating the 3D proposals.

3.3.1. Anchor generation and fusion

The 3D anchor boxes are encoded in an axis-aligned manner (Song & Xiao, 2016) and represented by six dimensional parameters (t_x, t_y, t_z, d_x, d_y, d_z). Among them, (t_x, t_y, t_z) represents the centroid, and (d_x, d_y,

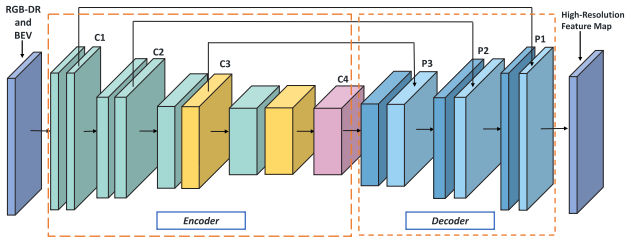


Fig. 4. Feature extraction network structure. The outputs of conv-2, conv-3, conv-4, and conv-5 in ResNet18 are denoted as C1, C2, C3, and C4, respectively. Deformable convolution is highlighted in the yellow block. We up-sample the feature map generated by C4 back to the input resolution, which is then horizontally concatenated with C3 and passed through a 3 × 3 convolutional operation to generate P3.

d_z) represents the size of the anchor box. The (t_x, t_y) pairs are sampled at 0.5-meter intervals in the BEV of the point cloud while (t_z) depends on the vertical distance between the sensor and the ground. In this paper, we use the K-means to obtain the size of anchor boxes. Under the autonomous driving scenario, two sets of clustered sizes are employed for each object category. In addition, we account for the orientation of objects in 3D space by establishing two directions (0° and 90°). This results in the generation of four anchor box types for each anchor point, achieved by combining the arranged anchor sizes and orientations.

High-dimensional feature maps occupy a large amount of memory and increase the computational burden on subsequent networks. To alleviate this issue, we first employ a 1 × 1 convolution to reduce the dimensionality of BEV and RGB-DR feature maps to obtain single-channel feature maps with the same resolution as the original image. Based on the spatial coordinate transformation relationship, the 3D anchor boxes are projected onto both the BEV and RGB-DR feature maps, and the corresponding regions of interest (ROI) are obtained for both views. Next, we adjust the ROI features corresponding to two feature maps with different resolutions to the same size and use the element-wise mean operation for ROI feature fusion.

3.3.2. 3D proposal generation

These fused feature maps are fed to two similar branches to perform 3D proposal regression and binary classification, respectively. Each branch consists of two fully connected layers. The classification branch outputs a 2D vector, which is used to represent the probability of an anchor being an object or background. Meanwhile, the regression branch estimates (Δt_x, Δt_y, Δt_z, Δd_x, Δd_y, Δd_z) to describe the differences between the centroids and dimensions of the anchor boxes and proposals. Non-maximum suppression (NMS) is applied to remove redundant regressed anchors box after initial screening. We keep the top 1024 proposals that are most likely to contain targets during the training stage. In the subsequent detection stages, NMS keeps the top 1024 3D proposals for the Pedestrian and Cyclist categories and the top 300 3D proposals for the Car category. The loss function during the training process of the 3D RPN will be explained in detail in Section 3.5.

3.4. Object detection network

As the second stage of MMAF-Net, the object detection network is utilized for further refinement and regression of the non-directional region proposals in the previous stage. The RAAF module is applied to adaptively adjust the paired proposals from multiple views. Subsequently, the fused region features are fed into specific branches for category classification, position regression, and orientation estimation.

3.4.1. Region attention adaptive fusion module

In current multi-sensor methods, it is a common practice to maintain fixed weights for proposals from different views during fusion. This approach inevitably impacts the representation of critical information and constrains the network's ability to generalize effectively. Since LiDAR and image modalities complement each other in parallel, their weights should be adaptive to various views, akin to how humans do not conflate color perception with geometric spatial perception when addressing visual problems. Hence, we introduce the attention mechanism into feature fusion and design a simple but effective module named Region Attention Adaptive Fusion (RAAF) as a multimodal feature selector to guide the proposal fusion. The RAAF module can adaptively adjust the weighting of BEV and RGB-DR to focus the network on the useful information for the task at hand. As shown in Fig. 5, F_i and F_b are the candidate region features after clipping and scaling, with a size of 7 × 7 × 64. By means of channel concatenation, we integrate the informative elements of the two channels to generate comprehensive feature F. Then, through global average pooling processing, we generate the channel descriptor s, defined by Eq. (1), where c is the number of channels of F.

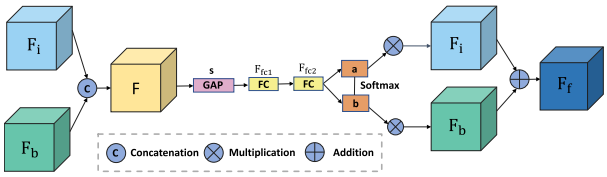


Fig. 5. The illustration of the region attention adaptive fusion (RAAF) module.

$$s_c = F_{gap}\left(F_c\right) = \frac{1}{H \times W}\sum_{i=1}^{H}\sum_{j=1}^{W} F_c(i,j) \tag{1}$$

The comprehensive features mentioned above are then fed into two fully connected layers. To further improve the efficiency, the first fully connected layer reduces the dimensionality of s and generates a vector F_{fc1} with a size of 1 × 64. After obtaining the vector F_{fc2} of 2×c in the second fully connected layer, the respective attention weights a_c, b_c of BEV, and RGB-DR are calculated using the Softmax function, as displayed in Eq. (2). A and B are the first and second-row vectors of F_{fc2}, respectively. Finally, the proposals are weighted and fused with the corresponding calculated weights to obtain F_f.

$$a_c = \frac{e^{A_c}}{e^{A_c} + e^{B_c}}, b_c = \frac{e^{B_c}}{e^{A_c} + e^{B_c}} \tag{2}$$

3.4.2. Bounding box regression

The fused feature maps undergo processing via fully connected layers to perform final category classification, position regression, and orientation estimation. Since the orientation of the bounding box needs to be considered in the final detection generation, the axis-aligned encoding method used in the 3D RPN is no longer applicable. Inspired by AVOD, the 3D bounding box is encoded with a ten-dimensional vector: (x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4, h_1, h_2). (x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4) denote the coordinates of the four corner points at the bottom of the bounding box. and represent the top and bottom corner offsets from the ground plane. The estimated orientation vector is represented as (cos(θ), sin(θ)), which allows each angle to have a unique corresponding vector. Moreover, the loss function of the object detection network is elaborated in Section 3.5.

3.5. Multi-task loss function

To describe the multi-task training process more clearly, we elaborate on the loss function in particular. The training task is a two-stage process, and the whole loss function L_{total} can be defined as Eq. (3):

$$L_{total} = L_{rpn} + L_{ref} \tag{3}$$

Algorithm 1 Joint Regression Loss Function Calculation Process.

