

Received March 28, 2019, accepted April 18, 2019, date of current version May 6, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2913433

Automatic Labeled LiDAR Data Generation and Distance-Based Ensemble Learning for Human Segmentation

WONJIK KIM^{1,2}, (Member, IEEE), MASAYUKI TANAKA^{1,2}, (Member, IEEE),
MASATOSHI OKUTOMI¹, (Member, IEEE), AND
YOKO SASAKI^{1,2}, (Member, IEEE)

¹Department of Systems and Control Engineering, School of Engineering, Tokyo Institute of Technology, Tokyo 152-8550, Japan

²Artificial Intelligence Research Center, National Institute of Advanced Industrial Science and Technology, Tokyo 135-0064, Japan

Corresponding author: Wonjik Kim (wkim@ok.sc.e.titech.ac.jp)

ABSTRACT Following the improvements in deep neural networks, state-of-the-art networks have been proposed for human segmentation using point clouds captured by light detection and ranging. However, the performance of these networks depends significantly on the training data. An issue with collecting training data is labeling. Labeling by humans is necessary to obtain ground-truth labels; however, labeling involves high costs. Therefore, we propose an automatically labeled data generation pipeline, for which we can change any parameters or data generation environments. Our approach uses a human model named Dhaiba and the background of Miraikan to generate realistic artificial data. We present 1M data generated by the proposed pipeline. Furthermore, we propose an ensemble learning based on generated data for utilizing our data generation pipeline. This paper proposes the specifications of the pipeline, data details, and explanation of ensemble learning with evaluations of various approaches.

INDEX TERMS Artificial intelligent, machine learning, computer vision, LiDAR.

I. INTRODUCTION

Deep neural networks have considerably affected computer vision, and robotics applications, among other fields. For example, various types of high-performance deep neural networks for the pixel-wise segmentation of RGB images have been proposed [1]–[3]. Following their success in RGB image segmentation, segmentation networks for depth map and point cloud data have been investigated vigorously [4]–[7]. Human segmentation is important in many types of robotics applications. Recently, light detection and ranging (LiDAR) has become a powerful tool for robot navigation. Therefore, in this study, we focus on human segmentation with three-dimensional point cloud data collected by LiDAR.

Deep neural networks typically require a considerably large amount of manually labeled training data to achieve high performance. Collecting a sufficient amount of labeled data may incur massive costs in terms of both time and

money. In this study, we develop a data generation pipeline that reduces manual labeling cost. The proposed data generation pipeline comprises three steps: 1) background data collection, 2) human model building, and 3) LiDAR data generation with human labels. The data generation pipeline can provide depth maps, point cloud data with xyz coordinates, and human labels. The generated data already includes the human labels; therefore, a manual labeling process is not required. Additionally, the proposed data generation pipeline can produce point cloud data with any human size, different types of LiDAR, and distance between LiDAR and human by changing several parameters.

We have built and trained a human segmentation network with LiDAR data generated by the proposed pipeline [8]. Subsequently, we observed that the human segmentation performance depends on the distance between the human and sensor. Generally, a distant object is more challenging to segment than a near object. We consider two reasons. First, the amount of LiDAR data associated with the distant object is less than that associated with the near object.

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

Next, the number of training samples associated with the distant object tends to be small, because other objects and/or walls occlude the distant object.

Subsequently, we develop a novel distance-based ensemble learning for human segmentation. First, we synthesize the LiDAR training data with different distance ranges. Subsequently, multiple networks are trained with the LiDAR training data with different distance range. In the inference phase, the outputs of those trained networks are assembled based on the distance. We call those training approaches distance-based ensemble training.

Our goal is to improve the performance of learning-based human segmentation with well-constructed training data. From the experimental result, we discovered that a simple training approach with the overall data might fail to perform well. Accordingly, we propose an ensemble method using networks tuned by different datasets. For the ensemble method, we generated five types of data divided in human model distance with the proposed data generation pipeline. We trained segmentation networks with the generated training data. The trained network has been evaluated with the generated test data and actual data collected by a real LiDAR sensor.

This paper is organized as follows. We review the related works in section II briefly. In section III, the entire data generation procedure is explained in detail with the description of each step. The concept of ensemble learning and implementation method are described in section IV. Section V-A contains a brief explanation of the segmentation and network model we used. The specific policy of the training and evaluation are described in section V. The conclusions are provided in section VI.

This study is an extension of the previous research published in international conference on robotics and automation 2019 [8]. In this study, we generate data by deploying humans in any distance range. This change allowed us to analyze the network performance in the depth direction. Based on the results of performance evaluation in the depth direction, we propose an ensemble learning with the generated data divided in distance. All generated data and labeled real data are presented in following url. Trained network weight and test sample code also included.

<http://www.ok.sc.e.titech.ac.jp/res/LHD/>

II. RELATED WORK

After the release of Microsoft Kinect in 2010, several RGB-D datasets have been published. RGB-D datasets for human recognition have been provided for the re-identification of a person with RGB-D sensors [9], BIWI RGBD-ID dataset [10], and UPCV Gait dataset [11]. As the Kinect cannot measure depths greater than 10 [m], LiDAR sensors were employed to handle depths over 10 [m]. In addition, LiDAR sensors are used in auto driving technology. In this field, the KITTI dataset [12] is used widely by many researchers [13], [14]. However, KITTI only provided 93K+ of depth data without labeling. Collecting labeled depth maps

is still challenging. Pixel or point-wise labeling for three-dimensional (3D) depth data is typically challenging as it involves massive cost. Hence, the video game Grand Theft Auto was deployed to collect data [4], [15], [16]. This approach may reduce the cost of data construction; nevertheless, limitations still exist. Grand Theft Auto is not designed for research purposes; therefore, we cannot control the specific properties of the circumstances in the simulation, such as human body type and model deployment location.

Hence, we constructed a data generation pipeline in which any parameters and environments can be changed. Furthermore, the proposed generator creates human labels during the process, thereby incorporating the human task into the computation cost. Consequently, we can generate data continuously if sufficient computational resources are available.

