ODD-M3D: Object-wise Dense Depth Estimation for Monocular 3D Object Detection

Chanyeong Park[1], Heegwang Kim[1], Junbo Jang[2], and Joonki Paik[1,2], *Senior Member, IEEE*

***Abstract*—Despite the significant benefits of low cost and scalability associated with monocular 3D object detection, accurately estimating depth from a single 2D image remains challenging due to the typical ill-posed nature of the problem. To address this issue, we propose a new method that improves depth estimation accuracy by randomly sampling object-wise points instead of relying on a single center point, which is a common practice in conventional methods. To generate the object-wise multiple reference points, we create a sampling space and obtain the ground truth by moving them from the sampling space to the object space. For this reason, the proposed approach is named ODD-M3D, which stands for Object-wise Dense Depth estimation for Monocular 3D object detection. In addition, we conduct an ablation study comparing LiDAR-guided and random sampling methods to identify the limitations of using point cloud data for image-based 3D object detection tasks. The proposed network achieved better performance by allowing for dense depth estimation instead of sparse depth estimation, which is typical in conventional networks.**

***Index Terms*—Monocular 3D object detection, Object detection, Convolutional neural network**

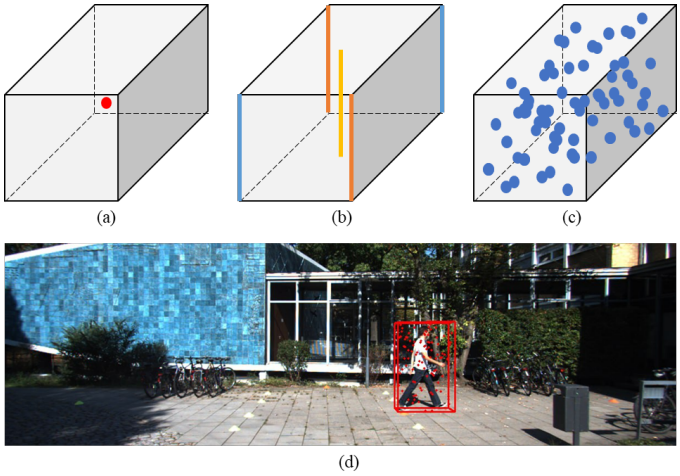

Fig. 1. Comparison of 3 different depth estimation methods. (a) Center point for direct depth, (b) object height for Keypoint Depths, (c) the proposed reference points and (d) the example of Reference Points projected into image plane.

I. Introduction

TECHNOLOGIES such as autonomous driving systems and indoor robot vision systems have gained significant attention for their ability to facilitate intelligent perception and safe movement in the surrounding environment without human resources. In recent years, advances in these technologies have emphasized the growing significance of three-dimensional (3D) object detection. 3D object detection is a crucial computer vision technology that has gained significant attention due to its applications in autonomous driving systems and indoor robot vision systems. Unlike conventional two-dimensional (2D) object detection methods, 3D object detection can accurately predict the location, size, and orientation of objects in 3D scenes. While various sensors such as RGB cameras, LiDAR, and radar have contributed to the development of 3D object detection, monocular 3D object detection, which utilizes a single RGB camera, has emerged as a promising approach due to its cost-effectiveness and ease of implementation. Monocular 3D object detection algorithms typically extend well-known 2D object detection networks to predict the 3D bounding box of objects in 2D RGB images. However, a major challenge in monocular 3D object detection is the estimation of lost 3D information from 2D images. Most existing methods rely on sparsely predicted depth based on the center points of objects, leading to inaccurate 3D localization [1]–[5].

Accordingly, several methods leverage additional data to alleviates the inaccurate depth estimation [6]–[9] in recent years. However, the sparsity of depth estimation, the computational complexity and slow inference time for monocular 3D object detection are still challenging. To address this issue, we propose an approach to enhance the accuracy of monocular camera-based 3D object detection networks by replacing conventional sparse depth estimation with object-wise dense depth estimation using point sampling. Main contributions of our method involves:

1) **Random point sampling and LiDAR-guided sampling:** We randomly sample points from the bounding box area of each object and utilize point cloud data from a LiDAR sensor to sample points around each object.
2) **Reference Point Depth Estimation:** We propose a new dense depth estimation method using pre-generated sampled points.
3) **The proposed method demonstrates superior performance compared to other state-of-the-art networks.**

By leveraging both the center coordinate of the object and its surrounding area, our method can estimate object-wise dense depth, significantly improving the accuracy of 3D localization in monocular 3D object detection.

Manuscript received ***

[1]Department of Image, Chung-Ang University, 84 Heukseok-ro, Seoul 06974, Korea. [2]Department of Artificial Intelligence, Chung-Ang University, 84 Heukseok-ro, Seoul 06974, Korea. (e-mail: chanyeong@ipis.cau.ac.kr, heegwang@ipis.cau.ac.kr, junbojang@ipis.cau.ac.kr, and paikj@cau.ac.kr)

II. Related Work

Monocular camera-based 3D object detection networks have been developed in several ways, based on whether additional data is employed during the training stage. Among the monocular 3D object detection networks without using extra data during training, CenterNet-style networks are the mainstream. These one-stage keypoint-based anchor-free 2D detectors are widely used [1]. Liu et al. referred to the difference between the 2D and 3D center points, and proposed SMOKE which regresses the keypoints based on the projected 3D center point instead of using the 2D center point [2]. MonoPair utilizes geometric information in a single 2D image by estimating pair-wise geometrical constraints on surrounding objects [3]. MonoDLE estimates the offset between the center point of the 2D bounding box and the projected 3D center point, and then uses the IoU-oriented optimization method for 3D size estimation [4]. MonoFlex considers geometric information in a 2D image and leverages the height information of an object to improve depth prediction accuracy. In contrast, prior methods rely only on direct regression using the center point of each object to estimate depth [5]. M3D-RPN, which is a one-stage anchor-based detectors, uses only RPNs to perform 3D object detection without using other sub-networks [10]. By defining 2D and 3D anchors together, a region proposal is generated by utilizing the correlation between 2D scale and 3D depth as a prior. For better 3D bounding box prediction, M3D-RPN generates a spatial-aware feature using depth-aware convolution. GrooMeD-NMS (Non Maximum Suppression) extracts 2D features and sets the best 3D box candidates with differentiable NMS [11]. In the case of a two-stage method, GS3D uses relatively accurate 2D candidates to generate the self-supervised 3D guidance, because the performance of the 2D detectors itself produces sufficiently reliable resultsNMS [12]. In addition, GS3D generates 3D features through inverse operations based on created 3D guidance. MonoDIS separates the training process of 2D and 3D features to alleviate the negative effect on optimization caused by the mismatch between 2D and 3D features [13].

