

Received August 7, 2021, accepted August 20, 2021, date of publication August 24, 2021, date of current version August 30, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3107041

Mapless LiDAR Navigation Control of Wheeled Mobile Robots Based on Deep Imitation Learning

CHI-YI TSAI¹, (Senior Member, IEEE), HUMAIRA NISAR², (Senior Member, IEEE),
AND YU-CHEN HU¹

¹Department of Electrical and Computer Engineering, Tamkang University, Tamsui, New Taipei City 251, Taiwan

²Department of Electronic Engineering, Universiti Tunku Abdul Rahman, Kampar, Perak 31900, Malaysia

Corresponding author: Chi-Yi Tsai (chiyi_tsai@mail.tku.edu.tw)

This work was supported by the Ministry of Science and Technology of Taiwan, R.O.C., under Grant MOST 109-2221-E-032-039 and MOST 110-2221-E-032-047.

ABSTRACT This paper addresses the problems related to the mapless navigation control of wheeled mobile robots based on deep learning technology. The traditional navigation control framework is based on a global map of the environment, and its navigation performance depends on the quality of the global map. In this paper, we propose a mapless Light Detection and Ranging (LiDAR) navigation control method for wheeled mobile robots based on deep imitation learning. The proposed method is a data-driven control method that directly uses LiDAR sensors and relative target position for mobile robot navigation control. A deep convolutional neural network (CNN) model is proposed to predict motion control commands of the mobile robot without the requirement of the global map to achieve navigation control of the mobile robot in unknown environments. While collecting the training dataset, we manipulated the mobile robot to avoid obstacles through manual control and recorded the raw data of the LiDAR sensor, the relative target position, and the corresponding motion control commands. Next, we applied a data augmentation method on the recorded samples to increase the number of training samples in the dataset. In the network model design, the proposed CNN model consists of a LiDAR CNN module to extract LiDAR features and a motion prediction module to predict the motion behavior of the robot. In the model training phase, the proposed CNN model learns the mapping between the input sensor data and the desired motion behavior through end-to-end imitation learning. Experimental results show that the proposed mapless LiDAR navigation control method can safely navigate the mobile robot in four unseen environments with an average success rate of 75%. Therefore, the proposed mapless LiDAR navigation control system is effective for robot navigation control in an unknown environment without the global map.

INDEX TERMS Deep imitation learning, end-to-end learning, mapless LiDAR navigation control, behavior cloning.

I. INTRODUCTION

Navigation control is one of the core functions of autonomous mobile robots; it enables the robots to successfully complete the task of motion control and obstacle avoidance in the working environment. Figure 1(a) shows the architecture of the traditional map-based mobile robot navigation control system [1], which is a layered architecture composed of mapping, planning, and tracking control systems. The mapping system is used to construct a global map of the unknown environment. The planning system includes a global planner and a local planner to provide the optimal motion trajectory for the mobile robot to perform navigation tasks such as target approaching or obstacle avoidance. Both global

planner and local planner need to use the global map, sensor information, and the absolute target position to compute the optimal path trajectory and the motion control commands, respectively. According to [2], each system of the traditional map-based navigation framework represents a challenging research topic, and their integration often leads to poor computing performance due to large computational errors. In addition, the navigation performance of the traditional map-based navigation framework depends on the quality of the global map, which is very sensitive to sensor noise. This requirement may limit the ability of the navigation system to manage unknown or dynamic environments [2].

To simplify the architecture of the traditional map-based navigation control system, the development of mapless LiDAR navigation technology for wheeled mobile robots has received more and more attention in recent years.

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

Figure 1(b) shows the system architecture of the existing deep learning-based mapless LiDAR navigation control system, which employs the neural network model to predict the motion control commands for the robot to approach the target while avoiding obstacles. Compared with traditional map-based navigation technology, the existing deep learning-based mapless navigation technology has two major advantages:

(1) It releases the requirement of global map information. This advantage makes the performance of the navigation system independent of the quality of the global map, but it depends on the diversity of the training dataset.

(2) It directly uses the sensor information and the relative target position to predict the corresponding motion control commands. This advantage simplifies the complexity of the navigation system by using the neural network model. However, a potential flaw of this simplification is that the user may need to retrain the entire neural network model if the hardware specification of the mobile platform is changed.

Several mapless navigation control methods have been proposed in the literature. Based on the types of sensors, we divide the existing mapless navigation control methods into two approaches, namely the vision-based approach and the LiDAR-based approach. The vision-based approach uses RGB cameras to capture environmental images and a deep CNN model to learn the mapping between the observed images and the corresponding motion control commands, which can directly drive the robot to perform the navigation control in an unknown environment. For example, Kanayama *et al.* [3] proposed a two-mode mapless visual navigation system for mobile robots based on deep CNN models. The proposed navigation system is composed of two modules, namely the approach module and the positioning module. The approach module detects the landmarks in the environment to determine the velocity which directs the robot to move close to the target. The positioning module consists of a deep CNN model to calculate the moving distance from the current position to the target position. Experimental results show that this visual mapless navigation system is able to navigate the robot in the first-time-seen environments. Similar research can also be found in the references [4] and [5].

On the other hand, the LiDAR-based approach uses LiDAR sensors to obtain point cloud information of the surrounding environment and a deep CNN model to learn the mapping between the observed point clouds and the corresponding navigation control commands without the global map information. Recently, some mapless LiDAR navigation control methods have been published in the literature. Figure 2 shows the existing mapless LiDAR navigation control methods, which can be divided into two categories. The first category is based on data clustering technology to deal with scene segmentation problems. The second category is based on deep learning technology, which can be further divided into training methods based on reinforcement learning [6] and imitation learning [7]. The former can be used to solve

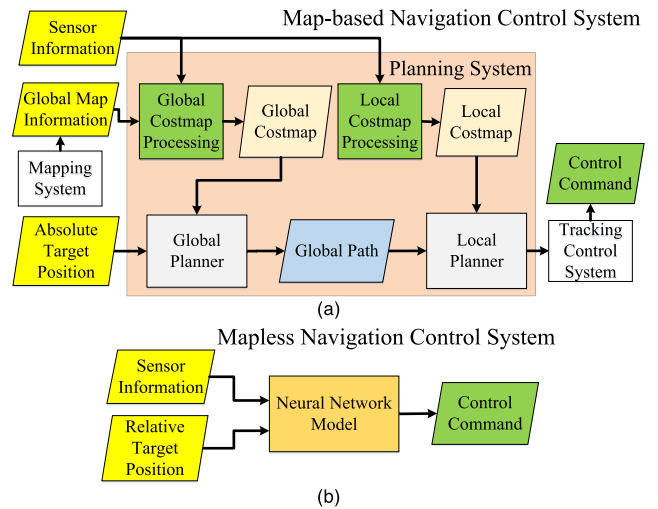


FIGURE 1. System architecture of (a) the traditional map-based navigation control system, which is a layered architecture composed of mapping, planning, and tracking control systems, and (b) the existing deep learning-based mapless navigation control system.

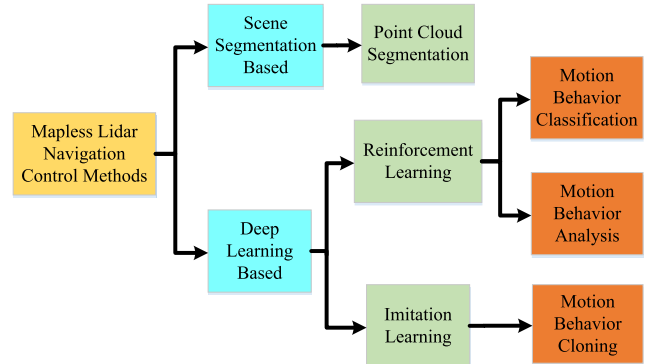


FIGURE 2. Categories of the existing mapless LiDAR navigation control methods.

the problem of motion behavior classification and motion behavior analysis; the latter is useful in dealing with the problem of motion behavior cloning.

The purpose of the mapless navigation method based on point cloud segmentation processing is to use the geometric surface features of the LiDAR point cloud to segment the road area in the scene and to plan the motion trajectory of the area. For example, Ort *et al.* proposed a mapless driving framework for autonomous vehicle navigation, which combines sparse topological maps for global navigation with a sensor-based perception system for local navigation [8]. The sensor-based perception system uses a 3D laser scanner mounted on the vehicle for planning safe trajectories in the local frame based on a road segmentation algorithm, which aims to obtain the road boundary points in the sensor swath of the vehicle. Experimental results show that this framework allows the vehicle to navigate road networks without detailed prior maps in a rural unknown environment. Similar research can also be found in the reference [9].

Due to the rise of deep learning technology in recent years, its applications have become more diverse. With deep

learning technology, mapless navigation control methods can achieve more robust navigation results. The main purpose of the reinforcement learning method is to determine the best action based on environmental feedback to maximize the overall reward. The classification problem of navigation control commands is to classify the output commands of the navigation system into several fixed categories according to the environmental conditions. The motion behavior classification method based on reinforcement learning can be applied to this classification problem to train the optimal classifier that can predict the correct motion command category based on the sensing information. For example, Qiang *et al.* [10] proposed a model-free mapless navigation method for mobile robots based on reinforcement learning. They designed an end-to-end navigation model that uses LiDAR sensor data as input and one of the five possible motion actions as output. After training the model through reinforcement learning, the mobile robot can safely reach the navigation target in an unknown environment without prior maps. References [11] and [12] also proposed similar research on this topic. A good review of reinforcement learning-based navigation techniques compared to traditional map-based navigation methods can be found in [2].

