IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 3 January 2024, accepted 13 January 2024, date of publication 17 January 2024, date of current version 30 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3355137

APPLIED RESEARCH

Automatic Pseudo-LiDAR Annotation: Generation of Training Data for 3D Object Detection Networks

CHANGSUK OH[1], YOUNGSEOK JANG[2], (Graduate Student Member, IEEE), DONGSEOK SHIM[3], (Graduate Student Member, IEEE), CHANGHYEON KIM[4], JUNHA KIM[2], AND H. JIN KIM[2], (Member, IEEE)

[1]Department of Aerospace Engineering, Seoul National University, Gwanak-gu, Seoul 08826, South Korea
[2]Department of Mechanical and Aerospace Engineering, Artificial Intelligence Institute, Seoul National University, Gwanak-gu, Seoul 08826, South Korea
[3]Interdisciplinary Program in Artificial Intelligence, Seoul National University, Gwanak-gu, Seoul 08826, South Korea
[4]Samsung Research, Seocho-gu, Seoul 06765, South Korea

Corresponding author: H. Jin Kim (hjinkim@snu.ac.kr)

This work was supported by the AI Institute at Seoul National University.

ABSTRACT Object detection in 3D is a key ingredient of various autonomous systems. Many 3D object detection methods rely on LiDAR, as it is robust to illumination conditions and provides accurate distance measurements. To apply LiDAR-based 3D object detection networks for new objects, we need new training datasets. However, because labeling target objects with 3D bounding boxes in LiDAR point clouds requires significant resources and open datasets contain annotations of only car-related classes, it is challenging to deploy LiDAR-based 3D object detectors for detecting objects not related to cars. We propose a system that automatically generates annotated pseudo-LiDAR (APL) data, which requires only stereo images to synthesize 3D bounding box annotations and pseudo-LiDAR points. Using the proposed method, we can dramatically reduce efforts and time for generating a LiDAR-based 3D object detection dataset. By utilizing classes in 2D image datasets, the proposed framework can annotate diverse objects beyond limited classes of existing LiDAR-based 3D object detection datasets. To verify the capability of the synthesized training data, we train 3D object detection networks with the APL data of new classes. The experiments show that the 3D object detection networks trained on the APL data can detect objects of the new classes in LiDAR point clouds, which demonstrates that the proposed method can help LiDAR-based 3D object detectors operate for various objects not covered in existing LiDAR-based 3D object detection datasets.

INDEX TERMS 3D object detection, light detection and ranging (LiDAR), pseudo-LiDAR point cloud.

I. INTRODUCTION

Object detection in 3D, which involves perceiving surrounding objects or obstacles and estimating their position, is an essential component for autonomous navigation [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16]. For robust and accurate 3D object detection, many methods [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [17], [18] utilize LiDAR due to its robustness to illumination conditions and accurate distance measurements.

We need annotated 3D data to train 3D object detection networks in a supervised manner. Typically, training datasets [19], [20], [21], [22], [23] for LiDAR-based 3D object detector are generated by obtaining raw LiDAR point clouds. Then, human annotators label the positions of the target objects using 3D bounding boxes. As the current data generation method demands laborious manual labeling and LiDAR point cloud obtaining processes, the public 3D object detection datasets [19], [20], [21], [22], [23] have annotations of very limited classes related to automobiles. Therefore, most of the LiDAR-based 3D object detectors [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13] are targeted to detect car-related objects. If an automatic training data generation

The associate editor coordinating the review of this manuscript and approving it for publication was Senthil Kumar.

method exists, we can easily deploy off-the-shelf LiDAR-based 3D object detection networks in various environments.

In order to automatically generate 3D object detector training data, [24] and [25], [26], [27] use deep learning methods and virtual environments, respectively. However, the data generation methods using deep learning models still have limitations on the class diversity, as they relies on a pretrained network trained on existing 3D object detection datasets. In the cases of using virtual environments, we can obtain 3D bounding box labels of various classes. However, creating a huge set of virtual environments which follows scene grammars of real environments still requires significant resources.

In this paper, we propose a data generation method which can *automatically* generate 3D bounding box annotations of various classes without considering scene grammars during data generation processes. We design a two-stage framework which requires only stereo images to automatically generate the training data for LiDAR-based 3D object detector. The proposed method can annotate various classes, as it utilizes pretrained networks trained on readily available 2D image datasets which cover a much broader range of labeled classes compared to limited classes of the existing LiDAR-based 3D object detection datasets. The proposed method does not require a module for mimicking real environments, as it utilizes images captured in real environments to generate point clouds.

In the first stage, the proposed framework yields two types of region proposals: a 3D bounding box and 3D semantic segmentation. Two 2D bounding boxes of stereo images are utilized to generate a 3D bounding box initialization, and a 2D semantic segmentation and a disparity map are employed to render 3D semantic segmentation. In the second stage, the framework identifies whether region proposals can generate reliable data. In the experiment, we synthesize annotated pseudo-LiDAR (APL) data of new classes which are not annotated in the existing LiDAR-based 3D object detection datasets and train 3D object detection networks to detect the target objects. The experimental results show the capability of the proposed method to generate datasets that can be used to train models for detecting classes not labeled in existing LiDAR-based 3D object detection datasets.

The proposed framework has the following contributions:

- The proposed framework automatically generates 3D bounding boxes from commonly available stereo images without human intervention saving resources for creating a new dataset.
- The proposed method can create 3D bounding boxes for diverse classes if recognized by a 2D semantic segmentation network and a 2D bounding box network.
- The proposed cross-verification method improves the data efficiency by removing unreliable data for training 3D object detection networks from two region proposals (3D bounding box and 3D semantic segmentation).
- Training with additional data generated by the proposed framework can improve the performances of LiDAR-based 3D object detection networks.

II. RELATED WORKS

In this section, we introduce approaches aimed at simplifying the generation of training data for LiDAR-based 3D object detectors.

A. UTILIZATION OF DEEP NEURAL NETWORKS

References [24], [28], [29], [30], [31], and [32] employ deep learning networks to reduce laborious tasks for generating training data for LiDAR-based 3D object detectors. Reference [28] reduces a search space for the annotation by removing points outside of the frustums obtained by a pretrained 2D object detector. Reference [29] simplifies the annotation process to a task of finding one point from a target object. When a human annotator selects a point of a target object, a pre-trained per-instance segmentation network [30] segments the target object, and a box regressor estimates 3D bounding boxes on the base of the segmentation results. Reference [24] proposes automatic training data generation method without human intervention. It first generates cylindrical object proposals and refines the cylindrical object proposals to obtain 3D bounding box annotations. References [31] and [32] generate an annotation by refining 3D bounding box estimations of an object obtained from consecutive frames.