On the other hand, several monocular 3D object detection methods that require prior knowledge in the form of additional data, such as depth maps, point cloud data, and shape information, have been the subject of active research. A 2D image is obtained by projecting the 3D scene and does not contain depth information. Therefore, using depth information with a single 2D image is expected to improve the accuracy in monocular 3D object detection tasks. To incorporate precise depth information during the training phase, an auxiliary depth estimation network is developed. One approach that uses depth information to train monocular 3D object detection networks is the Pseudo-LiDAR method [6], which converts a depth map into pseudo-LiDAR representation. The depth map is first estimated by monocular depth estimation network and then transformed into a pseudo-LiDAR point cloud data. Next, 3D object detection is performed using LiDAR-based detector. AM3D fuses the RGB features of 2D images with pseudo-LiDAR point cloud data by using attention mechanism as a gate function to amplify the flow of feature informatio nmethod [7]. Mono3DPLiDAR utilizes an instance mask instead of a bounding box and improves the performance of 3D object detection by reducing points that do not correspond to objects in point cloud frustum [8]. PCT refines the predictions by a confidence-aware localization boosting mechanism and uses a global context encoding to solve the problem of inaccurate localization of pseudo-LiDAR [14]. ForeSeE analyzes the data distribution of foreground and background features and detects the 3D objects by separating foreground and background using pseudo-LiDAR with the analyzed distribution [15]. D4LCN is an example of depth map-guided approach which employs depth maps to train a monocular 3D object detection network [16]. Instead of relying on a global kernel, the complete image is learned through local information from each pixel and channel, as well as depth maps. While pseudo-LiDAR-based 3D object detection experiences degraded performance when converted point cloud data acquired from monocular depth estimation network is inaccurate, D4LCN can maintain a relatively consistent level of 3D object detection performance even when the depth estimation results are inaccurate. MonoGRNet pointed out that existing methods do not focus on object localization, and to solve this problem, MonoGRNet divides 3D localization into several sub-tasks [9]. In addition, instance level depth estimation is used to increase the accuracy of depth estimation. CaDDN predicts the depth distribution and the feature extraction in parallel and uses the estimated depth distribution as a frustum feature grid [17]. The frustum feature is then transformed to voxel grid using the camera calibration parameter to generate a 3D voxel feature volume. When performing 3D object detection, CaDDN transforms the generated 3D voxel feature to BEV (Birds Eye View) feature and utilize BEV-based detector to detect the 3D objects. MonoPSR utilizes LiDAR data in the learning process to perform instance-wise 3D reconstruction through shape and scale information of objects [18]. MonoPSR estimates the 3D center point of the object and utilizes the reconstructed instance point cloud data to improve the 3D localization accuracy of monocular 3D object detection. MonoRUn has the effect of better estimating the shape of 3D objects and mitigating overfitting problem with LiDAR supervision in the proposed monocular 3D object detection network [19].

III. Proposed Method

A. Problem Statement

Monocular 3D object detection involves predicting the 3D information of an object and generating its 3D bounding box as output. To predict the 3D bounding box, we need to determine the object's position (x, y, z), dimensions (w, h, l), and orientation (θ). However, due to the use of a monocular camera, the z-axis, which corresponds to depth information, is lost as the 3D scene is projected onto a 2D image plane.

$$f(h(u, v)) = (x, y, z), \tag{1}$$

where f and h respectively represent a deep neural network and image plane. When a monocular 3D object detection network takes a 2D image as input and generates 3D bounding

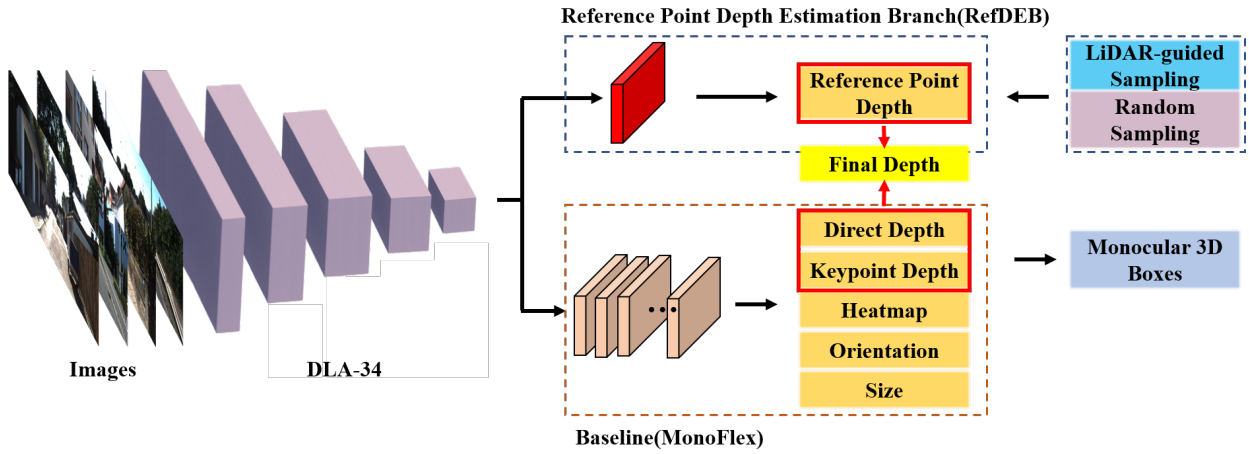

Fig. 2. The architecture of the proposed method. DLA-34 extracts feature maps from input 2D images. The extracted feature maps go through both baseline and proposed reference point depth branch. LiDAR-guided and random sampling method are used to train reference point depth estimation branch (RefDEB). Reference point depth is combined with direct depth and keypoint depth to obtain final depth with uncertainty-based soft ensemble.

boxes as output, the lack of depth information can lead to inaccurate 3D localization. To address this issue, additional information is needed to compensate for the lack of depth information in the 2D image. In our proposed method, we sample multiple points based on the 3D bounding box of each object to predict depth densely, in contrast to existing methods that only rely on regression based on the object's center point to predict depth. As a result, our proposed method alleviates the mismatch between the 2D image and 3D space by enabling object-wise dense depth estimation.

As shown in Fig. 2, the architecture of the proposed method involves using DLA-34 [20] extract feature maps from 2D input images. The extracted feature maps then go through two separate branches. The first branch estimates various 3D information including heatmap, direct depth, keypoint depth, orientation, and size. The second branch, which we propose, is the reference point depth estimation branch, consisting of dilated convolution and coordinate convolution. The objective of this branch is to estimate the object-wise dense depth map and output the depths of five reference points using obtained depth map.

