PERSPECTIVE-AWARE CONVOLUTION FOR MONOCULAR 3D OBJECT DETECTION

[1]*Jia-Quan Yu,* [2]*Soo-Chang Pei*

[1]Graduate Institute of Communication Engineering,
National Taiwan University, Taiwan
E-mail: kenyu910645@gmail.com
[2]Department of Electrical Engineering,
National Taiwan University, Taiwan
E-mail: peisc@ntu.edu.tw

ABSTRACT

Monocular 3D object detection is a crucial and challenging task for autonomous driving vehicle, while it uses only a single camera image to infer 3D objects in the scene. To address the difficulty of predicting depth using only pictorial clue, we propose a novel perspective-aware convolutional layer that captures long-range dependencies in images. By enforcing convolutional kernels to extract features along the depth axis of every image pixel, we incorporates perspective information into network architecture. We integrate our perspective-aware convolutional layer into a 3D object detector and demonstrate improved performance on the KITTI3D dataset, achieving a 23.9% average precision in the easy benchmark. These results underscore the importance of modeling scene clues for accurate depth inference and highlight the benefits of incorporating scene structure in network design. Our perspective-aware convolutional layer has the potential to enhance object detection accuracy by providing more precise and context-aware feature extraction.

***Index Terms*—** Dilation Convolution, Monocular 3D Object Detection, Perspective-aware

1. INTRODUCTION

Estimating object depth and understanding the scene structure are critical tasks in image recognition, especially in the context of autonomous driving where safety is paramount. While inferring object depth from a single camera image is challenging, existing approaches predominantly rely on costly active sensors like LiDAR, Radar, or infrared cameras that directly provide depth information. However, cameras offer a more cost-effective and practical alternative, given their ease of installation on vehicles. The main hurdle with cameras is the absence of depth information in 2D images, posing a significant challenge for depth estimation algorithms.

Fig. 1. Illustration of long-range dependency that aids in depth inference. Directly predicting the depth of the green dot is challenging. However, if we can determine the depth-axis for every pixel and extract the surrounding front and back pixels along this axis, it can significantly enhance the accuracy of predicting the depth for the green dot. To achieve this, we utilize the camera intrinsic matrix to derive the depth-axis and introduce a skewed convolutional kernel designed to capture features along this axis.

Despite the absence of depth information in camera images, we posit that the human perception system is capable of inferring depth from limited visual cues by leveraging other scene information. For instance, as depicted in Fig.1, humans can infer the location of objects based on the relative positions of other nearby objects in the scene, exploiting their understanding of scene structure. By comparing the distances between adjacent objects, depth can be estimated even in cases where occlusion occurs. Hence, we believe understanding scene structure plays a pivotal role in accurate depth estimation. Our objective is to integrate this perspective information into our convolutional neural network.

In this paper, we present a novel approach called perspective-aware convolution (PAC) to enhance the capability of convolutional neural networks in capturing perspective-related features. PAC extracts feature along the depth axis by adjusting the shape of the convolutional kernels. Additionally, we introduce a PAC module that integrates multiple dilation

rates within parallel convolutional branches. By incorporating the PAC module into 3D object detection networks, we enable them to generate perspective-aware feature maps, enhancing their ability to analyze objects in specific perspective scene structure.

Finally, to demonstrate the effectiveness of the PAC module, we evaluate it in KITTI 3D object detection challenge, where objects are defined as cuboids in the camera coordinate system. We train and evaluate our network on the KITTI dataset and achieve 23.53% AP on easy metric.

The remainder of this paper is organized as follows: Section 2 introduces the related work on 3D object detection and dilated convolution modules. Section 3 explains our proposed perspective-aware convolution. Subsequently, we report our experimental results on the KITTI dataset in Section 4.

2. RELATED WORKS

2.1. Convolutional Module

Convolutional layers are essential components of deep learning network. It plays a crucial role in extracting features from image by convolving a learnable kernel with the image pixel in a sliding-window fashion. While convolutional layers excel at recognizing local patterns, they have limitations when it comes to capturing long-range dependencies within an image. To address this limitation, researchers have explored various techniques to increase network's receptive field, which is, the size of the image region that the network considers when making predictions at a particular location.

