

MonoPGC: Monocular 3D Object Detection with Pixel Geometry Contexts

Zizhang Wu¹, Yuanzhu Gan¹, Lei Wang¹, Guilian Chen¹, Jian Pu²

Abstract—Monocular 3D object detection reveals an economical but challenging task in autonomous driving. Recently center-based monocular methods have developed rapidly with a great trade-off between speed and accuracy, where they usually depend on the object center’s depth estimation via 2D features. However, the visual semantic features without sufficient pixel geometry information, may affect the performance of clues for spatial 3D detection tasks. To alleviate this, we propose MonoPGC, a novel end-to-end Monocular 3D object detection framework with rich Pixel Geometry Contexts. We introduce the pixel depth estimation as our auxiliary task and design depth cross-attention pyramid module (DCPM) to inject local and global depth geometry knowledge into visual features. In addition, we present the depth-space-aware transformer (DSAT) to integrate 3D space position and depth-aware features efficiently. Besides, we design a novel depth-gradient positional encoding (DGPE) to bring more distinct pixel geometry contexts into the transformer for better object detection. Extensive experiments demonstrate that our method achieves the state-of-the-art performance on the KITTI dataset.

I. INTRODUCTION

Accurate detection of 3D objects in various scenarios has extensive applications such as autonomous driving and robotic manipulation [1], [2]. To obtain a precise perception of 3D information [3], many 3D object detection methods rely on 3D point clouds or stereo images that typically require costly system setups, such as LiDAR or stereo sensors [4], [5]. In contrast, monocular 3D object detection, which only uses a monocular RGB camera as a simpler and cheaper setting for deployment, has attracted increasing attention recently.

Most existing monocular 3D object detection methods can be roughly divided into two categories. The pseudo-LiDAR-based methods [1], [6] normally lift learned features from 2D space to 3D space and then conduct 3D object detection. The center-based methods [4], [7]–[10] directly adopt image features to predict object’s center location, orientation, dimension, depth and other auxiliary monocular tasks. Generally, the pseudo-LiDAR-based methods perform more accurately yet time-consuming for lifting images to 3D space. In contrast, the center-based methods achieve better speed-accuracy trade-off [4], [11]. However, the performance of center-based methods heavily depends on the estimation of depth. When incorrect estimation occurs, localization error easily affects the performance of the following detection task [3]. As a consequence, without considering 3D geometry information, the center-based methods are often inferior to those pseudo-LiDAR-based methods [1], [6], [12]. While

MonoDTR [13] introduces the pixel-level depth-aware auxiliary supervision to achieve great improvement, its pixel geometry information still appears some indistinct and smooth for object detection task, which may bring disturbance with different semantics of the same depth.

To address the issues of coarsely encoded 3D pixel geometry information, we propose **MonoPGC**, a novel end-to-end monocular 3D object detection with rich pixel geometry contexts, and surpass the prior state-of-the-art 3D monocular detection methods. Specifically, we introduce the pixel-level depth estimation as an auxiliary task [13], and design the Depth Cross-attention Pyramid Module (DCPM) to obtain more precise depth prediction from the multi-scale image features. Inspired by implicit neural representation (INR) methods [14]–[16], we further propose to encode 3D coordinates information into depth-aware features by the Depth-Space-Aware Transformer (DSAT), for enhancing the 3D position-aware pixel geometry information. In addition, we introduce a new term, depth gradient, for encoding the object boundary and propose the Depth-Gradient Positional Encoding (DGPE) to spotlight edges of objects, which serves as valuable hints for pixel geometry modeling. We summarize our contributions as follows:

(1) We propose a novel framework, MonoPGC, leveraging pixel-level geometry information to boost the 3D object detection. To achieve so, we introduce the pixel depth estimation to strengthen geometry knowledge. Moreover, we design the cross-attention pyramid module to avoid incorrect depth priors, and produce more robust depth-aware features integrated with local and global receptive field.

(2) We present the first depth-space-aware transformer to integrate 3D space coordinates and depth aware features efficiently. We propose a novel depth-gradient positional encoding (DGPE) to bring more distinct and rich hints into the transformer for the object detection task.

(3) Our MonoPGC framework achieves the state-of-the-art performance on the KITTI [17] dataset. Particularly, extensive experiment results illustrate that it achieves 24.68%, 17.17%, and 14.14% in terms of the AP₄₀ metric, on the easy, moderate, and hard test setting for Car category, respectively.

II. RELATED WORK

A. Monocular 3D Object Detection

Many prior works [18] [19] [20] [5] [21] [22] have explored the inherently ill-posed problem of detecting 3D objects from monocular images. Due to the lack of depth information from images, monocular 3D detection significantly falls behind Lidar-based and stereo-based counter-