III. PROPOSED DATA GENERATION PIPELINE

A primary contribution of this study is the automatic generation of labeled depth map without involving manual labeling. The labeled depth map generation comprises three steps: 1) background data collection, 2) human model building, and 3) depth map synthesis with automatic labeling. The background depth maps without human were collected in advance. We collected 6,281 background depth maps at the 3rd floor of Miraikan, which is National Museum of Emerging Science and Innovation in Tokyo, Japan [17], using a velodyne HDL-32E LiDAR sensor [18]. Precise and realistic 3D human models were synthesized based on a digital human model called Dhaiba [19]. The depth maps were generated by composing the collected background depth map and 3D human model. The label data can be obtained easily because the locations of the 3D human models are known. This implies that we can generate labeled depth maps automatically without a manual input. This is an advantage of the proposed approach. The details are described in the following subsections, and an overview of the pipeline is illustrated in Fig. 1.

A. HUMAN MODEL

A key component of the proposed data generation pipeline is the precise 3D human model. We employed a digital human model, called “Dhaiba,” of DhaibaWorks [19] to build precise 3D human models. The Dhaiba is a human body function model, and DhaibaWorks is a platform for producing digital human models based on Dhaiba. DhaibaWorks supports editing and visualizing basic models such as 3D meshes and skeletal structures, including human models with motion [20]. Using DhaibaWorks, we can generate a specific human model easily by setting the human parameters such as height, weight, and action status [21].

In real environment, there can be human with any pose including standing, walking, and running. However, we assume that the human model is walking during the depth map generation in this study. Walking motion data are required to build an artificial walking human model. We used one period of walking data collected in [22]. The walking

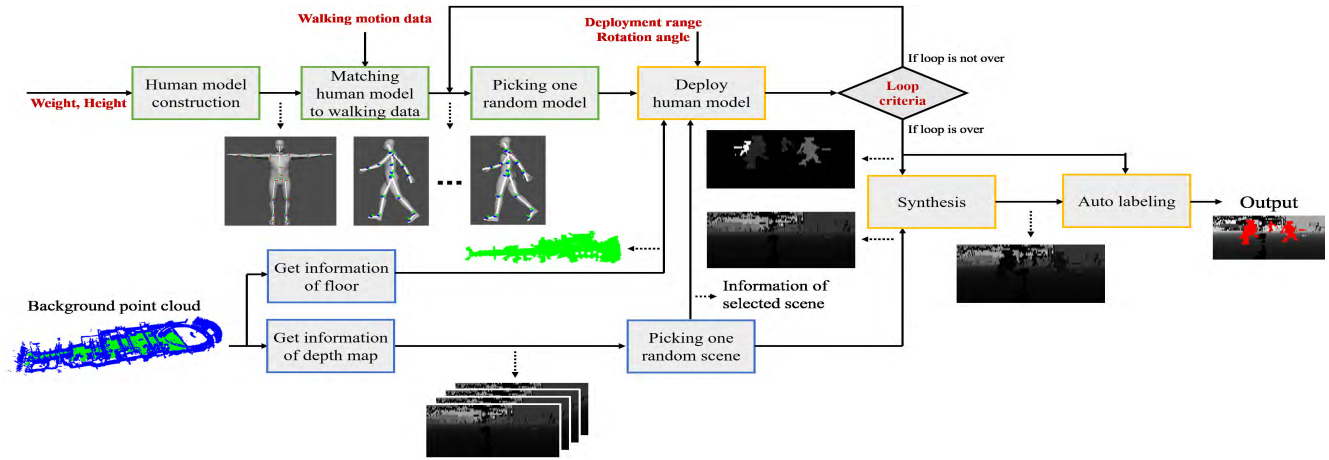


FIGURE 1. Overview of the pipeline. Red words represent controllable parameters. (Green boxes: Human model building) Human models are built by weight and height; subsequently, they are combined with a human walking motion. (Blue boxes: Background data processing) Floor pointcloud was estimated from precollected background point cloud. (Yellow boxes: LiDAR data generation) Walking human models are chosen and deployed into the estimated floor based on deployment range. Subsequently, the depth map with human models is synthesized with a randomly chosen background depth map. The depth map is labeled based on the information from the human model deployed location.

TABLE 1. Human model specifications.

Height (mm)	1200	1400	1600	1700	1800
Weight (kg)	15 20 30	20 30 40	40 50 70	50 60 80	50 70 90

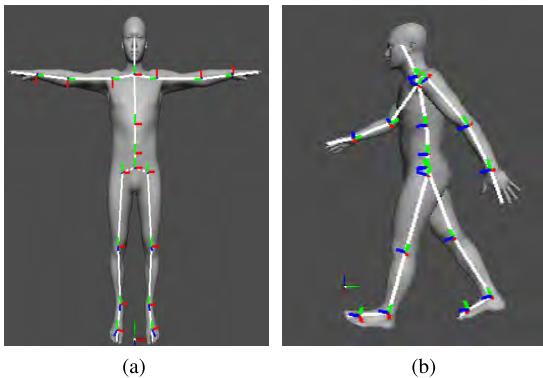


FIGURE 2. Example of built human mesh based on Dhaiba. (a) Human mesh. (b) Walking human mesh.

motion data contains 230 frames for a single walking motion. Subsequently, it is easy to build artificial walking human mesh models for depth map generation.

In this study, we used 15 typical combinations of height and weight, as summarized in Table 1. In our opinion, these combinations include a variety of scenarios. Therefore, we have 230 frames of walking motion data and 15 combinations of height and weight. As such, 3,450 different types of walking models can be generated. Figure 2 shows an example of the built human models.

B. BACKGROUND 3D DATA

Background data were collected on the 3rd floor of Miraikan with HDL-32E LiDAR. The collected data contains only the

xyz coordinates; subsequently, floor estimation is required for extracting the floor from background 3D data. Therefore, we estimated the floor based on a two-dimensional drivable map [23]. Not only floor data, but also LiDAR location are required for setting the human model deployment area. The location calculated by wheel odometry was recorded with the background point cloud. However, if only wheel odometry is used for estimating the location, the result may not be reliable because of the slip between the wheel and floor. Subsequently, we employed 'real-time 6DoF Monte-Carlo localization' [24] to obtain the accurate LiDAR location. The estimated LiDAR location is shown in Figure 4-(b)

C. LIDAR DATA GENERATION

In the LiDAR data generation step, the depth map of the built human model is first synthesized. Thereafter, the synthesized depth map of the human model and the randomly chosen background depth map are combined to generate the training depth map for human segmentation.