The motion behavior analysis method based on reinforcement learning is a regression analysis between the output motion control commands of the navigation system and the input environment perception and target position data. For example, Tai *et al.* [13] proposed a learning-based mapless motion planner, which takes the laser scan data and the relative target position as input and the continuous navigation commands as output. They applied an asynchronous deep reinforcement learning method to train an end-to-end mapless motion planner in the virtual environment. Experimental results show that the trained planner can be directly applied in unseen virtual and real environments. Pfeiffer *et al.* [14] proposed a target-driven mapless navigation policy based on a combination of imitation and reinforcement learning. They designed a neural network model, which uses the sensor measurements and the relative target position as input and the required control commands as output. The model was trained in a sequential fashion by using the result from imitation learning to initialize the reinforcement learning method. Their results show that target-driven demonstrations through imitation learning significantly improve the exploration during reinforcement learning.

Imitation learning means learning from the demonstration so that the network model can imitate the behavior of the demonstration. In the training process, the end-to-end learning method is used to make the neural network model learn the demonstration behavior given by the expert strategy, so that the predictive behavior of the model can become similar to the behavior of the expert strategy. Hence, the purpose of the navigation control method based on imitation learning is to learn the best mapping between the output motion action of the navigation controller and the input LiDAR sensing data and the relative target position. For example,

Khaksar *et al.* [15] proposed a mapless neuro-fuzzy motion planner learned from a virtual experience model, which generates enough training data to train the adaptive neuro-fuzzy inference system. Pfeiffer *et al.* [16] proposed a data-driven motion planning approach based on a feature-based maximum entropy model, which is trained to predict the joint navigation behavior of heterogeneous groups of agents based on the demonstration of human-human interactions. In [17], Pfeiffer *et al.* proposed a target-oriented end-to-end navigation model that can learn the complex mapping from raw 2D-laser range findings and a target position to the required motion commands for a robotic platform. Hamandi *et al.* [18] proposed a human-aware navigation model in which the robot mimics humans to navigate safely in the crowds. The proposed navigation model also processes LiDAR scans as input to navigate the robot to the target position. Hence, the main advantage of imitation learning is that it is very useful when an expert system is available to provide the desired behavior for the network model to learn the optimal policy by imitating the behavior of the expert. However, the efficiency of imitation learning depends on the diversity of behaviors of the expert system to cover various situations. Therefore, to find ways to increase the behavior diversity of the expert system is an important issue in the application of imitation learning.

In this paper, we propose a mapless LiDAR navigation control system based on an end-to-end CNN model, which is trained through imitation learning. The proposed navigation system directly uses the LiDAR sensor data and the relative target position as input, and outputs the motion control commands required by the mobile robot. In the training phase, we use a regression method to train the model end-to-end to learn the mapping between input data and output commands and apply the trained model to navigation control. Moreover, we also propose a novel data augmentation method to increase the diversity of training samples to improve the efficiency of imitation learning. The contributions of this paper are summarized in the following three points:

- (1) A new CNN-based navigation control model is proposed, which can obtain better training results by increasing the dimension of input LiDAR data.

- (2) We propose a data augmentation method that can increase the diversity of training samples so that the trained model can operate in an unknown environment.

- (3) By collecting training data in the real environment and using the proposed data augmentation method to increase the size of the training dataset, the trained model can obtain good navigation control performance in the real environment.

Experimental results show that the proposed LiDAR navigation control system based on end-to-end imitation learning achieves good navigation control performance of the wheeled mobile robot. In a known environment, the proposed navigation control system can safely control the mobile robot to avoid obstacles and reach the target position. When operating in an unknown environment, the proposed navigation control system can reach the target position with an average success

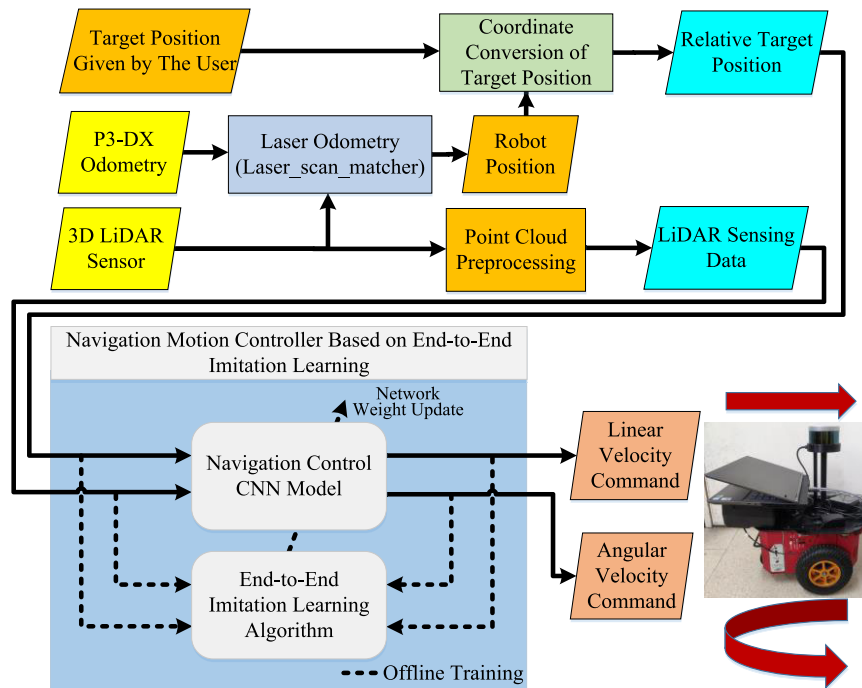


FIGURE 3. Architecture of the proposed mapless LiDAR navigation control system for wheeled mobile robots based on end-to-end imitation learning.

rate of 75%. Therefore, the proposed mapless navigation control system is practically applicable in real environments.

The rest of this paper is organized as follows. Section II describes the architecture of the proposed mapless navigation control system. Section III introduces the LiDAR signal CNN model and the motion prediction neural network model proposed in this paper. Section IV presents the collected training dataset and the proposed data augmentation method. Section V reports the experimental results to evaluate the navigation control performance of the proposed method in different environments. Section VI concludes this work and provides recommendations for future work.

II. SYSTEM ARCHITECTURE

Figure 3 shows the architecture of the proposed mapless LiDAR navigation control system based on end-to-end imitation learning, in which the wheeled mobile robot is equipped with a 3D laser scanner to capture the point cloud of the surrounding environment. When the 3D point cloud of the surrounding environment, has been received, the proposed mapless navigation control system performs point cloud preprocessing on the 3D point cloud information to obtain the LiDAR sensing data. In addition, the robot motion information is obtained from the odometry of the mobile platform. Next, we use the *Laser_scan_matcher* [19] provided in the Robot Operating System (ROS) to obtain the robot position information by fusing the LiDAR point cloud and the robot motion information. When the user manually gives the target position information, the proposed navigation control system performs/initiates a coordinate conversion process on the

given target position information to obtain the target position information relative to the current robot position.

As shown in Figure 3, the proposed navigation motion controller comprises of a navigation control CNN model and an end-to-end imitation learning algorithm. When performing navigation tasks, the proposed navigation control CNN model takes the extracted LiDAR sensing data and the relative target position as input, and outputs linear and angular velocity commands to the mobile platform. To learn the navigation control behavior provided by the expert strategy, we applied an end-to-end imitation learning algorithm to train the proposed navigation control CNN model offline. The detailed training process is described in Section III.

Figure 4 shows the architecture of the proposed navigation control neural network model, which is divided into two models. The first one is a LiDAR signal CNN model that includes a feature extraction module for extracting features of LiDAR sensing data, and a residual network module for generating key point cloud feature maps. The second one is a motion prediction neural network model for motion command prediction. In the design of this model, we preprocess the relative target position through a fully connected (FC) layer and input the result into the data fusion module together with the point cloud feature maps. Finally, we use the motion prediction module to predict motion control commands. Section III also introduces the detailed design of the proposed navigation control CNN model for motion command prediction.

Remark 1: In the proposed navigation system, the robot position is obtained from the *Laser_scan_matcher* node in ROS, which is a frame-to-frame laser odometry based on

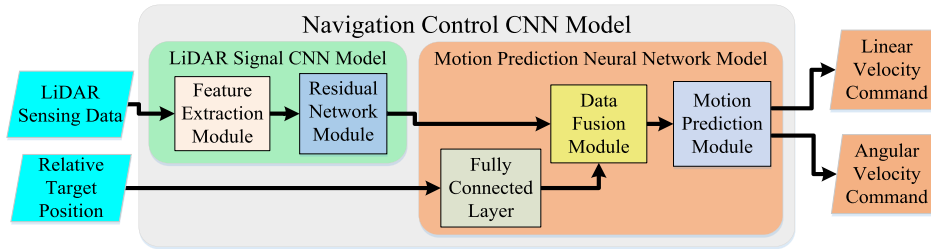


FIGURE 4. Architecture of the proposed navigation control CNN model.