¹Zongmu Technology

²Fudan University

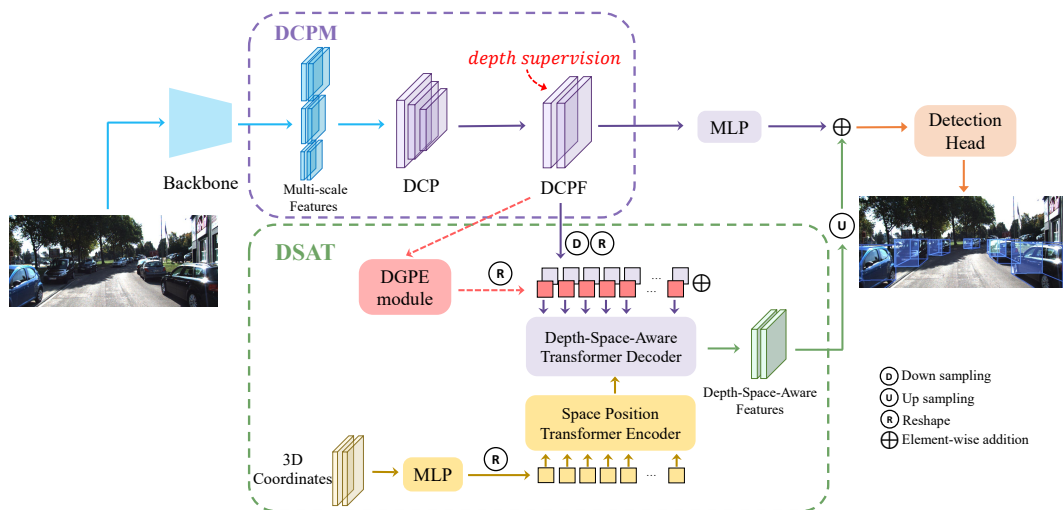


Fig. 1. The framework of our MonoPGC, which consists of the backbone, the depth cross-attention pyramid module (DCPM), the depth-space-aware transformer (DSAT) module, and the detection head. “DCP” denotes the depth cross-attention pyramid, “DCPF” denotes depth cross-attention pyramid features, and “DGPE” denotes the depth-gradient positional encoding.

parts. Many works [23] [24] [25] ease this problem by utilizing 2D-3D geometric constraints to improve 3D detection performance. Deep3Dbox [26] proposes MultiBin loss and enforces constraint between 2D and 3D boxes with geometric prior. Mono3D [18] is a seminal work that uses region proposals to deal with semantics and contexts. CenterNet [7] proposes a center-based anchor-free method but with restrained accuracy. Following this work, center-based series SMOKE [2], KM3D [4] and RTM3D [8] assist the regression of object depth by solving a Perspective-n-Point method and have achieved remarkable results. However, existing works, computing 3D locations explicitly based on 2D predictions, usually suffer from the well-known error amplification effect. Recent works [11], [25], [27], [28] try to use uncertainty modeling [27], sophisticated model ensemble [25] or auxiliary monocular contexts [11] to improve the performance.

B. Transformer-based Object Detection

Many recent works [29] [30] [15] [31] [13] take a step further and explore the usage of transformer in 3D object detection. DETR [32] brings Transformer into the field of object detection. DETR3D [29] proposes a new paradigm to address the ill-posed inverse problem of recovering 3D information from 2D images. Graph-DETR3D [30] improves the DETR3D by a novel dynamic graph feature aggregation module. BEVFormer [31] learns a unified BEV representation with spatiotemporal transformers. MonoDTR [13] proposes to globally integrate context- and depth-aware features with transformers and inject depth hints into the transformer.

C. Implicit Neural Representation

Early work has investigated the implicit neural representation of continuous 3D shapes by employing deep networks that map coordinates to signed distance functions [33] [34] or occupancy fields [35] [36]. NeRF [14] utilizes multi-layer perceptrons (MLPs) to encode 5D radiance fields (3D volumes with 2D view-dependent appearance), which is referred to as positional encoding and proved to be an

efficient way for modeling complex 3D scenes. Inspired by the success of NeRF [14], recent works [15] [16] take similar approach mapping both 3D coordinates and camera parameters (camera intrinsics and extrinsics) to positional encoding. These methods can be regarded as an extension of INR in 3D object detection [15]. Furthermore, they also use the popular Transformer architecture [37] to exploit the encoded information.

III. METHOD

A. The Overview

Figure 1 demonstrates the framework of our MonoPGC, which mainly consists of four components: the backbone, the depth cross-attention pyramid module (DCPM), the depth-space-aware transformer (DSAT) module, and the detection head. Given an image $I \in \mathbb{R}^{3 \times H_I \times W_I}$, we adopt the backbone network (DLA34 [38]) to extract its multi-scale 2D features $F_i^{2d} = \{F_i^{2d} \in \mathbb{R}^{C \times H_{F_i} \times W_{F_i}}, i = \frac{1}{4}, \frac{1}{8}, \frac{1}{16}\}$. The depth cross-attention pyramid module (DCPM) fuses multi-scale features for pixel depth estimation with the contribution of cross-attention and pyramid designs, then achieves DCP features (DCPF) (Section III-B). Furthermore, the 3D world space coordinates from the 3D coordinates generator, could be a strong indicator for the implicit neural representation. Thus, we integrate DCPF and space coordinates by the depth-space-aware transformer (DSAT) module and insert the informative depth-gradient positional encoding into the transformer through the depth-gradient positional encoding (DGPE) module (Section III-C). Finally, we apply DSA features to the detection head for object detection task (Section III-D).

B. Depth Cross-attention Pyramid Module

Existing monocular methods [2], [9], [11], [25] usually estimate the depth of objects’ center, which may bring inaccurate depth priors when objects stand in various poses and headings. Pixel-wise depth shall be helpful, since accurate

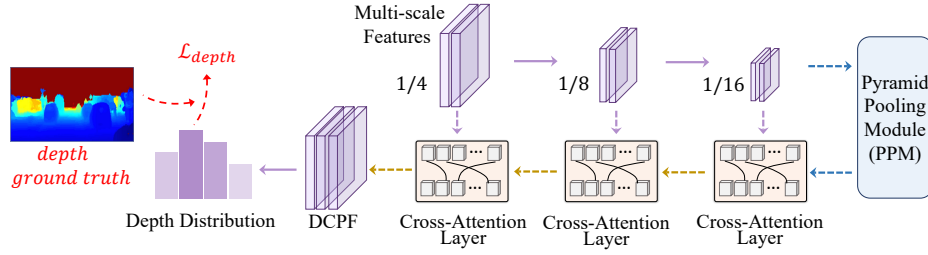


Fig. 2. The framework of Depth Cross-attention Pyramid Module (DCPM). We fuse the multi-scale features and introduce depth supervision to obtain the depth loss.