While the aforementioned methods simplify the process of 3D object detector training data generation methods, they require more resources to create the training data of new classes when compared to the proposed method. References [28] and [29] still need one or more human tasks to generate an annotation, while the proposed method generates the training data with no manual process. References [24] and [32] cannot automatically synthesize annotations of new classes, as it requires data with 3D bounding box annotations for training the auto-labeling network. However, the proposed method can generate training data for a much broader range of classes compared to those using LiDAR-based 3D object detectors, as our method leverages networks trained on 2D image datasets with diverse labeled classes. References [31] and [32] demand a high-performing odometry system as they necessitate highly accurate estimation of ego vehicle motion for labeling. The proposed method requires only a stereo camera for data generation.

B. UTILIZATION OF CAD MODEL

References [25], [26], and [27] generate 3D annotation using CAD models via graphic engines. The Dhaiba human model [26] and YCB model [27] are placed in virtual environments, and annotated data are acquired by virtual sensors. As human annotations are not required and the data acquisition process is simple, they can easily render the training data. However, the works do not contemplate the co-occurrence relationship between a target object and its environment. Therefore, it is difficult to say that the synthetic data accurately contains the context or a probabilistic scene grammar of a target object and its environment. This implies that domain

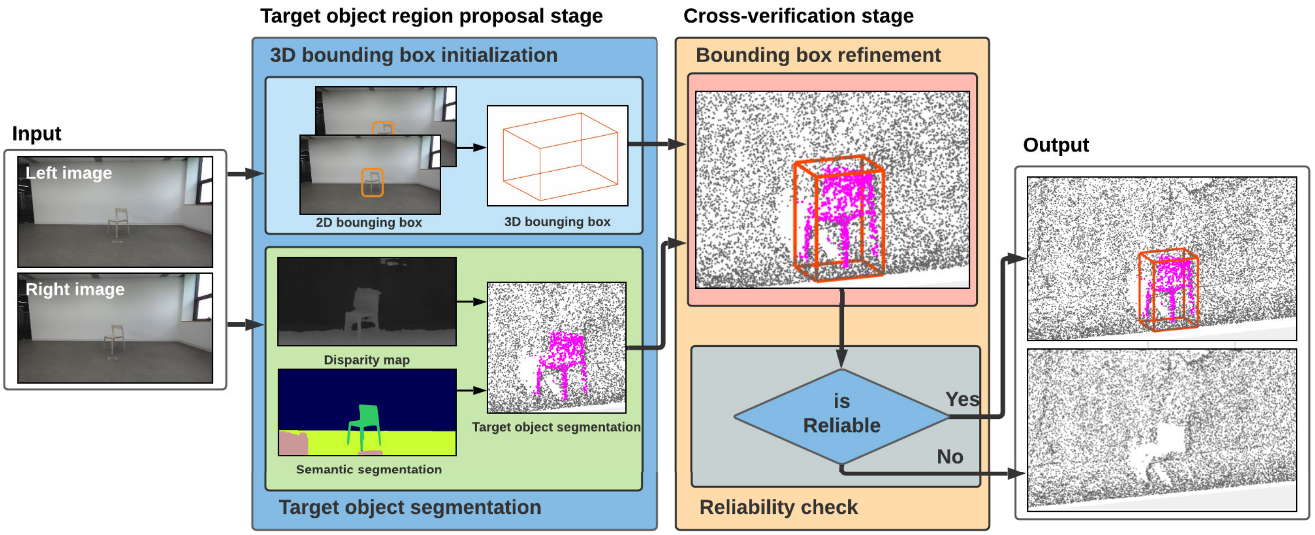


FIGURE 1. Flowchart of the proposed framework. The framework uses stereo images as input. Then, incorporating image-based networks to generate region proposals of target objects in the target object region proposal stage (Section III-A). In the next stage, the bounding box refinement module updates the region proposals to reduce the geometric error between two different types of region proposals (Section III-B1). Finally, the reliability check module determines whether the refined region proposals contain enough information to train 3D object detection networks (Section III-B2).

gaps between LiDAR point clouds obtained in real-world environments and those obtained in virtual environments may exists.

The proposed method utilizes real images containing target class objects for data generation. We underscore that these images adhere to scene grammar of target objects and their environments. Thus, there is no need for additional consideration of scene context during data generation. References [33], [34], and [35] also use real images for data generation. References [33] and [34] precisely estimate the position of target objects using CAD models and employ the estimated position as a label. However, as a CAD model typically represents only one shape of an object, these methods can only be applied when an object has a single pose and cannot generate annotations of complex objects with various poses, such as excavators or wheel loaders. Contrarily, the proposed method can also generate training data for objects that can have various postures by utilizing images containing objects in diverse poses. Reference [35] trains statistics about object co-occurrences and utilizes the statistics for generating training data for 3D object detection network in indoor environments. However, as [35] requires annotated 3D data for training, its applicability to various environments may be limited.

III. APPROACH

The main objective of the proposed framework is to automatically generate a dataset with 3D bounding boxes annotation using stereo images. The flow of the proposed framework is depicted in Fig. 1. In the first stage, two types of region proposals for a target object are generated by incorporating three types of existing image-based networks: 1) disparity estimation, 2) 2D semantic segmentation, and 3) 2D object detection networks. Then, in the cross-verification stage, the bounding box refinement module updates positions and heading angles of initial bounding boxes to reduce the estimation gap between region proposals. The reliability check module determines whether the updated region proposals have sufficient and accurate data to train 3D object detection networks.

A. TARGET OBJECT REGION PROPOSAL STAGE

In this stage, the proposed framework makes two different types of region proposals: a 3D bounding box and target object segmented points.

1) 3D BOUNDING BOX INITIALIZATION MODULE

To generate initial guesses of 3D bounding boxes using stereo images, the 3D bounding box initialization module utilizes 2D bounding boxes from stereo images. Using high-performance off-the-shelf 2D object detection networks, we can easily obtain accurate 2D bounding boxes of various classes. The 3D bounding box initialization module utilizes the Structural SIMilarity index (SSIM) [36] to associate 2D bounding boxes of a target instance from left and right images. Then, we estimate the three-dimensional position of an initial 3D bounding box using the disparity value between the centers of the associated 2D bounding boxes. As objects of the same class usually have similar dimensions, we utilize the pre-set dimension prior for each class. And the z-direction of the camera coordinate is used as the heading angle of an initial bounding box. For the classes directly related to autonomous vehicle environments, the image-based 3D object detection networks [37], [38] can be exploited as a 3D bounding box initialization module.