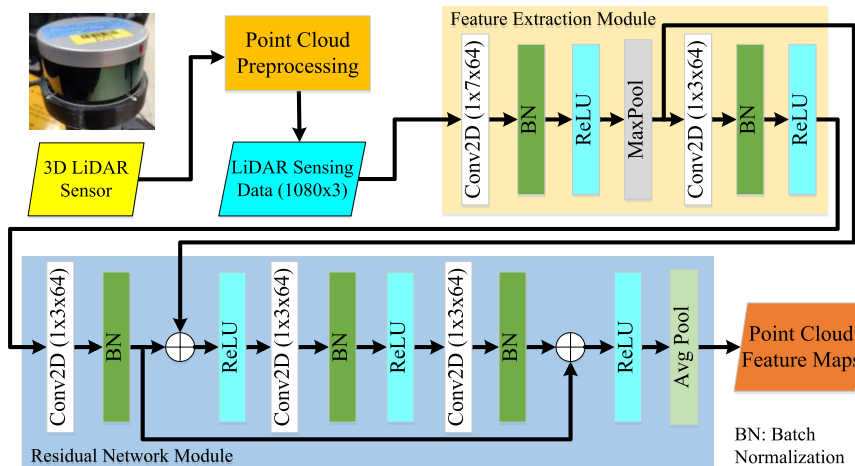


FIGURE 5. Detailed architecture of the LiDAR signal CNN model, which is implemented based on the design in [15].

laser scan match between consecutive laser scans. More specifically, the laser odometry used in this node applies a point-to-line iterative closest point algorithm [20] on two consecutive laser scans to estimate the motion of the robot while updating the robot position relative to the starting position. The Laser_scan_matcher node can operate with wheel odometry to speed up the scan match process by providing a guess for the current position of the sensor. Therefore, in an indoor environment suitable for scan match processing, this node can effectively solve the problem of accumulated odometry errors. However, in an open wide environment, this node may reduce the accuracy of robot localization due to the increase in the scan match error. Interested readers can refer to [19] and [20] for more detailed information.

III. THE PROPOSED NAVIGATION CONTROL CNN MODEL

In this section, we introduce the design of the proposed navigation control CNN model for mapless navigation of wheeled mobile robots.

A. LIDAR SIGNAL CNN MODEL

Figure 5 shows the detailed architecture of the proposed LiDAR signal CNN model, which is inspired from the design presented in [17]. The authors in [17] proposed a CNN model based on the ResNet architecture [21] to process laser scans obtained from a 2D LiDAR sensor, which provides a vector of distance values of the surrounding environment. In contrast, the proposed navigation control system uses a 3D LiDAR

sensor to capture a distance matrix of the surrounding environment and converts this distance matrix into a 3D point cloud matrix of the regional environment. Next, a point cloud preprocessing module is adopted to reduce the dimension of the 3D point cloud matrix that outputs a 1080-by-3 point cloud matrix, which is defined as the LiDAR sensing data. This data is the input data of the proposed CNN model.

As shown in Figure 5, the proposed CNN model is divided into a feature extraction module and a residual network (ResNet) module. The former is built by using two convolutional blocks and one maximum pooling (MaxPool) layer. Each convolutional block is composed of a 2D convolutional layer with ReLU [22] activation function and batch normalization (BN) layer for data normalization. The latter is designed and implemented based on the ResNet architecture to reduce the complexity of training and increase the speed of learning. Finally, we use an average pooling (AvgPool) layer to output the point cloud feature maps to complete the point cloud feature extraction process.

Figure 6 shows the processing steps of the proposed point cloud preprocessing, which is divided into two steps, namely point cloud extraction, and point cloud library (PCL) filtering. We used ROS to connect the LiDAR sensor and read laser scans from the LiDAR through RJ-45 network cable. In the point cloud extraction step, after receiving the laser scans from the LiDAR sensor node in ROS, we convert the received data into 3D point clouds, and then convert these point clouds into the PCL file format [23]. In the PCL filtering step,

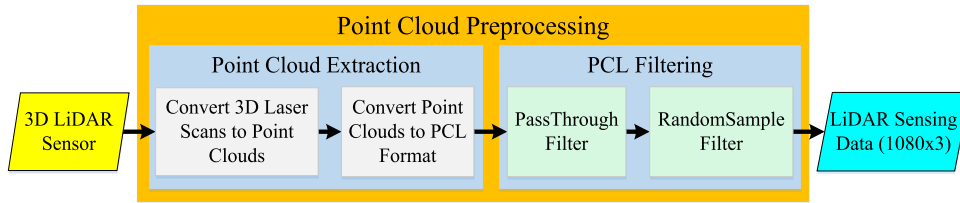


FIGURE 6. Processing steps of the proposed point cloud preprocessing.

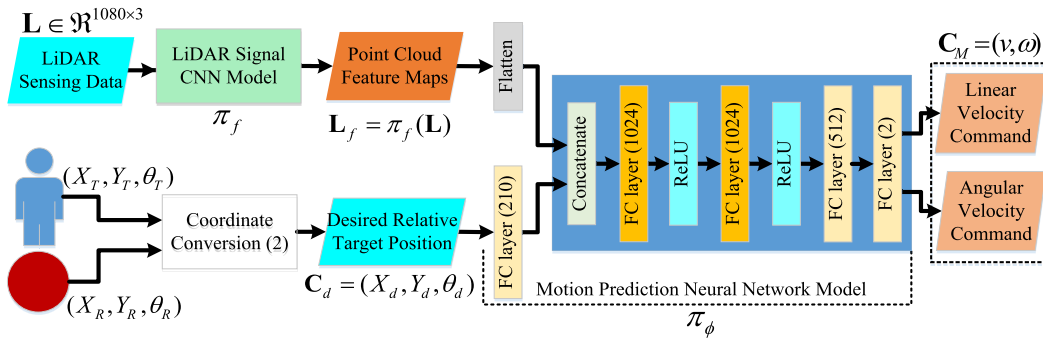


FIGURE 7. Detailed architecture of the proposed motion prediction neural network model.

we perform two filtering processes. The first one uses the PassThrough filter to set the size of a 3D cuboid to filter the point clouds outside the cuboid. The purpose of this process is to filter out the point clouds that are far away from the robot to retain the more important point cloud data. However, after processing by the PassThrough filter, the number of remaining point clouds is not fixed. Therefore, the second one uses the RandomSample filter to filter the remaining point clouds and output a fixed number of point clouds, which is the input data of the LiDAR signal CNN model.

B. MOTION PREDICTION NEURAL NETWORK MODEL

Figure 7 shows the architecture of the proposed motion prediction neural network model, which is modified from the neural network model presented in [17]. The proposed neural network model is built on five FC layers with dimensions (210, 1024, 1024, 512, 2). The first FC layer uses the linear activation function to increase the dimension of the input relative target position, which is calculated based on the coordinate conversion of the given target position with respect to the current robot location. After concatenating the high-dimensional target position information and the point cloud feature vector to form a hybrid feature vector, two FC layers with the ReLU activation function are used to perform nonlinear transformation and dimensionality reduction operations on the hybrid feature vector. The last two FC layers use the linear activation function to perform the mapping between the low-dimensional hybrid feature vector and the output linear and angular velocity commands.

Let \mathbf{L} and \mathbf{L}_f denote the LiDAR sensing data and the corresponding point cloud feature maps, respectively. Assume that the LiDAR signal CNN model is represented by a nonlinear mapping function π_f . Then, the relationship between \mathbf{L} and

\mathbf{L}_f can be expressed as $\mathbf{L}_f = \pi_f(\mathbf{L})$, where the dimension of \mathbf{L} is fixed to 1080×3 . Let $\mathbf{C}_d = (X_d, Y_d, \theta_d)$ denote the desired relative target position given by the user, where (X_d, Y_d) is the plane coordinate of the desired relative target point, and θ_d is the desired relative orientation angle; $\mathbf{C}_M = (v, \omega)$ the predicted motion control command, where v is the linear velocity command, and ω is the angular velocity command. Assume that the proposed motion prediction neural network model is formulated by a nonlinear mapping function π_ϕ . Then, the relationship between the input data \mathbf{L}_f , \mathbf{C}_d and the output command \mathbf{C}_M can be described by the following formula

$$\mathbf{C}_M = \pi_\phi(\mathbf{L}_f, \mathbf{C}_d), \tag{1}$$

where the input data \mathbf{C}_d is obtained from the user-defined target position (X_T, Y_T, θ_T) and the current robot position (X_R, Y_R, θ_R) . Note that the current robot position can be obtained from a mobile robot localization algorithm, such as the Laser_scan_matcher method used in the proposed navigation control system. Then, the desired relative target position \mathbf{C}_d can be calculated by the following coordinate conversion formula

$$\begin{bmatrix} X_d \\ Y_d \\ \theta_d \end{bmatrix} = \begin{bmatrix} \cos \theta_R & \sin \theta_R & 0 \\ -\sin \theta_R & \cos \theta_R & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_T - X_R \\ Y_T - Y_R \\ \theta_T - \theta_R \end{bmatrix}, \tag{2}$$

where the current robot position (X_R, Y_R, θ_R) is continuously updated during the navigation task. Note that the physical meaning of the coordinate conversion formula (2) is the transformation of the target position from the world frame to the robot base frame. Therefore, when the desired relative target position converges to zero, it indicates that the robot is already at the desired target position.

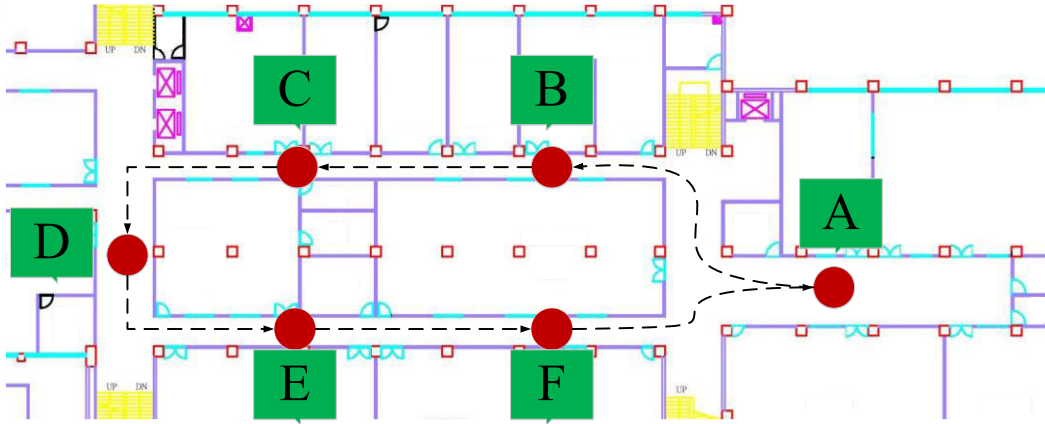


FIGURE 8. Evolution of MAE with respect to different dimensions of the input FC layer of the relative target position.