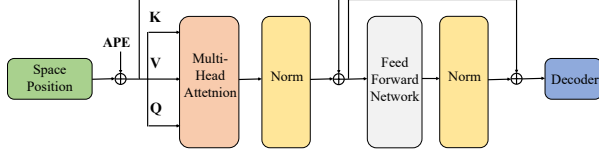


Fig. 3. The encoder of Depth-Space-Aware Transformer (DSAT) with space position as input. We adopt absolute positional encoding (APE) [41] as encoder’s positional encoding.

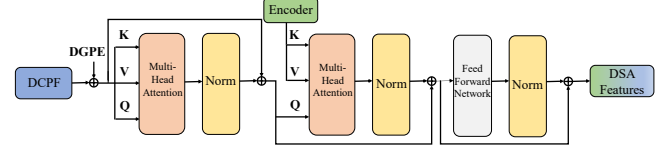


Fig. 4. The decoder of Depth-Space-Aware Transformer (DSAT) with DCPF and encoder’ output as input. Depth-gradient Positional Encoding (DGPE) assists pixel geometry modeling.

pixel object depth bring extra geometry context, such as it’s less possible to appear objects in a large depth.

In our work, we regard the pixel depth estimation as our auxiliary task, injecting the depth information to the visual features. We propose our depth cross-attention pyramid module (DCPM). Pyramid structure combined with cross-attention layer on different scales, improves the receptive field and multi-scale perception for better depth estimation. **Depth Cross-attention Pyramid Feature.** Figure 2 briefly shows the architecture of our depth cross-attention pyramid module. The input of this module remains the multi-scale features from the backbone. Firstly, we adopt the pyramid pooling module (PPM) [39] to aggregate the global and local information of the whole image from the top-level feature maps. Sequentially, we adopt three cross-attention layers to fuse the multi-level features, which finally outputs the enhanced depth cross-attention pyramid features $F_{DCP} \in R^{D \times H_{\hat{f}} \times W_{\hat{f}}}$, where $H_{\hat{f}}$ and $W_{\hat{f}}$ denote 0.25 scale of input image’s size.

$$F_{DCP} = DCPM(F_i^{2d}), i = \frac{1}{4}, \frac{1}{8}, \frac{1}{16} \quad (1)$$

By utilizing the cross-attention layers, the module can extract non-local context features with global and local perception and realize the efficient multi-level feature fusion. We intercept the cross-attention block from Linear transformer [40] as our cross-attention layer. We leave the detailed architecture of cross-attention layers in the supplementary material.

Depth estimation. As illustrated in Figure 1, accepting the depth cross-attention pyramid features (DCPF), we adopt a simple 3×3 convolution layers to achieve depth estimation. Following [1], [13], we predict the depth distribution: the probability of discretized depth bins $D_{gt} \in R^{D \times H_{\hat{f}} \times W_{\hat{f}}}$, where D is the number of depth bins. We adopt linear-increasing discretization (LID) [42], discretizing the depth ground truth to the depth bins. The relation between bin index and depth value states as follows:

$$d_c = d_{min} + \frac{d_{max} - d_{min}}{D(D+1)} \cdot d_i(d_i + 1) \quad (2)$$

where d_i denotes the bin index and d_c denotes the correspond depth value. Then we exploit the focal loss for the auxiliary depth task, trying to classify the depth to a correct depth bin and pay more attention to the foreground object pixels [1].

$$L_{depth} = \frac{1}{W_{\hat{f}} \cdot H_{\hat{f}}} \sum_{u=1}^{W_{\hat{f}}} \sum_{v=1}^{H_{\hat{f}}} FocalLoss(D_{pred}(u, v), D_{gt}(u, v)) \quad (3)$$

C. Depth-Space-Aware Transformer

Implicit neural representation (INR) proves to be an efficient way for 3D objects or 2D images modeling by mapping the coordinates to visual signal. Transformer architecture reveals a powerful weapon for self-modeling or cross-modeling. In this section, we exploit the effective transformer encoder-decoder structure to map the 3D space coordinates to our visual DCPF, enhancing the pixel’s geometry contexts. **3D Coordinates Grid.** Firstly, we need to obtain the 3D coordinates. Since coordinates are continuous, following DSGN [43], we discrete the 3D coordinates to the integer grid and project the points from camera frustum space to 3D space. Specifically, every pixel coordinate in the image indicates $P_j^i = (u_j, v_j, 1, 1)^T$, where (u_j, v_j) means the pixel coordinate in the image. Following [1], [13], we also apply LID [42] to transform the depth within ROI from continuous space to discretization intervals. Then we recovery the discrete bin indexes to depth values d_1, d_2, \dots, d_N (N denotes the number of depth bin), which acts as our depth grid coordinates along the axis orthogonal to the image plane. The camera frustum space points turn to $P_j^m = (u_j \times d_j, v_j \times d_j, d_j, 1)^T$, $d_j \in d_1, d_2, \dots, d_N$. Consequently, we reversely project our camera frustum space points to the 3D world space with the assistance of intrinsics and extrinsics calibration:

$$P_j^{3d} = K_E^{-1} K_I^{-1} P_j^m \quad (4)$$

where $K_E \in R^{4 \times 4}$ and $K_I \in R^{4 \times 4}$ indicate the extrinsics and intrinsics calibration matrix, and $P_j^{3d} = (x_j, y_j, z_j, 1)^T$ denotes 3D coordinates grid calculated from camera frustum