B. Architecture Overview

We propose the reference points depth branch, which can densely estimate the depth based on the 3D bounding box of each object using point random sampling. The baseline network is the MonoFlex [5], a CenterNet-style monocular 3D object detection network, with DLA-34 [20], which has the parameters of approximately 15.73M, serving as the backbone network. The deep features extracted from the backbone network then undergo processing through the proposed Reference point Depth Estimation Branch (RefDEB) and 3D branch, respectively. We constructed a 3D branch following the baseline network, and the details are as follows. First, the heatmap head predicts the 2D location of the center point of each object and the class information of each object. The predicted 2D center coordinate is used to obtain the 3D center coordinate of the object. We estimate the offset between the 2D and 3D center points based on the approximate 2D center point. To create more explicit bounding boxes, this offset is used to predict ten keypoints, including eight vertices $\{k_1, ..., k_8\}$ of the bounding box and the top and bottom center points $\{c_{btm}, c_{top}\}$ as:

$$\delta_{kpts} = \frac{kpts}{S} - \left\lfloor \frac{C_{2d}}{S} \right\rfloor, \quad kpts \in \{c_{3d}, c_{btm}, c_{top}, k_1, ..., k_8\}, \tag{2}$$

S and C_{2d} downsampling ratio between input image and the output feature map of the network, and 2D center point. The offsets between the keypoints, including the 3D center point of the object c_{3d}, and the 2D center coordinate c_{2d}, are calculated as shown in Figure 3. And, the loss function is defined as

$$L_{offset} = \sum_{kpts} |\delta^*_{kpts} - (\frac{kpts}{S} - |\frac{c_{2d}}{S}|)|, \tag{3}$$

where δ^* represent the ground truth offset of each $kpts$. The uncertainty-based depth estimation method has become popular in recent years, and many monocular 3D object detection networks have adopted this approach. In our proposed network, we also use the Laplacian likelihood method to model the uncertainty of all the depths we use, including one direct depth, three geometry depths, and five reference points depths. First, the direct depth represents the distance from the center point of objects in the image to the camera as shown in Figure 1(a), and the direct depth is calculated as:

$$z_r = \frac{1}{\sigma(z_o)} - 1, \sigma(x) = \frac{1}{1 + e^{-x}}, \tag{4}$$

z_r, z_0 and σ denote the absolute depth, network output and the uncertainty. And, the keypoint depth denotes the depth calculated using the height information of the predicted 3D bounding box of objects, and the keypoint depth can simply calculated as:

$$z_{kpt} = \frac{f \times H}{h_{2d}}, \tag{5}$$

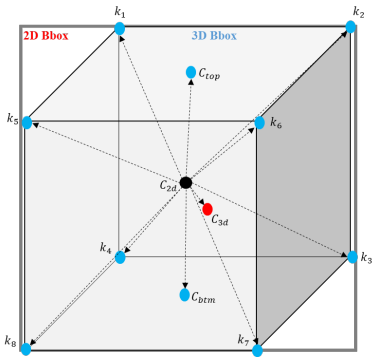

Fig. 3. Illustration of the offset.

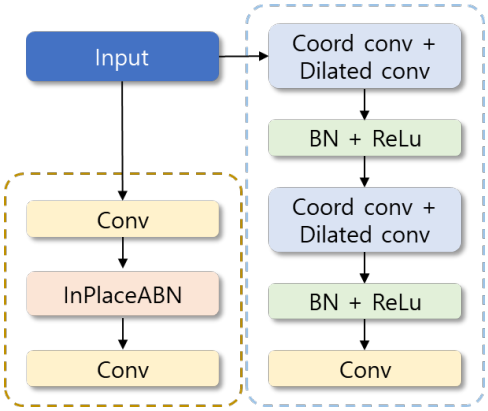

Fig. 4. The architecture of baseline (left dashed box) and RefDEB (right dashed box), which consists of coordinate convolution and dilated convolution blocks.

where z_{kpt}, f, H, and h_{2d} denote the keypoint depth, focal length, estimated object height and pixel height, respectively. The following loss function is defined as

$$L_{depth} = \lambda \frac{\|z^* - z\|_1}{\sigma_{pred}} + \log(\sigma_{pred}). \tag{6}$$

The uncertainty and following depth can be calculated as equation 4. The model is designed in such a way that when the confidence of the predicted depth is low, the uncertainty σ_{pred} increases. As a result, the loss value increases as can be seen in the form of equation 6.

Algorithm 1 Reference Point Generation using Random Sampling.

Input: The number of points n, 8 corner points $\{C_i\}_{i=1}^{8}$ and dimension $D = (w, h, l)$
Output: Ground truth random sampling-based reference points P_{3D}

1: Create an arbitrary sampling space $S \leftarrow (w, h, l)$
2: $\{P_i^S\}_{i=1}^{n} \leftarrow$ Sample n points in S
3: $\{P_i^D\} \leftarrow \{P_i^S\} \times D$
4: $\{P_i^D\}_{i=1}^{4} \leftarrow$ select 4 points in $\{P_i^D\}$ and corresponding $\{q_i\}_{i=1}^{4} \in C_i$
5: Transformation matrix $X \leftarrow ((p_i^D)^T p_i^D)^{-1}(p_i^D)^T q_i$
6: Sampled points in object space O
$P_i^D \leftarrow P_i^D \times X$
7: Points projected onto image plane I
$P_{xy}^I \leftarrow$ PROJECT(P_{3D}^O)
8: $P_{3D} \leftarrow$ CONCATENATE(P_{xy}^I, P_z^O)

C. Reference Point Generation

The process of generating reference points using random sampling is outlined in Algorithm 1. The algorithm is described in detail as follows:

Sampling Space. Our proposed method aims to estimate dense depth from a single 2D image by using various reference points, instead of relying on only one center point. To sample these reference points, we create a sampling space with the same size as the 3D cuboid of each object and randomly sample n points in this space, including the eight corner points and two center points of the cuboid to account for boundary effects. The range of n is set between 10 and 5500 as shown in Table VI and VII.

Transformation Matrix. In this step, the reference points that were generated in the sampling space need to be shifted to the object space. Since both sampling and object spaces have the same size and shape, the transformation matrix can be obtained using the corner points of the corresponding 3D cuboid. To obtain the transformation matrix, at least four points are needed, and we choose any four points from the eight corner points and two center points available. The procedure for obtaining the transformation matrix X can be expressed as:

$$\begin{bmatrix} p_1' \\ p_2' \\ p_3' \\ p_4' \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix} X, \tag{7}$$
$$X = (P^T P)^{-1} P^T P',$$

where P' represents $\{p_1', \ldots, p_4'\}$ the set of corner points in the object space, and P represents $\{p_1, \ldots, p_4\}$ the set of corner points in the sampling space. We obtain the matrix X using the pseudo-inverse method. After obtaining the transformation matrix, the reference points in the sampling space are shifted to the object space and then projected onto the image plane. The procedure for processing the sampled reference points on the image plane using camera metrics is as follows:

$$P_{2D} = K \begin{bmatrix} R & T \end{bmatrix} P_{3D} \tag{8}$$

where K represents the intrinsic matrix, $\begin{bmatrix} R & T \end{bmatrix}$ the extrinsic matrix, P_{2D} the projected sampled reference points in 3D space.