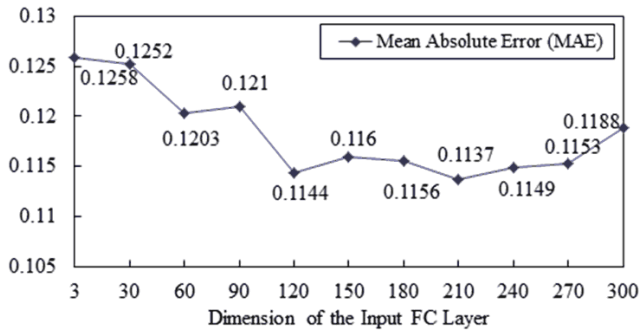


FIGURE 9. Experimental field for the preliminary training dataset collection.

In the model training phase, we first constructed a large-scale training dataset. Let $\Omega_{train} = \{(\mathbf{L}_i, \mathbf{C}_{di}, \hat{\mathbf{C}}_{Mi})\}_{i=1 \sim N_{train}}$ and $\Omega_{test} = \{(\mathbf{L}_j, \mathbf{C}_{dj}, \hat{\mathbf{C}}_{Mj})\}_{j=1 \sim N_{test}}$ denote, respectively, the constructed training and testing dataset, where \mathbf{L}_i is the i -th LiDAR sensing data, \mathbf{C}_{di} the i -th desired relative target position, and $\hat{\mathbf{C}}_{Mi}$ the i -th desired motion control commands. In this study, we collected a total of 250,000 training samples, of which 200,000 samples were used as the training dataset ($N_{train} = 200,000$), and the other 50,000 samples were used as the testing dataset ($N_{test} = 50,000$). During the model training process, we randomly divided the training dataset into several batches to train the proposed navigation control CNN model. We used the Mean-Absolute-Error (MAE) as the loss function for training

$$L_{MAE}(\Omega_{batch}, \pi_\phi, \pi_f) = \frac{1}{N_{batch}} \sum_{k=1}^{N_{batch}} \left| \hat{\mathbf{C}}_{Mk} - \pi_\phi(\mathbf{C}_{dk}, \pi_f(\mathbf{L}_k)) \right| \quad (3)$$

where $\Omega_{batch} = \{(\mathbf{L}_k, \mathbf{C}_{dk}, \hat{\mathbf{C}}_{Mk})\}_{k=1 \sim N_{batch}} \subset \Omega_{train}$ is a batch of the training dataset, and $N_{batch} \ll N_{train}$ is the batch size. Based on the MAE loss function (3), we optimized both the LiDAR signal CNN model and the motion prediction neural network model such that

$$\hat{\pi}_\phi, \hat{\pi}_f = \arg \min_{\pi_\phi, \pi_f} L_{MAE}(\Omega_{batch}, \pi_\phi, \pi_f). \quad (4)$$

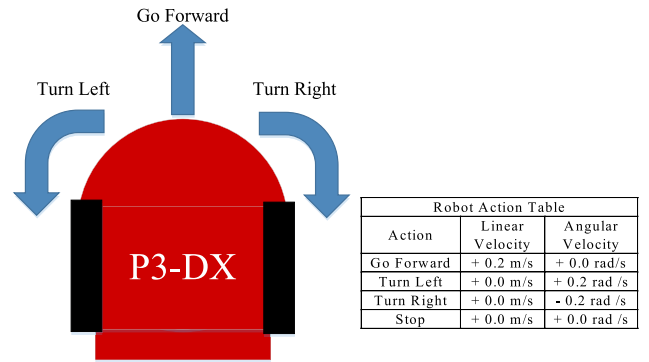


FIGURE 10. Four behaviors and the corresponding velocity control commands of the wheeled mobile robot defined in the training dataset.

Finally, the optimal navigation control CNN model is given by $\hat{\pi}_\phi(\mathbf{C}_d, \hat{\pi}_f(\mathbf{L}))$.

In the model testing phase, we evaluated the performance of the optimized navigation control CNN model based on the MAE loss function $L_{MAE}(\Omega_{test}, \hat{\pi}_\phi, \hat{\pi}_f)$ calculated from the testing dataset Ω_{test} . We observed from the experimental results that the MAE loss value of the optimized navigation control CNN model tested on the testing dataset is $L_{MAE}(\Omega_{test}, \hat{\pi}_\phi, \hat{\pi}_f) = 0.1137$.

C. INFLUENCE OF DIFFERENT DIMENSIONS OF THE INPUT FC LAYER OF THE RELATIVE TARGET POSITION ON MODEL TRAINING

In this section, we study the influence of the different dimensions of the input FC layer of the relative target position on the training of the proposed navigation control CNN model. Figure 8 shows the evolution of MAE with respect to different dimensions of this FC layer from 3 to 300. It can be seen from Figure 8 that the use of a 3-dimensional FC layer results in the worst performance of model training. In contrast, when the dimension of the input FC layer is increased to 210, the best performance of model training can be obtained. Therefore, we use the 210-dimensional input FC layer as the default setting of the proposed motion prediction neural network model.

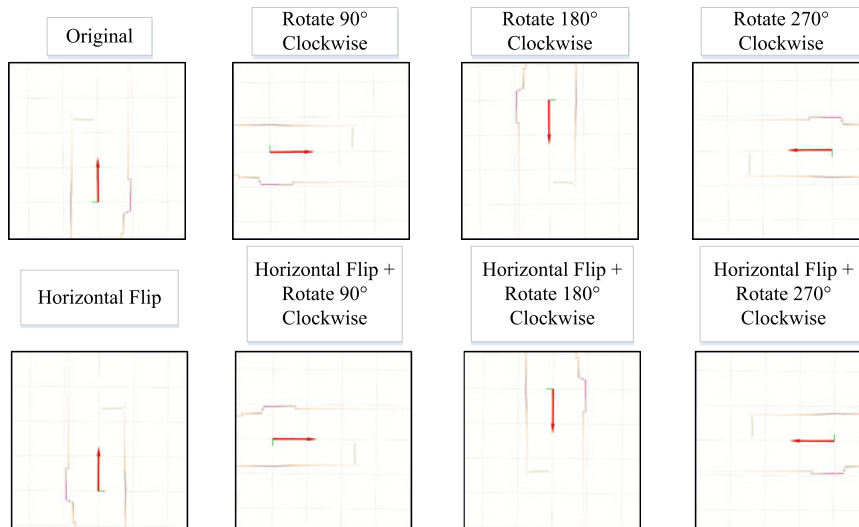


FIGURE 11. Some examples of affine transformations used in the proposed data augmentation method.

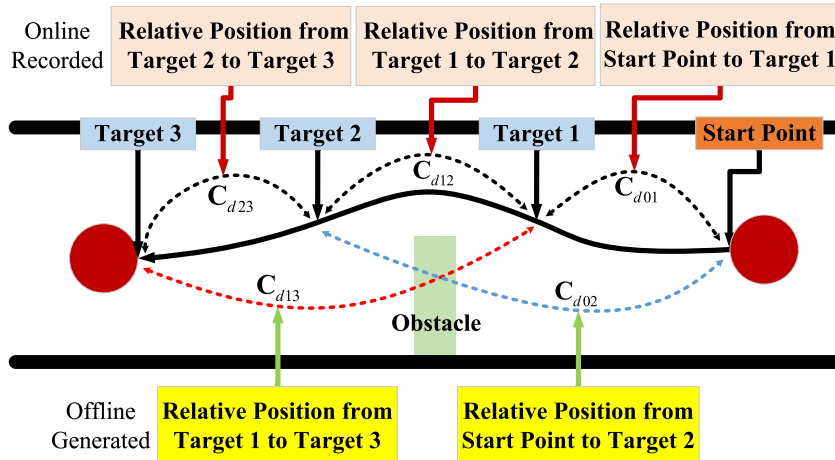


FIGURE 12. Graphic illustration of the data merging processing approach used in the proposed data augmentation method.

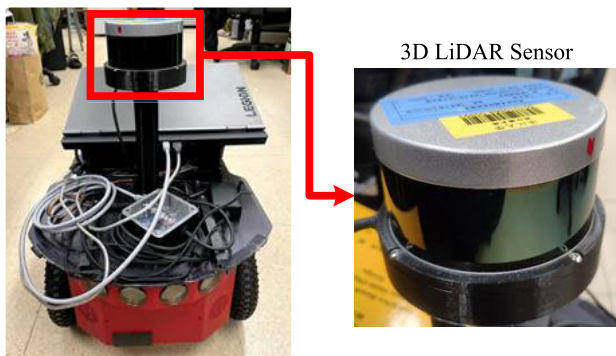


FIGURE 13. The wheeled mobile robot (P3-DX) equipped with a 3D LiDAR sensor (Velodyne-16) used in the experiments.

Remark 2: The difference between the proposed CNN model and the original model in reference [17] is twofold. First, the original model is only suitable for 2D LiDAR,

TABLE 1. Changes in the number of training samples before and after using the proposed data augmentation method.