To synthesize the human model depth map, we positioned the LiDAR sensor virtually at $(0, 0, z_s)$, where z_s was at the same height as that of the sensor for background collection. We assumed that the LiDAR sensor was installed horizontally. Subsequently, the human model depth map was synthesized for the given human model, by virtually inserting the human model at $(x_s, y_s, 0)$, where (x_s, y_s) was the sampled position in the floor point cloud. The human model direction was sampled randomly as well. Once the geometrical positions of the LiDAR sensor and the human model were provided, the associated depth map can be synthesized easily.

After synthesizing the human model depth map, it is combined with the background depth map by pixel-wise minimum depth selection. In this step, the human model farther than the background is deleted by the minimum depth selection. The depth map captured by the LiDAR sensor typically

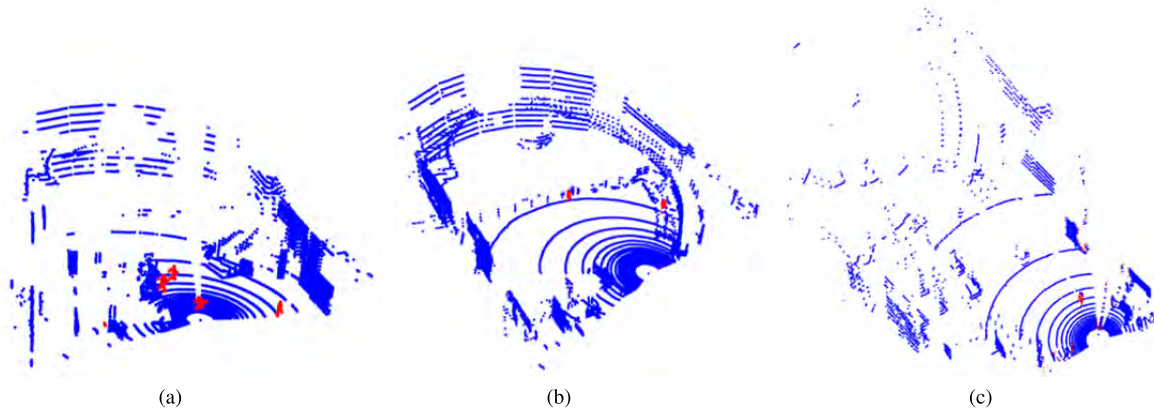


FIGURE 3. Example of data. Blue points denote background and red points denote human. Figures (a) and (b) show examples of generated data in different human model deployment ranges. Figure (c) shows an example of real data labeled manually. (a) Deployment range 1000 to 10000 [mm]. (b) Deployment range 5000 to 15000 [mm]. (c) Real data.

TABLE 2. Data generation properties.

Description	Specification
Number of detector pairs	32
Limitation in horizontal scanning	1024
Vertical scanning range	+10.67 to -30.67°
Angular resolution in vertical	1.33°
Angular resolution in horizontal	0.2°
Human number in one image	0 to 10
Rotation angle of Dhaiba	0 to 360 degrees
Output depth image size	$32 \times 1024 \times 1$
Output xyz map size	$32 \times 1024 \times 3$
Output label data size	$32 \times 1024 \times 1$
LiDAR position	(0, 0, 800) [mm]

includes holes or missing pixels whose the depth cannot be measured. These holes are used only for the synthesis process as they are equally obtainable in the real sensing process. In addition, the human labeling task can be performed simultaneously because we know which pixels correspond to the human model depth map.

The parameters in the LiDAR data generation are summarized in Table 2. The parameters in Table 2 are chosen based on the properties of velodyne HDL-32E [18] and the experimental environment of background data collection. Figure 3 shows an example of the generated and real point cloud data. The point cloud data are converted from the depth map.

D. PROVISION OF DATA

We generated 1M data using the proposed pipeline. They contain the depth, xyz coordinates, and human label in HDF5 format. Furthermore, we provide the specific information for each depth data in the form of an xml file. Xml files contain a human number in the depth scene, location, weight, and height of each Dhaiba. In the 1M data, 500K data are the same as those in [8]. The other 500K data consist of five different groups described in V, and each group contains 100K of data. Further, we generated the 0.5K dataset described in section V and 0.1K of manually labeled real data.

IV. DISTANCE BASED ENSEMBLE LEARNING

The properties of the point cloud data of the human model depends significantly on the distance between the LiDAR

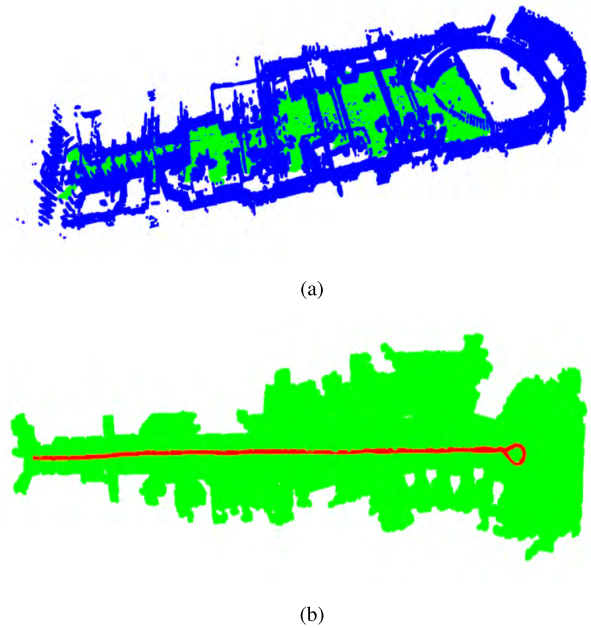


FIGURE 4. Background and LiDAR location. Green and blue points represent background, where green points denote floor and blue points denote other objects. Red points denote LiDAR location. (a) Background point cloud. (b) Floor point cloud and LiDAR location.

sensor and the human. It is limited to include whole distance ranges by a single network. It is advisable for the single network to focus on a certain distance range. Subsequently, we propose a distance-based ensemble learning. In the training phase, we train multiple networks focusing on the different distance ranges. Subsequently, the inferences of the multiple networks are assembled based on the measured distance information.