D. Reference Point Depth Estimation Branch (RefDEB)

We propose a branch of a neural network architecture, entitled "Reference point Depth Estimation Branch (RefDEB)", that is trained to predict depth values for each object in an image using ground truth values obtained in the previous

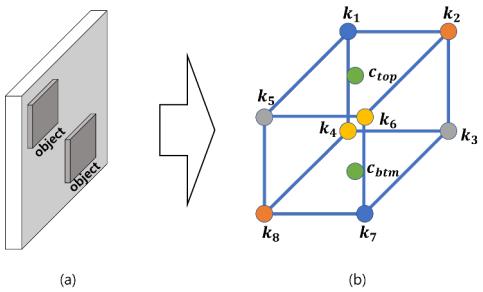

Fig. 5. Reference point Depth Estimation Branch (RefDEB): (a) object-wise dense depth map extracted from RefDEB and (b) the output of RefDEB that involves five reference point depths averaging the two depth values located on each diagonal line expressed as same color.

step. The training process for this branch involves predicting the depth value for each object and comparing it with a predefined ground truth mask to measure the accuracy of the predicted depth values. Object-wise dense depth map and the proposed the depth of reference points are shown in Fig. 5. In RefDEB, depth is predicted densely for each object, and the training process involves measuring the accuracy of the predicted depth value against a predefined ground truth mask. The RefDEB is composed of a dilated convolution and a coordinate convolution, as shown in Fig. 4.

Due to the limited resolution of 2D images, occlusion or object truncation can occur during the processing of 3D scenes. To address these issues, a dilated convolution is employed to capture more contextual information surrounding the object. Additionally, since the ground truth values used for training are obtained from sampled points, the model must predict the depth values for all points within the 3D bounding box of the object. Therefore, we use the coordinate convolution to precisely predict the positional coordinates of the sampled points for each object and obtain the corresponding depth values. Consequently, we generate an object-wise dense depth map, which is optimized using L_1 loss during the training process. The coordinate convolution technique concatenates the x, y coordinate map to the input feature map and is employed to predict the position of an image. Thus, we use the coordinate convolution to obtain precise depth predictions for the object-wise sampled points. The generated the object-wise depth map is combined with the direct depth and geometry depth to achieve more precise depth predictions. To accomplish this, we calculate the depth of the eight vertices of the bounding box and the depth corresponding to the upper and lower center coordinates based on the generated object-wise dense depth map. We then average the 10 depth values located on each diagonal line as shown in Fig. 5. The resulting depth of five reference points are then merged through the direct and geometry depths using a soft ensemble method, which can be expressed using the following equation:

$$z_{final} = (\sum_{i=1}^{9} \frac{z_i}{\sigma_i}) / (\sum_{i=1}^{9} \frac{1}{\sigma_i}) \tag{9}$$

E. LiDAR-guided Sampling

This section presents an analysis of the proposed network utilizing the LiDAR-guided sampling technique. The RefDEB is trained by sampling point cloud data, which enables to perform advanced depth estimation. Point cloud data acquired using a LiDAR sensor is characterized by precise depth information. As a result, we expect that incorporation of point cloud data in the learning process via sampling will enhance the performance of object-wise dense depth estimation. To use point cloud data in the learning process of the monocular 3D object detection network, we reduce the point cloud map acquired by rotating 360 degrees to only include points present in the same direction as the RGB image. We then exclude other points because we only need points that exist in the 3D bounding box of the object within the forward point cloud map. The ground truth data for LiDAR-based sampled points are obtained in the same manner as when obtaining ground truth data for randomly sampled points. The ground truth values of each LiDAR-based sampled point are obtained by concatenating the z values of each point among the (x, y, z) values of each point and the corresponding (u, v) image plane obtained by processing each point. The algorithm for generating ground truth of LiDAR-based sampled points is described as shown in Algorithm 2:

Algorithm 2 Generating LiDAR-guided Reference Points

Input: Reduced point cloud map P_i, 3D bounding box information B_{3D}
Output: Ground truth LiDAR-guided reference points P_{obj}

for $((x_1, y_1, z_1), ..., (x_i, y_i, z_i)) \in P_i$ **do**
  if (x_i, y_i, z_i) in B_{3D} **then**
    $I_{xy} \leftarrow$ PROJECT_TO_IMAGE(x_i, y_i, z_i)
    $P_{obj} \leftarrow$ STACK(CONCATENATE (I_{xy}, z_i))
  end if
end for

The proposed network has been trained using the LiDAR-based sampling method, and a performance comparison with the random sampling method can be found in Table VIII of the ablation study.

IV. Experimental Results

Dataset. The KITTI 3D object detection dataset [31] is utilized for training and evaluating the proposed network. The dataset basically consists of $7,481$ train sets and $7,518$ test sets. The train set is paired with images and the corresponding annotation files, while the test set does not provide annotation files. To evaluate the validation set, the training set is divided into a train set of $3,712$ and a validation set of $3,769$, following the approach used in previous studies [1]–[5]. Moreover, the KITTI raw data, which consists of multiple sequence images and is commonly used to train depth map-based approaches, is also used. The data can be accessed on the KITTI 3D official website. Inspired by other research works [32], [33], we incorporate the KITTI raw data in the training process and evaluate the trained model using the

TABLE I
QUANTITATIVE RESULTS FOR CAR CLASS ON KITTI TEST SET, EVALUATED BY $AP_{3D|R40}$ AND $AP_{BEV|R40}$ WITH IOU ≥ 0.7. THE BEST AND SECOND BEST RESULTS ARE HIGHLIGHTED IN **BOLD** AND UNDERLINE, RESPECTIVELY.

Methods	Reference	Extra Data	Runtime (ms)	Test, $AP_{3D\|R40}$			Test, $AP_{BEV\|R40}$		
				Easy	Mod	Hard	Easy	Mod	Hard
AM3D [7]	ICCV19	Depth	400	16.50	10.74	9.52	25.03	17.32	14.91
D4LCN [16]	CVPR20	Depth	-	16.65	11.72	9.51	22.51	16.02	12.55
PatchNet [21]	ECCV20	Depth	400	15.68	11.12	10.17	22.97	16.86	14.97
Neighbor-Vote [22]	ACMMM21	Depth	-	15.57	9.90	8.89	27.39	18.65	16.54
Kinem3D [23]	ECCV20	KITTI raw	120	19.07	12.72	9.17	26.69	17.52	13.10
PCT [14]	NeurIPS21	Depth	45	21.00	13.37	11.31	29.65	19.03	15.92
CaDDN [17]	CVPR21	Depth	63	19.17	13.41	11.46	27.94	18.91	17.19
AutoShape [24]	ICCV21	CAD	50	22.47	14.17	11.36	30.66	20.08	15.59
MonoSIM [25]	arXiv22	Depth	120	20.31	13.74	12.31	28.27	19.89	17.96
M3D-RPN [10]	ICCV19	-	160	14.76	9.71	7.42	21.02	13.67	10.23
SMOKE [2]	CVPRW20	-	30	14.03	9.76	7.84	20.84	14.49	12.75
MonoPair [3]	CVPR20	-	57	13.04	9.99	8.65	19.28	14.83	12.89
MonoDLE [4]	CVPR21	-	40	17.23	12.26	10.29	24.79	18.89	16.00
MonoRUn [19]	CVPR21	-	70	19.65	12.30	10.58	27.94	17.34	15.24
GrooMeD [11]	CVPR21	-	120	18.10	12.32	9.65	26.19	18.27	14.05
MonoRCNN [26]	ICCV21	-	70	18.36	12.65	10.03	25.48	18.11	14.10
GUPNet [27]	ICCV21	-	30	20.11	14.20	11.77	-	-	-
MonoFlex [5]	CVPR21	-	30	19.94	13.89	12.07	28.23	19.75	16.89
DEVIANT [28]	ECCV22	-	-	21.88	14.46	11.89	29.65	20.44	17.43
MonoEdge [29]	WACV23	-	-	21.08	14.47	12.73	28.80	20.35	17.57
MonoRCNN++ [30]	WACV23	-	-	20.08	13.72	11.34	-	-	-
Ours(Best)	-	KITTI raw	30	**28.27**	**18.60**	**16.08**	**38.02**	**24.58**	**22.46**

TABLE II
QUANTITATIVE RESULTS FOR CAR CLASS ON KITTI VALIDATION SET, EVALUATED BY $AP_{3D|R40}$ AND $AP_{BEV|R40}$ WITH IOU ≥ 0.7 AND IOU ≥ 0.5, RESPECTIVELY. THE BEST AND SECOND BEST RESULTS ARE HIGHLIGHTED IN **BOLD** AND UNDERLINE, RESPECTIVELY.