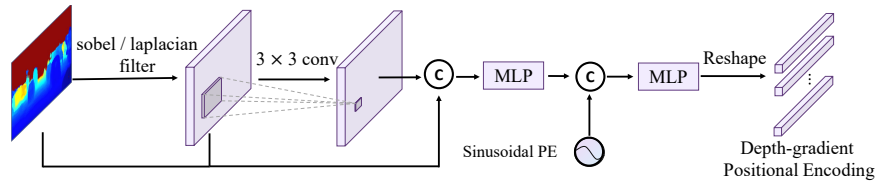


Fig. 5. The structure of Depth-Gradient Positional Encoding (DGPE). The input reveals our depth prediction in above DCPM. The operation of C indicates concatenation. The channels after the 3x3 convolution are three, of two channels for the sobel filter in X and Y axes and one channel for laplacian filter.

space coordinates grid. After normalizing and reshaping, we achieve the normalized 3D coordinates grid $P^{3d} = \{P_j^{3d} \in \mathbb{R}^{(D \times 4) \times H_f \times W_f}\}$.

Space Position Encoder. Implicit neural representation (INR) usually adopts MLPs as the mapping function [14]. Inspired by rapid-developed transformer architecture, which succeeds in modeling the long-range relationships, we adopt the encoder-decoder structure to map the 3d space coordinates to our visual features.

As shown in Figure 1, firstly we deliver our 3D coordinate grid to the MLPs for simply linear transformation. Sequently, we flatten them to $\hat{P}^{3d} \in \mathbb{R}^{N \times (D \times 4)}$, where $N = H_f \times W_f$ is the input of the space position (SP) transformer encoder.

$$F_E = \text{Encoder}(\hat{P}^{3d}) \quad (5)$$

To mitigate the time and memory budget, following [13], we also utilize the Linear transformer [40] to accelerate the attention operation. Figure 3 shows our encoder framework, which consists of transformer self-attention modules.

Depth-Space-Aware Decoder. We adopt the transformer decoder to complete the space coordinates mapping to our visual features DCPF. Figure 4 demonstrates our decoder structure. We feed our DCPF into the decoder instead of learnable embeddings (object query). Simple learnable embedding can't fully represent complex scale variant situations due to the perspective projection [13]. Besides, our DCPF contains depth knowledge, which is more suitable for the transformer decoder to enhance the coordinate mapping.

The encoder's output embeds into the decoder's cross-attention modules and the multi-layer transformer decoder efficiently models the transformation between space coordinates and DCPF, resulting in a more robust mapping. The output of decoder denotes depth-space-aware features (DSAF).

$$F_{DSA} = \text{Decoder}(F_{DCPF}, F_E) \quad (6)$$

Depth-Gradient Positional Encoding (DGPE). To increase position information for transformer's input, adding position embedding to features becomes conventional operation in transformer, such as sinusoidal positional encoding [37] and absolute position encoding [41]. MonoDTR [13] regards depth as hints to produce depth positional embedding. However the depth usually appears indistinct and smooth for object detection task, and is affected with same-depth-different-semantics confusion, which limits the depth's potential.

We propose a new hint, depth gradient, for encoding more clues for the object detection task. Our depth-gradient positional encoding (DGPE) introduces the gradient information to spotlight the edge of objects, which act as valuable

indications for anchor-free object detection. Specifically, we adopt sobel and laplacian filters, which are common detectors for image edge detection. Sobel filter calculates the first derivatives of the image separately for the X and Y axes and laplacian filter calculates second order derivatives in a single pass. As shown in Figure 5, we manipulate our depth prediction with two gradient-based edge filters. We also adopt a 3x3 convolution layer for local perception. Finally we combine all the edge features, depth knowledge with the the init positional embedding, to produce our depth-gradient positional encoding (DGPE).

D. Detection Head and Loss

We reserve the detection head and loss from [11]. Specifically, we adopt the anchor-offset style during training. We predict the class-specific heatmap for the 2D bounding box's center, orientation, dimension, object depth, depth uncertainty and other auxiliary monocular tasks. We also maintain the loss calculation, added with our depth loss. [11] reveals more details about the 3D detection heads and losses.

$$L = \lambda_{depth} L_{depth} + \lambda_{cls} L_{cls} + \lambda_{reg} L_{reg} \quad (7)$$

IV. EXPERIMENTS

A. Dataset, metrics and implementation

KITTI. The popular KITTI [17] dataset officially consists of 7,481 samples for training and 7,518 for testing. It is a common practice [1] to divide training samples into a training set with 3,712 samples and a validation set with 3,769 samples. We conduct ablation studies on the validation split with models trained on the training split. The 3D Average Precision (AP_{3D}) and BEV Average Precision (AP_{BEV}) are two vital evaluation metrics for KITTI 3D object detection. Following previous methods [1], our approach is compared on the test set using AP_{40} instead of AP_{11} . In addition, the Car category has an IoU threshold of 0.7 while the Pedestrian and Cyclist categories have an IoU threshold of 0.5.

Waymo Open Dataset. The Waymo Open Dataset [44] consists of 798 training sequences and 202 validation sequences for vehicles and pedestrians. We experiment on this dataset to showcase the generalization of our method. Due to the large dataset size and high frame rate [1], we use only a subset of the training and validation sets to form our training (3,000 samples) and validation (2,716 samples) sets, with only front camera images. We scale images to size 960×640 , and make labels along the KITTI standard, under which there are no 'hard' category objects. For evaluation, we use 0.7 IoU threshold for the Car category.