If we construct LiDAR data by real-world experiment, it is expensive to obtain sufficient data with different distance ranges. Therefore, data from real world are exposed to be unbalanced. This implies that a network trained by unbalanced data may be ineffective for insufficient data types. Meanwhile, it is possible to generate data including humans

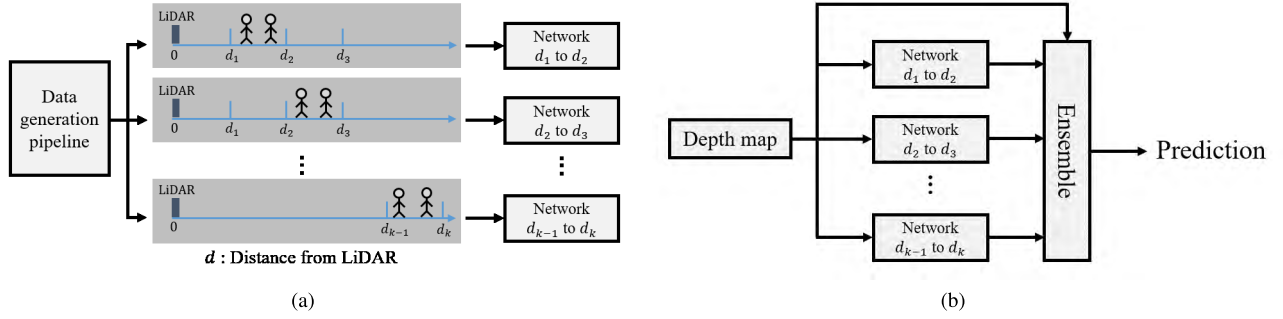


FIGURE 5. Training and Ensemble phase. Figure (a) describes the method to train networks. Data generation pipeline creates data divided in distance range and each network is tuned by different data. Figure (b) illustrates the method to ensemble trained networks. In this work, we employed depth map as input data. The networks optimized in the training phase provide their own predictions. Thereafter, the ensemble layer extracts the overall prediction based on each result and input data. (a) Training phase. (b) Ensemble phase.

in specific distances by the data generation pipeline proposed in section III. Accordingly, we propose a novel method for utilizing the generated data. The method includes two steps: 1) constructing networks specialized in different ranges, and 2) decision by ensemble networks.

Networks can be combined variously. Herein, we introduce three concepts for assigning the weights for the overall prediction. For distance-based Gaussian weight, the depth information is necessary. However, defects exist in the LiDAR data; therefore, we interpolated the defects using a low-pass filter when we used LiDAR data as the input for calculating the weight in section IV-1.

1) DISTANCE BASED GAUSSIAN WEIGHT

Distance-based Gaussian weight considers every output of the networks. From the input depth map, we can calculate the Gaussian kernel between the pixel value and distance in which the network is specialized. Using the calculation results, we can obtain the weights for every network. The pixel value is required in this approach; however, the LiDAR depth map typically contains defects. Thus, we interpolated the defects using a low-pass filter for only the weight calculation.

2) SIMPLE AVERAGE

The concept of a simple average is that each network can be considered equally. Subsequently, the overall output will be average of each network's prediction value.

3) TRAINED WEIGHT

We attempted to train another network specialized in weight. In this case, whole networks will become untrainable except the network for weight calculation. Thus, the weight network will be optimized by the depth map as the input and weights of each network as output.

V. EVALUATION

For evaluation, we employed five types of distance ranges for data generation: 01 [m] to 10 [m], 05 [m] to 15 [m], 10 [m] to 20 [m], 15 [m] to 25 [m], and 20 [m] to 30 [m]. We generated 55136

10.1K data for each distance range; subsequently, 10K data were used for training and 0.1K data were used for validation and estimation. Consequently, 50K data were used for training and 0.5K data for evaluation. We investigated the average human-labeled pixels per image. For the distance range of 01 [m] to 10 [m], 1402 pixels of one depth image were labeled as human on average. In contrast, 295 pixels per depth image were labeled as human for 05 [m] to 15 [m] data, 99 for 10 [m] to 20 [m], 48 for 15 [m] to 25 [m], and 27 for 20 [m] to 30 [m]. Therefore, the depth image of 20 [m] to 30 [m] contains only 27 human-labeled pixels on average, i.e., 2% of the 01 [m] to 10 [m] case and 0.08% of the pixel number in one image. Accordingly, the distance between human and sensor affects the number of human-labeled pixels and the difficulty in human recognition.

In this section, four different networks are described in section V-A, and the training method is shown in V-B. Section V-C contains V-C.1: the effect of trained data number, and V-C.2: the benchmarking of different networks. Section V-D contains the evaluation results of distance based ensemble learning. A more specific analysis of the generated data including the effectiveness of precision of the human model and a validation of different combinations of heights and weights are explained in [8].

A. SEGMENTATION NEURAL NETWORKS

Semantic segmentation has been investigated using various approaches for applications. The wide range of its application includes scene understanding, depth analysis, and autonomous driving. Semantic pixel-wise labeling has been garnering considerable interest owing to improvements in deep learning [2], [25]–[27]. Three different networks were selected in this study to evaluate the effectiveness of segmentation networks in the generated human depth maps.

1) FULLY CONVOLUTIONAL NETWORK

The fully convolutional network (FCN) for segmentation was proposed by Long [1]. Classification networks typically use fully connected layers at the end of a network to compute class scores. On the contrary, the FCN changes the fully

connected layer to a convolutional layer, and the output of the network is also changed from a predicted score to a heatmap. Thus, the classification network can be changed easily to a segmentation network using the concept of an FCN.

2) FULLY CONVOLUTIONAL DENSENETS

Fully convolutional DenseNets were proposed by Jégou [28]. They were inspired by densely connected convolutional networks (DenseNets) [29]. DenseNets exhibit good performance in image classification, and the concept of DenseNets is based on the observation that if each layer is connected directly to every other layer in a feedforward manner, subsequently the network will be more accurate and easier to train. Jégou proposed an extension to the concept of DenseNets to address problems in semantic segmentation. Herein, we refer to this network as the FCDN.

3) POINTNET

Pointnet was proposed by Charles [5], and was designed for addressing unordered point sets in 3D; it is not affected by rotation or translation. In addition, by adapting a transform network into Pointnet, it can gain interactions among points and capture local structures from neighbor points. Two versions of Pointnet are available: classification and segmentation. The architectures of both versions exhibit differences in the latter half. In this study, we employed Pointnet as a segmentation network.

