Fine-Grained Multilevel Fusion for Anti-Occlusion Monocular 3D Object Detection

He Liu, Huaping Liu, Yikai Wang, Fuchun Sun, *Fellow, IEEE*, and Wenbing Huang

***Abstract*— We propose a deep fine-grained multi-level fusion architecture for monocular 3D object detection, with an additionally designed anti-occlusion optimization process. Conventional monocular 3D object detection methods usually leverage geometry constraints such as keypoints, object shape relationships, and 3D to 2D optimizations to offset the lack of accurate depth information. However, these methods still struggle against directly extracting rich information for fusion from the depth estimation. To solve the problem, we integrate the monocular 3D features with the pseudo-LiDAR filter generation network between fine-grained multi-level layers. Our network utilizes the inherent multi-scale and promotes depth and semantic information flow in different stages. The new architecture can obtain features that incorporate more reliable depth information. At the same time, the problem of occlusion among objects is prevalent in natural scenes yet remains unsolved mainly. We propose a novel loss function that aims at alleviating the problem of occlusion. Extensive experiments have proved that the framework demonstrates a competitive performance, especially for the complex scenes with occlusion.**

***Index Terms*— Monocular 3D object detection, anti-occlusion, fine-grained multi-level.**

I. Introduction

THIS paper focuses on accurate 3D object detection based on a monocular image. Monocular 3D object detection is particularly essential and fundamental in robotics, autonomous driving, and machine cognition, involving diverse applications ranging from object detection and robot grasping to advanced augmented reality and automatic storage. One critical point of 3D object detection from a monocular image is to compute the 3D information, such as the depth between the object and the camera. Compared to using the point cloud data, multi-view geometry images, and CAD data, estimating 3D object information with a monocular image is very challenging due to the limited visual information available. In other words, the task remains primarily unsolved, though it has drawn much attention.

Manuscript received October 17, 2021; revised March 22, 2022 and May 6, 2022; accepted May 10, 2022. Date of publication June 9, 2022; date of current version June 15, 2022. This work was supported in part by the National Natural Science Fund for Distinguished Young Scholars under Grant 62025304. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Daniel L. Lau. *(Corresponding author: Huaping Liu.)*

The authors are with the Beijing National Research Center for Information Science and Technology, Institute for Artificial Intelligence, and the Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, China (e-mail: hpliu@mail.tsinghua.edu.cn).

Digital Object Identifier 10.1109/TIP.2022.3180210

Fig. 1. Sample numbers of different occlusion levels are collected based on the KITTI dataset. Occlusion happens all the time in real traffic, such as traffic jams, narrow roads, the parallel cars changing lanes. We categorize the level of occlusion into four classes: non-occlusion, part-occlusion, heavy-occlusion, and unknown.

Recent monocular 3D object detection can be roughly divided into two pipelines, including image-based methods [1]–[3] and pseudo-LiDAR methods [4]–[6]. The image-based methods utilize the relationship between the 2D bounding box and properties of a 3D object as the geometry constraints. These constraints are formulated as different convex optimization problems and different terms in loss function to improve detection results. The pseudo-LiDAR methods consider traditional 2D convolutional as the bottleneck as it cannot capture accurate depth information. Hence pseudo-LiDAR methods usually convert the RGB images into pseudo-LiDAR maps to mimic the LiDAR signal.

However, image-based methods still struggle with representing 3D structures, capturing meaningful local object scale in 3D, and structure information because of the following two factors. (1) Due to the loss of depth dimension in monocular images, the two objects with the same shape in the different distances can be seen at a different scale in the image coordinate. It is difficult for traditional 2D convolutional kernels to learn the objects with different scales simultaneously. (2) The two objects have overlap in the image coordinates. They are hard to distinguish given the monocular condition, as there is not enough physical information like depth in a natural scene. In that case, a convolutional filter is hard to distinguish whether a pixel belongs to the foreground or the background.

Although several promising pseudo-LiDAR approaches [4]–[6] have been proposed to give new perspectives, they remain two key issues. (1) The pseudo-LiDAR approaches heavily rely on the accuracy of the estimated depth maps. However, the point clouds generated by the monocular images are always coarse and far from accurate, resulting in inaccurate 3D predictions. In other words, the coarse and inaccurate pseudo-LiDAR is the bottleneck of monocular 3D object detection. (2) Although pseudo-LiDAR methods are good at extracting spatial information, it is difficult for them to employ high-level semantic information effectively. Unreliable features generate false-positive samples and reduce the performance of monocular 3D object detection.

Apart from the issue discussed above, different occlusion levels are tricky for monocular 3D object detection in real traffic. Most of the researches, such as [1], [3], [7], regard all the objects in different occlusion levels in the same way. In some datasets, such as KITTI, the sample numbers of different occlusion levels are imbalanced. The sample numbers of non-occlusion are always more extensive than that of part-occlusion or heavy-occlusion. Hence, part-occlusion and heavy-occlusion can be complex examples, and non-occlusion is easy. The network should pay more attention to complex examples to get better performance.

We propose a novel structure termed fine-grained multi-level fusion (HMF) to address the first two problems. These structures can compensate for the surrounding, spatial, and local scale information that cannot be learned in one branch. Besides, the structure of fine-grained multi-level layers utilizes the inherent multi-scale and promotes the depth and semantic information flow in different stages. We propose a novel loss termed anti-occlusion loss to address the third problem. The new loss can pay more attention to the hard examples and automatically adjust the importance and weight of the generated object during training.

In summary, our main contributions to this paper are the following:

- We propose a novel component for 3D object detection, termed fine-grained multi-level fusion. The fine-grained multi-level fusion structure can capture more reliable features, superior in obtaining features with more accurate depth information and more abundant semantic information.
- We propose a novel loss for occlusion in the actual scene, termed anti-occlusion loss. The new loss can automatically adjust the weight of the estimated object during training.
- We design a one-stage network architecture for monocular 3D object Detection surpassing all current methods on the challenging dataset.

II. Related Work

A. Pseudo-LiDAR Based Monocular 3D Detection

Previous pseudo-LiDAR monocular methods [4]–[6] convert the image-based depth maps to pseudo-LiDAR representations for mimicking the LiDAR signal. The pseudo-LiDAR monocular methods utilize LiDAR-based 3D detection. The problem transfers to use point cloud data to predict the 3D detection. For example, [4] first predicts the depth map from given monocular images, followed by back-projection it into a 3D point cloud in the LiDAR coordinate system. After obtaining the pseudo-LiDAR representations, the network can process the input exactly like LiDAR - any LiDAR-based 3D detection method can be applied. [5] leverages a standalone module to transform the input data from the 2D image plane to the 3D point clouds space for a better input representation, then perform the 3D detection using PointNet backbone net to obtain objects' 3D information. [5] proposes a multi-modal features fusion module to embed the complementary RGB cue into the generated point clouds representation. However, converting the monocular image to pseudo-LiDAR causes a depth loss of accuracy, and the methods rely heavily on the accuracy of the depth map. In contrast, our method considers pseudo-LiDAR representations as filter kernels to learn better 3D information from RGB images.

B. Image-Based Monocular 3D Detection

In the previous works, image-based monocular 3D detection methods use a similar framework with 2D detection [8], [9]. However, the methods are struggled with estimating the 3D coordinates (x, y, z) of the object center since the monocular image cannot decide the absolute physical location. Hence, previous monocular 3D detection methods [2], [3], [10] usually make assumptions about the relative geometry between the object in the scene and use the relative geometry as a constraint to train the 2D to 3D mapping. [3] encode spatial constraints for partially-occluded objects from their adjacent neighbors. The proposed detector computes uncertainty-aware predictions for object locations and 3D distances for the adjacent object pairs, subsequently jointly optimized by nonlinear least squares. MDPCNN [11] utilizes a fusion mechanism to combine deep learning features from different inputs. [1] predicts the nine perspective key points of a 3D bounding box in image space and then utilizes the geometric relationship of 3D and 2D perspectives to recover the dimension, location, and orientation in 3D space. [12] infer the 3D IoU between the 3D proposals and the object solely based on 2D cues. To introduce more prior information, [13]–[15] design different 3D anchor templates to get better object geometry. [16] proposes aggregate losses, including a novel projection alignment loss, and generates a point cloud in an object-centered coordinate system to learn local scale and shape information. [17] proposes a novel loss formulation by lifting 2D detection, orientation, and scale estimation into 3D space and uses 3D synthetic data augmentation via inpainting recovered meshes directly onto the 2D scenes. However, as 2D image features struggle with representing 3D structures, the above geometric constraints are not easy to restore accurate 3D information of objects from a single monocular image. Hence, our motivation is to utilize the depth information, which essentially complements the information between 2D and 3D representation, to guide learning the 2D to 3D feature representation.