TABLE I

COMPARISON ON THE KITTI TEST SET. HIGHEST RESULT IS MARKED WITH RED AND THE SECOND HIGHEST IS MARKED WITH BLUE. IOU = 0.7 IS ADOPTED AS THE THRESHOLD FOR THE 'CAR' CATEGORY. 'IMPROVEMENT' INDICATES THE OFFSET WITH MONOCON [11].

Method	Times(ms)	Car AP_{3D}			Car AP_{BEV}		
		Easy	Mod.	Hard	Easy	Mod.	Hard
SMOKE [2]	30	14.03	9.76	7.84	20.83	14.49	12.75
PatchNet [45]	400	15.68	11.12	10.17	22.97	16.86	14.97
PGD [9]	21	19.05	11.76	9.39	26.89	16.51	13.49
MonoDLE [28]	40	17.23	12.26	10.29	24.79	18.89	16.00
PCT [46]	45	21.00	13.37	11.31	29.65	19.03	15.92
CaDDN [1]	630	19.17	13.41	11.46	27.94	18.91	17.19
MonoEF [47]	30	21.29	13.87	11.71	29.03	19.70	17.26
MonoFlex [25]	35	19.94	13.89	12.07	28.23	19.75	16.89
AutoShape [48]	50	22.47	14.17	11.36	30.66	20.08	15.95
GUPNet [27]	34	20.11	14.20	11.77	-	-	-
Homo [49]	-	21.75	14.94	13.07	29.60	20.68	17.81
MonoDTR [13]	37	21.99	15.39	12.73	28.59	20.38	17.14
MonoDETR [50]	-	23.65	15.92	12.99	32.08	21.44	17.85
MonoDistill [51]	40	22.97	16.03	13.60	31.87	22.59	19.72
MonoCon [11]	26	22.50	16.46	13.95	31.12	22.10	19.00
Ours	46	24.68	17.17	14.14	32.50	23.14	20.30
Improvement	-	+2.18	+0.71	+0.19	+1.38	+1.04	+1.30

Implementation. The network is trained on a single Nvidia Tesla V100S GPU. We adopt an Adam optimizer with an initial learning rate of $2.25e-4$ with a one-cycle learning rate policy, which gradually increases the learning rate to $2.25e-3$. We train our model for 200 epochs with a batch size of 8. To maintain the real-time performance, we adopt the $\frac{1}{16}$ size features to generate 3D coordinates grid. The depth grid ranges from 2m to 46.8m for KITTI dataset and 2m to 55.76m for Waymo dataset, both with 64 depth bins.

B. Main Results

Results in Car category on the KITTI test set. Table I shows the results of our method on the KITTI test set. It compares our method with the state-of-the-art published monocular methods, where the ranking is based on the AP_{3D} metrics at the moderate level. Our proposed method obtains the first rank place among these methods. Compared to the 2nd ranked one, we achieves an improvement of [+1.03%, +0.71%, +0.19%] on AP_{3D} and [+0.42%, +0.55%, +0.58%] on AP_{BEV} . In addition, our method outperforms the baseline with large margins of [+2.18%, +0.71%, +0.19%] on AP_{3D} and [+1.38%, +1.04%, +1.30%] on AP_{BEV} , which proves the effectiveness of our network design.

Results in Pedestrian and Cyclist categories on the KITTI test set. We further present the performance of the Pedestrian and Cyclist categories, as shown in Table II. Our approach is superior to our baseline MonoCon [11] over 1%, and shows competitive performance with other methods ranking within the top 3.

Results in Car category on the subset of Waymo Open dataset. We conduct experiments on the subset of Waymo Open Dataset where we train 3,000 samples and test on the validation subset (2,716 samples), as shown in Table III. We adopt the official codes of CaDDN [1] and MonoCon [11] to perform experiments on the same subset of Waymo Open Dataset for a fair comparison. From Table

TABLE II

COMPARISON ON THE KITTI TEST SET. HIGHEST RESULT IS MARKED WITH RED AND THE SECOND HIGHEST IS MARKED WITH BLUE. IOU = 0.5 IS USED FOR 'CYCLIST' AND 'PEDESTRIAN' CATEGORIES. 'IMPROVEMENT' INDICATES THE OFFSET WITH MONOCON [11].

Method	Pedestrian AP_{3D}			Cyclist AP_{3D}		
	Easy	Mod.	Hard	Easy	Mod.	Hard
MonoDLE [28]	9.64	6.55	5.44	4.59	2.66	2.45
CaDDN [1]	12.87	8.14	6.76	7.00	3.41	3.30
MonoEF [47]	4.27	2.79	2.21	1.80	0.92	0.71
MonoFlex [25]	9.43	6.31	5.26	4.17	2.35	2.04
GUP Net [27]	14.72	9.53	7.87	4.18	2.65	2.09
Homo [49]	11.87	7.66	6.82	5.48	3.50	2.99
MonoDTR [13]	15.33	10.18	8.61	5.05	3.27	3.19
MonoDistill [51]	12.79	8.17	7.45	5.53	2.81	2.40
MonoCon [11]	13.10	8.41	6.94	2.80	1.92	1.55
Ours	14.16	9.67	8.26	5.88	3.30	2.85
Improvement	+1.06	+1.26	+1.32	+3.08	+1.38	+1.30

TABLE III

COMPARISON ON THE SUBSET OF WAYMO DATASET. HIGHEST RESULT IS MARKED WITH RED AND THE SECOND HIGHEST IS MARKED WITH BLUE. WE ADOPT IOU = 0.7 AS THE THRESHOLD FOR 'CAR' CATEGORY.