Input: Anchor box: $B_p \leftarrow (x_1^p, x_2^p, x_3^p, x_4^p, y_1^p, y_2^p, y_3^p, y_4^p, h_1^p, h_2^p)$,
Ground truth box: $B_g \leftarrow (x_1^g, x_2^g, x_3^g, x_4^g, y_1^g, y_2^g, y_3^g, y_4^g, h_1^g, h_2^g)$.
Output: Joint regression loss: $L_{ref-reg}$.

Calculate the volume of B^p: V^p. $V^p = \left(x_3^p - x_1^p\right) \times \left(y_3^p - y_1^p\right) \times \left(h_2^p - h_1^p\right)$.
Calculate the volume of B^g: V^g. $V^g = \left(x_3^g - x_1^g\right) \times \left(y_3^g - y_1^g\right) \times \left(h_2^g - h_1^g\right)$.
Calculate the intersection volume of B^p and B^g: V^i.

$$x_1^i = \max(x_1^p, x_1^g),\ x_2^i = \min(x_3^p, x_3^g),$$
$$y_1^i = \max(y_1^p, y_1^g),\ y_2^i = \min(y_3^p, y_3^g),$$
$$h_1^i = \max(h_1^p, h_1^g),\ h_2^i = \min(h_2^p, h_2^g).$$
$$V^i = \begin{cases} \left(x_2^i - x_1^i\right) \times \left(y_2^i - y_1^i\right) \times \left(h_2^i - h_1^i\right) & if \quad x_2^i > x_1^i, y_2^i > y_1^i \\ 0 & otherwise \end{cases}$$

Calculate the minimum enclosing box volume of B^p and B^g: V^c.

$$x_1^c = \min(x_1^p, x_1^g),\ x_2^c = \max(x_2^p, x_2^g),$$
$$y_1^c = \min(y_1^p, y_1^g),\ y_2^c = \max(y_2^p, y_2^g),$$
$$h_1^c = \min(h_1^p, h_1^g),\ h_2^c = \max(h_2^p, h_2^g).$$
$$V^c = \left(x_2^c - x_1^c\right) \times \left(y_2^c - y_1^c\right) \times \left(h_2^c - h_1^c\right)$$

Calculate the distance between the centroid of B^p and B^g: ρ, and the diagonal length of the minimum bounding box: c.

$$x_c^p = \left(x_2^p - x_1^p\right)/2, y_c^p = \left(y_3^p - y_1^p\right)/2, h_c^p = \left(h_2^p - h_1^p\right)/2,$$
$$x_c^g = \left(x_2^g - x_1^g\right)/2, y_c^g = \left(y_3^g - y_1^g\right)/2, h_c^g = \left(h_2^g - h_1^g\right)/2,$$
$$\rho^2 = \left(x_c^g - x_c^p\right)^2 + \left(y_c^g - y_c^p\right)^2 + \left(h_c^g - h_c^p\right)^2 \ , \ c^2 = \left(x_2^c - x_1^c\right)^2 + \left(y_2^c - y_1^c\right)^2 + \left(h_2^c - h_1^c\right)^2 \ .$$

Calculate the 3D_DIOU of B^p and B^g.

$$3D\_IOU = \frac{V^i}{V^p + V^g - V^i}$$
$$3D\_DIOU = 3D\_IOU - \frac{\rho^2(b, b^{gt})}{c^2} \qquad 3D\_DIOU \in [-1, 1]$$

$$L_{3D\_DIOU} = 1 - 3D\_DIOU \qquad 3D\_DIOU \in [0, 2]$$

$$L_{SmoothL1} = \sum Smooth_{L1}(t) \qquad t \in (\Delta x_1, \Delta x_2, \Delta x_3, \Delta x_4, \Delta y_1, \Delta y_2, \Delta y_3, \Delta y_4, \Delta h_1, \Delta h_2)$$

$$L_{ref-reg} = \lambda L_{3D\_DIOU} + (1 - \lambda) L_{SmoothL1}$$

return $L_{ref-reg}$

in which the L_{rpn} and the L_{ref} describe the optimizing goal of the 3D region proposal stage and the object detection stage, respectively. In addition to considering the loss terms common to both, L_{ref} also needs to take the directional angle loss L_{ang} into consideration. Therefore, the L_{total} can also be formulated as Eq. (4). α_1, α_2, β_1, β_2 and β_3 denote the weight value of each loss term.

$$L_{total} = \alpha_1 L_{rpn-cls} + \alpha_2 L_{rpn-reg} + \beta_1 L_{ref-cls} + \beta_2 L_{ref-reg} + \beta_3 L_{ref-ang} \quad (4)$$

Two stages of classification loss are implemented by the cross-entropy function, which can be defined as Eq. (5):

$$L_{rpn-cls} = L_{ref-cls} = -\frac{1}{N} \sum_{i=1}^{N} y_i \log\left(p_i\right) \quad (5)$$

where N is the number of samples, y_i denotes the label value of sample i; p_i is the predicted probability value of sample i. SmoothL1 loss function is used to calculate both $L_{rpn-reg}$ and $L_{ref-ang}$. To be specific, the loss term $L_{rpn-reg}$ can be formulated as Eq. (6), where (Δt_x, Δt_y, Δt_z, Δd_x, Δd_y, Δd_z) represents the difference between the anchor box and the ground truth. Similarly, the loss term $L_{ref-reg}$ can be defined as Eq. (7), where θ^g and θ^p indicate the direction angles of the ground truth and bounding box, respectively.

$$L_{rpn-reg} = \frac{1}{N_{reg}} \sum Smooth_{L1}(b) \ b \in \left(\Delta t_x, \Delta t_y, \Delta t_z, \Delta d_x, \Delta d_y, \Delta d_z\right) \quad (6)$$

$$L_{ref\text{-}ang} = \sum Smooth_{L1}\left(\sin\left(\theta^g - \theta^p\right)\right) \quad (7)$$

The mean square error represented by L1 Loss and L2 Loss is commonly used as the loss function for current 3D object detection tasks. However, it cannot fully reflect the overlap between the bounding box and the ground truth. The IoU score may not be the same even when the loss function values are consistent. The proposal of L_{IOU} (Yu et al., 2016) effectively solves this problem, but it fails to reflect the way the two intersect when the bounding box and ground truth have the same IoU score. L_{GIOU} (Rezatofighi et al., 2019) improves and optimizes L_{IOU} by adding a penalty term to it. L_{GIOU} better reflects the position of the bounding box in relation to the ground truth, but there is a risk that the penalty term is 0 when the two are enclosed with no convergence. L_{DIOU} (Zheng et al., 2020) takes into account the overlapping area and centroid distance of the two frames, further improving convergence speed. The L_{DIOU} can be calculated by Eq. (8), where ρ represents the distance between the centroid (b) of the bounding box and the centroid of the ground truth (b^{gt}), and c represents the diagonal distance of the smallest enclosing frame that contains both the bounding box and ground truth.

$$L_{DIOU} = 1 - IOU + \frac{\rho^2\left(b, b^{gt}\right)}{c^2} \quad (8)$$

Compared to 2D detection, the 3D IoU score is more complex to calculate. When the bounding box has a rotation angle, the overlapping part of the two boxes is an irregular shape (Zhou et al., 2019). Therefore, we introduce L_{DIOU} from 2D object detection into 3D object detection and establish a joint regression loss based on 3D_DIOU and SmoothL1, aiming to calculate the similarity degree between two 3D boundary boxes. For detail, the calculation process for $L_{ref-reg}$ is operated by the Algorithm 1. The positional offsets of the bounding box and the ground truth are represented by (Δx_1, Δx_2, Δx_3, Δx_4, Δy_1, Δy_2, Δy_3, Δy_4, Δh_1, Δh_2), and λ is a weighting coefficient.