Data Augmentation Method		Number of Training Samples	
		Before Augmentation	After Augmentation
Stage 1	Affine transformation	15,625	125,000
Stage 2	Data merging	125,000	250,000

whereas the proposed model is expanded to be used for 3D LiDAR by integrating a point cloud preprocessing module. Second, the original model does not have an input FC layer for the relative target position. On the contrary, the proposed model adds an input FC layer to improve the learning accuracy of the CNN model, as shown in Figure 8. Hence, the navigation performance of the proposed model can be improved.

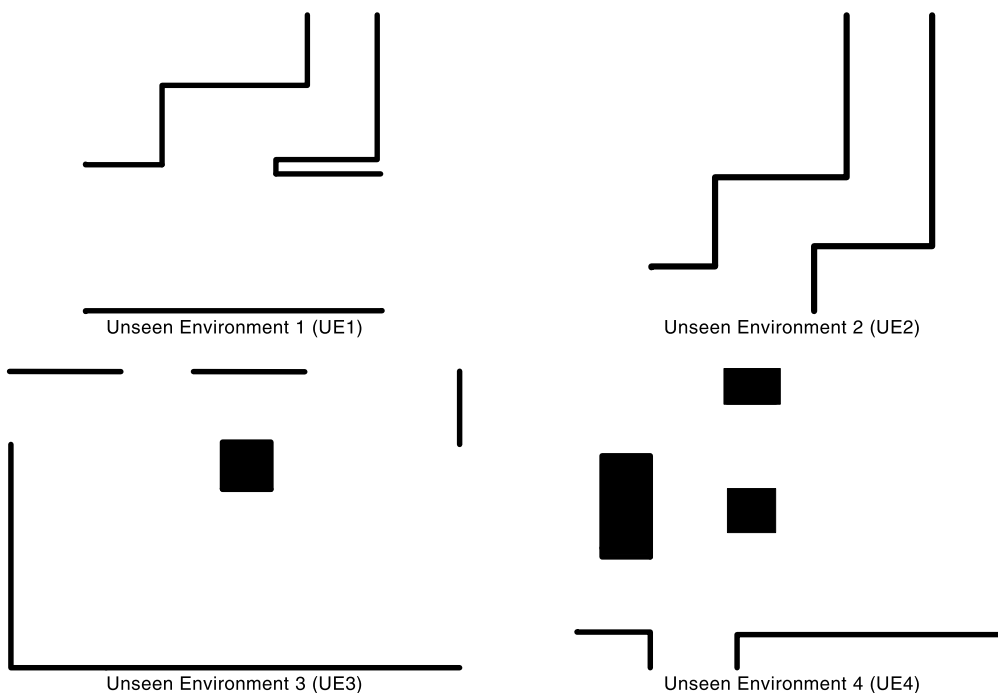


FIGURE 14. Four unseen environments used in the experiments to evaluate the performance of the proposed navigation control CNN model.

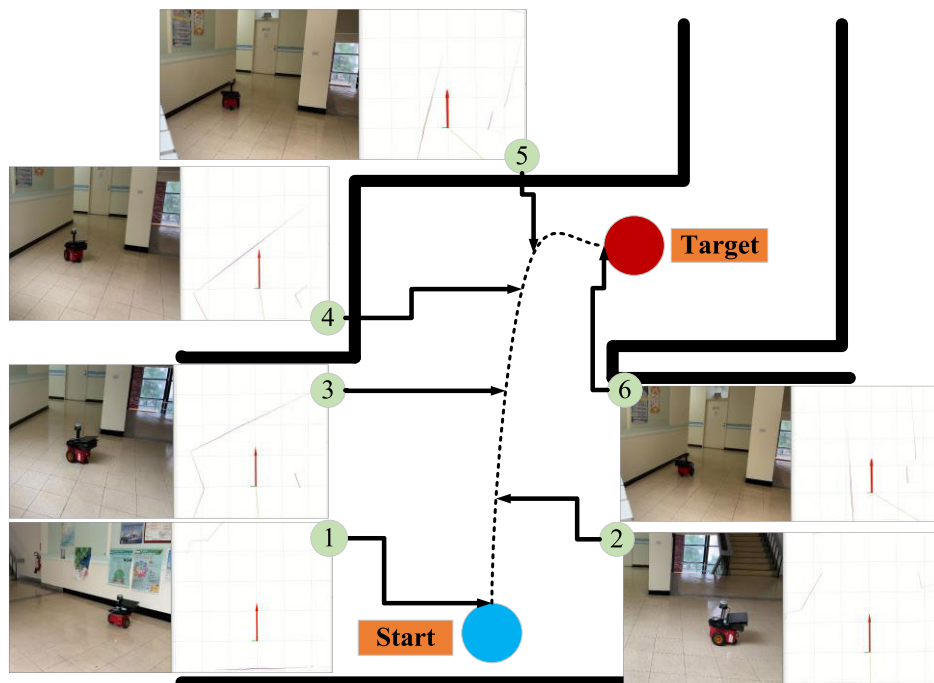


FIGURE 15. Experimental result of the proposed mapless navigation control system in the UE1.

IV. PREPARATION OF TRAINING DATASET

This section presents the collection of the training dataset and the proposed data augmentation methods to increase the number of training samples.

A. COLLECTION OF THE TRAINING DATASET

In this study, we first collected a preliminary training dataset by manually controlling a wheeled mobile robot (Pioneer P3-DX [24]) equipped with a 3D LiDAR sensor

TABLE 2. Experimental results of the five experiments of the robot mapless navigation in the UE1.

Experiments in UE 1	Desired Target Position	Final Robot Position	Error Distance	Navigation Distance	Succeeded/Failed
1st (Figure 15)	(5.0,-2.5)	(4.9915,-2.4030)	0.09 m	6.4116 m	Succeeded
2nd	(6.0,2.5)	X	X	X	Failed
3rd	(10.0,-0.5)	(10.0871,-0.3209)	0.19 m	10.2061 m	Succeeded
4th	(2.5,-3.0)	(2.5408,-3.0762)	0.08 m	5.2713 m	Succeeded
5th	(3.0,-3.0)	(2.9353,-3.0646)	0.09 m	5.3807 m	Succeeded
Average Success Rate					80%

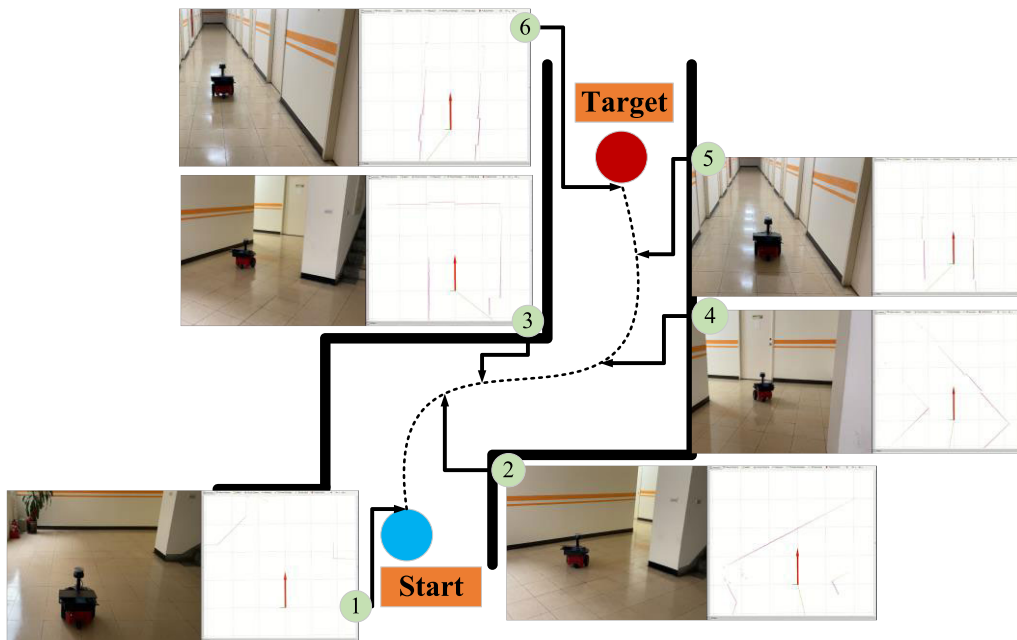


FIGURE 16. Experimental result of the proposed mapless navigation control system in the UE2.

TABLE 3. Experimental results of the five experiments of the robot mapless navigation in the UE2.

Experiments in UE 2	Desired Target Position	Final Robot Position	Error Distance	Navigation Distance	Succeeded/Failed
1st (Figure 16)	(7.0,-5.0)	(6.9917,-5.0131)	0.13 m	11.7015 m	Succeeded
2nd	(8.0,4.0)	(8.0853,3.9712)	0.09 m	11.9422 m	Succeeded
3rd	(8.5,-8.0)	X	X	X	Failed
4th	(7.5,7.0)	(7.5496,6.9596)	0.06 m	13.0101 m	Succeeded
5th	(4.0,-4.0)	(3.9404,-3.9267)	0.09 m	7.4032 m	Succeeded
Average Success Rate					80%

(Velodyne VLP-16 [25]). The robot navigated in an experimental field. Figure 9 shows the experimental field for the preliminary training dataset collection. A notebook computer was placed on the mobile robot to collect LiDAR sensor information, robot motion control commands, and the relative target positions.