2) TARGET OBJECT SEGMENTATION MODULE

The target object segmentation module produces a point cloud whose points from target objects are segmented. The

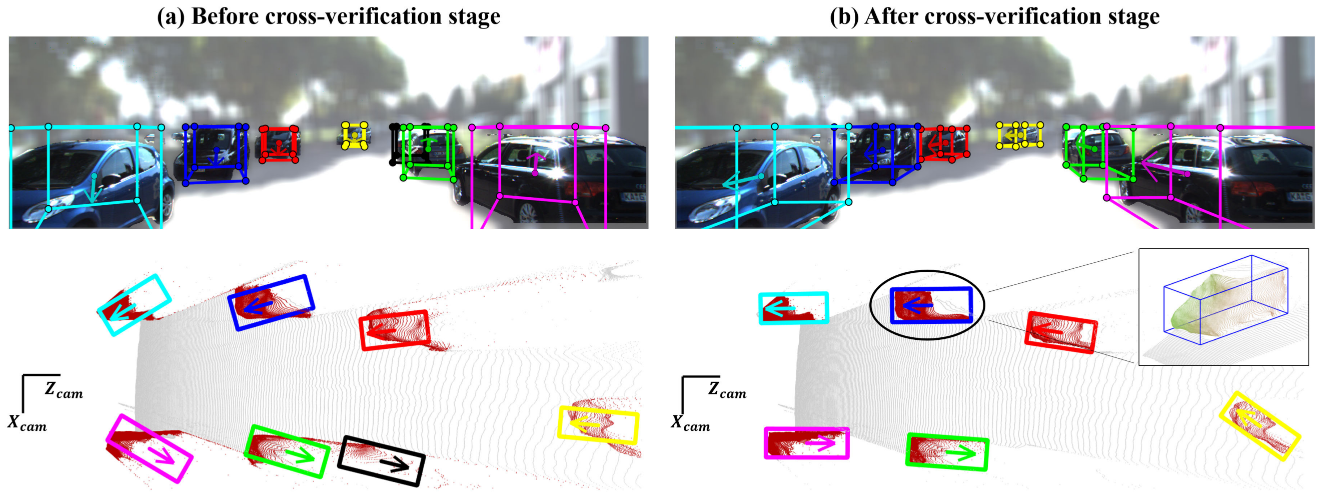

(a) Before cross-verification stage
(b) After cross-verification stage
Z_{cam}
X_{cam}
Z_{cam}
X_{cam}

FIGURE 2. The results of the cross-verification stage. The top figures show the 2D and projected 3D bounding boxes before and after the cross-verification stage onto an image. The bottom figures represent point clouds and 3D bounding boxes with heading angles in the x-z plane of camera coordinates. The black bounding box in (a) is removed in (b) because it is considered an unreliable label through the reliability check module. The configurations of the other labels' 3D bounding boxes are refined through the bounding box refinement module.

3D position of a pixel in the image plane can be inferred when a disparity map and extrinsic parameters of a stereo camera system are known. The pixels from target objects can be back-projected into 3D points using a disparity map. We can perform target object segmentation for various classes by utilizing 2D semantic or instance segmentation datasets.

B. CROSS-VERIFICATION STAGE

The proposed framework distinguish the reliable 3D bounding box annotations and the segmented 3D points among region proposals in the cross-verification stage. Because the framework does not have ground-truth that can be used to supervise two region proposals, two region proposals become each other's supervisor.

1) BOUNDING BOX REFINEMENT MODULE

Note that the initial 3D bounding box and segmented points from a target object are not perfectly matched as illustrated in Fig. 2(a). Through a conversion process from the image pixel domain $[u, v]^{\top} \in \mathbb{R}^2$ to the continuous 3D space $[x, y, z]^{\top} \in \mathbb{R}^3$, a small error in a disparity map can be inverse-proportionally propagated into a large 3D error:

$$x = \frac{(u - c_{u,l}) \times b}{D(u, v)} \tag{1}$$

$$y = \frac{f_{u,l} \times (v - c_{v,l}) \times b}{f_{v,l} \times D(u, v)} \tag{2}$$

$$z = \frac{f_{u,l} \times b}{D(u, v)}\ , \tag{3}$$

where $f_{u,l}$ and $f_{v,l}$ are the horizontal and vertical focal lengths of the left camera, and $(c_{u,l}, c_{v,l})$ is the left camera's principal point. b and $D(u, v)$ are the baseline between stereo cameras and the disparity of a left camera's pixel coordinates (u, v), respectively. The 3D bounding box initialization module cannot make bounding boxes that always accurately wrap target objects. To suppress uncertainties on region proposals, we propose a refinement step for pruning out outlier points and fitting 3D bounding box to inliers.

The refinement module searches the optimal position and heading angle for a 3D bounding box to maximize the number of segmented points within the box. We limit the search space of optimization parameters to a small region because when the optimal position is not found near the initial position, it implies that at least one region proposal has incorrect geometric information of a target object. Therefore, we find the optimal parameters within the limited search space. We discretize the continuous search space to solve the following optimization problem:

$$\begin{aligned} k^* &= \underset{k \in \{1,\ldots,M\}}{\operatorname{argmax}}\ card\left(\mathcal{B}_{3D}^{i,k} \cap \mathcal{S}_{2D}^{i}\right), \\ \text{where}\ M &= N_x \times N_y \times N_z \times N_\theta\ . \end{aligned} \tag{4}$$

Here, N_x, N_y, N_z, and N_θ are the numbers of bins discretizing continuous domains of the associated optimization parameters, and k is the index of an optimization variable set.

$\mathcal{B}_{3D}^{i,k}$ is the set of points within the i-th 3D bounding box determined by the k-th optimization variables, and $\mathcal{S}_{2D}^{i,k}$ is segmented points inside of the i-th 2D bounding box determined by the k-th optimization variables. $card(\cdot)$ is the cardinality of the specific set.

The refinement module solves the optimization problem with the exhaustive search method and updates the bounding boxes to the optimal positions and heading angles.