Methods	IoU ≥ 0.7						IoU ≥ 0.5					
	Val, $AP_{3D\|R40}$			Val, $AP_{BEV\|R40}$			Val, $AP_{3D\|R40}$			Val, $AP_{BEV\|R40}$		
	Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard
CenterNet [1]	0.60	0.66	0.77	3.46	3.31	3.21	20.00	17.50	15.57	34.46	27.91	24.65
MonoGRNet [9]	11.90	7.46	5.76	19.72	12.81	10.15	47.59	32.28	25.50	48.53	35.94	28.59
M3D-RPN [10]	14.53	11.07	8.65	25.94	21.18	17.90	48.53	35.94	28.59	53.35	39.60	31.76
MonoPair [3]	16.28	12.30	10.42	24.12	18.17	15.76	55.38	42.39	37.99	61.06	47.63	41.92
MonoDLE [4]	17.45	13.66	11.68	24.97	19.33	17.01	55.41	43.42	37.81	60.73	46.87	41.89
Kinem3D [23]	19.76	14.10	10.47	27.83	19.72	15.10	55.44	39.47	31.26	61.79	44.68	34.56
GrooMeD [11]	19.67	14.32	11.27	27.38	19.75	15.92	55.62	41.07	32.89	61.83	44.98	36.29
GUPNet [27]	22.76	16.46	13.72	31.07	22.94	19.75	57.62	42.33	37.59	61.78	47.06	40.88
MonoFlex [5]	23.64	17.51	14.83	31.65	23.29	20.02	60.70	45.65	39.91	66.26	49.30	44.42
DEVIANT [28]	24.63	16.54	14.52	32.60	23.04	19.99	61.00	46.00	40.18	65.28	49.63	43.50
Ours	**25.14**	**18.02**	**15.31**	**33.01**	**23.63**	**20.40**	**62.37**	**46.27**	**41.55**	**66.68**	**51.14**	**45.04**

test set provided on the official website. As the KITTI raw dataset does not offer annotations, we create and use pseudo-annotations using PV-RCNN [34], a LiDAR-based detector.

Evaluation metrics. We evaluated the object detection capability of the proposed network on three classes: 'Car', 'Pedestrian', and 'Cyclist', using two evaluation metrics, namely AP_{3D} and AP_{BEV}. These metrics represent the average precision of the predicted 3D bounding box and the average precision of the results in the Bird's Eye View map, respectively. For the 'Car' category, we set the intersection over union (IoU) threshold to 0.7, whereas for 'Pedestrian' and 'Cyclist', we set it to 0.5. We also evaluated the KITTI 3D dataset according to three levels of difficulty: 'Easy', 'Moderate', and 'Hard', which are defined based on the object's level of occlusion and translation.

Implementation details. The training of the proposed method is performed on an RTX 2080Ti GPU. The model is trained for a total of 100 epochs, and the initial learning rate is set to 3e-4. AdamW optimizer is used during the training process, and the learning rate is decayed at the $80th$ and $90th$ epochs. The input image size is $(384 \times 1280 \times 3)$, and the DLA-34 backbone network produces a feature map of size

$(96 \times 320 \times 256)$ with a down-sampling ratio of 4 ($S = 4$). The deep features from this network are then fed into both the 3D branch and the proposed RefDEB for detection.

TABLE III
QUANTITATIVE RESULTS FOR 'CAR' CLASS ON KITTI VALIDATION SET USING RAW DATA COMPARED TO OTHER SOTA NETWORKS, EVALUATED BY $AP_{3D|R40}$ WITH IOU ≥ 0.5 AND 0.7. THE BEST AND SECOND BEST RESULTS ARE HIGHLIGHTED IN **BOLD** AND UNDERLINE, RESPECTIVELY.

Methods	$AP_{3D\|R40}$, IoU ≥ 0.5			$AP_{3D\|R40}$, IoU ≥ 0.7		
	Easy	Mod	Hard	Easy	Mod	Hard
MonoGRNet	53.84	37.24	29.70	16.30	10.06	7.86
M3D-RPN [10]	62.92	47.14	42.03	26.17	19.61	16.80
RTM3D [10]	65.44	49.40	43.55	25.23	19.43	16.77
GrooMeD-NMS	68.27	50.80	45.14	27.79	20.46	17.75
MonoFlex [10]	69.16	54.27	48.37	31.15	23.42	20.60
ours	**74.16**	**58.00**	**53.65**	**39.58**	**29.28**	**25.97**

TABLE IV
FOUR DIFFERENT RUNS FOR CAR CLASS ON KITTI TEST SET.

Times	Test, $AP_{3D\|R40}$			Test, $AP_{BEV\|R40}$		
	Easy	Mod	Hard	Easy	Mod	Hard
1	24.82	15.17	13.46	34.21	21.46	18.73
2	20.36	12.87	11.14	29.33	18.18	16.71
3	28.65	17.59	15.66	38.51	24.38	21.35
4	28.27	18.60	16.08	38.02	24.58	22.46
Average	25.52	16.06	14.86	35.01	22.15	19.81

TABLE V
REFDEB ON/OFF TEST. THE BEST RESULT IS HIGHLIGHTED IN **BOLD**.

Methods		$AP_{3D\|R40}$, IoU ≥ 0.5			$AP_{3D\|R40}$, IoU ≥ 0.7		
DC	CC	Easy	Mod	Hard	Easy	Mod	Hard
-	-	21.76	16.19	13.63	28.98	22.25	19.38
✓	-	22.88	16.40	13.68	31.11	22.66	19.49
-	✓	23.36	16.55	14.48	29.65	21.38	19.15
✓	✓	**25.14**	**18.02**	**15.31**	**33.01**	**23.63**	**20.40**

TABLE VI
QUANTITATIVE RESULTS FOR 'PEDESTRIAN' AND 'CYCLIST' CLASS ON KITTI VALIDATION SET, EVALUATED BY $AP_{3D|R40}$ AND $AP_{BEV|R40}$ WITH IOU ≥ 0.5. THE BEST AND SECOND BEST RESULTS ARE HIGHLIGHTED IN **BOLD** AND UNDERLINE, RESPECTIVELY.