Method	Car AP_{3D}		Car AP_{BEV}	
	Easy	Mod.	Easy	Mod.
CaDDN [1]	61.48	54.33	67.29	57.09
MonoCon [11]	64.24	56.17	69.77	59.64
Ours	67.42	59.07	70.78	62.36
Improvement	+3.18	+2.90	+1.01	+2.72

III, our method outperforms MonoCon [11] with margins of [+3.18%, +2.90%] on Car AP_{3D} and [+1.01%, +2.72%] on Car AP_{BEV} , which indicates great generalization.

Time analysis. We measure our model's average inference time on a single Nvidia Tesla V100S GPU. As shown in Table I, our model can achieve real-time performance at 25 FPS, which validates the efficiency of our framework. Specifically, our MonoPGC runs $15.75\times$ faster than CaDDN [1], for our method avoids time-consuming lifting module. Our pixel depth estimation and transformer bring some additional computation with a little slower speed than MonoCon [11], but our result greatly overcomes the MonoCon [11].

C. Ablation Study

We conduct ablation studies on the KITTI validation split with models trained on the training split.

Importance of each proposed components. Table IV shows the ablation studies about our method, and our three modules perform improving the detection performance. When adding depth cross-attention pyramid module (c) and depth-gradient positional encoding (d), our AP_{3D} and AP_{BEV} both increase especially in the easy category, since DCPM enhances more robust depth-aware features and DCPE strengthens the unitary depth information, which enriches the pixel geometry contexts for 3D detection. Setting (b) introduces an individual depth-space-aware transformer where 2D visual features display the transformer's input. Without depth knowledge and other surround-view images to make up the scene information, 2D features can't adapt to the 3D position well, making little influence on our baseline. However, after

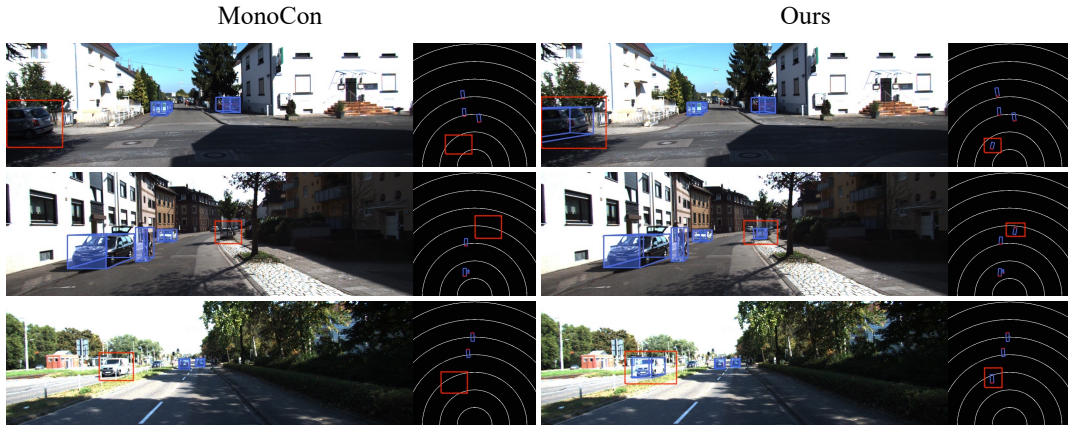


Fig. 6. Qualitative results on KITTI val set. The blue bounding box denotes the predicted results. In the BEV image, the red side denotes the cars’ head. Our proposed method dominates the baseline.

TABLE IV

ABLATION STUDIES OF EACH PROPOSED COMPONENTS.

	DCPM	DCPE	DSAT	$AP_{3D}@IoU=0.7$			$AP_{BEV}@IoU=0.7$		
				Easy	Mod.	Hard	Easy	Mod.	Hard
(a)	-	-	-	22.88	17.81	15.19	31.17	23.78	20.58
(b)	-	-	✓	22.85	17.84	15.12	30.96	23.57	20.57
(c)	✓	-	-	24.73	17.98	15.25	33.26	23.91	20.61
(d)	✓	✓	-	24.91	18.18	15.31	33.28	23.96	20.60
(e)	✓	-	✓	25.42	18.62	15.60	33.40	24.10	20.58
(f)	✓	✓	✓	25.67	18.63	15.65	34.06	24.26	20.78

TABLE V

ABLATION STUDIES OF DIFFERENT COORDINATES GRID SETTING ON THE KITTI VALIDATION SET. ‘D.V.’ DENOTES THE DIRECTION VECTOR.

Ablation	$AP_{3D}@IoU=0.7$			$AP_{BEV}@IoU=0.7$		
	Easy	Mod.	Hard	Easy	Mod.	Hard
(a) No coord. grid	22.88	17.81	15.19	31.17	23.78	20.58
(b) d.v. [16] w/o DCPF	22.85	17.87	15.22	31.24	23.81	20.60
(c) d.v. [16] w DCPF	24.77	18.16	15.28	33.00	23.85	20.62
(d) 3D coord. grid w/o DCPF	22.85	17.84	15.12	30.96	23.57	20.57
(e) 3D coord. grid w DCPF	25.67	18.63	15.65	34.06	24.26	20.78

introducing DCRM (e), depth-aware features match well with the space coordinates. Finally, our method achieves the highest accuracy applying all modules.

Comparison with different coordinates grid setting. We investigate different coordinates grid setting with/without depth cross-attention pyramid features (DCPF) in monocular detection, as shown in Table V. We refer to the work [16] to introduce the direction vector as the transformer encoder’s input. Setting (a) denotes our baseline. From Table V, depth-aware DCPF could maximize the potential of spatial encoding (c)(e), but the direction vector method doesn’t perform superior to our 3D coordinates grid. Its depth 1 setting may limit the feature enhancement and the monocular detection task doesn’t have other surround-view images to assist scene understanding. Setting (b) and (d) also reveal that visual features perform not well with spatial encoding in monocular detection task.