B. TRAINING POLICY

We trained all networks with 500 epochs. In each epoch, we randomly chose 500 scenes from the training datasets. We set a different batch size that was chosen experimentally for each network, 25 for the FCN, 16 for PointNet, and 5 for the FCDN. We subsequently coordinated the step for epoch as $batchsize \times step \text{ for epoch} = 500$, except for PointNet. As PointNet uses a one-dimensional vector, we divided one scene into 16 vectors. Therefore, PointNet is trained with 500 scenes in one epoch, with a batch size of 16. Further, we employed Adam [30] with a learning rate of 0.001 and decay of 0.001 for the optimizer and categorical cross-entropy, respectively, for the objective function. From Table 2, the size of the input image is 32×1024 . Therefore, we revised the FCN and FCDN algorithms in the pooling layers. As the height of the input image is only 32, we pooled only the input data in the width direction. For Pointnet, the input images were divided vertically into 16 pieces of images and addressed with 2048 pixels as points for the input data. For the FCDN, we revised the pooling layer to a width-pooling layer, the number of layers per dense block to three, and the growth rate to eight.

C. GENERATED LIDAR DATA EVALUATION

We analyzed the effectiveness of the produced data and verified them using various approaches. In this section, Avg. Acc denotes the average of the background accuracy and human accuracy. In addition, avg. acc., precision, recall, and threat scores [31] were calculated human is positive and background

TABLE 3. Real data for evaluation.

Description	Specification
Location	Miraikan [17]
Date	2017.Aug.04
LiDAR model	Velodyne HDL32E
LiDAR height from ground	800 [mm]
Data number	100
Labeling method	Manual

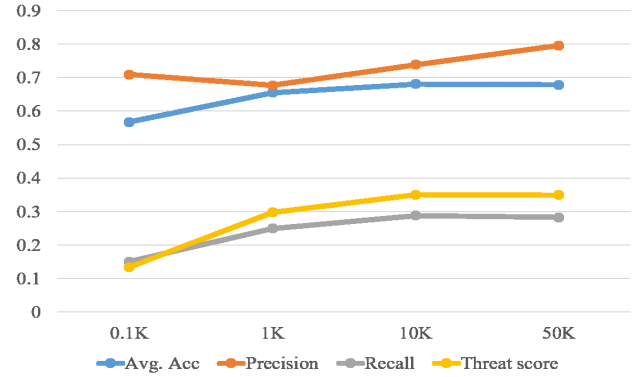


FIGURE 6. Effect of training data number.

is negative. We computed each score as a true negative (TN), false negative (FN), true positive (TP), and false positive (FP) as follows:

$$Avg.Acc = \frac{TN}{TN + FP} + \frac{TP}{TP + FN} \div 2$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$Threat\ score = \frac{TP}{TP + FP + FN}$$

In addition, we collected the real data and labeled them to compare and evaluate performances. Details of real data are described in table 3.

1) EFFECT OF TRAINING DATA NUMBER

In general, the performance of the learning-based approach can be improved by increasing the volume of the training data, especially in deep neural networks. However, in practice, it is difficult to collect a large volume of training data owing to the labeling cost. An advantage of the proposed depth map generation pipeline is automatic labeling. In this sense, the labeling cost of the proposed pipeline is significantly lower than that of manual labeling. Here, we clarify the effect of the number of training examples. We prepared four different sizes of training datasets comprising five different distance ranges equally, 0.1K, 1K, 10K, and 50K. Subsequently, the FCN networks were trained with the four datasets. The trained networks were evaluated with the labeled real data.

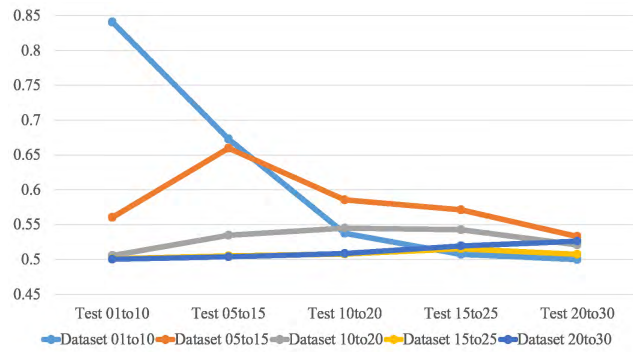
Figure 6 shows the evaluation results. As shown, the average accuracy, recall, and threat scores were improved by increasing the volume of the training data from 0.1K to

TABLE 4. Results of the networks for human segmentation.

Network Input		FCN [1]			FCDN [28]			Pointnet [5]
		depth	xyz	depth+xyz	depth	xyz	depth+xyz	
Generated data	Avg. Acc	0.6895	0.6317	0.6963	0.5292	0.6789	0.5449	0.5058
	Recall	0.1555	0.1291	0.1848	0.1858	0.8169	0.2361	0.0703
	Threat score	0.3205	0.2149	0.2882	0.0245	0.0298	0.0299	0.0116
Real data	Avg. Acc	0.6780	0.5143	0.5163	0.5137	0.5787	0.5175	0.4997
	Recall	0.2824	0.0347	0.0357	0.0724	0.4022	0.0830	0.0008
	Threat score	0.3490	0.0315	0.0332	0.0434	0.1209	0.0521	0.0001

TABLE 5. Performance of FCN trained by depth toward different distance range datasets.

Dataset	01to10	05to15	10to20	15to25	20to30
Avg. Acc	0.7271	0.6023	0.5419	0.5190	0.5056
Recall	0.3832	0.1907	0.0873	0.0449	0.0097
Threat score	0.4293	0.1697	0.0528	0.0142	0.0045

**FIGURE 7. Average accuracy of each specialized network toward each dataset.**

10K training. However, the precision exhibits a different tendency. When training with 0.1K of data, the network could only be regarded as human when it exhibited a high confidence. Therefore, the average accuracy, recall, and threat scores of the training 0.1K data were lower than those of any other cases; however, the precision was greater than 0.7. The network trained with 50K data demonstrates almost the same performance as that of the 10K training case in terms of average accuracy, recall, and threat scores. However, the precision increased from 10K to 50K data training. Hence, we conclude that a large volume of data facilitates in performance improvement.

2) BENCHMARKING EXISTING NETWORKS

We evaluated three different types of network learning with the training data generated by the proposed pipeline. As in [8], we employed U-net [26] for evaluation; however, U-net failed to train with our training policy. Therefore, we abbreviated the result of U-net. For the FCN and FCDN networks, we prepared three different types of inputs: the depth data only, the xyz data only, and the depth-and-xyz data. We used the xyz data for the input of the Pointnet owing to the network structure of Pointnet. The manually labeled 0.1K real data and the 0.5K test data generated by the proposed pipeline were used for the evaluation. It is noteworthy that the training data does not include the test data.