To enlarge the receptive field of a network, researchers often resort to building deeper networks or increasing number of downsampling to expand the receptive field size; however, these approaches can result in the loss of fine-grained details in feature maps. Alternatively, dilated convolutions, as introduced in [1], provide another solution. By skipping a certain number of image pixels during convolution, determined by the dilation rate, dilated convolutions enable the network to have larger receptive fields while no need of down-sampling. Expanding on the benefits of dilated convolutions, DeepLabv2[2] introduced atrous spatial pyramid pooling (ASPP) to further enhance feature extraction. The ASPP module incorporates multiple parallel dilated convolutions with different dilation rates, allowing for the extraction of multi-scale features from the same feature map. Another approach to enhancing the feature extraction capability of convolutional layers is the receptive field block(RFB)[3], which adjusts the kernel size based on the corresponding dilation rate of the convolutional module.

However, all the aforementioned methods rely on fixed kernel shapes that are predefined before the training process. To address this limitation, Dai et al. proposed a novel convolutional layer called deformable convolutional networks (DCN)[4], and its improved version DCNv2[5]. These convolutional modules enable the network to dynamically adjust the shape of convolutional kernel during training, making the feature extractor more adaptive to the scene structure. While this method introduces more flexibility to the training process, it also incurs a non-negligible overhead.

Fig. 2. 3D object detection. The cuboids in defined in camera coordinate and we need to find the location (x, y, z), dimension (w, h, l), and orientation θ of each cuboid.

Despite the extensive research on convolution layers, there are few methods that incorporate perspective information into the network. Therefore, drawing inspiration from the ASPP module and deformable convolution, we propose our perspective-aware convolution (PAC) module, which adjusts the kernel shape based on the depth axis in the image. This novel module allows the network to capture the underlying scene geometry and perspective, enhancing its ability to understand the 3D structure of the environment. In Section 3, we will provide a detailed explanation of our PAC module and its integration within our proposed method.

2.2. Monocular 3D Object Detection

Monocular 3D object detection is a rapidly evolving research field that utilize single camera images as input to estimate the 3D appearance of objects within the scene. In this section, we provide an overview of the fundamental concepts of 3D object detection and discuss some relevant prior works in the field.

In the field of 3D object detection, an object is represented by a cuboid which respect to camera coordinates. This cuboid is characterized by seven parameters: centroid coordinate (x, y, z), which specifies the location of the cuboid center relative to the camera center, and dimensions (w, h, l), which correspond to the width, height, and length of the cuboid, respectively. Additionally, the yaw angle θ denotes the orientation of the cuboid. Our objective in 3D object detection is to identify objects within images and accurately localize them in camera coordinates by predicting these seven variables $(x, y, z, w, h, l, \theta)$ for each object. This concept is illustrated in Fig2.

The related work in 3D object detection can be divided into two main branches, as we shown in Fig3: two-stage detectors and one-stage detectors. Two-stage detectors, such as Deep3DBox[6] and FQNet[7], rely on a predicted 2D bounding box as prior information. These methods assume that all

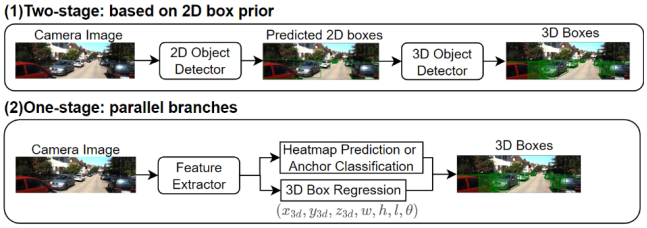

Fig. 3. Related work in 3D object detection can be categorized into two-stage and one-stage detectors. Two-stage detectors utilize a predicted 2D bounding box as a prior and extract features within the 2D box, whereas one-stage detectors treat the prediction of 2D and 3D boxes as a unified task, predicting both boxes in parallel branches.