C. Dynamic Networks

Some existing techniques are proposed to exploit the depth information for monocular 3D detection. [10] proposes depth-aware convolution, which designs several fixed non-shared kernels in the row-space to learn depth information spatially. However, the fixed and non-shared kernels can only extract the spatial division, failing to capture depth information accurately. [18] uses the sample-specific and position-specific filters to exploit more surrounding information; however, the method also fails to solve the problem of 2D convolutions in monocular 3D object detection. [7], [19], [20] are also fail to capture object scale and local depth information as well. [21] designs a structure termed depth-guided dynamic depthwise dilated LCN (D4LCN), where the filers and their receptive fields can be automatically learned. However, the method does not let the depth of information flow through different levels. [22], [23] are methods for exploring interaction of features in multi-modal fusion. However, our method incorporates dot product fusion, which introduces the quadratic feature interaction of the modal channel itself. In this work, our fine-grained multi-level fusion for anti-occlusion monocular 3D object detection is proposed to solve the two problems associated with the occlusion between the object and let the depth information more circulating between 2D convolution and pseudo-LiDAR based 3D processing.

D. Occlusion Handling

Occlusion is widespread in pedestrian-related tasks. Many loss algorithms [24]–[26] for pedestrian detection are designed to solve the problem of occlusion in crowded people. Bayesian loss [24] builds a density contribution probability model from the point annotations to solve the crowd counting. Repulsion loss [27] designs two terms named the attraction term and the repulsion term to solve the occlusion. The attraction term is used to shrink the gap between a proposal and its associated target. The repulsion term is designed for pushing away surrounding non-target objects. [25], [26] utilize the three-dimensional geometric relationship in space for mathematical modeling, establish the characteristics of the occlusion module, and improve the accuracy of the occlusion problem. Our anti-occlusion loss are motivated by [28], [29]. The methods of [28], [29] improve the imbalanced distribution of hard proposals in singe stage object detection.

III. Methodology

In this section, we detail our proposed one-stage 3D detector for monocular 3D object detection, which is comprised of two key components: a fine-grained multi-level fusion structure (see Figure 2), and an anti-occlusion optimization design.

A. Fine-Grained Multi-Level Fusion Structure

As shown in Figure 2, our overall network architecture can be summarized into a two-branch network: a feature extraction network and a filter generation network. The feature extraction network takes an input $I \in \mathbb{R}^{h\times w\times c}$ and output $I_n \in \mathbb{R}^{h_n\times w_n\times c_n}$, where h, w, c are the height, width and number of channels of the input I, respectively. h_n, w_n, c_n are height, width, and number of channels of the output I_n at layer n, respectively. The monocular depth estimation is applied to input $I \in \mathbb{R}^{h\times w\times c}$ to generate an depth map $D = H(I)$, with $D \in \mathbb{R}^{h\times w\times c/3}$. $H(\cdot)$ is monocular depth estimation network. We concatenate D three times on the third channel to let the D_n have the same channel with I_n and we can get $D^* \in \mathbb{R}^{h\times w\times c}$. The filter generation network takes an input $D^* \in \mathbb{R}^{h\times w\times c}$. It outputs activation $F_n \in R^{h_n\times w_n\times c_n}$. Different filters are applied to the different position outputs of the feature extraction network $I_n \in \mathbb{R}^{h_n\times w_n\times c_n}$: for each position (i, j) of the input I_n, a specific local filter $F_n{}^{(i,j)}$ is applied to the region centered around I_n to generate an output:

$$G_n^{(i,j)} = F_n{}^{(i,j)} \otimes I_n{}^{(i,j)}, \tag{1}$$

with $G_n \in R^{h_n\times w_n\times c_n}$. $\otimes$ means the operation of element-wise multiplication.

We propose the channel swap operation to encourage the interaction of feature extraction network and filter generation network information flow across channels. As shown in Figure 3, when given two inputs I_n and F_n, the channel swap fuses two inputs by exchanging inputs corresponding to half of the channels. The channel swap operation can be described as:

$$\begin{aligned} J(I_n, F_n) &= I_n[1, \ldots, \frac{c_n}{2}]||F_n[\frac{c_n}{2}, \ldots, c_n] \\ J(F_n, I_n) &= F_n[1, \ldots, \frac{c_n}{2}]||I_n[\frac{c_n}{2}, \ldots, c_n], \end{aligned} \tag{2}$$

where $J(\cdot,\cdot)$ is a fused operation between two inputs, $||$ indicates channel-wise concatenation.

After the channel swap operation, we can get feature pixels $\hat{I}_n \in R^{h_n\times w_n\times c_n}$ within the feature extraction map and filter pixels $\hat{F}_n \in R^{h_n\times w_n\times c_n}$ within the filter generation map. For simplicity, we omit subscript n in the following text. Following by the 1×1 convolutional network and activation function, we can get the $z^{k'} \in R^{h\times w}$ which is obtained after the channel swap module operation. k' indicates the k'-th channel of the output activation.

$$z^{k'} = (\sum_{k=1}^{\frac{c}{2}} \beta^k I^k + \sum_{k=\frac{c}{2}+1}^{c} \beta^k F^k) \otimes (\sum_{k=1}^{\frac{c}{2}} \delta^k F^k + \sum_{k=\frac{c}{2}+1}^{c} \delta^k I^k) \tag{3}$$

where β^k and δ^k are weights of the k'-th filters in the 1×1 convolutional layers.

From equation 3, we can conclude that our channel swap module is feature quadratic interactions. Besides, compared with the method not applying our channel swap module, our channel swap module could have feature quadratic interactions within the modality. The Figure 4 illustrates a sketched comparison.

According to the formulation, the channel swap module carries no additional parameters and is FLOP-efficient to the multi-fusion part. The channel swap module introduces channel-wise feature and filter interactions across inputs, making the networks more effective for information flow.

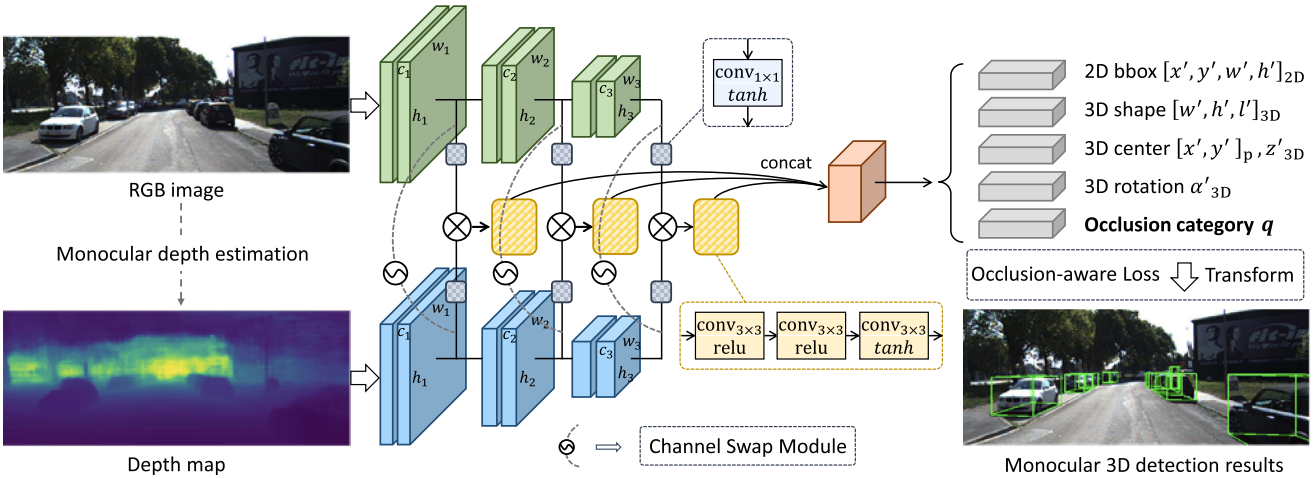

Fig. 2. Overview of the proposed network architecture approach for monocular 3D object detection. The depth map is first generated from the single monocular RGB image using method [30]. The fine-grained multi-level fusion structure is used to fuse two networks of each stage. The details of the feature channel swap module are shown in Figure 3. A one-stage detection head with Non-Maximum Suppression (NMS) and the transformation between the estimated parameters and the parameters in the camera coordinate system is used after the fine-grained multi-level fusion structure in the proposed network.