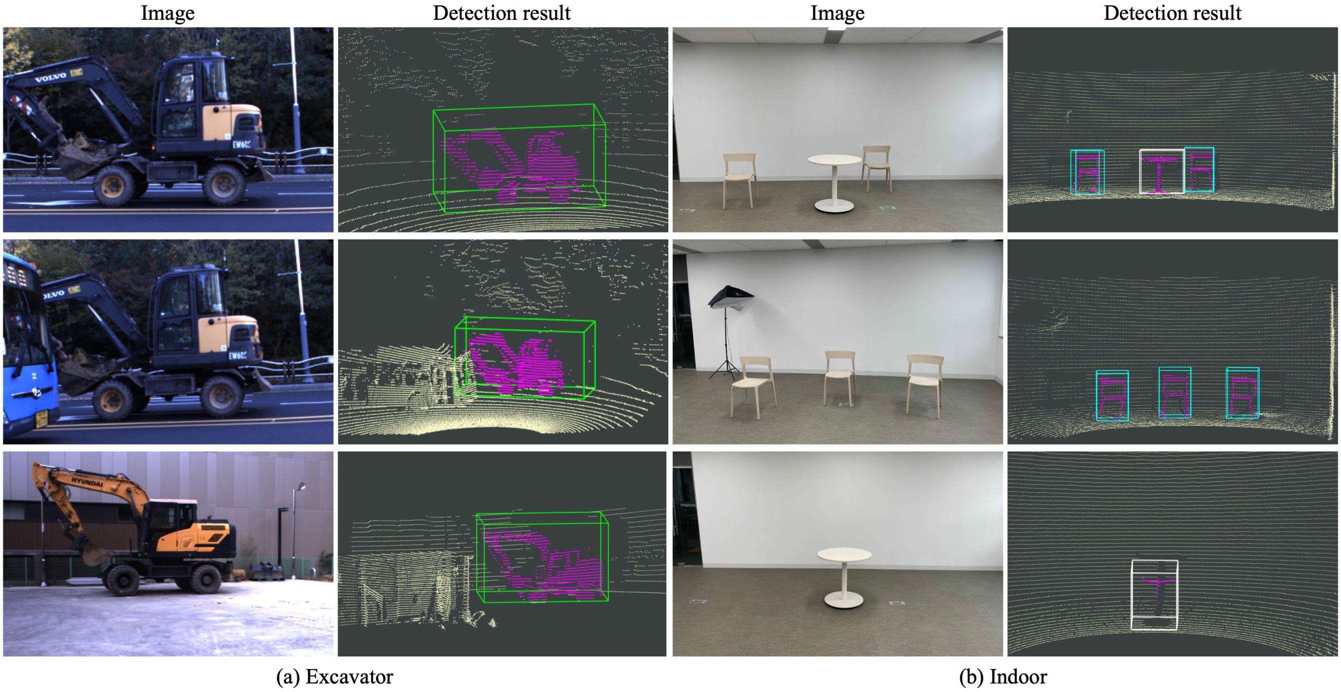


FIGURE 3. Qualitative results on the excavator and indoor datasets. We visualize point clouds and detected bounding boxes from the task network [4] trained on the APL data. Green, white, and cyan bounding boxes indicate excavator, table, and chair classes, respectively. Images are not used for inference and are included for visual purposes.

2) RELIABILITY CHECK MODULE

After the refinement of bounding boxes, the reliability check module counts the number of points in the updated boxes to identify whether the refined region proposals have sufficient and accurate information to train 3D object detection networks. When the number of segmented points in the refined bounding box exceeds a threshold N_{th}, the framework determines that the region proposals contain enough information. Then, the reliability check module approves the two results as reliable data; it enrolls the refined bounding box in the annotation list and the segmented points inside of the bounding box to a point cloud.

For the refined bounding boxes containing fewer segmented points than a threshold, the framework determines that the region proposals are defective to train 3D object detection networks; at least one proposal may include inaccurate geometric information of a target object, or the result may contain too little information to train a network. When region proposals from a target object are regarded as unreliable, the reliability check module does not update the annotation list and does not add segmented points to a point cloud. Therefore, a target object existing in an image may not exist in a point cloud when the region proposals from the target object are filtered out in the cross-verification stage, as shown in Fig. 2(b).

IV. EXPERIMENTS

We conduct two experiments to evaluate whether the proposed framework can generate 1) data of a new class and 2) reliable data which can train 3D object detection networks. In the first experiment, we generate 3D point clouds and bounding boxes of new classes utilizing the proposed method. We take stereo images of the two indoor and one outdoor classes, and train 3D object detection networks using the synthesized data to verify whether the trained networks can find target objects in LiDAR point clouds. As there is no open LiDAR-dataset containing the new classes, we test the performance of the trained detectors with the self-obtained LiDAR point clouds. In the second experiment, we train a 3D object detection network with the APL data created using the KITTI stereo images [21] to compare performance differences between the network trained on a real-LiDAR dataset and the network trained on APL data. Unlike previous pseudo-LiDAR-based 3D object detection networks [39], [40], we do not use 3D annotations of the KITTI dataset, but only stereo images are exploited to train 3D object detection networks.

A. DATA GENERATION UTILIZING SELF-ACQUIRED IMAGES

We synthesize training data of the indoor classes (chair and table) and one outdoor class (excavator) using the proposed method, and we use the synthesized data to verify whether the trained 3D object detection network can detect target objects.

1) TRAINING DATA GENERATION

We acquired 8,708 sequences of stereo images using ZED-2 in indoor environments. We utilize MSeg [41] trained on

TABLE 1. Performance of 3D object detection networks trained on the new dataset. Average precision (in %) is used as performance metrics.

Task network	Class	Average precision
PointRCNN [44]	Chair	83.82
	Table	74.48
	Excavator	94.44
SECOND [4]	Chair	88.13
	Table	65.48
	Excavator	86.57

the MSeg dataset for semantic segmentation, GA-Net [42] trained on the KITTI dataset for disparity estimation, and derive 2D bounding boxes from semantic segmentation images. The threshold N_{th} is set to 1,000. The proposed framework synthesizes 6,296 chair labels and 2,000 table labels using the images.

For the excavator class, we take 1,020 and 375 sequences of stereo images of an excavator in a construction site and public road, respectively. We take stereo images using two mvBlueCOUGAR-X-104iC cameras. We employ GA-Net and VideoProp-LabelRelax [43]. We finally generate 1289 excavator labels. We subsample points of a pseudo-LiDAR point cloud to match the number of points of a pseudo-LiDAR point cloud with the number of points of a real-LiDAR point cloud. The number of pseudo-LiDAR points is more than 273,000 while the number of real-LiDAR points inside of the stereo camera's field of view in the KITTI dataset is around 25,000. We randomly select 20,000 pseudo-LiDAR points to make training data.

2) TEST DATA GENERATION

We utilize Ouster OS1 GEN2 64-channel LiDAR for test data acquisition. We obtain 272 frames with a chair, 124 frames with a table and a chair, and 272 frames with a table and chairs. For the excavator experiment, we take 21 and 15 sequences of LiDAR point clouds including an excavator in the construction site and public road, respectively.

3) TRAINING

We train PointRCNN [44] and SECOND [4] on the APL data. For PointRCNN, we use [45] for training. We set the anchor generator size of the chair class to 0.5 m, 0.6 m, 0.9 m, the table class to 0.7 m, 0.7 m, 0.9 m, and the excavator class to 4.4 m, 7.5 m, 4.4 m for width, length, and height, respectively.

4) RESULT

The quantitative test results are summarized in Table 1, and the qualitative results are depicted in Fig. 3. For the indoor objects, PointRCNN records more than 74% average precision, and SECOND marks more than 65% average precision. For the excavator class, both detectors mark more than 86% average precision. The qualitative results show that all points considered to have been obtained from the target objects are inside the corresponding bounding boxes. Human annotators generate ground-truth bounding boxes of self-acquired test set, and we judge that detection is successful when the intersection over union is over 0.5. The high average precision rates and qualitative results indicate that the proposed framework synthesizes reliable training data.

Through the experiment, we confirm that we can train a new class to a 3D object detection network only with stereo images, while existing pseudo-LiDAR-based 3D object detection networks need stereo images and 3D labels for training. As far as we know, there is no open LiDAR-based 3D object detection dataset acquired in indoor environments. Also, there is no 3D annotation for an excavator among the datasets. Therefore, it is not possible to train a LiDAR-based 3D object detection network that can detect indoor objects or an excavator using existing 3D detectors and open datasets.