projected corners of a 3D bounding box must lie within its corresponding 2D bounding box. While these approaches reduce the complexity of the problem, they are sensitive to inaccuracies in the predicted 2D bounding box. On the other hand, other two-stage detector, including MonoDIS[8], ROI10D[9], MonoGRNet[10], and MonoPSR[11], utilize the predicted 2D bounding box as a region proposal to extract features and predict 3D box geometry from it. In specific, MonoDIS uses RoIAlign to extract fixed-length features from the regions of interest (ROIs) and proposes a disentangle loss function to avoid interference between each loss term and aid in faster convergence. MonoGRNet employs both early and deep features in the backbone network and uses deep features only for tasks that require a higher receptive field, such as depth estimation, to prevent excessive downsampling of the feature map. ROI10D and MonoPSR both utilize a pre-trained depth estimation model to improve their accuracy in estimating object depth.

One-stage detectors take a different approach compared to two-stage detectors as they do not rely on 2D box priors. Instead, they treat 3D objects as extensions of 2D objects and utilize a unified network to predict both 2D and 3D bounding boxes in parallel branches without the need for region-specific feature extraction. One example is M3D-RPN[12], which reformulates the 2D detector network to capture 3D proposals using a shared network for both tasks. M3D-RPN also incorporates statistical data from the training set to determine the 3D anchor box prior. Additionally, M3D-RPN proposes a depth-aware convolution, which employs separate kernels to extract features from different image rows, enabling the network to handle depth features separately. Another method, Ground-aware[13], focuses on locating and extracting the ground-contact point of each object to enhance its depth perception capability.

Another example of a one-stage detector is the keypoint-based network, as proposed by CenterNet[14]. By detecting keypoints such as the center point or corner points of the 3D objects, these methods can efficiently predict 3D bounding boxes. RTM3D[15] takes a similar approach by identifying eight corners and the object center of the cuboid. It also introduces a feature pyramid network to capture multi-scale keypoints. SMOKE[16] simplifies the network by eliminating the 2D box regression branch and incorporates MonoDIS's disentangle loss to improve convergence. MonoPair[17] focuses on leveraging the relationship between adjacent objects by predicting keypoints at the midpoint of each adjacent object pair. Additionally, MonoPair introduces uncertainty in the regression branch and utilizes it for post-processing optimization of the detection results. MonoFlex[18] addresses truncated objects whose centers lie outside the image by proposing an edge fusion module to separate feature learning from truncated object prediction. MonoFlex employs an ensemble approach for depth prediction, considering depth uncertainty and the predicted 3D bounding box's multiple pixel heights to enhance depth estimation accuracy.

Overall, two-stage detectors have been pioneering in the field of 3D object detection, using 2D bounding boxes as priors. However, they tend to have performance degradation when the 2D bounding box predictions are inaccurate. On the other hand, one-stage detectors offer a more unified network architecture with improved performance. Therefore, in our work, we choose to adopt an anchor-based one-stage detector for our experiments, leveraging its superior performance in 3D object detection tasks.

3. PROPOSED METHOD

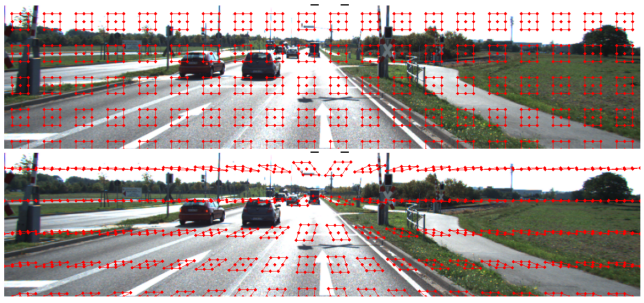

Fig. 4. Perspective-aware convolution. The red square in the image represent the kernel shape. The top image illustrates the kernel shape of dilation convolution, while the bottom image demonstrates the kernel shape of perspective-aware convolution. Our perspective-aware kernel dynamically adjusts its shape based on the depth axis at each pixel.