Section "Experiment" illustrates that our fine-grained multi-level fusion structure could improve predictions for different occlusion level objects.

We consider integrating the fine-grained multi-level fusion structure into convolutional networks. We adopt residual blocks of ResNet [31] as an example. We use the last feature map of every stage as the input of every level fine-grained multi-level fusion structure. After the ResNet network architecture, the network always uses the $ReLU$ activation function and the dropout layer after the convolutional and batch norm layers. If we use the feature after the operation of the dropout layer fuse with the filter kernels, many filter kernels would equal zero causing a considerable amount of information loss. Hence, we add a 1×1 convolution after the dropout layer, followed by the $Tanh$ activation function instead of $ReLU$. Because of the $Tanh$ activation function characters, few feature maps or the kernel filters are equal to zeros.

The filter generation network is trained and generated by the depth map based on a monocular image. Hence, it is both sample-specific and position-specific, capturing the surrounding depth information. In the fine-grained multi-level fusion structure, the depth feature maps are considered the filter kernels of the feature extraction network. The features generated by the feature extraction network are mutually considered the filter kernels of the depth feature maps. Hence, different kernels can have different functions. These structures can compensate for the surrounding, spatial, and local scale information that cannot be learned in one branch. Besides, the structure of fine-grained multi-level layers utilizes the inherent multi-scale and promotes the depth and semantic information flow in different stages. We can also understand the element-wise product operation of the network in another more intuitive way: the network tends to respond more obviously to the feature pixels, which have high responsiveness both in the feature extraction network and filter generation network.

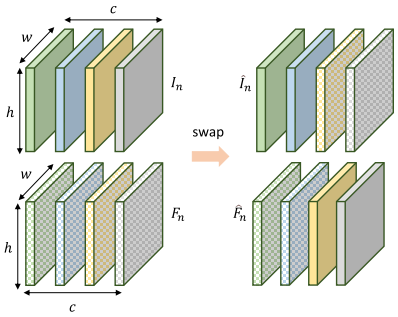

Fig. 3. Proposed channel swap module that are part of the fine-grained multi-level Fusion structure. In the figure, we utilize a hat sign to represent the fused features and filters, i.e. $\hat{I}_n = J(I_n, F_n)$, $\hat{F}_n = J(F_n, I_n)$. Channel swap module exchanges half of features and filters with respect to the same-indexed channels.

Our network architecture has a new branch that uses another 1×1 convolutional layer to generate $\boldsymbol{q}$, used in the next section. Besides the estimated parameters shown in Figure 2, no other parameters are predicted during training. However, we still have the best performance on the benchmark, illustrating our proposed network architecture's consistent effectiveness.

B. Anti-Occlusion Optimization

In this part, we propose an anti-occlusion optimization process with an anti-occlusion loss designed to address the one-stage monocular 3D object detection scenario with different occlusion levels. We introduce the anti-occlusion loss from the cross-entropy (CE) loss for multi-class occlusion classification. Given a probability vector $\boldsymbol{p} = \{p_y\}_{y=1}^{C}$, a multi-class CE loss with $softmax$ as activation function can be written as:

$$CE(\boldsymbol{p}, y) = -\log p_y = -\log \frac{\exp(f_y)}{\sum_{c=1}^{C} \exp(f_c)} \tag{4}$$

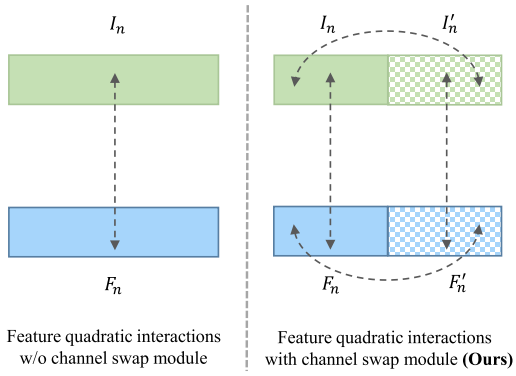

Fig. 4. A sketched comparison of feature quadratic interactions when not applying and applying the channel swap module. The channel swap module could not only have feature quadratic interactions between two modalities but also have feature quadratic interactions within one modality.

where $f = \{f_y\}_{y=1}^{C} \in \mathbb{R}^{\mathbb{C}}$ is the logit vector produced by the last fully-connected layer applying the linear classifier to the feature vector. $y \in \{1, \ldots, C\}$ specifies the ground-truth class label. C is the number of categories.

1) Balanced Occlusion Loss: A standard method for addressing occlusion in 3D object detection is introducing a new occlusion loss. A weighting factor $\alpha \in [0, +\infty)$ is introduced to balance the classification and occlusion losses. In practice, α may be treated as a hyperparameter, and we write the balanced occlusion loss as:

$$L_{balanced\_occ} = \alpha CE(\boldsymbol{p}, y) + (1 - \alpha) CE(\boldsymbol{q}, y'), \quad (5)$$

where $\boldsymbol{q}$ is the occlusion probability vector. y' specifies the ground-truth occlusion label.

This loss is a simple extension to CE that we consider an experimental baseline for our proposed anti-occlusion loss.

2) Anti-Occlusion Loss: From Equation 5, we can find the term $CE(\boldsymbol{p}, y)$ and the term $CE(\boldsymbol{q}, y')$ are two individual loss without any relationship. The hyperparameter α is fixed during training and cannot adjust according to different level occlusion samples. Easily classified non-occlusion samples comprise most of the loss and dominate the gradient. Motivated by [28], [29], while α balances the importance of $CE(\boldsymbol{p}, y)$ loss and $CE(\boldsymbol{q}, y')$ loss, it does not differentiate between non-occlusion examples and different level occlusion examples. Instead, we propose to reshape the classification loss function to up-weighted hard level occlusion examples and focus training on hard level occlusion examples.

More formally, we propose to add a modulating factor $\phi(\boldsymbol{q})$ to the classification loss, with $\boldsymbol{q}$ estimated by the network. We define the anti-occlusion loss as:

$$L_{class}(\boldsymbol{p}, \boldsymbol{q}) = -\phi(\boldsymbol{q}) \log p_y - \log q_y, \quad (6)$$

where $\phi(\boldsymbol{q})$ is defined as:

$$\phi(\boldsymbol{q}) = \boldsymbol{w}^{\boldsymbol{T}} \boldsymbol{q}, \quad (7)$$

where $\boldsymbol{w}$ is the weight vector of different level occlusion. We note two properties of the anti-occlusion loss. (1) When an example is heavy level occlusion, the modulating factor increases, affecting the loss. While an example is non-occlusion, the modulating factor is closed to 1, and the loss is unaffected. As shown in Figure 1, we assume that the occlusion category is non-occlusion, slight occlusion, and heavy occlusion, which are annotated as 1, 2 and 3, respectively. Then the value $\boldsymbol{w} = [0, 1, 2, 3]$, where 0 means the background category. If the model's estimated occlusion probability vector is $\boldsymbol{q} = [0, 0.1, 0.8, 0.1]$ which means the model consider the proposal as slight occlusion, we can get $\phi(\boldsymbol{q}) = 2.0$. The modulating factor can adjust the weight according to the level of occlusion. (2) The modulating factor smoothly adjusts the rate at which non-occlusion examples are down-weighted. If the model's estimated probability is $\boldsymbol{q} = [0, 1.0, 0.0, 0.0]$ which means the model consider the proposal as non-occlusion, we can get $\phi(\boldsymbol{q}) = 1$. The weight term of $\log p_y$ is degraded to the term without the modulating factor $\phi(\boldsymbol{q})$.