Methods	Pedestrian, $AP_{3D\|R40}$			Cyclist, $AP_{BEV\|R40}$		
	Easy	Mod	Hard	Easy	Mod	Hard
M3D-RPN [10]	4.75	3.55	2.79	3.10	1.49	1.17
MonoFlex [5] (retrained)	6.80	4.80	4.08	7.90	4.05	3.84
MDS-NET [35]	9.O4	6.27	4.91	3.17	1.80	1.64
MonoDistill [36]	8.95	6.84	5.32	5.38	2.67	2.53
OBMO [37]	**12.80**	**9.55**	**7.40**	7.81	4.06	3.60
MVC-MonoDet [38]	8.04	6.26	6.94	6.94	4.04	3.94
Ours(500)	10.90	8.12	6.60	7.05	3.45	3.29
Ours(1000)	10.86	8.92	7.19	**9.25**	**4.71**	**4.34**
Ours(2500)	10.28	7.65	6.41	5.33	2.69	2.23
Ours(5500)	8.04	6.01	4.99	3.72	1.96	1.69

A. Quantitative Results

Table I presents a performance comparison between various state-of-the-art (SOTA) models on the KITTI test set. The dataset used to train the model and to evaluate the test set is created using the KITTI raw dataset and pseudo-label generation techniques. AP_{3D} and AP_{BEV} are performance evaluation metrics for 3D object detection that increase with the accuracy of depth estimation, and the accuracy of depth estimation is crucial for 3D object detection. The proposed method achieved better results than recent SOTA models, whether trained with additional data or not. To achieve more accurate depth estimation, we proposed RefDEB to estimate depth in a dense manner, whereas it was previously estimated sparsely. As a result, the proposed method yields higher depth estimation accuracy, leading to improve 3D localization accuracy, and thus achieves excellent performance in terms of AP_{3D} and AP_{BEV}. Notably, compared to MonoFlex [5], a baseline network that employs the geometry-based depth estimation method, the proposed method achieves a higher performance of 4.88, 1.28, and 1.39 in 'easy', 'moderate', and 'hard' levels, respectively. Similarly, the proposed method achieves improved performance compared to MonoRCNN [26] and MonoRCNN++ [30], which use object height information.

Table II presents a performance comparison between various SOTA models using the KITTI validation set. For evaluating the KITTI validation set, the proposed method is trained on a dataset comprising $7,481$ samples, including both the train and the validation sets. The table summarizes the official performance results for several SOTA models on the KITTI validation set, as reported in their respective papers. The proposed method achieves higher performance compared to the CenterNet-style methods such as SMOKE [2], MonoPair [3], MonoDLE [4], and MonoFlex [5], including CenterNet [1]. Notably, the proposed method achieves 1.5, 0.51, and 0.48 higher performance than MonoFlex in AP_{3D} with IoU ≥ 0.7 and 1.36, 0.34, and 0.38 higher than MonoFlex in AP_{3D} with IoU ≥ 0.5, and the performance difference is greater in terms of IoU ≥ 0.5.

Table III presents the quantitative evaluation of the 'Car' category on the KITTI validation set using raw data, assessed by $AP_{3D|R40}$ with IoU thresholds of 0.5 and 0.7. The proposed method achieves 5, 3.73, and 5.28 higher performance than MonoFlex in AP_{3D} with IoU ≥ 0.7 and 8.43, 5.86, and 5.37 higher than MonoFlex in AP_{BEV} with IoU ≥ 0.7. The results indicate that our proposed method outperforms other state-of-the-art techniques when trained under identical conditions with raw data.

In Table IV, we calculated the median outcomes in a sequence of four different experiments for 'Car' class on KITTI test set. Although there were variations across the four tests, it is noteworthy that our method consistently showed enhanced performance relative to other state-of-the-art networks.

B. Qualitative Results

We conducted a qualitative evaluation as shown in Fig. 6. We use the KITTI 3D validation set and compare the predicted boxes of the proposed method and the baseline using both 2D and BEV images. The proposed method shows more sophisticated 3D localization in both 2D and BEV images, which can be attributed to the improved depth estimation method.

Fig. 6. The proposed method (right) and the baseline (left) are compared in terms of their predictions on both the image view and BEV, as shown in the qualitative results. The predicted boxes are represented by green, while the ground truth boxes are in red.

TABLE VII
RESULTS OF ABLATION STUDY FOR VARYING NUMBER OF SAMPLED POINTS. THE BEST RESULT IS HIGHLIGHTED IN **BOLD**.

Num of samples	$AP_{3D\|R40}$, IoU ≥ 0.7			$AP_{BEV\|R40}$, IoU ≥ 0.7		
	Easy	Mod	Hard	Easy	Mod	Hard
Baseline [5] (retrained)	21.76	16.19	13.63	28.98	22.25	19.38
500	21.25	15.13	12.96	28.46	21.01	17.80
1000	**25.13**	**17.51**	**14.65**	**33.00**	**23.21**	**19.83**
2500	21.28	15.95	13.33	30.34	22.49	19.38
5500	14.96	11.24	9.73	22.19	16.64	9.74

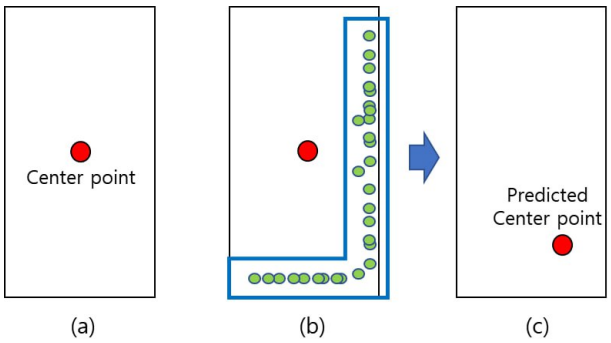

Fig. 7. LiDAR-guided sampling. (a) bounding box with ground truth center point, (b) LiDAR-guided sampled points and (c) the predicted center point trained by using LiDAR-guided sampling.

C. Ablation Study

1) RefDEB On/off test: Table V displays a comparative analysis of the individual contributions of dilated convolution and coordinate convolution in the proposed RefDEB tested on KITTI validation set for 'Car' class. This is demonstrated by toggling these components on and off. The performance of baseline network, MonoFLEX, is represented by the results obtained without utilizing either dilated or coordinate convolution. A marginal improvement in performance is noted when each component is activated separately. However, the optimal performance is attained when both dilated convolution and coordinate convolution are used in conjunction. In section 3.D, as mentioned earlier, dilated convolution is employed to capture a broader range of contextual information surrounding the object. Furthermore, coordinate convolution is utilized to accurately predict the positional coordinates of sampled points for each object and derive the corresponding depth values. Consequently, the combination of dilated convolution and coordinate convolution is employed to acquire accurate positional coordinates and depth values for object-wise sampled points.