B. DATA GENERATION UTILIZING OPEN DATASET IMAGES

We analyze the performance differences between the 3D object detection network trained on real-LiDAR point clouds with human-made labels and the network trained on APL data. We also verify if there is a performance improvement when using the proposed method as a data augmentation method.

1) DATA GENERATION

We create 3D points and 3D bounding boxes of the car class with KITTI stereo images. We divide training data into the train (3,712 frames) and validation (3,769 frames) sets according to [46]. We employ GA-Net [42] for disparity estimation and VideoProp-LabelRelax [43] for 2D segmentation, and StereoRCNN [37] is utilized in the 3D bounding box initialization module. The image-based subnetworks are trained on the KITTI dataset separately. We randomly select 20,000 pseudo-LiDAR points to make training data.

When an updated bounding box contains more segmented points than the threshold, the proposed framework uses the bounding box and segmented points as the APL data. The threshold N_{th} is set to 1,000. The framework yields 6,614 annotations in the training set. Considering that 14,359 annotations exist in the training set, 46% of the target objects are recreated in the APL data.

2) TRAINING

We train PointRCNN [44] on the APL data generated from the KITTI stereo images. The open-code settings are used without alteration, and the RPN stage and the RCNN stage are trained for 200 and 70 epochs, respectively. Next, SECOND [4] is trained on the KITTI dataset and additional data (APL). We follow open-code settings, and the network is trained for 50 epochs.

3) RESULTS

Detection results of the 3D object detection networks trained on the KITTI dataset and APL data are summarized in Table 2. The networks trained on APL data cannot outperform those trained on the real-LiDAR point clouds and human-made labels. However, for IoU threshold 0.5, the task

TABLE 2. Performance comparison of 3D object detection networks trained on the KITTI dataset and APL data, respectively. We report AP_{3D} / AP_{BEV} and how much degradations (in %) have occurred (D_{3D} / D_{BEV}). The evaluation metric is average precision (in %) for 3D bounding boxes (AP_{3D}) and BEV boxes (AP_{BEV}) with IoU over 0.7 and 0.5 for the car class. The largest and smallest degradation results are underlined with bold letters.

Task network	Training data	IoU = 0.5			IoU = 0.7		
		Easy	Moderate	Hard	Easy	Moderate	Hard
SECOND	KITTI	90.41 / 90.42	89.33 / 89.42	88.73 / 88.90	87.43 / 89.96	76.48 / 87.07	69.10 / 79.66
	APL	85.85 / 86.40	68.73 / 73.47	67.26 / 68.52	61.55 / 76.50	45.11 / 58.91	40.34 / 57.17
	(degradation)	(5.04 / 4.44)	(23.06 / 17.84)	(24.20 / 22.92)	(29.60 / 14.96)	(41.02 / 32.34)	(**42.62** / 28.23)
PointRCNN	KITTI	96.12 / 96.21	89.74 / 89.78	89.30 / 89.39	88.88 / 90.19	78.63 / 87.80	77.38 / 85.29
	APL	93.24 / 93.32	78.14 / 78.54	73.73 / 74.20	83.46 / 89.32	66.01 / 74.02	60.69 / 69.73
	(degradation)	(3.00 / 3.00)	(12.93 / 12.52)	(17.44 / 16.99)	(6.10 / **0.95**)	(12.62 / 13.78)	(16.69 / 15.56)

TABLE 3. Performance of 3D object detection networks trained on the KITTI and augmented data, respectively. The evaluation metric is average precision (in %) for 3D bounding boxes (AP_{3D}) and BEV boxes (AP_{BEV}) with IoU over 0.7 for the car class. KITTI* stands for the augmented data consisting of the KITTI dataset and APL data. We obtain the test set results by submitting inference results to the official test server.

Task network	Evaluation set	Training data	AP_{3D}			AP_{BEV}		
			Easy	Moderate	Hard	Easy	Moderate	Hard
SECOND	val	KITTI	87.43	76.48	69.10	89.96	87.07	79.66
		KITTI*	**88.79**	**78.45**	**77.12**	**90.33**	**87.95**	**87.35**
	test	KITTI	83.13	73.66	66.20	88.07	79.37	77.95
		KITTI*	**84.26**	**75.75**	**70.65**	**91.45**	**86.16**	**81.08**
PointRCNN	val	KITTI	78.96	60.98	52.09	81.05	62.84	53.87
		KITTI*	**81.14**	**63.03**	**53.99**	**90.28**	**72.19**	**62.60**