In conclusion, the modulating factor extends the loss contribution from occlusion objects and reduces the influence in which an example is easy to detect. In our experiment, learned from [28], we use the loss layer's implementation combining the $softmax$ operation for computing $\boldsymbol{p}$ and $\boldsymbol{q}$ with the loss computation, resulting in more excellent numerical stability.

3) Overall Loss: Our overall loss contains the anti-occlusion loss for classification, a 2D regression loss, a 3D regression loss:

$$L = L_{class} + L_{2D} + L_{3D}, \quad (8)$$

L_{class}, L_{2D}, L_{3D} are the anti-occlusion loss, 2D regression loss and 3D regression loss, respectively.

In our work, for both 2D regression loss and 3D regression loss, we use the traditional $Smooth_{L1}$ regression losses. The specific definition is as follows:

$$L_{2D} = \text{Smooth}_{L1}\left(\left[x', y', h', w'\right]_{2D}, [x, y, h, w]_{2D}\right), \quad (9)$$

where $\left[x', y', h', w'\right]_{2D}$ denotes the estimated 2D information. $[x, y, h, w]_{2D}$ denotes the corresponding ground truth. x, y means the upper left corner coordinates, taking the upper left corner of the image as the origin of the coordinate axis. h, w means the weight and height of the object.

$$L_{3D} = \text{Smooth}_{L1}\left(\left[w', h', l', z', \alpha\right]_{3D}, [w, h, l, z, \alpha]_{3D}\right) + \text{Smooth}_{L1}\left(\left[x', y'\right]_{p}, [x, y]_{p}\right), \quad (10)$$

where $[x, y]_p$ denotes the projected center in image coordinates of the ground truth 3D box, z is its ground truth depth. $[w, h, l]_{3D}$ denotes the weight, height, length of the object in image coordinates. $[\alpha]_{3D}$ is the relative rotation to the camera.

Besides the terms in Equation 8, we do not use any other losses for training. In this way, we reduce the number of parameters and the complexity of the model.

IV. Experiments

In order to verify the generalization and consistent effectiveness of our proposed network architecture, we conduct experiments with two criteria, including 3D object detection and bird's eye view. The benchmark is performed on the KITTI dataset and ApolloCar3D with natural traffic scenes, mainly urban city scenes.

TABLE I

COMPARATIVE RESULTS ON THE KITTI 3D OBJECT DETECTION DATASET. MOST OF THE METHOD ONLY PROVIDES $AP|R_{11}$ IN THE PREVIOUS OFFICIAL EVALUATION METRICS. WE THUS SHOW THE RESULTS ON THE TEST SET IN $AP|R_{40}$ AND SPLIT1/SPLIT2 IN $AP|R_{11}$. WE USE RED TO HIGHLIGHT THE HIGHEST RESULT AND THE RELATIVE IMPROVEMENT COMPARED WITH THE SECOND-HIGHEST RESULT. THE BLUE NUMBERS ARE THE SECOND-HIGHEST RESULT. ALL OF THE RESULTS ARE EVALUATED IN THE CLASS CAR. OUR METHOD ACHIEVES FIVE FIRST AND 1 SECOND IN 6 ITEMS

Method	Test set			Split1			Split2		
	Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
OFT-Net [32]	1.61	1.32	1.00	4.07	3.27	3.29	-	-	-
FQNet [12]	2.77	1.51	1.01	5.98	5.50	4.75	5.45	5.11	4.45
Deep3DBox [33]	4.32	2.02	1.46	-	-	-			
ROI-10D [17]	4.32	2.02	1.46	10.25	6.39	6.18	-	-	-
GS3D [34]	4.47	2.90	2.47	13.46	10.97	10.38	11.63	10.51	10.51
Shift R-CNN [35]	6.88	3.87	2.83	13.84	11.29	11.08	-	-	-
MonoGRNet [2]	9.61	5.74	4.25	13.88	10.19	7.62	-	-	-
MonoPSR [16]	10.76	7.25	5.85	12.75	11.48	8.59	13.94	12.24	10.77
Mono3D-PLiDAR [4]	10.76	7.50	6.10	31.5	21.00	17.50	-	-	-
SS3D [36]	10.78	7.68	6.51	14.52	13.15	11.85	9.45	8.42	7.34
MonoDIS [37]	10.37	7.94	6.40	11.06	7.60	6.37	-	-	-
Pseudo-LiDAR [4]	-	-	-	19.50	17.20	16.20	-	-	-
M3D-RPN [10]	14.76	9.71	7.42	20.27	17.06	15.21	20.40	16.48	13.34
MonoPair [3]	13.04	9.99	8.65	16.28	12.30	10.42	-	-	-
RTM3D [1]	14.41	10.34	8.77	20.77	16.86	16.63	19.47	16.29	15.57
D4LCN [7]	16.65	11.72	9.51	26.97	21.71	18.22	24.29	19.54	16.38
Ours	20.28(+3.63)	13.12(+1.40)	9.56(+0.05)	29.67(+2.70)	22.96(+1.25)	18.97(+0.75)	26.64(+2.35)	22.43(+2.89)	18.50(+2.12)

A. Dataset and Setting

1) KITTI Dataset: The KITTI 3D object detection dataset [38] is extensively used for evaluating monocular 3D object detection. The dataset consists of 7,481 training images and 7518 test images and the corresponding calibration parameters. Our experiments do not need extra data such as point clouds or Bird's eye view images in the KITTI 3D object detection. The dataset comprises 80256 labeled objects, including three object classes: Car, Pedestrian, and Cyclist. The official evaluation code defines each detected object as three difficulty classes (easy, moderate, hard) according to the object heights, occlusion, and truncation levels. If the object height is larger than 40, the occlusion level is 0, and the truncation level is smaller than 0.15, the object is assigned as "Easy." In contrast, if the object height is larger than 25, occlusion levels are 0 or 1, and the truncation level is smaller than 0.30, the object is assigned as "Moderate." If the object height is larger than 25, occlusion levels are 0, 1, or 2, and the truncation level is smaller than 0.50, the object is assigned as "Hard." If the detected object is none of the above, it is assigned as "Unknown." We use two train-val splits of KITTI: the split1 [39] containing 3,712 training and 3,769 validation images for validation while the split2 [15] uses 3,682 images for training and 3,799 images for validation. The dataset provides three tasks about object detection: 2D object detection, 3D object detection, and Bird's eye view, among which 3D object detection is the focus of detection methods.

2) ApolloCar3D Dataset: The ApolloCar3D dataset is about autonomous driving, which was released in 2019. The dataset includes more than 60,000 annotated objects with 5,277 images, and each object has a corresponding CAD model. However, we did not use the CAD model in training and testing. ApolloCar3D dataset is about monocular 6D pose estimation. However, we convert the annotation of 6D pose estimation to monocular 3D object detection since we know the CAD model.

3) Evaluation Metrics: We use the official evaluation, PASCAL criteria, precision-recall curves with the IoU threshold of 0.7 for evaluating our method. Prior to the research of [37], 11-point Interpolated Average Precision (AP) metric $AP|R_{11}$ proposed in the PASCAL VOC benchmark is used. Proposed by [37], the official evaluation adopts the 40 recall positions-based metric $AP|R_{40}$ instead of $AP|R_{11}$, which evaluates the result more accurately. We evaluate our method at three difficulty settings: easy, moderate, and hard, according to the object's occlusion, truncation, and height in the image mentioned above.

4) Implementation Details: Using PyTorch, we train our network with the machine i9-7900X CPU and four 1080Ti GPUs (12G for each). We test our network with one 1080Ti GPU. All networks use ResNet50 as the backbone, initialized by a classification model pre-trained on the ImageNet dataset. The fine-grained multi-level fusion structure is used three times on every stage of the last block of ResNet50. The method by [30], whose model has already been pre-trained based on the monocular images, is used for depth estimation. We pad and resize the original image to 1720×512 for training and testing. We run stochastic gradient descent (SGD) optimizer with a momentum 0.9 and a weight decay 0.0005. The iteration number for the training process is set to 40,000 using the "poly" learning rate policy, the base learning rate to 0.01, and power to 0.9. Because we find our method remains convergent and can continue lifting the performance, we add the extra 10,000 iteration number whose learning rate is not adjusted. Finally, the model costs about one day with batch size eight training.