2) Pedestrian and Cyclist: Table VI shows the results of an ablation study for the performance of the 'Pedestrian' and 'Cyclist' classes evaluated by other models and the proposed method on the KITTI validation set. First, we retrained the baseline network MonoFlex [5] under the same conditions as the proposed method. The proposed method achieved significantly higher performance for the 'Pedestrian' class, recording 4.06, 4.12, and 3.11 higher performance than the baseline, and 1.35, 0.66, and 0.5 higher performance for the 'Cyclist' class. Additionally, the proposed method outperformed other networks in terms of overall performance for 'Cyclist' class and achieved the second-best performance for the 'Pedestrian' class.

3) The number of points: We conducted an ablation study on the number of sampling points using the KITTI validation set for 'Car' class. In Table VI We observed that performance improved as the number of sampling points increased from 500 to 1000. However, we also observed that

the performance started to decrease at $2,500$ sampling points and decreased significantly at $5,500$. We found that as the number of sampling points increased, sampling various points in a narrow space had an adverse effect on regularization. In addition, for pedestrians, the performance is consistently favorable compared to the baseline network, with similar performance observed at 500 and 2500 sampling points as shown in Table VII. Notably, even at 5500 sampling points, our method outperforms the baseline network. However, for cyclists, the optimal performance is achieved at 1000 sampling points, while the performance diminishes compared to the baseline network at other sampling point counts. Therefore, when considering the car class, it is evident that utilizing 1000 sampling points yields the optimal performance. Although the sampled points may differ in size corresponding to the dimensions (w, h, l) of the object in 3D space, the location difference of sampling points may be closer when projected onto the image. As points that differ in depth value in the actual 3D space are projected onto the image, there is little difference in distance, and we confirmed that performance decreases as the number of sampling points increases.

TABLE VIII
RESULTS OF ABLATION STUDY FOR COMPARING LIDAR-GUIDED AND RANDOM SAMPLING METHODS. THE BEST RESULT IS HIGHLIGHTED IN **BOLD**.

Types of sampling	$AP_{3D\|R40}$, IoU $\geq$0.7			$AP_{BEV\|R40}$, IoU $\geq$ 0.7		
	Easy	Mod	Hard	Easy	Mod	Hard
Baseline [5] (retrained)	21.76	16.19	13.63	28.98	22.25	19.38
LiDAR	22.16	16.13	14.29	32.10	23.39	20.21
Random	**25.14**	**18.02**	**15.31**	**33.01**	**23.63**	**20.40**

4) LiDAR-guided Sampling: Table VIII presents the results of an ablation study comparing the performance of baseline, random sampling, and LiDAR-based sampling methods using the KITTI validation set. LiDAR-based sampling methods are generally more accurate in terms of depth estimation than random sampling methods as LiDAR sensor provides accurate depth information. However, as shown in Fig. 7(b), point cloud data tends to gather in the outer parts of objects, which can result in object-wise dense depth estimation regressing the depth value of the outer part rather than the center point of the object, as shown in Fig. 7(c). As a result, in table VIII, the LiDAR-based sampling method outperforms the baseline but performs worse than the random sampling method. Specifically, for the 'Car' class in the KITTI validation set, the LiDAR-based method shows a 2.98, 1.89, and 1.02 worse performance in AP_{3D} compared to the random sampling method.

5) Three different depth estimation methods: Table IX provides a comparative analysis of three different depth estimation methods using the KITTI 3D validation set. Initially, the direct depth estimation method with uncertainty outperformed the geometry depth estimation method with uncertainty. Nevertheless, a notable boost in performance emerged when both depth estimation methods were employed concurrently. This shows that the combination of diverse depth estimation techniques with uncertainty enhances the overall accuracy of depth estimation. Moreover, when integrated into the ensemble of depth estimation methods, the proposed RefDEB demonstrated the highest level of performance. In summary, it is clear that the object-wise dense depth estimation method provides a more accurate estimation of depth compared to the sparse depth estimation method.

TABLE IX
COMPARISON OF DIFFERENT DEPTH ESTIMATION METHODS. THE BEST RESULT IS HIGHLIGHTED IN **BOLD**.

Types	$AP_{3D\|R40}$, IoU $\geq$ 0.7		
	Easy	Mod	Hard
Direct [5]	15.86	12.60	11.38
Direct + σ	19.63	14.83	13.25
Geometry	15.45	12.18	10.73
Geometry + σ	18.42	14.76	12.49
Direct + Geometry + σ	23.64	17.51	14.83
ours	**25.14**	**18.02**	**15.31**

V. CONCLUSION

We propose a monocular 3D object detection approach based on dense depth estimation using object-wise sampling, which allows for the substitution of the sparse depth estimation method with a more precise dense depth estimation. We use both random sampling and LiDAR-guided sampling methods to estimate object-wise dense depth in the proposed approach. We also propose a ground truth data generation method using these two sampling methods. The random sampling method defines an arbitrary sampling space and obtains ground truth data using sampled points, while the LiDAR-guided sampling method obtains ground truth data by reducing the point cloud map according to the camera frontal view. Major contribution of the proposed approach includes: i) significantly improving the accuracy of monocular 3D object detection by improving the accuracy of depth estimation, ii) addressing some of the key limitations associated with sparse sampling and a single center point, and iii) object-wise sampling and a ground truth data generation method that leverages both random and LiDAR-guided sampling. We conducted comparative experiments using the LiDAR-guided sampling and random sampling methods to analyze the limitations of applying point cloud data to image-based 3D object detection tasks. To compare and experiment with these methods, we performed several experiments and demonstrated the superiority of our proposed approach through various evaluation metrics. Looking ahead, we believe that our approach holds significant potential for improving the performance of monocular 3D object detection systems in a wide range of real-world scenarios.

ACKNOWLEDGMENTS

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (2021-0-01341, Artificial Intelligence Graduate School Program(Chung-Ang University)), and Korea Research Institute for defense Technology planning and advancement through Defense Innovation Vanguard Enterprise Project, funded by Defense Acquisition Program Administration(R230106).

REFERENCES

[1] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," *arXiv preprint arXiv:1904.07850*, 2019.

[2] Z. Liu, Z. Wu, and R. 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.

[3] Y. Chen, L. Tai, K. Sun, and M. 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. 12 093–12 102.

[4] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4721–4730.

[5] Y. Zhang, J. Lu, and J. 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.

[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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8445–8453.

[7] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6851–6860.

[8] X. Weng and K. Kitani, "Monocular 3d object detection with pseudo-lidar point cloud," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[9] Z. Qin, J. Wang, and Y. Lu, "Monogrnet: A geometric reasoning network for monocular 3d object localization," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 8851–8858.

[10] G. Brazil and X. 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.

[11] A. Kumar, G. Brazil, and X. Liu, "Groomed-nms: Grouped mathematically differentiable nms for monocular 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8973–8983.

[12] B. Li, W. Ouyang, L. Sheng, X. Zeng, and X. Wang, "Gs3d: An efficient 3d object detection framework for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1019–1028.

[13] A. Simonelli, S. R. Bulo, L. Porzi, M. López-Antequera, and P. Kontschieder, "Disentangling monocular 3d object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1991–1999.

[14] L. Wang, L. Zhang, Y. Zhu, Z. Zhang, T. He, M. Li, and X. Xue, "Progressive coordinate transforms for monocular 3d object detection," *Advances in Neural Information Processing Systems*, vol. 34, pp. 13 364–13 377, 2021.