4. PERSPECTIVE-AWARE CONVOLUTION

We propose a novel perspective-aware convolutional layer that extracts features along the perspective lines at each pixel

location. Our motivation arises from the recognition that the depth-axis-adjacent objects contain essential information for object depth estimation. Conventional convolutional layers often struggle to capture long-range dependencies in images. To overcome this limitation, we attempt to explicitly inject the perspective information into the network by guiding the convolutional kernels to capture features along the depth axis of each pixel. These axis are straight lines parallel to the camera's depth axis in the camera coordinate system. By projecting these lines onto the image plane using the camera pinhole model, we obtain an indication of how each pixel would move on the image if its depth value were to change. Additionally, we can derive the angle between the depth axis and the u-axis, which we refer to as the perspective angle, and use it to represent the perspective line for simplicity. In the following part of this section, we will elaborate on how we derive the perspective angle based on the camera pinhole model.

To get the perspective angle, we begin with the formulation of the pinhole camera model:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{Z_w} \begin{bmatrix} f_x & 0 & C_x \\ 0 & f_y & C_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} \tag{1}$$

In Equation 1, (X_w, Y_w, Z_w) represents the coordinates of a point in the camera coordinate system, while (u, v) represents the projected pixel coordinates on the image plane. The parameters f_x and f_y denote the focal lengths expressed in pixel units for the u and v axes. The values C_x and C_y correspond to the principal point, which is the intersection point of the optical axis with the image plane.

To determine the amount of pixel displacement caused by a change in depth, we differentiate Equation 1 with respect to Z_w and obtain:

$$\begin{cases} \dfrac{du}{dZ_w} = -\dfrac{X_w f_x}{Z_w^2} \\ \dfrac{dv}{dZ_w} = -\dfrac{Y_w f_y}{Z_w^2} \end{cases} \tag{2}$$

where (X_w, Y_w, Z_w) represents a 3D point in camera coordinates, which is inversely projected from a pixel (u, v). Since the depth of each pixel is unknown, we assume that all image pixels lie on the ground plane and set Y_w equal to the height of the ground plane, denoted as Y_0. By applying Equation 1, we can inversely project (u, v) back to the camera coordinate system and the inversely projection equation is as following:

$$\begin{cases} X_0 = \dfrac{(u_0 - C_x) Y_0 f_y}{(v_0 - C_y) f_x} \\ Y_0 = Y_0 \\ Z_0 = \dfrac{Y_0 f_y}{v_0 - C_y} \end{cases} \tag{3}$$

Here, (u_0, v_0) represents a specific pixel on the image, and (X_0, Y_0, Z_0) represents the inversely projected 3D point in camera coordinates. By substituting (X_0, Y_0, Z_0) into (X_w, Y_w, Z_w) in Equation 2, we can calculate the derivatives and the perspective angle ϕ is determined by the following equation:

$$\phi = \operatorname{atan2}\left(\frac{dv}{dZ_w}, \frac{du}{dZ_w}\right) \tag{4}$$

Using Equations 2 and 4 , we can calculate the perspective angle for each image pixel (u, v). This perspective angle allows us to guide kernel to change their shape according to its pixel coordinate and perspective angle, as illustrated in Fig. 4.

4.1. Perspective-aware Convolutional Module

In addition to incorporating a single PAC convolution layer, we employ a multiple-branch design inspired by ASPP[2], utilizing different dilation rates for each branch. Our objective is to capture multi-scale features along each perspective line with the PAC module. Furthermore, to ensure the preservation of regular features, we include a branch with a standard 3x3 kernel in our PAC module. This design choice guarantees that the regular feature map passes through our module without any alterations. A comparison between ASPP module and PAC module is depicted in Fig. 6.