The backbone of feature extraction network, i.e. ResNet50, is pre-trained based on ImageNet classification dataset [40]. To reduce the parameters, we do not use the FC layers. The max-pooling layer damages the depth information in the deep feature layer; hence, we only use the max-pooling layer in the first block of ResNet50. The filter generation network also uses the ResNet50 backbone and has the same number

TABLE II

COMPARISON OF OUR METHOD TO MONOCULAR 3D DETECTION FRAMEWORKS FOR CAR 3D DETECTION, EVALUATED USING METRIC AP_{BEV} ON THE KITTI SPLIT1

Method	$AP\|R_{11}$			$AP\|R_{40}$		
	Easy	Moderate	Hard	Easy	Moderate	Hard
M3D-RPN [10]	25.94	21.18	17.90	-	-	-
FQNet [12]	9.50	8.02	7.71	-	-	-
D4LCN [7]	34.82	25.83	23.53	31.53	22.58	17.87
Deep3DBox [33]	9.99	7.71	5.30	-	-	-
RTM3D [1]	25.56	22.12	20.91	-	-	-
Ours	37.70	26.99	24.29	34.95	23.93	18.17

TABLE III

ABLATION STUDY ON THE CLASS CAR ON THE KITTI SPLIT1 EVALUATED USING METRIC AP_{3D}. "W/O" MEANS NOT USING LOSS (ANTI-OCCLUSION LOSS) AND HMF (FINE-GRAINED MULTI-LEVEL FUSION STRUCTURE). HMF* MEANS HMF WITHOUT CHANNEL SWAP MODULE

Method	$AP\|R_{11}$			$AP\|R_{40}$		
	Easy	Moderate	Hard	Easy	Moderate	Hard
w/o	26.97	21.71	18.22	22.32	16.20	12.30
w loss	27.71	22.80	18.88	24.69	17.52	13.83
w HMF*	28.08	22.17	18.43	24.86	16.75	12.72
w HMF	29.17	22.73	18.86	25.63	17.31	13.04
w loss+HMF	29.67	22.96	18.97	26.41	17.65	14.01

TABLE IV

ABLATION STUDY ON THE CLASS CAR ON THE KITTI SPLIT1 EVALUATED USING METRIC AP_{BEV}. "W/O" MEANS NOT USING LOSS (ANTI-OCCLUSION LOSS) AND HMF (FINE-GRAINED MULTI-LEVEL FUSION STRUCTURE). HMF* MEANS HMF WITHOUT CHANNEL SWAP MODULE

Method	$AP\|R_{11}$			$AP\|R_{40}$		
	Easy	Moderate	Hard	Easy	Moderate	Hard
w/o	35.39	22.10	17.68	31.99	18.76	14.75
w loss	35.96	26.43	21.24	32.55	22.53	17.48
w HMF*	36.16	25.85	19.14	33.25	20.42	16.86
w HMF	37.61	26.87	21.41	34.68	22.94	17.77
w loss+HMF	37.70	26.99	24.29	34.95	23.93	18.71

TABLE V

WE COMPARE DIFFERENT FUSION METHODS FOR CAR 3D DETECTION ON THE KITTI SPLIT1. THE "CONCATENATE" METHOD DIRECTLY CONCATENATES THE FEATURE MAPS OF THE FEATURE EXTRACTION AND FILTER GENERATION NETWORKS. THE "ADD" METHOD USES AN ELEMENT-WISE "ADD" OPERATION BETWEEN THE OUTPUT OF TWO BRANCH NETWORKS

Method	$AP\|R_{11}$			$AP\|R_{40}$		
	Easy	Moderate	Hard	Easy	Moderate	Hard
Concatenate	24.64	21.10	17.95	21.60	15.98	11.67
Add	24.94	19.92	15.84	21.95	15.31	10.39
D4LCN [7]	26.97	21.71	18.22	22.32	16.20	12.30
HMF*	28.08	22.17	18.43	24.86	16.75	12.72
HMF	29.17	22.73	18.86	25.63	17.31	13.04

TABLE VI

WE COMPARE 3D DETECTION AP_{3D} BASED ON DIFFERENT TRAINING SAMPLE WEIGHTS FOR THE CAR CATEGORY ON THE KITTI SPLIT1. N, P, H ARE SHORT FOR THE NON-OCCLUSION, PART OCCLUSION, AND HEAVY OCCLUSION. N MEANS WE DO NOT USE ANTI-OCCLUSION DURING TRAINING, WHILE P MEANS WE ONLY ADD THE TRAINING SAMPLE OF PART-OCCLUSION WEIGHT

Method	$AP\|R_{11}$			$AP\|R_{40}$		
	Easy	Moderate	Hard	Easy	Moderate	Hard
N	26.97	21.71	18.22	22.32	16.20	12.30
P	24.30	22.27	18.40	23.49	16.78	12.67
N+P	27.71	22.80	18.88	24.69	17.52	13.83
N+P+H	24.36	22.44	18.79	23.73	17.06	12.82

TABLE VII

COMPARISON OF OUR METHOD TO MONOCULAR 3D DETECTION FRAMEWORKS FOR CAR 3D DETECTION, EVALUATED USING METRIC AP_{3D} ON THE APOLLOCAR3D

Method	$AP\|R_{11}$	$AP\|R_{40}$
Pseudo-LiDAR [4]	10.64	9.66
M3D-RPN [10]	12.48	11.57
D4LCN [7]	14.72	13.91
Ours	18.85(+4.13)	16.46(+2.55)

of channels and blocks as the feature extraction network. However, there is a little difference between them: In the last layer of each block of the filter generation network, we use the activation function $Tanh$ instead of $ReLU$ because of the fine-grained multi-level fusion character. To make the filter generation network obtain a larger reception field and keep the filter feature's size, we set the dilation rate 2, padding two, and stride 1 in the last convolutional layer of ResNet50.

B. Comparative Results

We conduct experiments on two splits of the validation set of the KITTI dataset and the ApolloCar3D dataset. Table I includes the top 16 monocular methods in the leaderboard. We achieve five first and 1 second in 6 items. We can observe that: (1) Our method outperforms the second-best competitor based on monocular 3D car detection by a large margin (relatively 11.9% for 13.12 vs. 11.72 in KITTI test set, 5.8% for 22.96 vs. 21.71 in KITTI split1 and 14.8% for 22.43 vs. 19.54 in KITTI split2) under the moderate setting, which considers as the most crucial setting of KITTI. (2) Our model is trained end-to-end using the standard ImageNet pre-trained model, unlike most detectors, such as [6], [10], [35], [37], pre-train on COCO/KITTI or use the data augmentation such as multi-stage. However, we still achieve the state-of-the-art 3D detection results, validating our structure and loss's effectiveness to learn 3D information. (3) After Aug. 2019, the official code uses $AP|R_{40}$ instead of $AP|R_{11}$, however, all existing methods except D4LCN report the results under the old metric $AP|R_{11}$. For a fair comparison, all existing methods report the results under the $AP|R_{11}$ on the validation set.

We also compare our method with current image-based SOTA approaches on AP_{BEV} whose precision-recall curves are for Birds Eye View, as shown in Table II. However, it is not realistic to list $AP|R_{40}$ of all previous methods because most of them are published before the metric of $AP|R_{40}$. From Table II, our method outperforms the second-best competitor for monocular 3D bird-eye view by a large margin (we get $+2.88$, $+1.16$, $+0.76$, $+3.42$, $+1.35$, $+0.84$ improvement on the KITTI split1, respectively). From Table VII, our method lift up 4.13, 2.55 points on the task of 3D detection in ApollorCar3D dataset.