4. Experiments

4.1. Setup

4.1.1. Dataset

As an authoritative 3D object detection dataset for autonomous driving, the KITTI dataset (Geiger et al., 2012) consists of 7481 training pairs and 7518 testing pairs of RGB images and LiDAR point clouds. Since the test set is not publicly available, we follow the experimental setting in (Chen et al., 2015, 2017; Ku et al., 2018; Qin et al., 2019) to split the training set into 3712 training samples and 3769 validation samples. The training and testing samples can be defined at three levels

Table 1
Performance comparison of 3D object detection with state-of-the-art methods on KITTI test split in the Car category. "L" and "R" mean the LiDAR and RGB images, "E"," M", and "H" are the simplified representation of three levels: easy, moderate, and hard. "mAP" indicates the mean average precision for three levels.

Method	Modality	Time (s)	$AP_{3D\|40}$				$AP_{BEV\|40}$			
			E	M	H	mAP	E	M	H	mAP
MonoGRNet (Qin et al., 2019)	R	0.06	13.88	10.19	7.62	10.56	24.97	19.44	16.30	20.23
Stereo RCNN (Li et al., 2019)	R	0.41	47.58	30.23	23.72	33.84	68.50	48.30	41.47	52.75
VoxelNet (Zhou & Tuzel, 2018)	L	0.23	77.47	65.11	57.73	66.77	89.35	79.26	77.39	82.00
SECOND (Yan et al., 2018)	L	0.05	83.13	73.66	66.20	74.33	79.37	77.95	79.37	78.90
PointPillar (Lang et al., 2019)	L	0.02	82.58	74.31	68.99	75.29	86.56	82.81	79.58	82.98
PointRCNN (Shi et al., 2019)	L	0.10	86.96	75.64	70.70	77.77	87.39	82.72	80.32	83.48
TANet (Liu et al., 2020)	L	0.03	84.39	75.94	68.82	76.38	91.58	86.54	81.19	86.43
Pointformer (Pan et al., 2021)	L	–	87.13	77.06	69.25	77.81	–	–	–	–
MV3D (Chen et al., 2017)	L+R	0.36	74.97	63.63	54.00	64.20	86.62	78.93	69.80	78.45
AVOD-FPN (Ku et al., 2018)	L+R	0.10	83.07	71.76	65.73	73.52	91.17	84.67	74.77	83.54
F-PointNet (Qi et al., 2018)	L+R	0.17	82.19	69.79	60.59	70.86	90.99	84.82	79.62	85.14
Cont-Fuse (Liang et al., 2018)	L+R	0.06	83.68	68.78	61.67	71.38	94.07	85.35	75.88	85.10
F-ConvNet (Wang & Jia, 2019)	L+R	0.47	85.88	76.51	68.08	76.82	89.69	83.08	74.56	82.44
3D-CVF (Yoo et al., 2020)	L+R	0.06	89.20	80.05	73.11	80.79	93.52	89.56	82.45	88.51
PI-RCNN (Xie et al., 2020)	L+R	0.10	84.37	74.82	70.03	76.41	91.44	85.81	81.00	86.08
PointPainting (Vora et al., 2020)	L+R	0.40	82.11	71.70	67.08	73.63	92.45	88.11	83.36	87.97
CM3D (Zhu et al., 2021)	L+R	–	87.22	77.28	72.04	78.85	–	–	–	–
DMMF (Guo et al., 2021)	L+R	–	83.76	74.51	68.43	75.57	89.70	86.98	79.99	85.56
MMAF-Net (Ours)	L+R	0.16	87.34	74.98	73.37	78.56	88.53	87.95	83.52	86.67

Table 2
Performance comparison of 3D object detection with state-of-the-art methods on KITTI validation split in the Car category. "L" and "R" mean the LiDAR and RGB images, "E"," M", and "H" are the simplified representation of three levels: easy, moderate, and hard. "mAP" indicates the mean average precision for three levels.

Method	Modality	$AP_{3D\|40}$				$AP_{BEV\|40}$			
		E	M	H	mAP	E	M	H	mAP
VoxelNet (Zhou & Tuzel, 2018)	L	81.97	65.46	62.85	70.09	89.60	84.81	78.57	84.33
SECOND (Yan et al., 2018)	L	88.61	78.62	77.22	81.48	89.96	87.07	79.66	85.56
PointPillar (Lang et al., 2019)	L	86.46	77.28	74.65	79.46	–	–	–	–
PointRCNN (Shi et al., 2019)	L	88.72	78.61	77.82	81.72	–	–	–	–
TANet (Liu et al., 2020)	L	85.27	77.64	72.13	78.35	–	–	–	–
MV3D (Chen et al., 2017)	L+R	71.29	62.68	56.56	63.51	86.55	78.10	76.67	80.44
AVOD-FPN (Ku et al., 2018)	L+R	84.41	74.44	68.65	75.83	89.37	86.09	79.13	84.86
F-PointNet (Qi et al., 2018)	L+R	83.76	70.92	63.65	72.28	88.16	84.02	76.44	82.87
PointFusion (Xu et al., 2018)	L+R	77.92	63.00	53.27	64.73	–	–	–	–
SIFRNet (Zhao et al., 2019)	L+R	85.62	72.05	64.19	73.95	88.63	83.45	76.08	82.72
3D-CVF (Yoo et al., 2020)	L+R	89.67	79.88	78.47	82.67	–	–	–	–
MMAF-Net (Ours)	L+R	86.04	76.59	72.56	78.40	89.70	87.96	84.53	87.40

(Easy, Moderate, and Hard) based on occlusion, truncation, and 3D bounding box height criteria. In this paper, we focus on evaluating the three categories of objects: Cars, Pedestrians, and Cyclists. We evaluate both 3D object detection and BEV object detection tasks at 0.7 IoU threshold for the Car class and 0.5 IoU threshold for the Pedestrian and Cyclist classes.