While collecting the training data, we placed the robot in the experimental field and manually controlled the robot to navigate among six specific locations (the red dots from A to F), as shown in Figure 9. In the manual navigation control task, we recorded the surrounding point cloud information, the relative target positions, and all the control commands of the navigation process as expert strategies. The starting point of the navigation task starts from point A, passes

through point B to point F in order (sequence), and finally returns to point A. When performing navigation control, the motion trajectory of the robot is maintained at the midpoint of the movable space. If the robot encounters an obstacle, we controlled the robot to perform obstacle avoidance actions.

In order to control the motion of the mobile robot, we used the ROS system to connect the robot through the RS232 interface. In this work, we defined four control behaviors to navigate the mobile robot in the experimental field. Figure 10 shows these four behaviors and the corresponding velocity control commands of the wheeled mobile robot. These commands are defined as the output ground truth in the training dataset.

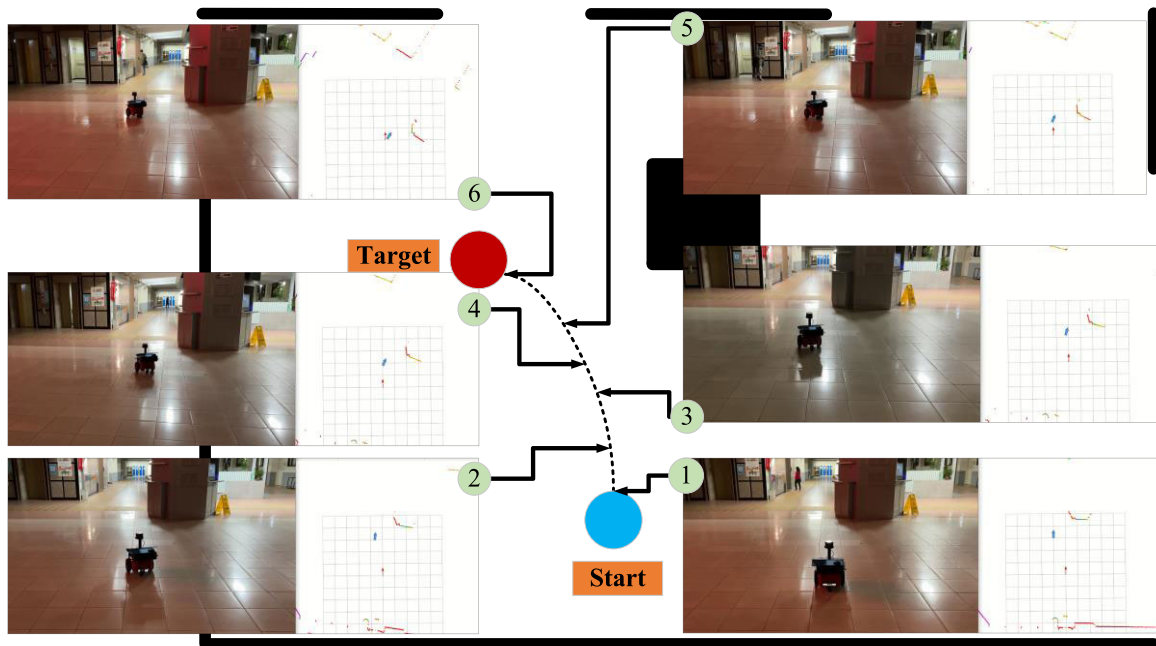


FIGURE 17. Experimental result of the proposed mapless navigation control system in the UE3.

TABLE 4. Experimental results of the five experiments of the robot mapless navigation in the UE3.

Experiments in UE 3	Desired Target Position	Final Robot Position	Error Distance	Navigation Distance	Succeeded/Failed
1st (Figure 17)	(3.0,1.0)	(2.8746,0.9403)	0.14 m	3.1009 m	Succeeded
2nd	(3.0,-1.0)	X	X	X	Failed
3rd	(4.5,0.5)	(4.0537,0.6337)	0.14 m	4.6529 m	Succeeded
4th	(8.0,-7.0)	X	X	X	Failed
5th	(5.0,-0.5)	(4.9210,-0.6808)	0.19 m	5.0476 m	Succeeded
Average Success Rate					60%

B. THE PROPOSED DATA AUGMENTATION METHOD

After obtaining the preliminary training dataset recorded in the real environment, we performed a data augmentation processing on the recorded training samples to form a large-scale training dataset. First, we applied multiple affine transformations, such as rotating 90 degrees clockwise, flipping 180 degrees horizontally, flipping 180 degrees vertically, etc., to the recorded LiDAR point cloud data to obtain multiple sets of new data. Figure 11 shows some examples of affine transformations used in the proposed data augmentation method. Note that the output ground truth of each augmented new data is the same as that of the original data.

Second, we propose a new data augmentation method based on a data merging process to increase the training samples by merging multiple consecutive training data into a new training data. Figure 12 illustrates the concept of the proposed data augmentation method. Let C_{dij} denote the relative position from target point i to target point j . In Figure 12, there are three target points to guide the mobile robot to avoid an obstacle in the corridor. When we sequentially recorded the training samples of the robot navigation from C_{d01} to C_{d23} , we perform the data merging process on the recorded data to generate new training samples between two different

target points offline, such as the new robot navigation path C_{d02} and C_{d13} . By doing so, we can efficiently increase the navigation distance of the robot in the training sample so that the diversity of the augmented training dataset can be improved. Table 1 records the changes in the number of training samples before and after using the proposed data augmentation method. At the beginning, we recorded 15,625 training samples in the preliminary training dataset through manual navigation control of the mobile robot. After applying multiple affine transformations to the preliminary training samples, the number of the training samples increased to 125,000 samples. Finally, a large-scale training dataset with 250,000 samples was created by applying the data merging process on the augmented 125,000 samples.

V. EXPERIMENTAL RESULTS

We trained the proposed navigation control CNN model using Tensorflow Keras framework running on a desktop computer equipped with an Nvidia RTX 2080Ti GPU, and then implemented it on a laptop to control a wheeled mobile robot equipped with a 3D LiDAR sensor. Figure 13 shows the wheeled mobile robot and the 3D LiDAR sensor used in the experiment. We used the ROS framework to control the

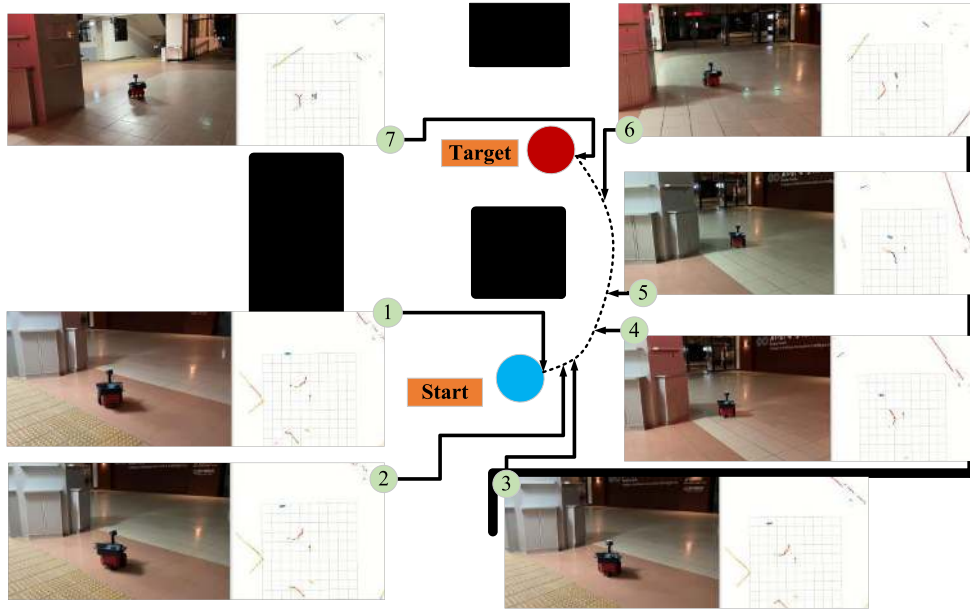


FIGURE 18. Experimental result of the proposed mapless navigation control system in the UE4.

TABLE 5. Experimental results of the five experiments of the robot mapless navigation in the UE4.

Experiments in UE 4	Desired Target Position	Final Robot Position	Error Distance	Navigation Distance	Succeeded/Failed
1st (Figure 18)	(5.0,2.0)	(4.7285,1.8776)	0.29 m	6.6683 m	Succeeded
2nd	(6.0,-1.0)	X	X	X	Failed
3rd	(7.0,3.0)	(6.8836,2.9120)	0.17 m	8.2839 m	Succeeded
4th	(5.0,3.0)	(4.8105,2.8997)	0.21 m	6.0166 m	Succeeded
5th	(4.0,3.0)	(4.2873,3.1980)	0.33 m	7.2868 m	Succeeded
Average Success Rate					80%

mobile robot and the LiDAR sensor to test and evaluate the performance of the proposed navigation control method.

Figure 14 shows four unseen environments used to evaluate the performance of mobile robot navigation control based on the proposed navigation control CNN model. These unseen environments are considered as common scenes in the indoor environments. For example, UE1 and UE2 are indoor corridor scenes, and UE3 and UE4 are indoor hall scenes. For each unseen environment, we performed five times of the mapless navigation control on the mobile robot and recorded the desired target position and the final robot position to evaluate the success rate of the navigation control task in an unseen environment. In the experiment, we define the condition of successful navigation as the error distance within 0.4m between the desired target position and the final robot position.

A. EXPERIMENTAL RESULTS OF THE ROBOT MAPLESS NAVIGATION IN THE UE1

Table 2 lists the results of five experiments of the mapless navigation control of the mobile robot in the UE1. As shown in Table 2, after five experiments, the robot successfully navigated to the desired target position four times. Therefore, the average success rate of the mapless navigation control task in the UE1 reaches 80%. In addition, in the four

successful navigation experiments, the longest navigation distance was about 10.21 meters, and the maximum error distance was 0.19 meters.