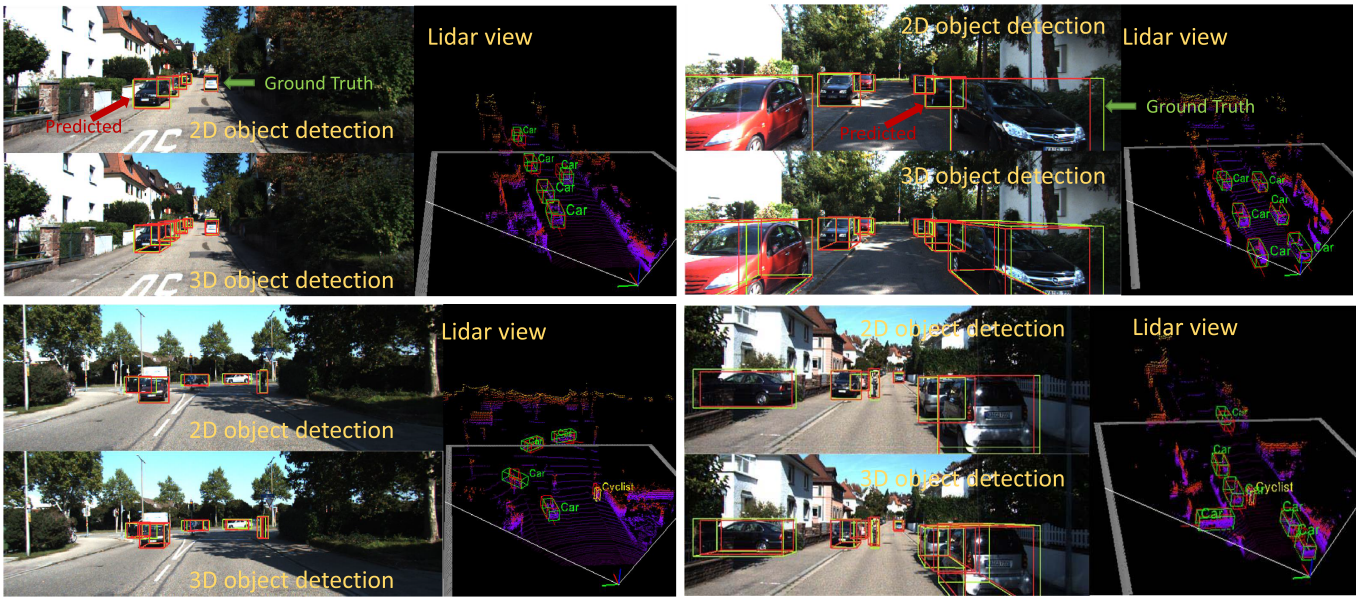

Fig. 5. Qualitative Results on KITTI. The ground truth of the car category is in green, and the ground truth of the cyclist category is in yellow, while predicted 3D bounding box results are drawn in red. The Lidar point cloud data are only plotted for reference but not used in our method. For each scene, we visualize the 2D object detection (top left), monocular 3D object detection (bottom left), and 3D localization shown in lidar point cloud (black right).

TABLE VIII

COMPARISON OF OUR METHOD TO MONOCULAR 3D DETECTION FRAMEWORKS FOR CAR 3D DETECTION, EVALUATED USING METRIC AP_{BEV} ON THE APOLLOCAR3D

Method	$AP\|R_{11}$	$AP\|R_{40}$
Pseudo-LiDAR [4]	16.49	12.47
M3D-RPN [10]	22.19	19.76
D4LCN [7]	27.15	24.36
Ours	31.76(+4.61)	28.34(+3.98)

TABLE IX

RUNTIME COMPARISON OF OUR METHOD TO MONOCULAR 3D DETECTION FRAMEWORKS FOR CAR 3D DETECTION ON THE KITTI AND APOLLOCAR3D DATASET

Method	$Runtime_{kitti}$ (sec./image)	$Runtime_{apollo}$ (sec./image)
Pseudo-LiDAR [4]	0.49	0.67
M3D-RPN [10]	0.20	0.35
D4LCN [7]	0.22	0.40
Ours	0.16	0.25

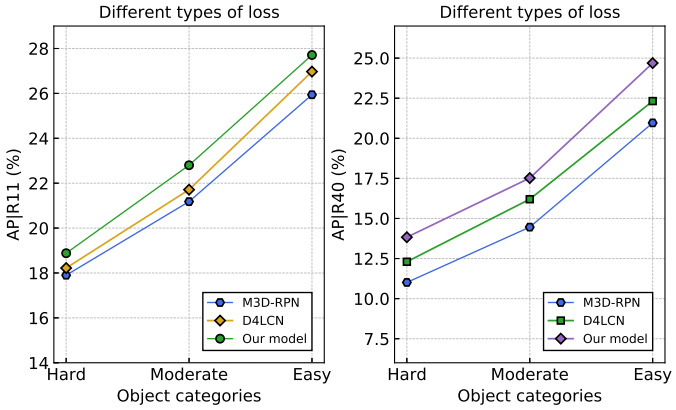

Fig. 6. Ablation Study. According to the occlusion level, the KITTI split1 dataset can be split into three categories: easy, moderate, and hard. We compare different methods in three categories. Enabled by the anti occlusion loss, our simple monocular 3D detector outperforms previous detectors. We show variants with different types of loss.

C. Ablation Study

This section provides ablation studies to verify the validity of our proposed components, i.e., the fine-grained multi-level fusion architecture and the anti-occlusion losses.

1) Validity for Fine-Grained Fusion Structure: We make a comparison among six versions of our model: (1) w/o: the baseline model without using anti-occlusion loss and fine-grained multi-level fusion structure (HMF); (2) the baseline model using anti-occlusion loss without using HMF; (3) using HMF without using channel swap module; (4) using HMF with channel swap module; (5) both using anti-occlusion loss and HMF with channel swap module.

From Table III and Table IV, we can observe that: (1) The performance continuously increases when the two novel components are used for 3D object detection and bird's-eye view, showing the effectiveness of each component. (2) Our anti-occlusion loss increases the 3D detection AP scores of moderate and hard from 21.71, 16.20 to 22.80, 17.52 and from 18.22, 12.30 to 18.88, 13.83 w.r.t the $AP|R_{11}$ and $AP|R_{40}$ metrics, respectively. Our loss also lifts the bird's-eye view scores of moderate and hard from 22.10, 18.76 to 26.43, 22.53 and from 17.68, 14.75 to 21.24, 17.48 w.r.t the $AP|R_{11}$ and $AP|R_{40}$ metrics, respectively. The performance of only using the HMF structure is worse than that of only using anti-occlusion loss on the 3D detection AP scores of moderate and hard. These suggest that our proposed loss effectively