[15] X. Wang, W. Yin, T. Kong, Y. Jiang, L. Li, and C. Shen, "Task-aware monocular depth estimation for 3d object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 12 257–12 264.

[16] M. Ding, Y. Huo, H. Yi, Z. Wang, J. Shi, Z. Lu, and P. Luo, "Learning depth-guided convolutions for monocular 3d object detection," in *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition workshops*, 2020, pp. 1000–1001.

[17] C. Reading, A. Harakeh, J. Chae, and S. L. Waslander, "Categorical depth distribution network for monocular 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8555–8564.

[18] J. Ku, A. D. Pon, and S. 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. 11 867–11 876.

[19] H. Chen, Y. Huang, W. Tian, Z. Gao, and L. Xiong, "Monorun: Monocular 3d object detection by reconstruction and uncertainty propagation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 379–10 388.

[20] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2403–2412.

[21] X. Ma, S. Liu, Z. Xia, H. Zhang, X. Zeng, and W. Ouyang, "Rethinking pseudo-lidar representation," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*. Springer, 2020, pp. 311–327.

[22] X. Chu, J. Deng, Y. Li, Z. Yuan, Y. Zhang, J. Ji, and Y. Zhang, "Neighbor-vote: Improving monocular 3d object detection through neighbor distance voting," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 5239–5247.

[23] G. Brazil, G. Pons-Moll, X. Liu, and B. Schiele, "Kinematic 3d object detection in monocular video," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16*. Springer, 2020, pp. 135–152.

[24] Z. Liu, D. Zhou, F. Lu, J. Fang, and L. Zhang, "Autoshape: Real-time shape-aware monocular 3d object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 641–15 650.

[25] H. Sun, Z. Fan, Z. Song, Z. Wang, K. Wu, and J. Lu, "Monosim: Simulating learning behaviors of heterogeneous point cloud object detectors for monocular 3d object detection," 2022. [Online]. Available: https://arxiv.org/abs/2208.09446

[26] X. Shi, Q. Ye, X. Chen, C. Chen, Z. Chen, and T.-K. Kim, "Geometry-based distance decomposition for monocular 3d object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 172–15 181.

[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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3111–3121.

[28] A. Kumar, G. Brazil, E. Corona, A. Parchami, and X. Liu, "Deviant: Depth equivariant network for monocular 3d object detection," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IX*. Springer, 2022, pp. 664–683.

[29] M. Zhu, L. Ge, P. Wang, and H. Peng, "Monoedge: Monocular 3d object detection using local perspectives," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 643–652.

[30] X. Shi, Z. Chen, and T.-K. Kim, "Multivariate probabilistic monocular 3d object detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 4281–4290.

[31] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset. the international journal of robotics research," *Int J Rob Res*, pp. 1–6, 2013.

[32] L. Peng, F. Liu, Z. Yu, S. Yan, D. Deng, Z. Yang, H. Liu, and D. Cai, "Lidar point cloud guided monocular 3d object detection," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part I*. Springer, 2022, pp. 123–139.

[33] Y. Hong, H. Dai, and Y. Ding, "Cross-modality knowledge distillation network for monocular 3d object detection," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part X*. Springer, 2022, pp. 87–104.

[34] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, "Pv-rcnn: Point-voxel feature set abstraction for 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 529–10 538.

[35] Z. Xie, Y. Song, J. Wu, Z. Li, C. Song, and Z. Xu, "Mds-net: A multi-scale depth stratification based monocular 3d object detection algorithm," *arXiv preprint arXiv:2201.04341*, 2022.

[36] Z. Chong, X. Ma, H. Zhang, Y. Yue, H. Li, Z. Wang, and W. Ouyang, "Monodistill: Learning spatial features for monocular 3d object detection," *arXiv preprint arXiv:2201.10830*, 2022.

[37] C. Huang, T. He, H. Ren, W. Wang, B. Lin, and D. Cai, "Obmo: One bounding box multiple objects for monocular 3d object detection," *arXiv preprint arXiv:2212.10049*, 2022.

[38] Q. Lian, Y. Xu, W. Yao, Y. Chen, and T. Zhang, "Semi-supervised monocular 3d object detection by multi-view consistency," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VIII*. Springer, 2022, pp. 715–731.

Chanyeong Park was born in Seoul, South Korea, in 1997. He received a B.S. degree in Computer Science from Coventry University, in 2021. He received an M.S. degree in Image Science - AI Imageing from Chung-Ang University, Korea, in 2023. Currently, he is pursuing a Ph.D. degree in Image Science - AI Imaging at Chung-Ang University. His research interests include Computer Vision, Domain Generalization and Monocular 3D Object Detection.

Heegwang Kim was born in Seoul, Korea, in 1992. He received a B.S. degree in electronic engineering from Soongsil University, Korea, in 2016. He received an M.S. degree in Image Science from Chung-Ang University, Korea, in 2018. Currently, he is pursuing a Ph.D. degree in image engineering at Chung-Ang University. His research interests include object detection and knowledge distillation.

Junbo Jang was born in Seoul, South Korea, in 1998. He received a B.S. degree in Industrial management engineering from Hankuk University of Foreign Studies, South Korea, in 2023. He is currently pursuing the M.S degree with the Department of Artificial Intelligence, Chung-Ang. His research interests include Tiny object detection, light-weight and EO/IR Fusion object detection.

Joonki Paik was born in Seoul, South Korea, in 1960. He received a B.S. degree in control and instrumentation engineering from Seoul National University in 1984 and M.Sc. and Ph.D. degrees in electrical engineering and computer science from Northwestern University in 1987 and 1990, respectively. From 1990 to 1993, he joined Samsung Electronics, where he designed image stabilization chipsets for consumer camcorders. Since 1993, he has been a member of the faculty of Chung-Ang University, Seoul, Korea, where he is currently a professor with the Graduate School of Advanced Imaging Science, Multimedia, and Film. From 1999 to 2002, he was a visiting professor with the Department of Electrical and Computer Engineering, University of Tennessee, Knoxville. Since 2005, he has been the director of the National Research Laboratory in the field of image processing and intelligent systems. From 2005 to 2007, he served as the dean of the Graduate School of Advanced Imaging Science, Multimedia, and Film. From 2005 to 2007, he was the director of the Seoul Future Contents Convergence Cluster established by the Seoul Research and Business Development Program. In 2008, he was a full-time technical consultant for the System LSI Division of Samsung Electronics, where he developed various computational photographic techniques, including an extended depth of field system. He has served as a member of the Presidential Advisory Board for Scientific/Technical Policy with the Korean Government and is currently serving as a technical consultant for the Korean Supreme Prosecutor's Office for computational forensics. He was a two-time recipient of the Chester-Sall Award from the IEEE Consumer Electronics Society, the Academic Award from the Institute of Electronic Engineers of Korea, and the Best Research Professor Award from Chung-Ang University. He has served the Consumer Electronics Society of the IEEE as a member of the editorial board, vice president of international affairs, and director of sister and related societies committee.