5. EXPERIMENTS

In this section, we present the experimental results of our proposed PAC module in the task of 3D object detection. We integrate the PAC module into the baseline network of Ground-aware[13] and aim to enhance its performance. The training and evaluation of the detector are conducted on the KITTI3D dataset, consisting of 3711 training images and a validation set of 3768 images. We adopt to the data split recommended by Chen et al.[19]. During training, we set the batch size to 8 and utilize the Adam optimizer with a learning rate of 10^{-4}. To speed up the network, we crop the top 100 pixels from the images and resize them to 288x1280. Additionally, we apply horizontal flipping and photometric distortion techniques to improve the diversity of the training data.

We present our experiment results in Table 1 where we compare with other 3D object detection networks. Our proposed method surpasses all other detectors in terms of average precision, showing the effectiveness of our PAC module.

Additionally, we compare PAC module with other convolutional module mentioned in Section 2 and report the result in 2. We adopt Ground-aware network as the baseline and applied the dilation convolution in the last three convolutional layers of the backbone and set the dilation rate to (2,2,2) and (3,3,3) in our experiment. We also add DCN, RFB, ASPP, and PAC after the feature extractor, following their respective

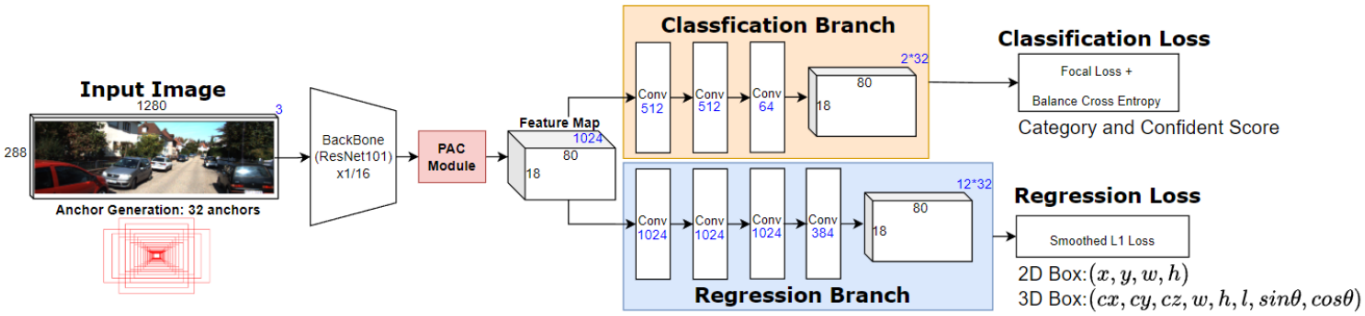

Fig. 5. Our proposed architecture for 3D object detection. We adopt the network proposed by Ground-aware[13], which is a one-stage anchor-based architecture. We add our PAC module after the feature extractor, which is ResNet-101 in our implementation, to obtain the perspective-aware feature map. The top branch focuses on classifying positive and negative anchor predictions, while the lower branch is responsible for regressing the geometry of both 2D and 3D bounding boxes. This architecture enables accurate object detection by leveraging the benefits of the PAC module in capturing scene structure and improving 3D box regression.

Table 1. Experimental results of the 3D object detection algorithm on the KITTI3D validation dataset. The best performance in each column is indicated in bold font.

Methods	Car AP 3D (IoU=0.7)			Car AP BEV (IoU=0.7)		
	Easy	Moderate	Hard	Easy	Moderate	Hard
MonoGRNet[10]	12.28	7.76	5.91	19.89	12.94	10.31
SMOKE[16]	6.96	4.30	3.98	12.73	7.93	6.94
DD3D[20]	19.16	15.27	13.37	25.72	20.78	18.38
MonoFlex[18]	22.14	16.19	14.18	29.30	21.91	18.82
Ground-aware[13]	21.90	16.06	13.17	28.29	20.98	17.59
Ours(Ground-aware+PAC Module)	**23.53**	**17.23**	**14.33**	**30.57**	**22.44**	**19.07**