4.1.2. Evaluation metrics

To provide accurate and comprehensive performance evaluation, the KITTI dataset offers a series of evaluation metrics, including mean Average Precision (*mAP*) and Average Orientation Similarity (*AOS*) for both 3D and the BEV detection benchmarks. We choose AP_{3D}, AP_{BEV}, and *mAP* to provide objective reports when conducting performance comparisons and ablation studies. AP_{3D} and AP_{BEV} represent the average accuracy of 3D detection frames in 3D space and bird's eye view, respectively. For fair comparison, we adopt the AP with 40 recall points to give an objective report when conducting performance comparison and ablation studies. Furthermore, the times refer to the inference times for processing one image.

4.1.3. Training details

The whole detection pipeline is trained with an AMD Ryzen 7 5800H CPU and a Tesla K80 GPU in the TensorFlow deep learning framework. We train the model for 120k epochs, with weight decay as 0.8 and decay applied every 30k epochs of training. The initial learning rate is set to $1e^{-4}$, and we use an Adam optimizer with a batch size of 4.

4.2. Main results

In this section, we conduct quantitative, qualitative, and visual analyses to show the performance of our proposed method on the KITTI dataset from multiple aspects.

4.2.1. Quantitative results

We compare MMAF-Net with state-of-the-art methods on both validation and test sets. As shown in Table 1, the best and second-best results on the test set are highlighted in red and blue, respectively. It can be seen from Table 1 that using solely images for 3D object detection is not good enough, as images alone cannot provide accurate depth information. While stereo images improve accuracy compared to monocular images, there remains a significant performance gap compared to methods utilizing point clouds and multimodal data fusion. Compared to other state-of-the-art methods, MMAF-Net attains outstanding results in both the easy and hard levels of the metric of AP_{3D}, particularly excelling in the hard level with a notable 0.26% improvement. As for the metric of AP_{BEV}, our method indicates remarkable detection results, highlighting its significant performance in challenging scenes. While our method may not consistently achieve the best performance across all levels, it consistently ranks at the top in terms of mAP, underscoring its overall effectiveness. Regarding runtime, MMAF-Net requires only 0.16 s for inference, surpassing the speed of most previous methods. It is reasonable to conclude that our method can guarantee real-time detection while better balancing the

Table 3
Performance comparison of 3D object detection with state-of-the-art methods on KITTI test split in the Pedestrian and Cyclist category.

Method	Modality	Time (s)	$AP_{3D\|40}$ (Pedestrian)				$AP_{3D\|40}$ (Cyclist)			
			E	M	H	mAP	E	M	H	mAP
VoxelNet (Zhou & Tuzel, 2018)	L	0.23	39.48	33.69	31.51	34.89	67.17	47.65	45.11	53.31
PointPillar (Lang et al., 2019)	L	0.02	51.45	41.92	38.89	44.09	77.10	58.65	51.92	62.56
TANet (Liu et al., 2020)	L	0.03	53.72	44.34	40.49	46.18	75.70	59.44	52.53	62.56
AVOD-FPN (Ku et al., 2018)	L+R	0.10	50.46	42.27	39.04	43.92	63.76	50.55	44.93	53.08
F-PointNet (Qi et al., 2018)	L+R	0.17	50.53	42.15	38.08	43.59	72.27	56.12	49.01	59.13
F-ConvNet (Wang & Jia, 2019)	L+R	0.47	52.16	43.38	38.80	44.78	81.98	65.07	56.54	67.86
PointPainting (Vora et al., 2020)	L+R	0.40	50.32	40.97	37.84	43.04	77.63	63.78	55.89	65.77
DMMF (Guo et al., 2021)	L+R	–	57.57	53.17	47.69	52.81	–	–	–	–
MMAF-Net (Ours)	L+R	0.16	56.54	52.91	48.14	52.53	78.60	67.89	56.83	67.77

Fig. 6. The visualized comparison of the 3D object detection results between our method and the same type of AVOD-FPN in three different scenes. (a) presents the detection results of AVOD-FPN, while (b) shows the detection results of MMAF-Net.

trade-off between detection accuracy and speed. Furthermore, we also evaluate the methods on the KITTI validation set for the Car category. The detailed metric comparison results are presented in Table 2. It is evident that MMAF-Net still offers remarkable performance gains over most previous arts, affirming its efficacy and competitiveness.

Given that pedestrians and cyclists are relatively small objects, estimating their 3D properties is a more challenging task. Nonetheless, we report a performance comparison on the KITTI test set for the Pedestrian and Cyclist classes in Table 3 as well. It is noteworthy that MMAF-Net ranks first in many cases, particularly on the hard level for pedestrians and cyclists. The superior 3D AP performance gains exhibited by MMAF-Net over DMMF (Guo et al., 2021) on the hard level for pedestrians demonstrate our method's ability to effectively detect overlapping or smaller objects.

4.2.2. Qualitative results

In addition to the quantitative results presented above, we conduct a qualitative analysis of the MMAF-Net by comparing it with the comparable two-stage algorithm AVOD-FPN (Ku et al., 2018). The results are shown in Fig. 6. For the detection results, we use 3D bounding boxes in the camera image to represent the detection performance in an intuitive visualization. As demonstrated in Fig. 6, we label the car category with green 3D boxes, the pedestrian category with blue 3D boxes, and the cyclist category with yellow 3D boxes. The missed objects in AVOD-FPN are marked with red 3D boxes. From the visualization results, we find that our detector outperforms AVOD-FPN by more accurately recognizing and locating objects in the real world. MMAF-Net has fewer false and missed objects than AVOD-FPN. Additionally, the 3D bounding box regression is more accurate, resulting in a more pronounced improvement in detection.

4.2.3. Visualization results

In Fig. 7, we select six representative scenarios from the KITTI dataset to visualize the detection results of MMAF-Net. Fig. 7(a)–(f) corresponds to a typical roadway scene, a small object detection scene with pedestrians, a crowded roadway scene, a scene with obscured objects, an overexposure scene, and a low light scene, respectively. For each scene, the detection results are mapped to images and point clouds through coordinate transformation for visualization. The visualization results demonstrate that most objects can be correctly detected by MMAF-Net, even in complex traffic scenarios with dense roads and occlusions. For scenes with more complicated lighting conditions, such as (e) and (f), which are even more difficult to be recognized by human eyes, MMAF-Net can still accurately detect the objects. This is primarily due to the fusion of dense reflection intensity information and RGB image in data preprocessing, which enhances the LiDAR front view's ability to resist light interference. MMAF-Net demonstrates its effectiveness and robustness by coping well with complex and changing road conditions and correctly detecting vehicles, pedestrians, and other objects in different scenarios.

4.3. Ablation studies

We carry out several ablation experiments to verify the effectiveness and rationality of each component of the proposed method. All the ablation experiments are performed on the KITTI validation datasets.