Comparison with different positional encodings. We discuss the effects of the proposed depth-gradient positional encoding (DGPE) in Table VI. We compare our DGPE with several previous positional encodings, including without using positional encoding (No PE), sinusoidal positional encoding [37], absolute positional encoding (APE) [41],

TABLE VI

ABLATION STUDIES OF DIFFERENT POSITIONAL ENCODINGS ON THE KITTI VALIDATION SET.

Method	$AP_{3D}@IoU=0.7$			$AP_{BEV}@IoU=0.7$		
	Easy	Mod.	Hard	Easy	Mod.	Hard
No PE	23.23	17.33	14.74	31.32	23.26	20.07
Sinusoidal [37]	24.78	18.17	15.09	33.00	23.85	20.33
APE [41]	24.34	18.36	15.31	32.42	23.98	20.41
DPE [13]	25.42	18.50	15.57	33.48	24.00	20.59
DGPE	25.67	18.63	15.65	34.06	24.26	20.78

and depth positional encoding (DPE) [13]. Our proposed DGPE achieves better results on KITTI validation set, for DGPE extracts more meticulous and attentive clues with the gradient-based filter, which enhances the pixel-level geometry information compared to the other encodings.

D. Qualitative Results

We provide the qualitative examples on the KITTI validation set, as shown in Figure 6. Compared with MonoCon [11], our method’s prediction reveals more accurate and robust. MonoCon [11], short of geometry clues, fails to detect vehicles affected by too dark or too bright illumination and occlusion. Our method succeeds in these hard cases, contributed from the features’ sufficient pixel geometry contexts. The supplementary material includes more qualitative results.

V. CONCLUSION

In this paper, we propose **MonoPGC**, a novel end-to-end monocular 3D object detection framework with adequate pixel geometry contexts. The proposed depth cross-attention pyramid module (**DCPM**) expands the receptive field enabling a more comprehensive and precise perception of depth geometry content. The depth-space-aware transformer (**DSAT**) efficiently incorporate 3D geometry position with DCPF, mapping geometry contents to pixel features. Finally depth-gradient positional encoding (**DGPE**) acts as a valuable indication for anchor-free object detection. Extensive experiments on the KITTI dataset validate that our model achieves the state-of-the-art performance on the KITTI dataset and a portion of the Waymo Open dataset. In future work, we would like to explore novel monocular architecture without the explicit depth estimation.