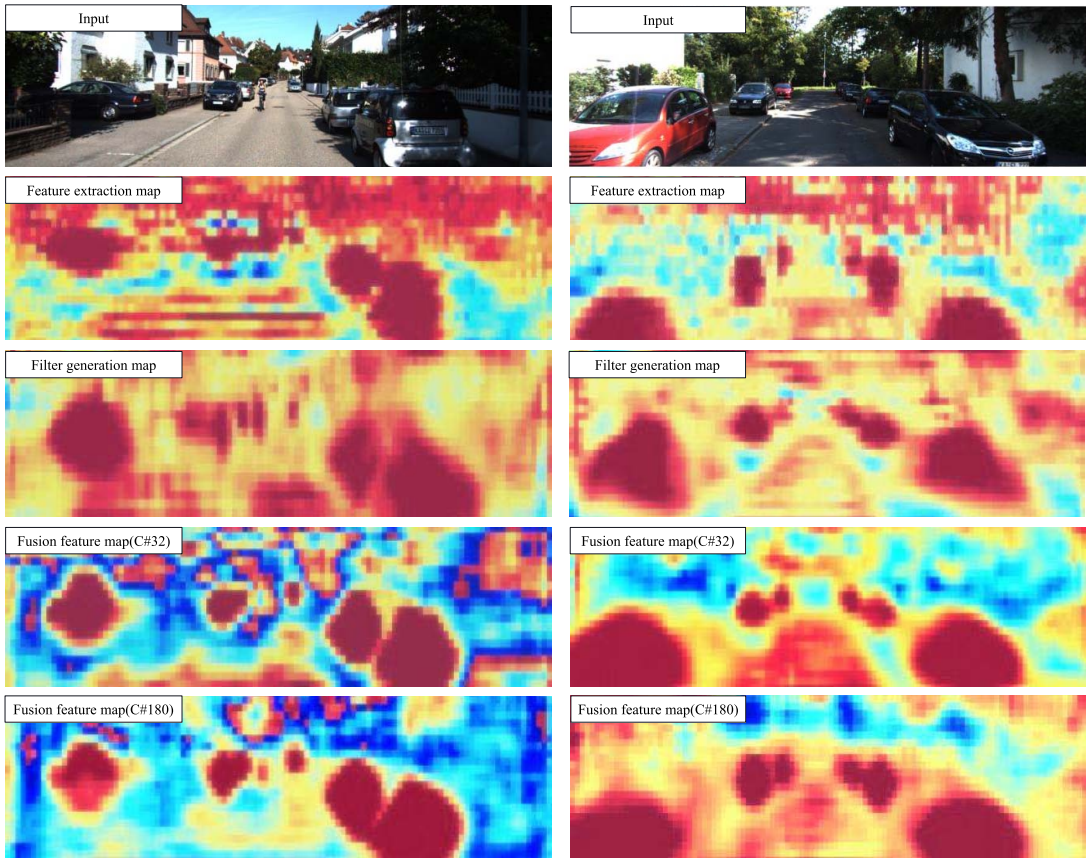

Fig. 7. Visualization of feature maps. The first row is the input. The second row is the last feature extraction map. The third row is the last filter generation map, and the fourth and fifth rows are the last fusion feature maps. C means the feature map channel and we select the 32th and 180th channels of fusion feature maps for visualization. The response area is dark red. The feature pixels within the object are responsive both in the feature extraction network and filter generation network. The fusion module could respond more obviously to the feature pixels belonging to the object.

improves the occlusion problem for 3D object detection. (3) The HMF structure also lifts the performance of the 3D detection AP scores especially of easy from 26.97, 22.32 to 29.17, 25.63 w.r.t the $AP|R_{11}$ and $AP|R_{40}$ metrics, respectively. After combining the component of anti-occlusion loss with HMF structure, the performance continuously increases compared with not adding HMF structure. These suggest that the HMF structure effectively captures the meaningful and representative local information for 3D object detection.

2) Evaluation of Different Fusion Methods: We compare three classical fusion methods with our fusion methods. The first comparative fusion method is "Concatenate," which means directly concatenating the features of the feature extraction network and the filter generation network. The second comparative fusion method is "Add," which means adding the features generated by the filter generation network at every feature pixel. The third comparative fusion method is D4LCN reproduced from [7]. As shown in Table V, the results show that our method surpasses other fusion methods by a large margin (Compared to the state of the art fusion methods in monocular 3D object detection, we get relatively +2.2, +1.02, +0.64, +3.31, +1.11, +0.74 improvement on the KITTI split1, respectively.). The essence of the fusion method we designed can be understood as taking the "And" between different feature pixels. This operation is in line with our intuitive reasoning: we should make many responses on the feature pixels, which are more responsive in filter generation networks and map filter generation networks.

3) Evaluation of Different Loss Settings: As visualized in Figure 1, the KITTI dataset labels each car object into 4 occlusion categories: 0 (non-occlusion), 1 (part-occlusion), 2 (heavy-occlusion), 3 (unknown). The unknown category is always not counted during the training and testing in monocular 3D object detection. According to the number and level of categories, we design four settings. (1) The baseline model without using the anti-occlusion loss. (2) The occlusion classification network predicts the probability of non-occlusion and part occlusion. The weight setting (w^T) is $[0, 0, 1]^T$ (corresponding to [background, non-occlusion, part occlusion]), which means solely increasing the weight of part occlusion and do not enlarge the weight of non-occlusion. (3) Similar to (2), the occlusion classification network predicts the probability of non-occlusion and part occlusion. In contrast, the weight setting is $[0, 1, 2]^T$. (4) The occlusion classification network predicts the probability of non-occlusion, part occlusion, and heavy occlusion. The weight setting is $[0, 1, 2, 3]^T$. From Table VI, we can find the performance of "N + P + H" is worse than that of "N + P." The results are not inconsistent with our intuition, which adds more samples and enlarges the hard samples' weight, increasing the performance.

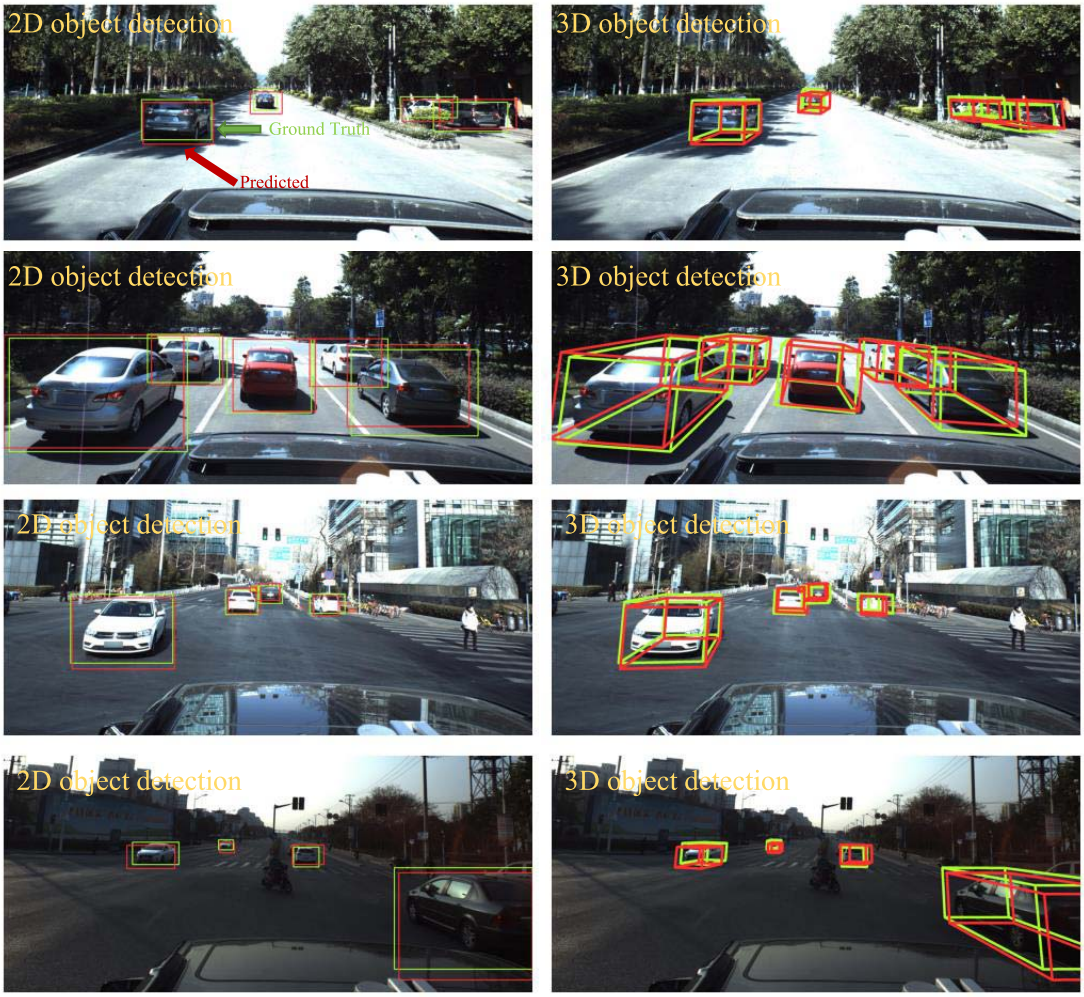

Fig. 8. Qualitative Results on ApolloCar3D. The ground truth of the car category is in green, while predicted 3D bounding box results are drawn in red. We visualize the 2D object detection and monocular 3D object detection for each scene.

Because each proposal is generated by a corresponding pixel in one stage detector, some pixels may confuse to regress non-occlusion object to the heavy occlusion object. Supposing one pixel belonging to part occlusion object needs to regress to part occlusion object and regress to a heavy occlusion object. In that case, the network is confused about the pixel belonging, such as foreground and background. Hence, the best setting is (3).