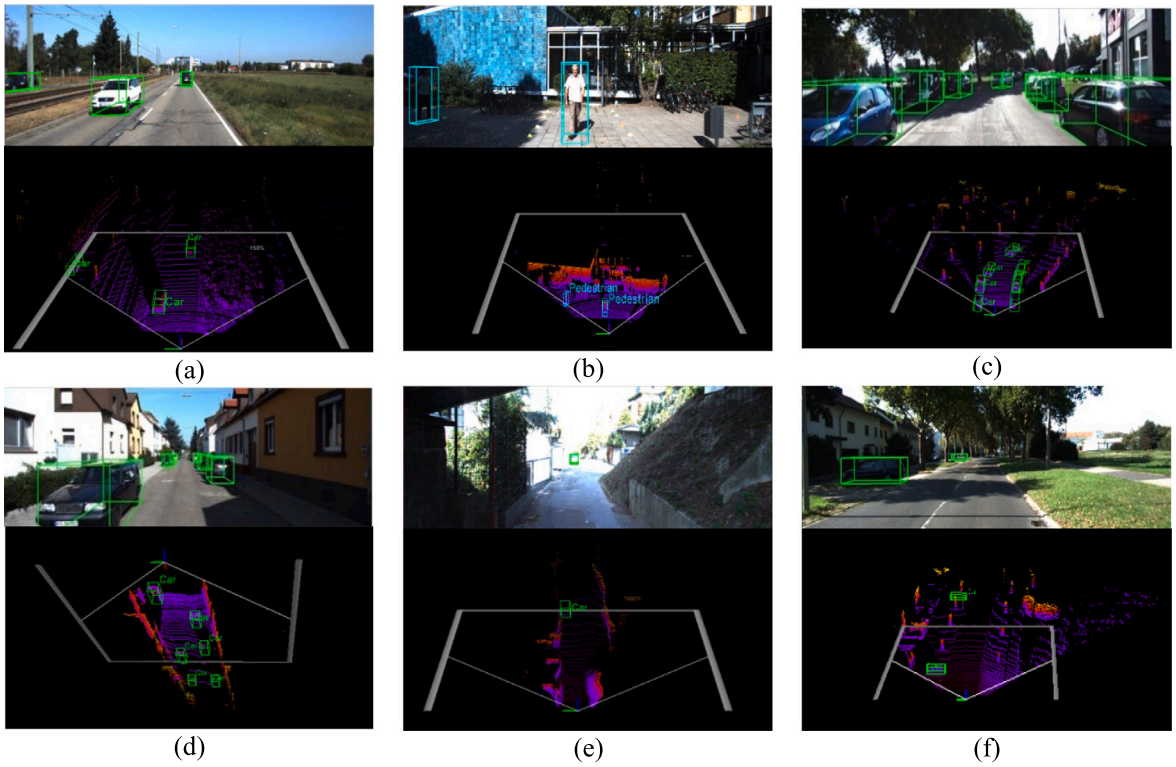


Fig. 7. Visualizations of MMAF-Net detection results on the KITTI validation set (best viewed with zoom-in). For each scene, the upper part is the RGB image, the lower part is a view of the corresponding point cloud, and detection boxes are marked in green. The word beside each box indicates the instance and its class.

Table 4
Performance comparison of different input forms. "mAP" means the mean average precision for three levels.

Input	$AP_{3D\|40}$ (Car)			$AP_{3D\|40}$ (Pedestrian)			$AP_{3D\|40}$ (Cyclist)			mAP
	E	M	H	E	M	H	E	M	H	
RGB+BEV (Baseline)	85.73	75.92	69.94	67.93	63.96	56.90	77.35	57.21	54.87	67.75
RGB-DR+BEV	**86.04**	**76.59**	**72.56**	**68.95**	**64.88**	**59.21**	**78.14**	**57.89**	**55.92**	**68.90**

Table 5
Performance comparison of different feature fusion strategies.

Fusion strategy	$AP_{3D\|40}$ (Car)			$AP_{3D\|40}$ (Pedestrian)			$AP_{3D\|40}$ (Cyclist)			mAP
	E	M	H	E	M	H	E	M	H	
Element-wise mean (Baseline)	85.36	75.91	70.26	68.53	63.49	56.62	76.84	57.01	54.08	67.56
Concatenation	85.74	76.18	70.64	68.31	63.24	57.83	77.55	57.36	54.41	67.91
CBAM	85.61	76.32	71.35	68.36	64.03	58.15	76.93	57.77	54.39	68.21
RAAF	**86.04**	**76.59**	**72.56**	**68.95**	**64.88**	**59.21**	**78.14**	**57.89**	**55.92**	**68.90**

4.3.1. Effectiveness of the RGB-DR input representation

We investigate the effects of the RGB-DR input representation by using RGB, and RGB-DR combined with BEV as inputs separately. We take MMAF-Net (RGB+BEV) as the baseline model for this evaluation. As shown in Table 4, using only the fusion of RGB and BEV improves the detection performance to a limited extent. However, when RGB-DR and BEV are used as inputs, the metric of mAP improves by 1.15%. As for 3D object detection in hard levels, the detection accuracy for the three categories improved by 2.62%, 2.31%, and 1.05%, respectively. This suggests that the initial fusion of image features with dense point cloud reflection intensity information to generate RGB-DR is indeed conducive to learning more features and facilitating detection accuracy.

4.3.2. Effectiveness of the RAAF module

In Table 5, we take MMAF-Net with the element-wise mean method as the basic model, comparing its 3D detection results with those obtained using the concatenation, CBAM (Woo et al., 2018), and RAAF fusion strategies, respectively. CBAM combines spatial and channel attention, focusing on both content and its respective location. In contrast, our RAAF module concentrates on the feature itself that contains the object information. The results in Table 5 reveal that integrating the RAAF module into the network achieves the best performance. This can be attributed to the fact that RAAF can adaptively adjust the weights of image and point cloud features through the attention mechanism, thus reducing interference from extraneous information.

Table 6
Performance comparison of different regression loss functions.

Loss function	$AP_{3D\|40}$ (Car)			$AP_{3D\|40}$ (Pedestrian)			$AP_{3D\|40}$ (Cyclist)			mAP
	E	M	H	E	M	H	E	M	H	
SmoothL1 (Baseline)	84.76	75.62	69.54	67.83	62.96	55.62	76.85	48.71	53.62	66.10
3D_DIOU	85.51	76.07	71.27	68.22	63.73	57.56	77.32	49.71	55.34	67.19
Joint regression loss function	**86.04**	**76.59**	**72.56**	**68.95**	**64.88**	**59.21**	**78.14**	**57.89**	**55.92**	**68.90**

4.3.3. Effectiveness of the joint regression loss function

To thoroughly evaluate the effectiveness of our proposed joint regression loss function, we set MMAF-Net with SmoothL1 loss function as the baseline. Based on the results in Table 6, we can find that our joint regression loss function improves the accuracy of 3D object detection at all levels, especially at the moderate and hard levels. This demonstrates that the joint regression loss function further optimizes the shape similarity and position distance between the detection box and the ground truth box. It also has a stronger bounding box regression capability. Therefore, our proposed joint loss function is meaningful for the detection of moderate and hard instances in real scenarios.