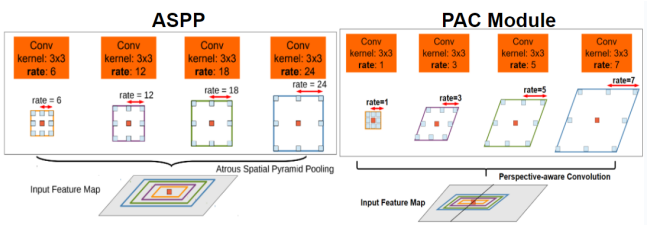

Fig. 6. Comparison of PAC module and ASPP module. Both modules utilize parallel branches to capture multi-scale features. However, the key distinction lies in the kernel shape employed for feature extraction. While the ASPP module utilizes a regular kernel shape, the PAC module incorporates a tilted kernel shape to guide feature extraction along the perspective line.

recommended settings in each paper. We conducted experiments with a single layer of PAC, where the dilation rate is set to two. As for the PAC module, we set the dilation rate to 2, 4, 6, and 8 in each parallel branch.

As shown in our experimental results in Table 2, our proposed PAC module outperforms all other methods and achieves an improvement of +1.59% AP compare to baseline in the 3D metric with moderate difficulty. In contrast, dilation convolution showed no improvement compared to the baseline. Deformable convolution had a slight improvement with a single layer, but its performance dropped after using more than one DCN layer. We suspect this is because DCN introduces too many parameters to train, making it easier to overfit the training data, especially since KITTI's training set is small. RFB showed roughly the same performance as the baseline, while ASPP showed a fair improvement, especially with hard-difficulty objects. This indicates that far or truncated objects require long-range information to predict their depth accurately.

5.1. Qualitative Result

To facilitate a fair comparison of different 3D object detectors, we present some inference outcome example in the KITTI3D validation set, as shown in Figures 7. In the figure, the left column shows the predicted 3D bounding box projected onto the image plane. However, to accurately evaluate the 3D box location, we recommend that readers use the bird's-eye-view (BEV) provided in the right column. In the

Table 2. Experimental results of the 3D object detection algorithm on the KITTI3D validation dataset. The best performance in each column is indicated in bold font.

Methods	Car AP 3D (IoU=0.7)			Car AP BEV (IoU=0.7)		
	Easy	Moderate	Hard	Easy	Moderate	Hard
Baseline	22.08	15.64	13.00	28.56	21.08	17.48
Dilation Convolution(2,2,2)	20.32	14.07	12.06	27.23	19.70	16.72
Dilation Convolution(3,3,3)	17.93	12.93	10.71	24.93	17.93	15.68
DCNv2[5](n=1)	22.13	16.20	13.44	29.19	21.58	18.57
DCNv2(n=2)	20.21	15.52	13.23	29.03	21.78	18.93
DCNv2(n=3)	20.21	14.58	12.24	28.12	19.79	17.11
RFB[3]	21.32	15.62	12.94	28.53	21.26	18.35
ASPP[21]	22.44	16.96	14.23	29.69	22.20	19.03
PAC(Ours)	22.71	15.73	13.05	30.55	21.85	18.56
PAC Module(Ours)	**23.53**	**17.23**	**14.33**	**30.57**	**22.44**	**19.07**

BEV figures, the yellow box represents the ground truth and the green box represents the predicted result.

Based on these results, we can observe some interesting traits of each detector. Firstly, despite the impressive accuracy of Pseudo-LiDAR in 3D box estimation, it can sometimes incorrectly predict the object orientation in pretty obvious cases. This is because Pseudo-LiDAR converts the image to a point cloud, sacrificing some advantages that are only available when the data is in image form. As for MonoFlex and Ground-aware, they perform roughly the same in this experiment, showing keypoint-based and anchor-based method both has potential in 3D object detection. DD3D, on the other hand, tends to generate too many false positives, although their confidence scores are low. This also highlights the advantage of anchor-based methods, where non-maximum suppression is applied to avoid similar issues.