Table 4 shows the evaluation results. From these comparisons, the FCN network with depth information generally

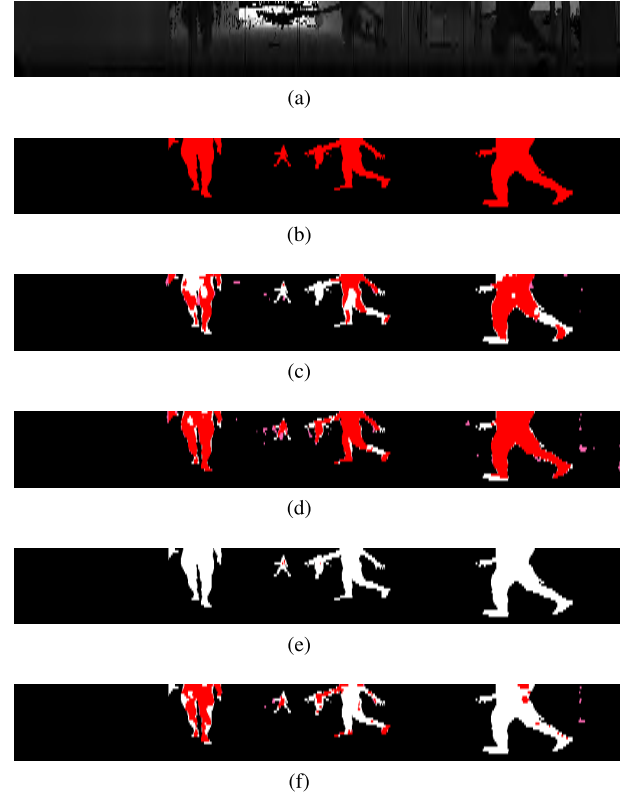


FIGURE 8. Example of segmentation with generated data. In (b), red area denotes human label. From (c) to (g), red and black areas denote true positive and true negative respectively, while pink and white areas denote false positive and false negative, respectively. (a) Example of generated data. (b) Ground truth. (c) Prediction with single FCN. (d) Prediction with gaussian weight. (e) Prediction with simple average. (f) Prediction with trained weight.

resulted in high performance, except for the average accuracy and recall in the generated data. In terms of recall, the FCDN network with depth and xyz information performed the best with the generated data. As shown, if a network exhibits any metric over 0.3 except precision in the test data generated by the proposed pipeline, that network also exhibits a value over 0.3 for that metric with the real data. Similarly, if the metric except precision for the generated test data is less than 0.3, the metric for the real data is also less than 0.3. Therefore, we can conclude that the evaluation with the generated test data helps to predict the effectiveness for the real data.

D. DISTANCE BASED ENSEMBLE LEARNING EVALUATION

Because FCN demonstrates sufficient performance, as shown in Table 4, we employed the FCN for evaluating distance-based ensemble learning. In addition, we generated five different datasets in the human deployment range. “Dataset D_A -to- D_B ” denotes that the human model was

TABLE 6. Result of ensemble networks for human segmentation with whole test data.

		Single FCN [1]	Gaussian weight	Ensemble Trained weight	Simple average
Generated data	Avg. Acc	0.6895	0.7813	0.6419	0.5056
	Recall	0.1555	0.2383	0.1606	0.0304
	Threat score	0.3205	0.4207	0.2486	0.0111
Real data	Avg. Acc	0.6780	0.6912	0.5619	0.5002
	Recall	0.2824	0.3263	0.1336	0.0097
	Threat score	0.3490	0.3656	0.1234	0.0005

TABLE 7. Average accuracy of ensemble networks for human segmentation with each test dataset.

Dataset	Single FCN [1]	Gaussian weight	Ensemble Trained weight	Simple average
01to10	0.7271	0.8351	0.6585	0.5035
05to15	0.6023	0.6654	0.6230	0.5150
10to20	0.5419	0.5529	0.5422	0.5093
15to25	0.5190	0.5283	0.5389	0.5043
20to30	0.5056	0.5182	0.5306	0.5013

deployed in the distance between D_A [m] and D_B [m] from LiDAR. We prepared Datasets 01 to 10, 05 to 15, 10 to 20, 15 to 25, and 20 to 30 that contain 10K training data and 0.1K test data. We set the minimum deployment distance as 1 [m] for preventing collision between the human and LiDAR.

Initially, we examined the performance of the FCN trained by depth in Table 4 toward each dataset. According to Table 5, the performances have degraded significantly from dataset 01to10 to dataset 20to30. It is obvious that a simple training plan with the overall data might be ineffective for part of the data. Consequently, our approach is based on a concept for improving network performance toward each dataset equally.

In addition, we investigated the performance of a specialized network in distance. Subsequently, five networks whose algorithm is the FCN trained by each dataset were prepared. From Figure 7, we observed that the network trained by Dataset 01to10 demonstrated the best performance with Dataset 01to10 and the performance degraded with other Datasets. This tendency is also detected in other networks. Therefore, we conclude that the network will demonstrate peak performance toward a specialized data range and degrade when the range is beyond the specialized area.

After validating the effectiveness of each network, we evaluated the ensemble learning method. For comparison, we used the FCN trained with depth, similar to that shown in Table 4. The evaluation result is described in Table 6 with average accuracy, recall, and threat scores. All generated test data (0.5K) were used for the calculated values shown in Table 6, generated data row. “Single FCN” in Table 6 denotes that the network was trained by all datasets (50K) in distance. We empirically set the value of sigma for the Gaussian distribution in Gaussian weight as 2.5, and the value of mu is the input pixel value converted into meters. Further, we employed the FCN as the weight training network for the trained weight. When we tuned the FCN for weight

training, we set the depth map as the input and used the network output as the weight with five channels. Subsequently, we multiplied each channel of the output and each specialized network output, and treated the multiplied result as the overall output. Therefore, we optimized the weight training network by backpropagation based on the categorical cross-entropy between the ground truth and overall output. Further, we set the weight to the human label as *background pixel number / human pixel number* for each of the scenes. In addition, we investigated the performance of the ensemble networks toward different datasets. As shown in Table 7, the ensemble with Gaussian weight performed the best in datasets 01to10, 05to15, and 10to20. Further, the ensemble with Gaussian weight performed better than the single FCN. For datasets 15to25 and 20to30, the ensemble with trained weight achieved the best score. According to the results, binary weight generally demonstrated high performance, except only for the threat score in the generated data. The Gaussian weight did not demonstrate the best performance; however, it indicated a better score than the single FCN. Regarding the simple average, most pixels were judged as the background because each of specialized network would not be effective to unknown data. We expected the trained weight to demonstrate a better performance than the single FCN; however, the results of the single FCN were better than those of the trained weight except for recall in the generated data. Meanwhile, the trained weight exhibited the best results in datasets 15to25 and 20to30. Therefore, we conclude that the trained weight attempted to improve the performance for the whole dataset; subsequently, the weight network was tuned for a balanced performance. From these results, we can decide whether the ensemble learning based on our generated dataset divided in distance is more effective than only training by the whole data. The example of prediction with the generated data is illustrated in Figure 8 and their point cloud is shown in Figure 9.