5. Discussion

In response to the limitations of prior fusion techniques, including imprecise methods, inadequate accuracy in detecting small and occluded objects, and limited detection capabilities for distant objects in 3D object detection, we propose the MMAF-Net. While previous approaches have shown improved performance in 3D object detection tasks, they have not adequately addressed the need for precise and robust fusion of LiDAR and camera features. In contrast, our approach focuses on optimizing the fusion of these features at different stages of the network. We also employ various fusion strategies at each stage to fully leverage the benefits of multimodal data.

The RGB-DR input representation proposes encoding the reflectance intensity of the dense point cloud onto the image to leverage the strengths of both LiDAR and camera features. This approach enables our detection pipeline to efficiently fuse advantageous image clues and selectively supplement spectral information while avoiding the computational burden and redundancy inherent in full data-level fusion. Furthermore, we have found that incorporating an attention mechanism is a powerful strategy when working with multimodal data. Drawing on this insight, we devise the RAAF module, which enhances the algorithm's adaptability and resilience to environmental variations. Acting as a modal feature selector, the RAAF module models the relationships between input feature elements, readjusts the weights associated with BEV and RGB-DR elements, and assigns different views varying degrees of importance during subsequent fusion and detection processes. This enables the module to spotlight key elements while simultaneously suppressing unwanted interference. In addition, we recognize the negative impact of unreliable 3D IoU scores during the training process and develop a simple yet effective solution that involves extending the 2D_DIOU loss function to 3D and constructing a joint regression loss based on 3D_DIOU and SmoothL1.

The current study yields various noteworthy discoveries. Firstly, our findings indicate that RGB-DR has a more pronounced influence on feature integration than RGB images. This suggests that incorporating the reflection intensity of dense point clouds strengthens the input representation and enriches the spectral information of the image. Additionally, whether directly or indirectly, the RAAF module was found to have a stronger effect compared to original fusion techniques without attention mechanisms, which highlights the importance of adjusting the weight of multi-sensor fusion according to different scenarios and situations. Finally, the joint regression loss function has been shown to significantly improve the similarity between detection boxes and ground truth boxes, leading to improved performance in 3D object detection.

The widespread availability of high-performance sensors has provided support for incorporating more modalities in 3D object detection. Our proposed method has potential applications in a range of fields, including autonomous driving, robot navigation, and virtual reality. Some questions remain. For example, the network models proposed in this paper all belong to the category of supervised learning, which requires the dataset to be high quality. However, the current multimodal datasets are not yet fully adequate for algorithm training and learning. In the future, we will exploit unsupervised learning to overcome the reliance on annotation and further improve the self-learning and generalization ability of the MMAF-Net. Moreover, it would be of interest to incorporate continuous frame data to further improve the performance of 3D object detection by utilizing contextual information in consecutive frames.

6. Conclusion

In this paper, we present a novel 3D object detection framework named MMAF-Net. Our method takes 3D point clouds and 2D images as input and uses a multi-view fusion strategy to generate a new input format called RGB-DR. Moreover, we introduce an attention mechanism and construct the RAAF module to adaptively adjust the weight of certain features. We also design a loss function based on 3D DIOU and SmoothL1 to optimize the joint regression loss for more effective detection of 3D objects in autonomous driving scenarios. Experimental results on the KITTI dataset, including ablation experiments, demonstrate that MMAF-Net achieves higher detection accuracy for distant and obscured objects while maintaining real-time detection.

CRediT authorship contribution statement

Wensheng Zhang: Concepualization, Methodology, Writing – review & editing, Investigation, Validation, Project administration. **Hongli Shi:** Software, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Yunche Zhao:** Software, Writing – review & editing. **Zhenan Feng:** Writing – review & editing, Supervision. **Ruggiero Lovreglio:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This study was supported by the Central Guidance for Local Science and Technology Development Fund Program, China under Grant 226Z6101G, the Introducing Foreign Intelligence Program for Hebei Province and Shijiazhuang City, China (2022–2024) and the Shijiazhuang Science and Technology Planning Program, China under Grant 221130134A and 236130217A.

Appendix. Abbreviations

See Table A.1.

Table A.1
Abbreviation explanation.

Full text	Abbreviation
Two/Three Dimensional	2D/3D
RGB-Dense Reflectivity	RGB-DR
Region Attention Adaptive Fusion Module	RAAF
Distance Intersection over Union	DIOU
Intersection over Union	IoU
Bird's Eye View	BEV
Region Proposal Network	RPN
Feature Pyramid Network	FPN
Region of Interest	ROI
Non-maximum Suppression	NMS
Mean Average Precision	mAP
Average Orientation Similarity	AOS
Average Precision	AP
Average Precision with 40 Recall Points	AP_{40}
Average Precision for 3D object Detection	AP_{3D}
Average Precision for Bird's Eye View	AP_{BEV}

References

Alaba, S. Y., & Ball, J. E. (2023). Deep learning-based image 3-D object detection for autonomous driving. *IEEE Sensors Journal*, *23*(4), 3378–3394. http://dx.doi.org/10.1109/JSEN.2023.3235830.

Arnold, E., Al-Jarrah, O. Y., Dianati, M., Fallah, S., Oxtoby, D., & Mouzakitis, A. (2019). A survey on 3d object detection methods for autonomous driving applications. *IEEE Transactions on Intelligent Transportation Systems*, *20*(10), 3782–3795. http://dx.doi.org/10.1109/TITS.2019.2892405.

Ashraf, I., Hur, S., & Park, Y. (2017). An investigation of interpolation techniques to generate 2D intensity image from LIDAR data. *IEEE Access*, *5*, 8250–8260. http://dx.doi.org/10.1109/ACCESS.2017.2699686.

Chang, J.-R., & Chen, Y.-S. (2018). Pyramid stereo matching network. In *2018 IEEE/CVF conference on computer vision and pattern recognition* (pp. 5410–5418). http://dx.doi.org/10.1109/cvpr.2018.00567.

Charles, R. Q., Su, H., Kaichun, M., & Guibas, L. J. (2017). PointNet: Deep learning on point sets for 3D classification and segmentation. In *2017 IEEE conference on computer vision and pattern recognition* (pp. 77–85). http://dx.doi.org/10.1109/CVPR.2017.16.

Chen, X., Kundu, K., Zhang, Z., Ma, H., Fidler, S., & Urtasun, R. (2016). Monocular 3D object detection for autonomous driving. In *2016 IEEE conference on computer vision and pattern recognition* (pp. 2147–2156). http://dx.doi.org/10.1109/CVPR.2016.236.

Chen, X., Kundu, K., Zhu, Y., Berneshawi, A. G., Ma, H., Fidler, S., & Urtasun, R. (2015). 3D object proposals for accurate object class detection. *Advances in Neural Information Processing Systems*, *28*, http://dx.doi.org/10.5555/2969239.2969287.

Chen, W., Li, P., & Zhao, H. (2022). MSL3D: 3D object detection from monocular, stereo and point cloud for autonomous driving. *Neurocomputing*, *494*, 23–32. http://dx.doi.org/10.1016/j.neucom.2022.04.075.