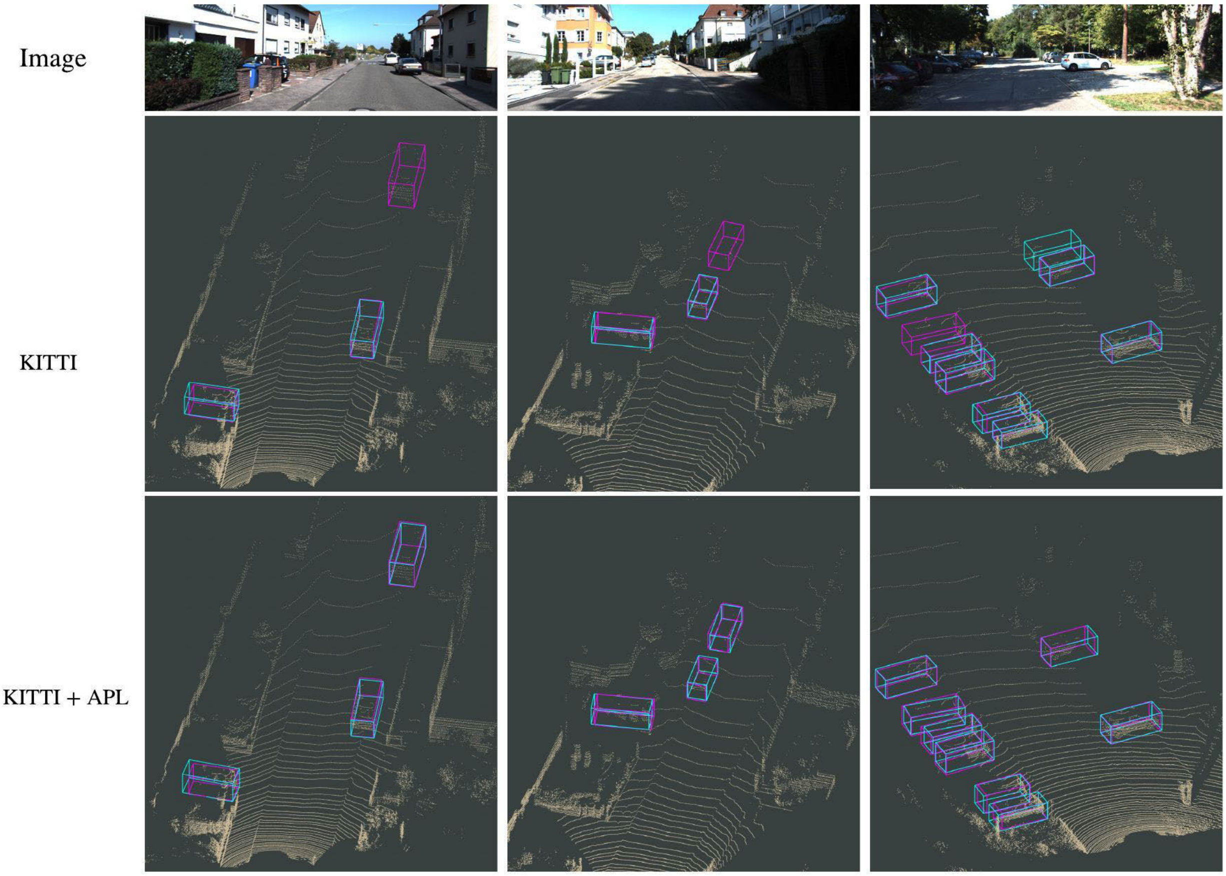


FIGURE 4. Qualitative results on KITTI validation set. We compare inferences from the task network [4] trained on KITTI and the task network using APL as additional training data. Bounding boxes in magenta are ground-truth and bounding boxes in cyan are predictions. Images are not used for inference and are included for visual purposes.

networks trained on the APL data can perform about 99 % of the networks trained on the KITTI dataset on low-difficulty instances. The results show that synthesized point clouds and automatically generated labels from stereo images can train

TABLE 4. Performance of the 3D object detection networks [4] trained on the inferior APL data. AP_{3D} represents average precision (in %) for 3D bounding boxes of easy, moderate, and hard instances of the KITTI car class. The tests are conducted on the KITTI validation set.

Training data	AP_{3D} (IoU $\geq$ 0.7) Easy	Moderate	Hard	Submodule degradation	Detection degradation	Sensitivity
APL-R	43.87	33.65	29.01	10.00	25.41	2.54
APL-P	40.09	29.56	26.31	16.82	34.48	2.05
APL-S	38.94	27.48	24.17	27.16	39.08	1.44
APL	61.55	45.11	40.34	-	-	-

TABLE 5. Performance of 3D object detection networks trained on the three different APL sets. The evaluation metric is average precision (in %) with IoU over 0.7. The tests are conducted on KITTI validation set.

Task network	Refinement	Reliability check	AP_{3D} Easy	Moderate	Hard	AP_{BEV} Easy	Moderate	Hard
SECOND	✓		38.88	28.93	25.48	70.25	52.02	47.25
		✓	55.62	39.79	35.89	72.31	56.19	51.70
	✓	✓	61.55	45.11	40.34	76.50	58.91	57.17

the task networks to detect target objects even in challenging environments where various classes exist.

We also test the proposed framework as a data augmentation method for training a LiDAR-based 3D object detection network. When a LiDAR-based 3D object detection dataset [19], [21] provides stereo images, APL data can be worked as additional data for training LiDAR-based 3D object detection networks. We train task networks on the KITTI 3D data and APL data, and the trained networks are tested on the KITTI test set and validation set. The result is presented in Table 3. The networks trained on the augmented data mark higher performance in all six evaluation metrics. Fig. 4 shows the qualitative results of the networks [4] train on KITTI dataset and the augmented dataset, respectively. In the first and second columns of Fig. 4, the task network trained on the default KITTI cannot detect the farthest cars, but the network trained on the augmented dataset can detect them. In the third column, several cars are parked close to each other, in which case the network trained on the KITTI fails to detect the car located in the middle, but when using the additional data, all cars are successfully detected. As both networks' structure and training method are identical, the performance difference is attributed to the training data. In other words, the proposed framework can be utilized as a data augmentation method for training the 3D object detection network.

V. ABLATION STUDIES

A. SENSITIVITY TO SUBTASK

The proposed framework is based on the 2D object detection, 2D segmentation, and disparity estimation network. Therefore, we demonstrate how sensitively the quality of synthesized data depends on the performance of subnetworks. We exploit GA-Net [42], ResNet101 [47], and VideoProp-LabelRelax [43] to generate default APL data. To evaluate how sensitively the quality of data changes with respect to the subnetworks performance, we select subnetworks which have about 80% of the performance of the subnetworks used in the proposed method.

We change the 2D object detection part from ResNet101 [47] to ResNet18 [47] to generate APL-R data. We utilize Top-1 error (10-crop testing) on ImageNet validation [47] to compare subnetwork performance. To create APL-P data, we alter only the proposed method's disparity estimation part from GA-Net to PSMNet [48]. D1-all [49] is utilized to evaluate the subnetwork's performance. APL-S is obtained by converting the 2D segmentation part from VideoProp-LabelRelax [43] to SGDepth [50]. We use IoU class score [51] for performance comparison. To calculate detection degradation, we use the mean of AP_{BEV} and AP_{3D} in moderate cases as a representative of a detector's performance. The sensitiviy is obtained as follows:

$$sensitivity = \frac{100 - D_d}{100 - D_s}, \tag{5}$$

where D_d and D_s are performance degradation (in %) of a detection network and subnetwork, respectively.

When the task network is trained on the APL data synthesized by the inferior subnetworks, the performance degradation occurs. Among the subtasks, the 3D bounding box initialization has the most impact on the quality of APL data, as shown in Table 4.

B. EFFECTS OF THE CROSS-VERIFICATION STAGE

The proposed framework refines region proposals and selects reliable data in the cross-verification stage. To verify the contribution of the cross-verification stage, we compare the 3D object detection networks trained on the two different APL sets; one is generated without the refinement module and the other without the reliability check module. We employ SECOND as a task network and exploit KITTI stereo images to generate APL sets. The result is shown in the Table 5. Without the refinement module or reliability check module, precisions have fallen. This shows that the two modules improve APL data's quality to better train the task network.

When the reliability check module is not used, more performance degradation has occurred, and in particular, the average precisions of 3D bounding box decrease more. We believe that this is because inadequate data for training remains even if they passed the refinement module. Without the reliability check module, there remain points that

TABLE 6. **Performance comparison of 3D object detection networks [4] trained on the excavator data with subsampling and without subsampling. The evaluation metric is an average precision (%) of 3D bounding box.**

Training data	AP_{3D}	
	Validation	Test
w/o subsampling	90.91	0
w/ subsampling	99.18	86.57

incorrectly represent the target objects' appearance or do not have sufficient information to train the task network. On the other hand, when the refinement module is not used, the number of labels passing the reliability check module will decrease, but they have sufficient information and relatively good representations of the target object's appearance.