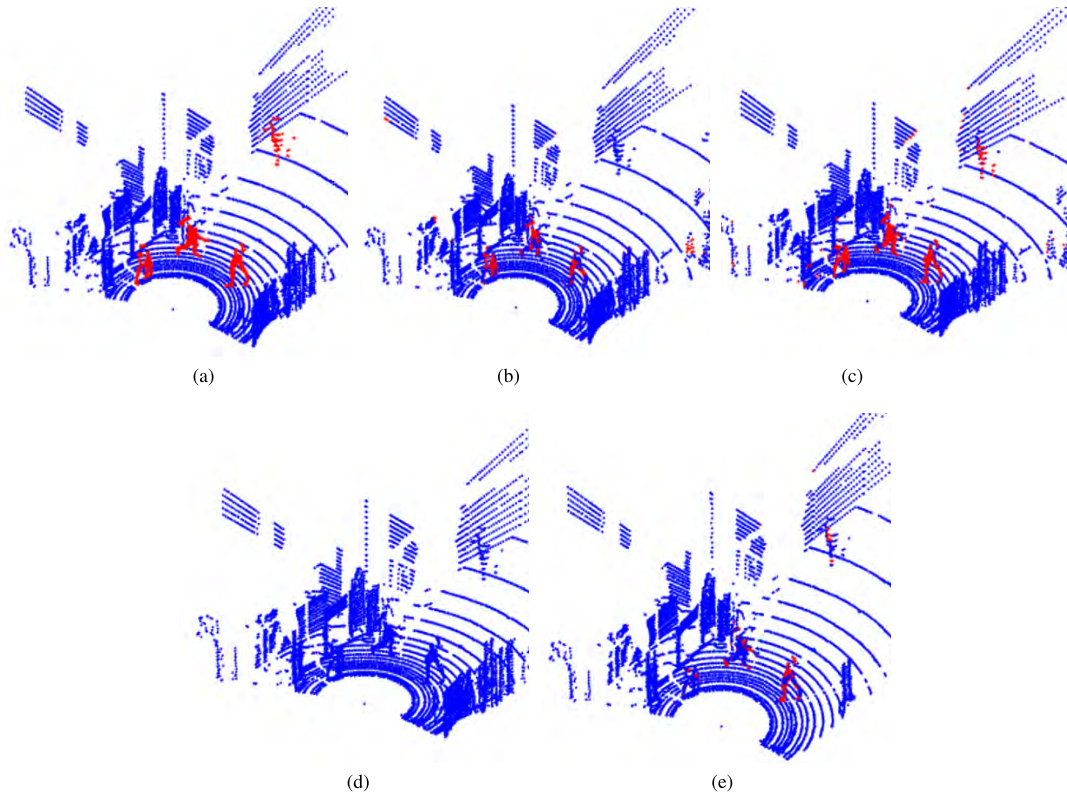


FIGURE 9. Example of segmentation in point cloud. In (a), blue points denote background and red points denote human. In figures (b) to (e), points are colored as network output: blue for background and red for human. (a) Ground truth. (b) Prediction with single FCN. (c) Prediction with gaussian weight. (d) Prediction with simple average. (e) Prediction with trained weight.

VI. CONCLUSION

We herein proposed a fully automated data generation pipeline for human detection using LiDAR initially. With this process, we can generate labeled data easily with any properties for LiDAR. Following the result, we concluded that our generated data improved the training for human detection. We presented 0.1K labeled real datasets and 1M generated datasets with human labels. In the generated datasets, 500K data were generated by human deployment in the range between 1 [m] and 25 [m], whereas the other 500K data were generated in a detailed range. We hope that this dataset will support studies in other fields, e.g., robotics and computer vision. Apart from a data generation pipeline, we also proposed ensemble learning based on generated data divided in distance. This method demonstrated a better performance than simple training using whole data.

We considered four aspects for improving our work. The first aspect pertained to the human model; although we used a confirmed method to produce the human model, it was not entirely representative of the real world. Subsequently, we will attempt to consider clothes, backpacks, and other conditions for accurate simulations. The second aspect pertains to the background. Only data from Miraikan were used in the current investigation; in the future, we will employ other backgrounds for the diversity of data generation. The third aspect was related with the number of human pixels. The number of human-labeled pixels in one depth map

decreased significantly when the distance between LiDAR and human increased. For example, the average number of human-labeled pixels in Dataset 01to10 was 1,401 and that of 20to30 was 26. Therefore, we will attempt to improve the performance even if only few human-labeled points are available. The fourth aspect is the comparison with real data. In this study, networks were only trained with generated data. The comparison of network performance in manually labeled training data and generated training data would be beneficial to this study. However, it would involve considerable costs.

REFERENCES

- [1] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.
- [2] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1915–1929, Aug. 2013.
- [3] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *CoRR*, Jun. 2017. [Online]. Available: <https://arxiv.org/abs/1706.05587>
- [4] B. Wu, A. Wan, X. Yue, and K. Keutzer, "SqueezeSeg: Convolutional neural nets with recurrent CRF for real-time road-object segmentation from 3D LiDAR point cloud," *CoRR*, Oct. 2017. [Online]. Available: <http://arxiv.org/abs/1710.07368>
- [5] C. R. Qi, H. Su, K. Mo, 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.
- [6] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *Proc. IEEE 4th Int. Conf. 3D Vis. (3DV)*, Oct. 2016, pp. 239–248.