6. CONCLUSIONS

In this paper, we introduced a novel perspective-aware convolution layer to address the limitations of traditional convolutional kernels in capturing long-range dependencies in images. The PAC module enforces the convolutional kernel to extract features along the perspective lines, making it able to extract perspective-aware features. We integrated the PAC module into a 3D object detector and evaluated its performance on the KITTI3D dataset. The experimental results demonstrate that our approach achieved significant improvements, achieving a 23.9% AP in the easy difficulty of the dataset and surpassing other 3D object detectors. Our findings highlight the importance of modeling scene clues for accurate depth inference in camera images and the benefits of incorporating perspective information into the neuron network. We believe that our proposed methods have the potential to application in autonomous driving, to enhance 3D object detection accuracy and driving safety.

REFERENCES

[1] Fisher Yu and Vladlen Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[3] Songtao Liu, Di Huang, et al., "Receptive field block net for accurate and fast object detection," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 385–400.

[4] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei, "Deformable convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 764–773.

[5] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai, "Deformable convnets v2: More deformable, better results," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 9308–9316.

[6] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Kosecka, "3d bounding box estimation using deep learning and geometry," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 7074–7082.

[7] Lijie Liu, Jiwen Lu, Chunjing Xu, Qi Tian, and Jie Zhou, "Deep fitting degree scoring network for monocular 3d object detection," in *Proceedings of the*