C. DOMAIN GAP BETWEEN PSEUDO- AND REAL-LIDAR POINT CLOUDS

The major domain shift seems to occur when the number of points between the training data (pseudo-LiDAR point clouds) and the test data (real-LiDAR point clouds) is not balanced. Without balancing the number of points between the training and testing point clouds, the task network trained on nearly 10 times more points than points provided by a real-LiDAR sensor cannot detect instances on a real-LiDAR point cloud, shown in Table 6. So the proposed method randomly samples points of an APL point cloud so that the number of points of an APL point cloud is similar to the average number of points in the test set point clouds. Using the sampled pseudo-LiDAR point clouds and automatically generated annotations, the 3D object detector trained on the APL data can detect the new class and achieves the high average precision value.

Considering unique point patterns of a LiDAR point cloud, the performance of a trained object detector can be improved by mimicking the point patterns in test data (real-LiDAR point cloud) during an APL point cloud generation. However, since point patterns vary across LiDAR sensors, our study does not include any method for generating point patterns.

VI. CONCLUSION

We propose a two-stage framework that automatically generates pseudo-LiDAR points and 3D bounding box annotations using stereo images. Unlike existing pseudo-LiDAR-based 3D object detection networks which exploit 3D labels of an open dataset for training, the proposed framework can synthesize 3D labels using image-based networks and stereo images only. It removes the needs to obtain 3D LiDAR point clouds and annotate 3D bounding boxes manually. Experiments show that the networks trained on APL data can perform about 99% of the network trained on the LiDAR point clouds and human-made labels on low-difficulty instances of KITTI dataset with IoU threshold 0.5, and the networks trained on the APL data of the new classes achieve high precisions. Moreover, we also verify that the proposed framework can be utilized as a data augmentation method for training LiDAR-based 3D object detection networks.

LiDAR-based 3D object detectors usually outperform monocular or stereo-based 3D object detectors because LiDAR-based methods can leverage highly accurate depth measurements. Although LiDAR-based 3D object detectors trained on the APL dataset may not outperform those trained on real LiDAR point clouds and human-made labels, we can easily train high-performing off-the-shelf LiDAR-based models only with stereo images using our method. Therefore, the proposed method facilitates the use of a LiDAR-based 3D object detector as one option for performing 3D object detection in situations where creating new human-made labels and acquiring real LiDAR point clouds are challenging due to resource or time constraints.

REFERENCES

[1] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast encoders for object detection from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12689–12697.

[2] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum PointNets for 3D object detection from RGB-D data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 918–927.

[3] W. Shi and R. Rajkumar, "Point-GNN: Graph neural network for 3D object detection in a point cloud," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1708–1716.

[4] Y. Yan, Y. Mao, and B. Li, "SECOND: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, Oct. 2018.

[5] M. Wang, L. Zhao, and Y. Yue, "PA3DNet: 3-D vehicle detection with pseudo shape segmentation and adaptive camera-LiDAR fusion," *IEEE Trans. Ind. Informat.*, vol. 19, no. 11, pp. 10693–10703, Nov. 2023, doi: 10.1109/TII.2023.3241585.

[6] S. Li, K. Geng, G. Yin, Z. Wang, and M. Qian, "MVMM: Multi-view multimodal 3-D object detection for autonomous driving," *IEEE Trans. Ind. Informat.*, vol. 20, no. 1, pp. 845–853, Jan. 2024, doi: 10.1109/TII.2023.3263274.

[7] L.-H. Wen and K.-H. Jo, "Three-attention mechanisms for one-stage 3-D object detection based on LiDAR and camera," *IEEE Trans. Ind. Informat.*, vol. 17, no. 10, pp. 6655–6663, Oct. 2021.

[8] Z. Zhang, Z. Liang, M. Zhang, X. Zhao, H. Li, M. Yang, W. Tan, and S. Pu, "RangeLVDet: Boosting 3D object detection in LiDAR with range image and RGB image," *IEEE Sensors J.*, vol. 22, no. 2, pp. 1391–1403, Jan. 2022.

[9] J. Lin, H. Yin, J. Yan, W. Ge, H. Zhang, and G. Rigoll, "Improved 3D object detector under snowfall weather condition based on LiDAR point cloud," *IEEE Sensors J.*, vol. 22, no. 16, pp. 16276–16292, Aug. 2022.

[10] X. Zhao, P. Sun, Z. Xu, H. Min, and H. Yu, "Fusion of 3D LiDAR and camera data for object detection in autonomous vehicle applications," *IEEE Sensors J.*, vol. 20, no. 9, pp. 4901–4913, May 2020.

[11] Z. Gong, Z. Wang, G. Yu, W. Liu, S. Yang, and B. Zhou, "FecNet: A feature enhancement and cascade network for object detection using roadside LiDAR," *IEEE Sensors J.*, vol. 23, no. 19, pp. 23780–23791, Oct. 2023.

[12] Y. Wu, S. Zhang, H. Ogai, H. Inujima, and S. Tateno, "Realtime single-shot refinement neural network with adaptive receptive field for 3D object detection from LiDAR point cloud," *IEEE Sensors J.*, vol. 21, no. 21, pp. 24505–24519, Nov. 2021.

[13] L. Wen and K.-H. Jo, "Fast LiDAR R-CNN: Residual relation-aware region proposal networks for multiclass 3-D object detection," *IEEE Sensors J.*, vol. 22, no. 12, pp. 12323–12331, Jun. 2022.

[14] W. Zhangyu, Y. Guizhen, W. Xinkai, L. Haoran, and L. Da, "A camera and LiDAR data fusion method for railway object detection," *IEEE Sensors J.*, vol. 21, no. 12, pp. 13442–13454, Jun. 2021.

[15] M. Chen, P. Liu, and H. Zhao, "BCAF-3D: Bilateral content awareness fusion for cross-modal 3D object detection," *Knowl.-Based Syst.*, vol. 279, Nov. 2023, Art. no. 110952.

[16] M. Chen, P. Liu, and H. Zhao, "M3DGAF: Monocular 3D object detection with geometric appearance awareness and feature fusion," *IEEE Sensors J.*, vol. 23, no. 11, pp. 11232–11240, Jun. 2023, doi: 10.1109/JSEN.2022.3189174.

[17] S. Y. Alaba and J. E. Ball, "A survey on deep-learning-based LiDAR 3D object detection for autonomous driving," *Sensors*, vol. 22, no. 24, p. 9577, Dec. 2022.

[18] J. Wang, X. Lin, and H. Yu, "POAT-Net: Parallel offset-attention assisted transformer for 3D object detection for autonomous driving," *IEEE Access*, vol. 9, pp. 151110–151117, 2021.

[19] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "NuScenes: A multimodal dataset for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11618–11628.

[20] Y. Choi, N. Kim, S. Hwang, K. Park, J. S. Yoon, K. An, and I. S. Kweon, "KAIST multi-spectral day/night data set for autonomous and assisted driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 3, pp. 934–948, Mar. 2018.

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

[22] *Waymo Open Dataset: An Autonomous Driving Dataset*, Waymo LLC, Mountain View, CA, USA, 2019.

[23] A. Patil, S. Malla, H. Gang, and Y.-T. Chen, "The H3D dataset for full-surround 3D multi-object detection and tracking in crowded urban scenes," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 9552–9557.