Figure 15 shows the image sequence of the mobile robot performing mapless navigation in the first experiment. First, we placed the robot at the starting point shown in Figure 15 and set the desired target position to (5.0m, -2.5m). From Point (2) to Point (4), the robot autonomously approached the desired target position with a smooth motion trajectory. At Point (5), the robot autonomously performed a right-turn motion to correct its orientation angle to face the desired target position. Finally, the robot successfully reached the desired target position at Point (6) with an error distance of about 0.09m.

B. EXPERIMENTAL RESULTS OF THE ROBOT MAPLESS NAVIGATION IN THE UE2

Table 3 lists the results of the five experiments of the mapless navigation control of the mobile robot in the UE2, which is an unseen corridor environment. After five experiments, the robot also successfully navigated to the desired target position four times. Therefore, the average success rate of the mapless navigation control task in the UE2 is 80%. In the four successful navigation experiments, the longest navigation

TABLE 6. Experimental results of the method proposed in [17] for the four unseen environments for comparison.

Unseen Environment	Experiments	Desired Target Position	Final Robot Position	Error Distance	Navigation Distance	Succeeded/Failed
UE1	1st	(4.0,-2.0)	(3.8745,-1.8530)	0.19 m	4.6254 m	Succeeded
	2nd	(6.0,2.5)	X	X	X	Failed
	3rd	(10.0,0.0)	(9.8747,-0.1532)	0.31 m	9.9043 m	Succeeded
	4th	(2.0,5.0)	(2.2594,4.8604)	0.29 m	6.0335 m	Succeeded
	5th	(6.0,2.5)	(5.7347,2.5600)	0.26 m	6.7529 m	Succeeded
UE2	1st	(7.0,-5.0)	(6.9749,-5.0813)	0.08 m	12.8410 m	Succeeded
	2nd	(8.5,4.0)	8.3496,4.0545)	0.16 m	11.7754 m	Succeeded
	3rd	(6.0,-8.0)	(6.2031,-7.7239)	0.34 m	11.8583 m	Succeeded
	4th	(9.0,8.0)	(8.9423,8.2794)	0.29 m	18.1658 m	Succeeded
	5th	(5.5,4.0)	X	X	X	Failed
UE3	1st	(3.0,1.0)	(2.8331,0.9946)	0.17 m	3.0607 m	Succeeded
	2nd	(3.0,-1.0)	X	X	X	Failed
	3rd	(4.0,1.0)	(4.0899,0.7405)	0.27 m	4.2123 m	Succeeded
	4th	(8.0,-7.0)	X	X	X	Failed
	5th	(5.0,-0.5)	X	X	X	Failed
UE4	1st	(5.0,2.0)	X	X	X	Failed
	2nd	(6.0,-1.0)	X	X	X	Failed
	3rd	(7.0,3.0)	X	X	X	Failed
	4th	(5.0,3.0)	(5.2917,2.6124)	0.48 m	7.3305 m	Succeeded
	5th	(4.0,3.0)	(4.1632,2.8033)	0.26 m	6.8798 m	Succeeded
Average Success Rate						60%

TABLE 7. Comparison of the proposed method with other state-of-the-art methods.

Method	Learning Type	Environment Type	Average Success Rate
c1000 [14]	IL	Virtual	~45%
CPO [14]	RL	Virtual	~66%
c1000-CPO [14]	R-IL	Virtual	~79%
SPN {360 10 5 0} [27]	RL	Virtual	78%
SPN-v2 {360 10 5 0} [27]	RL	Virtual	71%
Ours	IL	Real	75%

distance was about 13.01 meters, and the maximum error distance was 0.13 meters.

Figure 16 presents the image sequence of the first navigation experiment in the UE2. In this experiment, we tested whether the robot could complete the mapless navigation control task in the unseen corridor environment. At the beginning of the experiment, we placed the robot at the starting point and set the desired target position to the position where the robot needs to turn twice in the corridor. From Point (1) to Point (3), the robot successfully completed the first autonomous steering action with smooth motion trajectory. From Point (3) to Point (6), the robot completed the second autonomous steering action and successfully reached the desired target position of Point (6) with an error distance of about 0.13m.

C. EXPERIMENTAL RESULTS OF THE ROBOT MAPLESS NAVIGATION IN THE UE3

Table 4 records the results of the five experiments of the mapless navigation control of the mobile robot in the UE3,

which is an unseen wide open environment. After five experiments, the robot successfully reached the desired target position only three times. Therefore, the average success rate of the mapless navigation control task in the UE3 is 60%. In the three successful navigation experiments, the longest navigation distance was about 5.05 meters, and the maximum error distance was 0.19 meters.

Figure 17 shows the image sequence of the first navigation experiment in the UE3. In this experiment, we tested the performance of the proposed mapless navigation control system in the unseen open wide environment. At the beginning of the experiment, we placed the robot at Point (1) shown in Figure 17 and set the desired target position to (3.0m, 1.0m). From Point (2) to Point (5), the robot approached the desired target position with a smooth motion trajectory. At Point (6), the robot successfully reached the desired target position with an error distance of about 0.14m. During the experiments in the UE3, we found that the navigation performance of the proposed mapless navigation control system in an unseen open wide environment may be reduced due to the increased difficulty of the robot localization.

D. EXPERIMENTAL RESULTS OF THE ROBOT MAPLESS NAVIGATION IN THE UE4

Table 5 records the results of the five experiments of the mapless navigation control of the mobile robot in the UE4, which is an unseen wide-open environment with an architectural structure conducive to robot localization. After five experiments, the robot successfully reached the desired target position four times. Therefore, the average success rate of the mapless navigation control task in the UE4 is 80%. In the four successful navigation experiments, the longest navigation

distance was 8.28 meters, and the maximum error distance was 0.33 meters.

Figure 18 shows the image sequence of the first navigation experiment in the UE4. In this experiment, the mobile robot performed navigation tasks near a cylindrical structure in the unseen open wide environment. At the beginning of the experiment, we placed the robot at Point (1) shown in Figure 18 and set the desired target position to (5.0m, 2.0m). From Point (2) to Point (6), the robot tried to avoid the obstacle while approaching the desired target position with a smooth motion trajectory. At Point (7), the robot successfully reached the desired target position with an error distance of about 0.29m. The experimental results in the UE4 show that by improving the reliability of the robot localization, the navigation performance of the proposed mapless navigation control system in an unseen open wide environment can be improved. For a video clip of the experimental results, kindly refer to the online webpage in [26].

E. COMPARISON OF RESULTS WITH SOME EXISTING METHODS

In this section, we first compare the performance of the proposed method with the CNN model introduced in the literature [17], which is highly relevant to this work. We implemented the CNN model in [17] in Tensorflow Keras and trained the model using the proposed large-scale training dataset. After training the model, we tested its navigation performance in the same four unseen environments, each of which contains five experiments. Table 6 records the experimental results of twenty experiments conducted for the purpose of the comparison using the method proposed in [17]. It can be seen from Table 6 that the comparison method performs well in the first two unseen environments with an average success rate of 80%. However, in UE3 and UE4, the average success rate of the comparison method reduced to 40%. This problem is also mentioned in the literature [17] that one limitation of the comparison method is wide open spaces with a lot of glass and clutter around, which is like the last two unseen environments in our proposed work. Therefore, the overall average success rate of the comparison method [17] only reaches 60%, which is lower than the success rate of the proposed method which is 75%.

We next compare the performance of the proposed method with other state-of-the-art methods. Table 7 presents comparison results between the proposed method and all comparison methods, in which the data of all comparison methods are obtained directly from the literature [14] and [27]. In Table 7, the learning types are divided into three types: imitation learning (IL), reinforcement learning (RL), and reinforced imitation learning (R-IL). In addition, the average success rate results of all comparison methods were calculated in virtual environments. The c1000 method is a neural network model trained by IL training with 1000 trajectories in a complex map. The CPO method uses constrained policy optimization (CPO) to train the neural network model during RL training. The c1000-CPO method uses the c1000 model

as the pre-trained model and uses the CPO to update the model during RL training. It can be seen from Table 7 that the proposed method is superior to the c1000 and CPO methods and provides comparable performance to the c100-CPO method.

On the other hand, SPN (support point-based navigation) and SPN-v2 are two recently proposed LiDAR navigation network models that can be used with different range sensors and different installation positions. The SPN {360|10|5|0} method represents the SPN model that uses a LiDAR configuration with 360-degree field of view, 10-degree angular resolution, a maximum detection range of 5 meters, and a distance of 0 meter between the center point of the range sensor and the mobile robot. It can be seen from Table 7 that the proposed method provides comparable performance to the two SPN models configured with low-resolution LiDAR sensors.

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose a mapless LiDAR navigation control method for a wheeled mobile robot based on deep imitation learning to solve the mapless navigation control problem in unknown environments. The proposed navigation control method is a data-driven control method based on a deep CNN model, which consists of a LiDAR signal CNN model for generating point cloud feature maps and a motion prediction neural network model for predicting motion velocity commands. In addition, a data augmentation method is proposed to increase the number of training samples in the preliminary training dataset recorded in the real environment. After the data augmentation process, a large-scale training dataset with 250,000 samples was generated in this study.