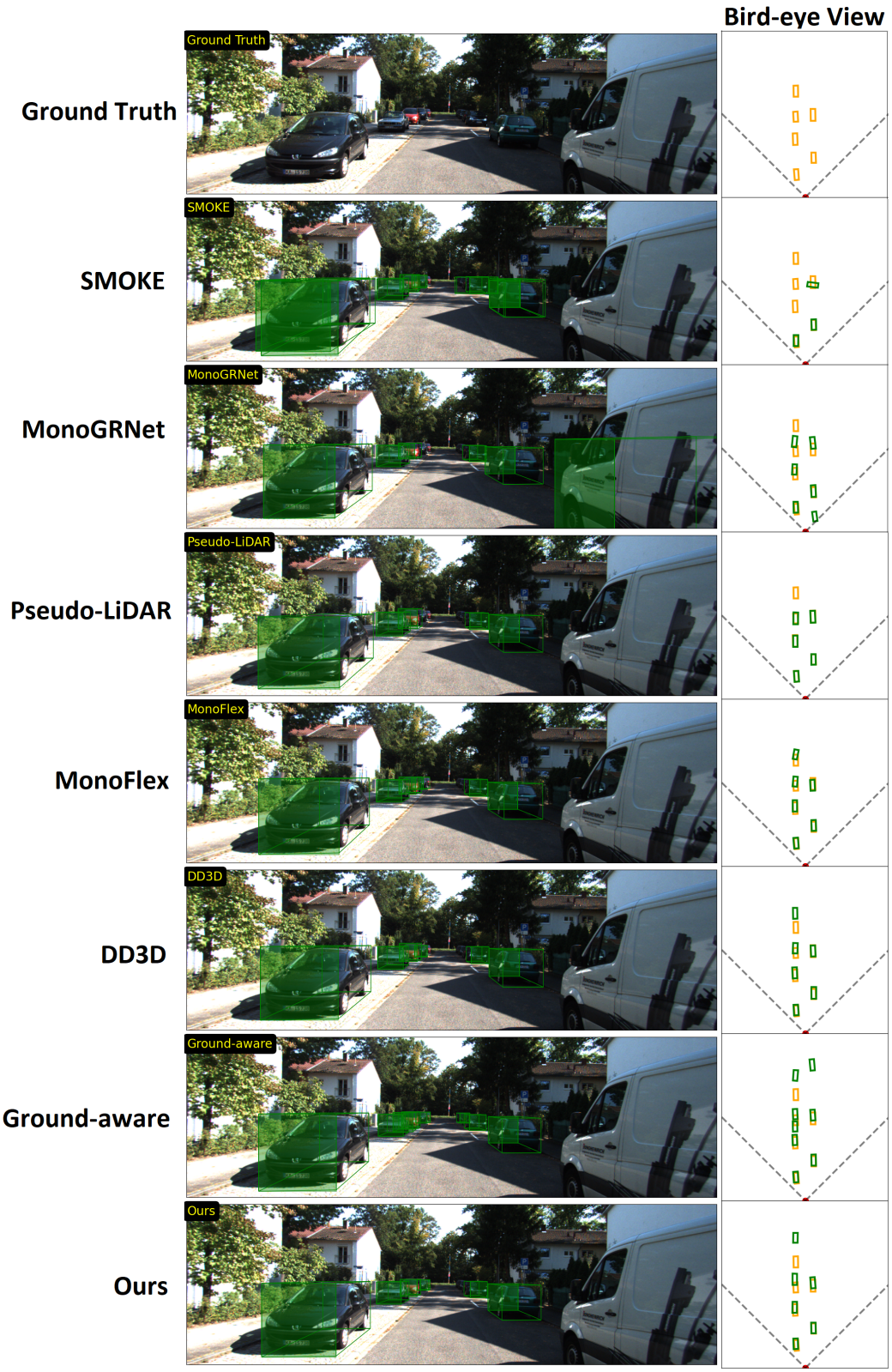

Fig. 7. Inference example of 3D object detectors. The left column shows the predicted 3D bounding boxes in the image plane, while the right column displays the bounding boxes projected onto the bird's-eye-view (BEV) plane. The yellow boxes on the BEV represent the ground truth, while the green boxes indicate the predictions. In this experiment, we observe that most methods exhibit inaccuracies in predicting object depth, whereas our proposed method demonstrates more accurate depth estimation.

IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 1057–1066.

[8] Andrea Simonelli, Samuel Rota Bulo, Lorenzo Porzi, Manuel López-Antequera, and Peter Kontschieder, "Disentangling monocular 3d object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1991–1999.

[9] Fabian Manhardt, Wadim Kehl, and Adrien Gaidon, "Roi-10d: Monocular lifting of 2d detection to 6d pose and metric shape," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2069–2078.

[10] Zengyi Qin, Jinglu Wang, and Yan Lu, "Monogrnet: A geometric reasoning network for monocular 3d object localization," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, vol. 33, pp. 8851–8858.

[11] Jason Ku, Alex D Pon, and Steven L Waslander, "Monocular 3d object detection leveraging accurate proposals and shape reconstruction," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 11867–11876.

[12] Garrick Brazil and Xiaoming Liu, "M3d-rpn: Monocular 3d region proposal network for object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9287–9296.

[13] Yuxuan Liu, Yuan Yixuan, and Ming Liu, "Ground-aware monocular 3d object detection for autonomous driving," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 919–926, 2021.

[14] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl, "Objects as points," *arXiv preprint arXiv:1904.07850*, 2019.

[15] Peixuan Li, Huaici Zhao, Pengfei Liu, and Feidao Cao, "Rtm3d: Real-time monocular 3d detection from object keypoints for autonomous driving," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*. Springer, 2020, pp. 644–660.

[16] Zechen Liu, Zizhang Wu, and Roland Tóth, "Smoke: Single-stage monocular 3d object detection via keypoint estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 996–997.

[17] Yongjian Chen, Lei Tai, Kai Sun, and Mingyang Li, "Monopair: Monocular 3d object detection using pairwise spatial relationships," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12093–12102.

[18] Yunpeng Zhang, Jiwen Lu, and Jie Zhou, "Objects are different: Flexible monocular 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3289–3298.

[19] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Andrew G Berneshawi, Huimin Ma, Sanja Fidler, and Raquel Urtasun, "3d object proposals for accurate object class detection," *Advances in neural information processing systems*, vol. 28, 2015.

[20] Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon, "Is pseudo-lidar needed for monocular 3d object detection?," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3142–3152.

[21] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.