REFERENCES

- [1] C. Reading, A. Harakeh, J. Chae, and S. L. Waslander, "Categorical depth distribution network for monocular 3d object detection," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2021, pp. 8555–8564.
- [2] Z. Liu, Z. Wu, and R. Tóth, "SMOKE: Single stage monocular 3d object detection via keypoint estimation," in *IEEE Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020, pp. 996–997.
- [3] X. Guo, S. Shi, X. Wang, and H. Li, "LIGA-Stereo: Learning lidar geometry aware representations for stereo-based 3d detector," in *IEEE Int. Conf. on Computer Vision (ICCV)*. IEEE, 2021, pp. 3153–3163.
- [4] P. Li and H. Zhao, "Monocular 3d detection with geometric constraint embedding and semi-supervised training," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5565–5572, 2021.
- [5] G. Brazil and X. Liu, "M3D-RPN: monocular 3d region proposal network for object detection," in *IEEE Int. Conf. on Computer Vision (ICCV)*. IEEE, 2019, pp. 9287–9296.
- [6] 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 *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2019, pp. 8445–8453.
- [7] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," *arXiv preprint arXiv:1904.07850*, 2019.
- [8] P. Li, H. Zhao, P. Liu, and F. Cao, "Rtm3d: Real-time monocular 3d detection from object keypoints for autonomous driving," in *European Conf. on Computer Vision Workshops (ECCVW)*. Springer, 2020, pp. 644–660.
- [9] T. Wang, X. Zhu, J. Pang, and D. Lin, "Probabilistic and geometric depth: Detecting objects in perspective," in *Conference on Robot Learning*. PMLR, 2021.
- [10] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3d object detection and tracking," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2021, pp. 11 784–11 793.
- [11] X. Liu, N. Xue, and T. Wu, "Learning auxiliary monocular contexts helps monocular 3d object detection," in *AAAI Conf. on Artificial Intell. (AAAI)*, 2021.
- [12] X. Weng and K. Kitani, "Monocular 3d object detection with pseudo-lidar point cloud," in *IEEE Int. Conf. on Computer Vision Workshops (ICCVW)*. IEEE, 2019, pp. 0–0.
- [13] K.-C. Huang, T.-H. Wu, H.-T. Su, and W. H. Hsu, "MonoDTR: Monocular 3d object detection with depth-aware transformer," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [14] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *European Conf. on Computer Vision Workshops (ECCVW)*. Springer, 2020, pp. 405–421.
- [15] Y. Liu, T. Wang, X. Zhang, and J. Sun, "PETR: Position embedding transformation for multi-view 3d object detection," *arXiv preprint arXiv:2203.05625*, 2022.
- [16] B. Zhou and P. Krähenbühl, "Cross-view transformers for real-time map-view semantic segmentation," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [17] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012, pp. 3354–3361.
- [18] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, "Monocular 3d object detection for autonomous driving," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016, pp. 2147–2156.
- [19] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, "SSD-6D: Making rgb-based 3d detection and 6d pose estimation great again," in *IEEE Int. Conf. on Computer Vision (ICCV)*, 2017, pp. 1521–1529.
- [20] J. Ku, A. D. Pon, and S. L. Waslander, "Monocular 3d object detection leveraging accurate proposals and shape reconstruction," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 11 867–11 876.
- [21] A. Simonelli, S. R. Buló, L. Porzi, M. López-Antequera, and P. Kontschieder, "Disentangling monocular 3d object detection," in *IEEE Int. Conf. on Computer Vision (ICCV)*, 2019, pp. 1991–1999.
- [22] Y. Liu, Y. Yixuan, and M. Liu, "Ground-aware monocular 3d object detection for autonomous driving," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 919–926, 2021.
- [23] L. Liu, J. Lu, C. Xu, Q. Tian, and J. Zhou, "Deep fitting degree scoring network for monocular 3d object detection," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 1057–1066.
- [24] Y. Cai, B. Li, Z. Jiao, H. Li, X. Zeng, and X. Wang, "Monocular 3d object detection with decoupled structured polygon estimation and height-guided depth estimation," in *AAAI Conf. on Artificial Intell. (AAAI)*, 2020, pp. 10 478–10 485.
- [25] Y. Zhang, J. Lu, and J. Zhou, "Objects are different: Flexible monocular 3d object detection," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 3289–3298.
- [26] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka, "3d bounding box estimation using deep learning and geometry," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7074–7082.
- [27] Y. Lu, X. Ma, L. Yang, T. Zhang, Y. Liu, Q. Chu, J. Yan, and W. Ouyang, "Geometry uncertainty projection network for monocular 3d object detection," in *IEEE Int. Conf. on Computer Vision (ICCV)*. IEEE, 2021, pp. 3111–3121.
- [28] X. Ma, Y. Zhang, D. Xu, D. Zhou, S. Yi, H. Li, and W. Ouyang, "Delving into localization errors for monocular 3d object detection," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 4721–4730.
- [29] Y. Wang, V. C. Guizilini, T. Zhang, Y. Wang, H. Zhao, and J. Solomon, "DETR3D: 3d object detection from multi-view images via 3d-to-2d queries," in *Conference on Robot Learning*. PMLR, 2022, pp. 180–191.
- [30] Z. Chen, Z. Li, S. Zhang, L. Fang, Q. Jiang, and F. Zhao, "Graph-DETR3D: Rethinking overlapping regions for multi-view 3d object detection," *arXiv preprint arXiv:2204.11582*, 2022.
- [31] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Q. Yu, and J. Dai, "BEVFormer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers," *arXiv preprint arXiv:2203.17270*, 2022.
- [32] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European Conf. on Computer Vision Workshops (ECCVW)*. Springer, 2020, pp. 213–229.
- [33] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "DeepSDF: Learning continuous signed distance functions for shape representation," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 165–174.
- [34] C. Jiang, A. Sud, A. Makadia, J. Huang, M. Nießner, T. Funkhouser, et al., "Local implicit grid representations for 3d scenes," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 6001–6010.
- [35] K. Genova, F. Cole, A. Sud, A. Sarna, and T. Funkhouser, "Local deep implicit functions for 3d shape," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 4857–4866.
- [36] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3d reconstruction in function space," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4460–4470.
- [37] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Conference and Workshop on Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.
- [38] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 2403–2412.
- [39] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2881–2890.
- [40] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret, "Transformers are rnns: Fast autoregressive transformers with linear attention," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5156–5165.
- [41] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Int. Conf. on Learning Representations (ICLR)*, 2021.
- [42] Y. Tang, S. Dorn, and C. Savani, "Center3d: Center-based monocular 3d object detection with joint depth understanding," in *DAGM German Conference on Pattern Recognition*. Springer, 2020, pp. 289–302.

- [43] Y. Chen, S. Liu, X. Shen, and J. Jia, "DSGN: Deep stereo geometry network for 3d object detection," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 12 536–12 545.
- [44] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 2446–2454.
- [45] X. Ma, S. Liu, Z. Xia, H. Zhang, X. Zeng, and W. Ouyang, "Rethinking pseudo-lidar representation," in *European Conf. on Computer Vision Workshops (ECCVW)*, 2020, pp. 311–327.
- [46] L. Wang, L. Zhang, Y. Zhu, Z. Zhang, T. He, M. Li, and X. Xue, "Progressive coordinate transforms for monocular 3d object detection," in *Conference and Workshop on Neural Information Processing Systems (NeurIPS)*, 2021, pp. 13 364–13 377.
- [47] Y. Zhou, Y. He, H. Zhu, C. Wang, H. Li, and Q. Jiang, "Monocular 3d object detection: An extrinsic parameter free approach," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2021, pp. 7556–7566.
- [48] Z. Liu, D. Zhou, F. Lu, J. Fang, and L. Zhang, "Autoshape: Real-time shape-aware monocular 3d object detection," in *IEEE Int. Conf. on Computer Vision (ICCV)*. IEEE, 2021, pp. 15 641–15 650.
- [49] J. Gu, B. Wu, L. Fan, J. Huang, S. Cao, Z. Xiang, and X.-S. Hua, "Homography loss for monocular 3d object detection," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [50] R. Zhang, H. Qiu, T. Wang, X. Xu, Z. Guo, Y. Qiao, P. Gao, and H. Li, "MonoDETR: Depth-aware transformer for monocular 3d object detection," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [51] Z. Chong, X. Ma, H. Zhang, Y. Yue, H. Li, Z. Wang, and W. Ouyang, "MonoDistill: Learning spatial features for monocular 3d object detection," in *Int. Conf. on Learning Representations (ICLR)*, 2022.