4) Evaluation of Different Types of Loss Function: As shown in Figure 6, we can split the dataset into three categories (easy, moderate, and hard) according to the occlusion level of the object. We evaluate our anti-occlusion loss on the three categories, and the results show our novel loss has better performance over other methods.

D. Qualitative Results

As shown in Figure 5 and 8, given a single RGB image, our approach can obtain accurate 2D object detection results and 3D object detection results, which include 3D location, object dimension, and 3D orientation. From the results of 3D localization shown in the lidar point cloud, after using the proposed method, the 3D object results are accurately detected even when the objects are crowded and occluded by each other in the 2D image. As shown in Figure 7, the feature pixels within the object are responsive both in the feature extraction network and filter generation network. The fusion module could respond more obviously to the feature pixels belonging to the object.

V. Conclusion and Future Work

This paper proposes a fine-grained multi-level fusion for anti-occlusion monocular 3D object detection. Our structure dynamically generates depth kernels and lets the depth kernels work on different levels and scales of feature maps. The structure can extract more reliable local representative features and have better performance. Besides, we propose an online anti-occlusion loss to make the network architecture pay more attention to the occluded object samples. Extensive experiments show that our method better captures 3D information and improves the occlusion problem. The comparative results have proved that each module is promising and effective. After evaluating the online leader-board test set, our method

demonstrates competitive performances on the KITTI dataset's current monocular 3D object detection methods. We hope our work could be applied as the perception part in the field of autonomous driving. One future direction is to extend the current fusion method to multi-modal tasks and explore the benefits of multiple modalities. Another extension could be the exploration of the occlusion problem in videos, which requires the spatial correspondence between objects in temporal images.

REFERENCES

[1] P. Li, H. Zhao, P. Liu, and F. Cao, "RTM3D: Real-time monocular 3D detection from object keypoints for autonomous driving," 2020, *arXiv:2001.03343*.

[2] Z. Qin, J. Wang, and Y. Lu, "MonoGRNet: A geometric reasoning network for monocular 3D object localization," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Jan. 2019, pp. 8851–8858.

[3] Y. Chen, L. Tai, K. Sun, and M. Li, "MonoPair: Monocular 3D object detection using pairwise spatial relationships," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12093–12102.

[4] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger, "Pseudo-LiDAR from visual depth estimation: Bridging the gap in 3D object detection for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8445–8453.

[5] X. Ma, Z. Wang, H. Li, P. Zhang, W. Ouyang, and X. Fan, "Accurate monocular 3D object detection via color-embedded 3D reconstruction for autonomous driving," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6851–6860.

[6] X. Weng and K. Kitani, "Monocular 3D object detection with pseudo-LiDAR point cloud," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019.

[7] M. Ding *et al.*, "Learning depth-guided convolutions for monocular 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1000–1001.

[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[9] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[10] G. Brazil and X. Liu, "M3D-RPN: Monocular 3D region proposal network for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9287–9296.

[11] Z. Gao, H. Xue, and S. Wan, "Multiple discrimination and pairwise CNN for view-based 3D object retrieval," *Neural Netw.*, vol. 125, pp. 290–302, May 2020.

[12] L. Liu, J. Lu, C. Xu, Q. Tian, and J. Zhou, "Deep fitting degree scoring network for monocular 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1057–1066.

[13] F. Chabot, M. Chaouch, J. Rabarisoa, C. Teuliere, and T. Chateau, "Deep MANTA: A coarse-to-fine many-task network for joint 2D and 3D vehicle analysis from monocular image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2040–2049.

[14] A. Kundu, Y. Li, and J. M. Rehg, "3D-RCNN: Instance-level 3D object reconstruction via render-and-compare," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3559–3568.

[15] Y. Xiang, W. Choi, Y. Lin, and S. Savarese, "Data-driven 3D voxel patterns for object category recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1903–1911.

[16] J. Ku, A. D. Pon, and S. L. Waslander, "Monocular 3D object detection leveraging accurate proposals and shape reconstruction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11867–11876.

[17] F. Manhardt, W. Kehl, and A. Gaidon, "ROI-10D: Monocular lifting of 2D detection to 6D pose and metric shape," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2069–2078.

[18] X. Jia, B. De Brabandere, T. Tuytelaars, and L. V. Gool, "Dynamic filter networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 667–675.

[19] Y. Li, Y. Chen, N. Wang, and Z.-X. Zhang, "Scale-aware trident networks for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6054–6063.

[20] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable ConvNets v2: More deformable, better results," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9308–9316.

[21] J. Tang, F.-P. Tian, W. Feng, J. Li, and P. Tan, "Learning guided convolutional network for depth completion," 2019, *arXiv:1908.01238*.

[22] Y. Wang, F. Sun, M. Lu, and A. Yao, "Learning deep multimodal feature representation with asymmetric multi-layer fusion," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 3902–3910.

[23] Y. Wang, W. Huang, F. Sun, T. Xu, Y. Rong, and J. Huang, "Deep multimodal fusion by channel exchanging," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 4835–4845.

[24] Z. Ma, X. Wei, X. Hong, and Y. Gong, "Bayesian loss for crowd count estimation with point supervision," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6142–6151.

[25] Y. Xiang and S. Savarese, "Object detection by 3D aspectlets and occlusion reasoning," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Dec. 2013, pp. 530–537.

[26] Y. Tian, P. Luo, X. Wang, and X. Tang, "Deep learning strong parts for pedestrian detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1904–1912.

[27] X. Wang, T. Xiao, Y. Jiang, S. Shao, J. Sun, and C. Shen, "Repulsion loss: Detecting pedestrians in a crowd," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7774–7783.

[28] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[29] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 761–769.

[30] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2002–2011.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[32] T. Roddick, A. Kendall, and R. Cipolla, "Orthographic feature transform for monocular 3D object detection," 2018, *arXiv:1811.08188*.

[33] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka, "3D bounding box estimation using deep learning and geometry," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7074–7082.

[34] B. Li, W. Ouyang, L. Sheng, X. Zeng, and X. Wang, "GS3D: An efficient 3D object detection framework for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1019–1028.

[35] A. Naiden, V. Paunescu, G. Kim, B. Jeon, and M. Leordeanu, "Shift R-CNN: Deep monocular 3D object detection with closed-form geometric constraints," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 61–65.

[36] E. Jörgensen, C. Zach, and F. Kahl, "Monocular 3D object detection and box fitting trained end-to-end using intersection-over-union loss," 2019, *arXiv:1906.08070*.

[37] A. Simonelli, S. R. Bulo, L. Porzi, M. López-Antequera, and P. Kontschieder, "Disentangling monocular 3D object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1991–1999.

[38] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[39] X. Chen *et al.*, "3D object proposals for accurate object class detection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 424–432.

[40] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

He Liu is currently pursuing the Ph.D. degree with the Department of Computer Science and Technology, Tsinghua University, Beijing, China. His research interests include 2D/3D object detection, active perception, and multimodal fusion.

Huaping Liu is currently a Tenured Associate Professor with the Department of Computer Science and Technology, Tsinghua University, Beijing, China. He is also with the State Key Laboratory of Intelligent Systems and Technology and the Beijing National Research Center for Information Science and Technology. In 2020, he was recognized as a Distinguished Young Scholar by the Natural Science Foundation of China. His research interests include robotics, dynamic systems, and machine learning, with particular emphasis on robotic perception, learning, and control.

Fuchun Sun (Fellow, IEEE) is currently a Professor with the Department of Computer Science and Technology, the President of the Academic Committee, Tsinghua University, and the Deputy Director of the State Key Laboratory of Intelligent Technology and Systems, Beijing, China. In 2006, he was recognized as a Distinguished Young Scholar by the Natural Science Foundation of China. His research interests include intelligent control and robotics, information sensing and processing in artificial cognitive systems, and networked control systems.

Yikai Wang is currently pursuing the Ph.D. degree with the Department of Computer Science and Technology, Tsinghua University, Beijing, China. His major research interests include machine learning, computer vision, especially multimodal fusion and efficient network design.

Wenbing Huang is currently a Research Assistant Professor at the Institute for AI Industry Research (AIR), Tsinghua University. His research interests include graph neural networks and graphical models with applications in representation/decision-making for physical systems and drug discovery.