Chen, Y., Liu, S., Shen, X., & Jia, J. (2020). DSGN: Deep stereo geometry network for 3D object detection. In *2020 IEEE/CVF conference on computer vision and pattern recognition* (pp. 12533–12542). http://dx.doi.org/10.1109/CVPR42600.2020.01255.

Chen, M., Liu, P., & Zhao, H. (2022). M3DGAF: Monocular 3D object detection with geometric appearance awareness and feature fusion. *IEEE Sensors Journal*, http://dx.doi.org/10.1109/JSEN.2022.3189174.

Chen, M., Liu, P., & Zhao, H. (2023). LiDAR-camera fusion: Dual transformer enhancement for 3D object detection. *Engineering Applications of Artificial Intelligence*, *120*, Article 105815. http://dx.doi.org/10.1016/j.engappai.2022.105815.

Chen, X., Ma, H., Wan, J., Li, B., & Xia, T. (2017). Multi-view 3D object detection network for autonomous driving. In *2017 IEEE conference on computer vision and pattern recognition* (pp. 6526–6534). http://dx.doi.org/10.1109/cvpr.2017.691.

Deng, J., & Czarnecki, K. (2019). MLOD: A multi-view 3D object detection based on robust feature fusion method. In *2019 IEEE intelligent transportation systems conference* (pp. 279–284). IEEE, http://dx.doi.org/10.1109/ITSC.2019.8917126.

Dou, J., Xue, J., & Fang, J. (2019). SEG-VoxelNet for 3D vehicle detection from RGB and LiDAR data. In *2019 International conference on robotics and automation* (pp. 4362–4368). http://dx.doi.org/10.1109/ICRA.2019.8793492.

Geiger, A., Lenz, P., & Urtasun, R. (2012). Are we ready for autonomous driving? The KITTI vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition* (pp. 3354–3361). http://dx.doi.org/10.1109/CVPR.2012.6248074.

Ghasemieh, A., & Kashef, R. (2022). 3D object detection for autonomous driving: Methods, models, sensors, data, and challenges. *Transportation Engineering*, *8*, Article 100115. http://dx.doi.org/10.1016/j.treng.2022.100115.

Girshick, R. (2015). Fast R-CNN. In *2015 IEEE international conference on computer vision* (pp. 1440–1448). http://dx.doi.org/10.1109/ICCV.2015.169.

Guo, R., Li, D., & Han, Y. (2021). Deep multi-scale and multi-modal fusion for 3D object detection. *Pattern Recognition Letters*, *151*, 236–242. http://dx.doi.org/10.1016/j.patrec.2021.08.028.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE conference on computer vision and pattern recognition* (pp. 770–778). http://dx.doi.org/10.1109/CVPR.2016.90.

Ku, J., Mozifian, M., Lee, J., Harakeh, A., & Waslander, S. L. (2018). Joint 3D proposal generation and object detection from view aggregation. In *2018 IEEE/RSJ international conference on intelligent robots and systems* (pp. 1–8). http://dx.doi.org/10.1109/IROS.2018.8594049.

Lang, A. H., Vora, S., Caesar, H., Zhou, L., Yang, J., & Beijbom, O. (2019). PointPillars: Fast encoders for object detection from point clouds. In *2019 IEEE/CVF conference on computer vision and pattern recognition* (pp. 12689–12697). http://dx.doi.org/10.1109/CVPR.2019.01298.

Li, P., Chen, X., & Shen, S. (2019). Stereo R-CNN based 3D object detection for autonomous driving. In *2019 IEEE/CVF conference on computer vision and pattern recognition* (pp. 7636–7644). http://dx.doi.org/10.1109/CVPR.2019.00783.

Liang, M., Yang, B., Chen, Y., Hu, R., & Urtasun, R. (2019). Multi-task multi-sensor fusion for 3D object detection. In *2019 IEEE/CVF conference on computer vision and pattern recognition* (pp. 7337–7345). http://dx.doi.org/10.1109/CVPR.2019.00752.

Liang, M., Yang, B., Wang, S., & Urtasun, R. (2018). Deep continuous fusion for multi-sensor 3d object detection. In *Proceedings of the European conference on computer vision* (pp. 641–656). http://dx.doi.org/10.1007/978-3-030-01270-0_39.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In *2017 IEEE conference on computer vision and pattern recognition* (pp. 936–944). http://dx.doi.org/10.1109/CVPR.2017.106.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector. In *Computer vision–ECCV 2016: 14th European conference, Amsterdam, the Netherlands, October 11–14, 2016, Proceedings, Part I 14* (pp. 21–37). Springer, http://dx.doi.org/10.1007/978-3-319-46448-0_2.

Liu, Z., Zhao, X., Huang, T., Hu, R., Zhou, Y., & Bai, X. (2020). Tanet: Robust 3d object detection from point clouds with triple attention. In *Proceedings of the AAAI conference on artificial intelligence (vol. 34)* (pp. 11677–11684). http://dx.doi.org/10.1609/aaai.v34i07.6837.

Mao, J., Xue, Y., Niu, M., Bai, H., Feng, J., Liang, X., Xu, H., & Xu, C. (2021). Voxel transformer for 3D object detection. In *2021 IEEE/CVF international conference on computer vision* (pp. 3144–3153). http://dx.doi.org/10.1109/iccv48922.2021.00315.

Mohapatra, S., Yogamani, S., Gotzig, H., Milz, S., & Mader, P. (2021). BevDetNet: Bird's eye view LiDAR point cloud based real-time 3D object detection for autonomous driving. In *2021 IEEE international intelligent transportation systems conference* (pp. 2809–2815). http://dx.doi.org/10.1109/ITSC48978.2021.9564490.

Mousavian, A., Anguelov, D., Flynn, J., & Košecká, J. (2017). 3D bounding box estimation using deep learning and geometry. In *2017 IEEE conference on computer vision and pattern recognition* (pp. 5632–5640). http://dx.doi.org/10.1109/CVPR.2017.597.

Mukhtar, A., Xia, L., & Tang, T. B. (2015). Vehicle detection techniques for collision avoidance systems: A review. *IEEE Transactions on Intelligent Transportation Systems*, *16*(5), 2318–2338. http://dx.doi.org/10.1109/TITS.2015.2409109.

Pan, X., Xia, Z., Song, S., Li, L. E., & Huang, G. (2021). 3D object detection with pointformer. In *2021 IEEE/CVF conference on computer vision and pattern recognition* (pp. 7459–7468). http://dx.doi.org/10.1109/CVPR46437.2021.00738.

Pang, S., Morris, D., & Radha, H. (2020). CLOCs: Camera-LiDAR object candidates fusion for 3D object detection. In *2020 IEEE/RSJ international conference on intelligent robots and systems* (pp. 10386–10393). http://dx.doi.org/10.1109/IROS45743.2020.9341791.

Qi, C. R., Liu, W., Wu, C., Su, H., & Guibas, L. J. (2018). Frustum PointNets for 3D object detection from RGB-d data. In *2018 IEEE/CVF conference on computer vision and pattern recognition* (pp. 918–927). http://dx.doi.org/10.1109/CVPR.2018.00102.

Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in Neural Information Processing Systems*, *30*, http://dx.doi.org/10.48550/arXiv.1706.02413.

Qian, R., Lai, X., & Li, X. (2022). 3D object detection for autonomous driving: A survey. *Pattern Recognition*, *130*, Article 108796. http://dx.doi.org/10.1016/j.patcog.2022.108796.

Qin, Z., Wang, J., & Lu, Y. (2019). MonoGRNet: A geometric reasoning network for monocular 3D object localization. In *Proceedings of the AAAI conference on artificial intelligence (vol. 33)* (pp. 8851–8858). http://dx.doi.org/10.1609/aaai.v33i01.33018851.

Ranft, B., & Stiller, C. (2016). The role of machine vision for intelligent vehicles. *IEEE Transactions on Intelligent vehicles*, *1*(1), 8–19. http://dx.doi.org/10.1109/TIV.2016.2551553.

Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., & Savarese, S. (2019). Generalized intersection over union: A metric and a loss for bounding box regression. In *2019 IEEE/CVF conference on computer vision and pattern recognition* (pp. 658–666). http://dx.doi.org/10.1109/CVPR.2019.00075.

Shi, S., Wang, X., & Li, H. (2019). PointRCNN: 3D object proposal generation and detection from point cloud. In *2019 IEEE/CVF conference on computer vision and pattern recognition* (pp. 770–779). http://dx.doi.org/10.1109/CVPR.2019.00086.

Song, S., & Xiao, J. (2016). Deep sliding shapes for amodal 3D object detection in RGB-d images. In *2016 IEEE conference on computer vision and pattern recognition* (pp. 808–816). http://dx.doi.org/10.1109/CVPR.2016.94.

Vora, S., Lang, A. H., Helou, B., & Beijbom, O. (2020). PointPainting: Sequential fusion for 3D object detection. In *2020 IEEE/CVF conference on computer vision and pattern recognition* (pp. 4603–4611). http://dx.doi.org/10.1109/CVPR42600.2020.00466.

Wang, Z., & Jia, K. (2019). Frustum ConvNet: Sliding frustums to aggregate local point-wise features for amodal 3D object detection. In *2019 IEEE/RSJ international conference on intelligent robots and systems* (pp. 1742–1749). http://dx.doi.org/10.1109/IROS40897.2019.8968513.

Wang, Z., Zhan, W., & Tomizuka, M. (2018). Fusing bird's eye view lidar point cloud and front view camera image for 3d object detection. In *2018 IEEE intelligent vehicles symposium (IV)* (pp. 1–6). IEEE, http://dx.doi.org/10.1109/IVS.2018.8500387.

Wang, J., Zhu, M., Sun, D., Wang, B., Gao, W., & Wei, H. (2019). MCF3D: Multi-stage complementary fusion for multi-sensor 3D object detection. *IEEE Access*, *7*, 90801–90814. http://dx.doi.org/10.1109/ACCESS.2019.2927012.

Woo, S., Park, J., Lee, J.-Y., & Kweon, I. S. (2018). Cbam: Convolutional block attention module. In *Proceedings of the european conference on computer vision* (pp. 3–19). http://dx.doi.org/10.1007/978-3-030-01234-2_1.

Xie, L., Xiang, C., Yu, Z., Xu, G., Yang, Z., Cai, D., & He, X. (2020). PI-RCNN: An efficient multi-sensor 3D object detector with point-based attentive cont-conv fusion module. In *Proceedings of the AAAI conference on artificial intelligence (vol. 34)* (pp. 12460–12467). http://dx.doi.org/10.1609/aaai.v34i07.6933.

Xu, D., Anguelov, D., & Jain, A. (2018). PointFusion: Deep sensor fusion for 3D bounding box estimation. In *2018 IEEE/CVF conference on computer vision and pattern recognition* (pp. 244–253). http://dx.doi.org/10.1109/CVPR.2018.00033.

Yan, Y., Mao, Y., & Li, B. (2018). Second: Sparsely embedded convolutional detection. *Sensors*, *18*(10), 3337. http://dx.doi.org/10.3390/s18103337.

Yang, B., Luo, W., & Urtasun, R. (2018). PIXOR: Real-time 3D object detection from point clouds. In *2018 IEEE/CVF conference on computer vision and pattern recognition* (pp. 7652–7660). http://dx.doi.org/10.1109/CVPR.2018.00798.

Yang, Z., Sun, Y., Liu, S., & Jia, J. (2020). 3DSSD: Point-based 3D single stage object detector. In *2020 IEEE/CVF conference on computer vision and pattern recognition* (pp. 11037–11045). http://dx.doi.org/10.1109/CVPR42600.2020.01105.

Yoo, J. H., Kim, Y., Kim, J., & Choi, J. W. (2020). 3D-cvf: Generating joint camera and lidar features using cross-view spatial feature fusion for 3d object detection. In *Computer vision–ECCV 2020: 16th European conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16* (pp. 720–736). Springer, http://dx.doi.org/10.1007/978-3-030-58583-9_43.

Yu, J., Jiang, Y., Wang, Z., Cao, Z., & Huang, T. (2016). Unitbox: An advanced object detection network. In *Proceedings of the 24th ACM international conference on multimedia* (pp. 516–520). http://dx.doi.org/10.1145/2964284.2967274.

Zhang, X., Zou, Z., Li, Z., Liu, H., & Li, J. (2020). Deep multi-modal fusion in object detection for autonomous driving. *CAAI Transactions on Intelligent Systems*, *15*(4), 758–771. http://dx.doi.org/10.11992/tis.202002010.

Zhao, X., Liu, Z., Hu, R., & Huang, K. (2019). 3D object detection using scale invariant and feature reweighting networks. In *Proceedings of the AAAI conference on artificial intelligence (vol. 33)* (pp. 9267–9274). http://dx.doi.org/10.1609/aaai.v33i01.33019267.

Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., & Ren, D. (2020). Distance-IoU loss: Faster and better learning for bounding box regression. In *Proceedings of the AAAI conference on artificial intelligence (vol. 34)* (pp. 12993–13000). http://dx.doi.org/10.1609/aaai.v34i07.6999.

Zhou, D., Fang, J., Song, X., Guan, C., Yin, J., Dai, Y., & Yang, R. (2019). Iou loss for 2D/3D object detection. In *2019 International conference on 3D vision* (pp. 85–94). http://dx.doi.org/10.1109/3DV.2019.00019.

Zhou, Y., & Tuzel, O. (2018). VoxelNet: End-to-end learning for point cloud based 3D object detection. In *2018 IEEE/CVF conference on computer vision and pattern recognition* (pp. 4490–4499). http://dx.doi.org/10.1109/CVPR.2018.00472.

Zhu, M., Ma, C., Ji, P., & Yang, X. (2021). Cross-modality 3D object detection. In *2021 IEEE winter conference on applications of computer vision* (pp. 3771–3780). http://dx.doi.org/10.1109/WACV48630.2021.00382.