- [7] F. Liu, C. Shen, and G. Lin, "Deep convolutional neural fields for depth estimation from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 5162–5170.
- [8] 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.*, 2019.
- [9] I. B. Barbosa, M. Cristani, A. D. Bue, L. Bazzani, and V. Murino, "Re-identification with RGB-D sensors," in *Proc. Eur. Conf. Comput. Vision*, Springer, Oct. 2012, pp. 433–442.
- [10] M. Munaro, A. Basso, A. Fossati, L. Van Gool, and E. Menegatti, "3D reconstruction of freely moving persons for re-identification with a depth sensor," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2014, pp. 4512–4519.
- [11] D. Kastaniotis, I. Theodorakopoulos, C. Theoharatos, G. Economou, and S. Fotopoulos, "A framework for gait-based recognition using Kinect," *Pattern Recognit. Lett.*, vol. 68, pp. 327–335, Dec. 2015.
- [12] 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.
- [13] C. Prenebida, J. Carreira, J. Batista, and U. Nunes, "Pedestrian detection combining RGB and dense LIDAR data," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2014, pp. 4112–4117.
- [14] D. Levi, N. Garnett, E. Fetaya, and I. Herizlyia, "Stixelnet: A deep convolutional network for obstacle detection and road segmentation," in *Proc. BMVC*, Sep. 2015, pp. 1–109.
- [15] M. Johnson-Roberson, C. Barto, R. Mehta, S. N. Sridhar, K. Rosaen, and R. Vasudevan, "Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 746–753.
- [16] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *Proc. Eur. Conf. Comput. Vis.*, Springer, Oct. 2016, pp. 102–118.
- [17] *Miraikan*. Accessed: May 27, 2018. [Online]. Available: <https://www.miraikan.jst.go.jp>
- [18] *LiDAR*. Accessed: May 25, 2018. [Online]. Available: <https://velodynelidar.com/hdl-32e.html>
- [19] M. Mochimaru *et al.*, "Dhaiba: Functional human models to represent variation of shape, motion and subjective assessment," SAE, Warrendale, PA, USA, Tech. Rep., 2006.
- [20] Y. Endo, M. Tada, and M. Mochimaru, "Hand modeling and motion reconstruction for individuals," *Int. J. Autom. Technol.*, vol. 8, no. 3, pp. 1–12, 2014.
- [21] Y. Endo, M. Tada, and M. Mochimaru, "Estimation of arbitrary human models from anthropometric dimensions," in *Proc. Int. Conf. Digit. Hum. Modeling Appl. Health, Safety, Ergonom. Risk Manage.*, Springer, Aug. 2015, pp. 3–14.
- [22] Y. Kobayashi, H. Hobara, T. A. Heldoorn, M. Kouchi, and M. Mochimaru, "Age-independent and age-dependent sex differences in gait pattern determined by principal component analysis," *Gait Posture*, vol. 46, pp. 11–17, May 2016.
- [23] S. Nijima, Y. Sasaki, and H. Mizoguchi, "Autonomous navigation of electric wheelchairs in urban areas on the basis of self-generated 2D drivable maps," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics (AIM)*, Jul. 2018, pp. 1081–1086.
- [24] A. Ohsato, Y. Sasaki, and H. Mizoguchi, "Real-time 6DoF localization for a mobile robot using pre-computed 3D laser likelihood field," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2015, pp. 2359–2364.
- [25] S. Hong, H. Noh, and B. Han, "Decoupled deep neural network for semi-supervised semantic segmentation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1495–1503.
- [26] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput. Assist. Intervent.*, Oct. 2015, pp. 234–241.
- [27] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2650–2658.
- [28] S. Jégou, M. Drozdal, D. Vazquez, A. Romero, and Y. Bengio, "The one hundred layers Tiramisu: Fully convolutional DenseNets for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 1175–1183.
- [29] F. Iandola, M. Moskewicz, S. Karayev, R. Girshick, T. Darrell, and K. Keutzer. (2014). "DenseNet: Implementing efficient Convnet descriptor pyramids." [Online]. Available: <https://arxiv.org/abs/1404.1869>
- [30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, Dec. 2014. [Online]. Available: <https://arxiv.org/abs/1412.6980>
- [31] I. T. Jolliffe and D. B. Stephenson, *Forecast Verification: A Practitioner's Guide in Atmospheric Science*. Hoboken, NJ, USA: Wiley, 2003.



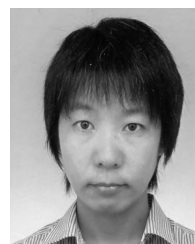
WONJIK KIM received the B.E. degree from the Department of Control and Systems Engineering, Tokyo Institute of Technology, in 2018, where he is currently pursuing the master's degree with the Department of Systems and Control Engineering. He has been a Research Assistant with the Artificial Intelligent Research Center, National Institute of Advanced Industrial Science and Technology, since 2018. His research interests include artificial intelligence, machine learning, robotics, and optimization.



MASAYUKI TANAKA received the bachelor's and master's degrees in control engineering and the Ph.D. degree from the Tokyo Institute of Technology, in 1998, 2000, and 2003, respectively. He was a Software Engineer at Agilent Technology, from 2003 to 2004. He was a Research Scientist with the Tokyo Institute of Technology, from 2004 to 2008. He was an Associated Professor with the Graduate School of Science and Engineering, Tokyo Institute of Technology, from 2008 to 2016. He was a Visiting Scholar with the Department of Psychology, Stanford University, from 2013 to 2014. He was an Associate Professor with the School of Engineering, Tokyo Institute of Technology, from 2016 to 2017. Since 2017, he has been a Senior Researcher with the National Institute of Advanced Industrial Science and Technology.



MASATOSHI OKUTOMI received the B.Eng. degree from the Department of Mathematical Engineering and Information Physics, The University of Tokyo, Tokyo, Japan, in 1981, the M.Eng. degree from the Department of Control Engineering, Tokyo Institute of Technology, Tokyo, Japan, in 1983, and the D.Eng. degree for his research on stereo vision from the Tokyo Institute of Technology. He joined the Canon Research Center, Canon, Inc., Tokyo, Japan, in 1983. From 1987 to 1990, he was a Visiting Research Scientist with the School of Computer Science, Carnegie Mellon University, USA. Since 1994, he has been with the Tokyo Institute of Technology, where he is currently a Professor with the Department of Systems and Control Engineering, School of Engineering.



YOKO SASAKI received the Ph.D. degree in mechanical engineering from the Tokyo University of Science for her research on auditory system for a mobile robot, in 2009. She joined the Digital Human Research Center, National Institute of Advanced Industrial Science and Technology (AIST). Since 2015, she has been a Senior Researcher with the Artificial Intelligent Research Center, AIST, involved in developing autonomous mobile robot and its sensing and navigation technology, especially in human living environment.

• • •