[24] Q. Meng, W. Wang, T. Zhou, J. Shen, L. Van Gool, and D. Dai, "Weakly supervised 3D object detection from LiDAR point cloud," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2020, pp. 515–531.

[25] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1325–1339, Jul. 2014.

[26] W. Kim, M. Tanaka, M. Okutomi, and Y. Sasaki, "Automatic labeled LiDAR data generation based on precise human model," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 43–49.

[27] J. Tremblay, T. To, and S. Birchfield, "Falling things: A synthetic dataset for 3D object detection and pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2018, pp. 2038–2041.

[28] D. Feng, X. Wei, L. Rosenbaum, A. Maki, and K. Dietmayer, "Deep active learning for efficient training of a LiDAR 3D object detector," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2019, pp. 667–674.

[29] J. Lee, S. Walsh, A. Harakeh, and S. L. Waslander, "Leveraging pre-trained 3D object detection models for fast ground truth generation," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2504–2510.

[30] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 77–85.

[31] I. Mouawad, N. Brasch, F. Manhardt, F. Tombari, and F. Odone, "Time-to-label: Temporal consistency for self-supervised monocular 3D object detection," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 8988–8995, Oct. 2022.

[32] C. R. Qi, Y. Zhou, M. Najibi, P. Sun, K. Vo, B. Deng, and D. Anguelov, "Offboard 3D object detection from point cloud sequences," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 6130–6140.

[33] S. Zakharov, W. Kehl, A. Bhargava, and A. Gaidon, "Autolabeling 3D objects with differentiable rendering of SDF shape priors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12221–12230.

[34] Z. Liu, D. Zhou, F. Lu, J. Fang, and L. Zhang, "AutoShape: Real-time shape-aware monocular 3D object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 15621–15630.

[35] A. Handa, V. Patraucean, S. Stent, and R. Cipolla, "SceneNet: An annotated model generator for indoor scene understanding," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 5737–5743.

[36] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[37] P. Li, X. Chen, and S. Shen, "Stereo R-CNN based 3D object detection for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7636–7644.

[38] J. Sun, L. Chen, Y. Xie, S. Zhang, Q. Jiang, X. Zhou, and H. Bao, "Disp R-CNN: Stereo 3D object detection via shape prior guided instance disparity estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10548–10557.

[39] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger, "Pseudo-LiDAR from visual depth estimation: Bridging the gap in 3D object detection for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8437–8445.

[40] A. D. Pon, J. Ku, C. Li, and S. L. Waslander, "Object-centric stereo matching for 3D object detection," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 8383–8389.

[41] J. Lambert, Z. Liu, O. Sener, J. Hays, and V. Koltun, "MSeg: A composite dataset for multi-domain semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2876–2885.

[42] F. Zhang, V. Prisacariu, R. Yang, and P. H. S. Torr, "GA-Net: Guided aggregation net for end-to-end stereo matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 185–194.

[43] Y. Zhu, K. Sapra, F. A. Reda, K. J. Shih, S. Newsam, A. Tao, and B. Catanzaro, "Improving semantic segmentation via video propagation and label relaxation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8848–8857.

[44] S. Shi, X. Wang, and H. Li, "PointRCNN: 3D object proposal generation and detection from point cloud," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 770–779.

[45] MMDetection3D Contributors. (2020). *MMDetection3D: OpenMMLab Next-generation Platform for General 3D Object Detection*. [Online]. Available: https://github.com/open-mmlab/mmdetection3d

[46] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D object detection network for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6526–6534.

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

[48] J.-R. Chang and Y.-S. Chen, "Pyramid stereo matching network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5410–5418.

[49] M. Menze, C. Heipke, and A. Geiger, "Joint 3D estimation of vehicles and scene flow," *ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 2, p. 427–434, Aug. 2015.

[50] M. Klingner, J.-A. Termöhlen, J. Mikolajczyk, and T. Fingscheidt, "Self-supervised monocular depth estimation: Solving the dynamic object problem by semantic guidance," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2020, pp. 582–600.

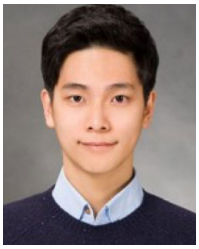
[51] H. A. Alhaija, S. K. Mustikovela, L. Mescheder, A. Geiger, and C. Rother, "Augmented reality meets computer vision: Efficient data generation for urban driving scenes," *Int. J. Comput. Vis.*, vol. 126, no. 9, pp. 961–972, Sep. 2018.

CHANGSUK OH received the B.S. and M.S. degrees from the Department of Mechanical and Aerospace Engineering, Seoul National University, Seoul, South Korea, in 2019 and 2021, respectively, where he is currently pursuing the Ph.D. degree in aerospace engineering. His current research interests include deep learning and computer vision.

YOUNGSEOK JANG (Graduate Student Member, IEEE) received the B.S. degree in mechanical engineering from Sungkyunkwan University, Suwon, South Korea, in 2017. He is currently pursuing the integrated M.S./Ph.D. degree with the Department of Mechanical and Aerospace Engineering, Seoul National University, Seoul, South Korea. His research interests include visual navigation for multirobot systems, sensor fusion for navigation, and perception-aware path planning.

DONGSEOK SHIM (Graduate Student Member, IEEE) received the B.S. and M.S. degrees in mechanical and aerospace engineering from Seoul National University, Seoul, South Korea, in 2020 and 2022, respectively, where he is currently pursuing the Ph.D. degree in artificial intelligence. His current research interests include deep learning and computer vision.

CHANGHYEON KIM received the B.S. and M.S. degrees from the Department of Mechanical and Aerospace Engineering, Seoul National University, Seoul, South Korea, in 2016 and 2018, respectively, and the Ph.D. degree in aerospace engineering from Seoul National University, in 2023. He is currently a Research Engineer with Samsung Research, Seoul. His research interests include 3-D reconstruction, visual navigation, and camera-IMU-LiDAR fusion.



JUNHA KIM received the B.S. degree in automotive engineering from Hanyang University, Seoul, South Korea, in 2019. He is currently pursuing the integrated M.S./Ph.D. degree in mechanical and aerospace engineering with Seoul National University, Seoul. His research interests include camera and LiDAR odometry, 3-D reconstruction, and camera-LiDAR fusion.

H. JIN KIM (Member, IEEE) received the B.S. degree from the Korea Advanced Institute of Technology (KAIST), in 1995, and the M.S. and Ph.D. degrees in mechanical engineering from the University of California at Berkeley (UC Berkeley), in 1999 and 2001, respectively. From 2002 to 2004, she was a Postdoctoral Researcher of electrical engineering and computer science with UC Berkeley. In 2004, she joined the Department of Mechanical and Aerospace Engineering, Seoul National University, as an Assistant Professor, where she is currently a Professor. Her research interests include intelligent control of robotic systems and motion planning.