In this experiment, four unseen environments are used to evaluate the navigation performance of the proposed mapless navigation control method. Experimental results show that the proposed mapless navigation control method has been tested twenty times in the four unseen environments, and the average success rate reached 75%. Therefore, the proposed mapless navigation control method is effective for robot navigation control in an unknown environment without the global map. We also compared the proposed method with some existing methods, and the experimental results show that the proposed method outperforms the two existing IL-based methods and provides comparable performance to the R-IL-based method in terms of the overall average success rate. Moreover, compared with the recently proposed SPN-based method, the proposed method also provides comparable performance to the SPN model equipped with a low-resolution LiDAR sensor.

In future work, we will try to increase the data diversity of the training database so that the trained navigation network model can have better navigation performance in unknown and complex environments. In addition, we also consider using reinforcement learning to improve the robustness of the navigation network model and achieve better mapless navigation performance.

REFERENCES

- [1] *ROS Move_Base*. Accessed: Aug. 23, 2021. [Online]. Available: http://wiki.ros.org/move_base
- [2] K. Zhu and T. Zhang, "Deep reinforcement learning based mobile robot navigation: A review," *Tsinghua Sci. Technol.*, vol. 26, no. 5, pp. 674–691, Oct. 2021.
- [3] H. Kanayama, T. Ueda, H. Ito, and K. Yamamoto, "Two-mode mapless visual navigation of indoor autonomous mobile robot using deep convolutional neural network," in *Proc. IEEE/SICE Int. Symp. Syst. Integr. (SII)*, Honolulu, HI, USA, Jan. 2020, pp. 536–541, doi: [10.1109/SII46433.2020.9025851](https://doi.org/10.1109/SII46433.2020.9025851).
- [4] T. Xue and H. Yu, "Model-agnostic metalearning-based text-driven visual navigation model for unfamiliar tasks," *IEEE Access*, vol. 8, pp. 166742–166752, Sep. 2020, doi: [10.1109/ACCESS.2020.3023014](https://doi.org/10.1109/ACCESS.2020.3023014).
- [5] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi, "Target-driven visual navigation in indoor scenes using deep reinforcement learning," Sep. 2016, *arXiv:1609.05143*. [Online]. Available: <http://arxiv.org/abs/1609.05143>
- [6] S. Arikian, K. Varanasi, and D. Stricker, "Surface defect classification in real-time using convolutional neural networks," Jan. 2019, *arXiv:1904.04671*. [Online]. Available: <http://arxiv.org/abs/1904.04671>
- [7] A. Hussein, M. M. Gaber, E. Ellyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Comput. Surv.*, vol. 50, no. 2, pp. 1–35, Jun. 2017, doi: [10.1145/3054912](https://doi.org/10.1145/3054912).
- [8] T. Ort, L. Paull, and D. Rus, "Autonomous vehicle navigation in rural environments without detailed prior maps," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Brisbane, QLD, Australia, May 2018, pp. 2040–2047, doi: [10.1109/ICRA.2018.8460519](https://doi.org/10.1109/ICRA.2018.8460519).
- [9] Y. Zhang, J. Wang, X. Wang, and J. M. Dolan, "Road-segmentation-based curb detection method for self-driving via a 3D-LiDAR sensor," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 12, pp. 3981–3991, Dec. 2018, doi: [10.1109/TITS.2018.2789462](https://doi.org/10.1109/TITS.2018.2789462).
- [10] L. Qiang, D. Nanxun, L. Huican, and W. Heng, "A model-free mapless navigation method for mobile robot using reinforcement learning," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Shenyang, China, Jul. 2018, pp. 3410–3415, doi: [10.1109/CCDC.2018.8407713](https://doi.org/10.1109/CCDC.2018.8407713).
- [11] S. Hossain, O. Doukhi, Y. Jo, and D.-J. Lee, "Deep reinforcement learning-based ROS-controlled RC car for autonomous path exploration in the unknown environment," in *Proc. 20th Int. Conf. Control. Automat. Syst. (ICCAS)*, Busan, South Korea, Dec. 2020, pp. 1231–1236, doi: [10.23919/ICCAS50221.2020.9268370](https://doi.org/10.23919/ICCAS50221.2020.9268370).
- [12] B. Zuo, J. Chen, L. Wang, and Y. Wang, "A reinforcement learning based robotic navigation system," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, San Diego, CA, USA, Oct. 2014, pp. 3452–3457.
- [13] L. Tai, G. Paolo, and M. Liu, "Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Vancouver, BC, Canada, Dec. 2017, pp. 31–36, doi: [10.1109/IROS.2017.8202134](https://doi.org/10.1109/IROS.2017.8202134).
- [14] M. Pfeiffer, S. Shukla, M. Turchetta, C. Cadena, A. Krause, R. Siegwart, and J. Nieto, "Reinforced imitation: Sample efficient deep reinforcement learning for mapless navigation by leveraging prior demonstrations," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 4423–4430, Oct. 2018, doi: [10.1109/LRA.2018.2869644](https://doi.org/10.1109/LRA.2018.2869644).
- [15] W. Khaksar, M. Z. Uddin, and J. Torresen, "Learning from virtual experience: Mapless navigation with neuro-fuzzy intelligence," in *Proc. Int. Conf. Intell. Syst. (IS)*, Funchal, Portugal, Sep. 2018, pp. 117–124, doi: [10.1109/IS.2018.8710525](https://doi.org/10.1109/IS.2018.8710525).
- [16] M. Pfeiffer, U. Schwesinger, H. Sommer, E. Galceran, and R. Siegwart, "Predicting actions to act predictably: Cooperative partial motion planning with maximum entropy models," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 2096–2101, doi: [10.1109/IROS.2016.7759329](https://doi.org/10.1109/IROS.2016.7759329).
- [17] M. Pfeiffer, M. Schauble, J. Nieto, R. Siegwart, and C. Cadena, "From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Singapore, May 2017, pp. 1527–1533, doi: [10.1109/ICRA.2017.7989182](https://doi.org/10.1109/ICRA.2017.7989182).
- [18] M. Hamandi, M. D'Arcy, and P. Fazli, "DeepMoTion: Learning to navigate like humans," in *Proc. 28th IEEE Int. Conf. Robot Hum. Interact. Commun. (RO-MAN)*, New Delhi, India, Oct. 2019, doi: [10.1109/RO-MAN46459.2019.8956408](https://doi.org/10.1109/RO-MAN46459.2019.8956408).
- [19] *Laser_Scan_Matcher*. Accessed: Aug. 23, 2021. [Online]. Available: http://wiki.ros.org/laser_scan_matcher
- [20] A. Censi, "An ICP variant using a point-to-line metric," in *Proc. IEEE Int. Conf. Robot. Automat.*, Pasadena, CA, USA, May 2008, pp. 19–25, doi: [10.1109/ROBOT.2008.4543181](https://doi.org/10.1109/ROBOT.2008.4543181).
- [21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778, doi: [10.1109/CVPR.2016.90](https://doi.org/10.1109/CVPR.2016.90).
- [22] A. F. Agarap, "Deep learning using rectified linear units (ReLU)," Mar. 2018, *arXiv:1803.08375*. [Online]. Available: <http://arxiv.org/abs/1803.08375>
- [23] *The PCL (Point Cloud Data) File Format*. Accessed: Aug. 23, 2021. [Online]. Available: https://pointclouds.org/documentation/tutorials/pcd_file_format.html
- [24] *Pioneer 3-DX Mobile Robot*. Accessed: Aug. 23, 2021. [Online]. Available: <https://www.generationrobots.com/media/Pioneer3DX-P3DX-RevA.pdf>
- [25] *Velodyne LiDAR PUCK*. Accessed: Aug. 23, 2021. [Online]. Available: <https://www.amtechs.co.jp/product/VLP-16-Puck.pdf>
- [26] *Experimental Results of the Proposed Mapless LiDAR Navigation Control Method of Wheeled Mobile Robots*. Accessed: May 30, 2021. [Online]. Available: <https://www.youtube.com/watch?v=P9cpFEFW3QI>
- [27] W. Zhang, N. Liu, and Y. Zhang, "Learn to navigate maplessly with varied LiDAR configurations: A support point-based approach," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 1918–1925, Apr. 2021, doi: [10.1109/LRA.2021.3061305](https://doi.org/10.1109/LRA.2021.3061305).



CHI-YI TSAI (Senior Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from the National Yunlin University of Science and Technology, Yunlin, Taiwan, in 2000 and 2002, respectively, and the Ph.D. degree in electrical and control engineering from the National Chiao Tung University, Hsinchu, Taiwan, in 2008.

He joined the Department of Electrical Engineering, Tamkang University, New Taipei City, Taiwan, in 2010, where he is currently a Professor.

His research interests include image processing, color image enhancement processing, visual tracking, visual servoing, computer vision, and deep learning.



HUMAIRA NISAR (Senior Member, IEEE) received the B.S. degree in electrical engineering from the University of Engineering and Technology, Lahore, Pakistan, the M.S. degree in nuclear engineering from Quaid-i-Azam University, Islamabad, Pakistan, and the M.S. degree in mechatronics and the Ph.D. degree in information and mechatronics from Gwangju Institute of Science and Technology, Gwangju, South Korea.

She has over 20 years of research experience.

She is currently an Associate Professor with the Department of Electronic Engineering, Universiti Tunku Abdul Rahman, Malaysia. She is also the Head of Program for the Master of Engineering (Electronic Systems) Program. She has authored a number of international journal articles and conference papers. Her research interests include signal and image processing, bio-medical imaging, brain signal and image analysis, and image analysis for wastewater treatment. She has served on technical committees of various conferences and journals.

YU-CHEN HU received the B.S. and M.S. degrees in electrical and computer engineering from Tamkang University, New Taipei City, Taiwan, in 2018 and 2021, respectively. He is currently in military service. His research interests include robotics, navigation control of mobile robots, and deep learning.

• • •