

Automatic Reconstruction of Building-Scale Indoor 3D Environment with a Deep-Reinforcement-Learning-Based Mobile Robot

Menglong Yang and Katashi Nagao*

Graduate School of Informatics, Nagoya University, IB South Building 395, Furo-sho Chikusa-ku, Nagoya, Aichi, 464-8603, Japan

Abstract: The aim of this paper is to digitize the environments in which humans live, at low cost, and reconstruct highly accurate three-dimensional environments that are based on those in the real world. This three-dimensional content can be used such as for virtual reality environments and three-dimensional maps for automatic driving systems.

In general, however, a three-dimensional environment must be carefully reconstructed by manually moving the sensors used to first scan the real environment on which the three-dimensional one is based. This is done so that every corner of an entire area can be measured, but time and costs increase as the area expands. Therefore, a system that creates three-dimensional content that is based on real-world large-scale buildings at low cost is proposed. This involves automatically scanning the indoors with a mobile robot that uses low-cost sensors and generating 3D point clouds.

When the robot reaches an appropriate measurement position, it collects the three-dimensional data of shapes observable from that position by using a 3D sensor and 360-degree panoramic camera. The problem of determining an appropriate measurement position is called the "next best view problem," and it is difficult to solve in a complicated indoor environment. To deal with this problem, a deep reinforcement learning method is employed. It combines reinforcement learning, with which an autonomous agent learns strategies for selecting behavior, and deep learning done using a neural network. As a result, 3D point cloud data can be generated with better quality than the conventional rule-based approach.

Keywords: Indoor 3D reconstruction, deep reinforcement learning, mobile robot, 3D point cloud, next best view.

1. INTRODUCTION

In recent years, as progress continues to be made on virtual reality (VR) and augmented reality (AR) technologies, the demand for three-dimensional content has explosively increased. However, to create three-dimensional content, experts must devise a model from scratch or manually scrutinize objects finely with a 3D scanner from every angle, which is very time-consuming. It is clear that there are huge costs involved in creating large-scale three-dimensional content in the real world. In addition to using real-world 3D content in VR/AR applications, 3D content that is based on a real-world environment can also be used by intelligent robots as a map for use in automatic driving. However, such robots must be manually moved, and three-dimensional content must be carefully created so that the entire operation area is contained in the map. This becomes time intensive and costly as the map area expands.

In other words, in the VR and robotics fields, real-world 3D content is required, but the cost of creation, especially the cost of creating large-scale 3D content,

is very high. Therefore, this research is aimed at creating three-dimensional content that is based on large-scale buildings in the real world, and an efficient method is proposed that is low-cost.

To reduce costs, a system is needed that automatically collects the three-dimensional data of an entire environment without the need for humans. Therefore, in this research, the focus is on equipment costs, and a robot is used that can be run autonomously with low-cost devices. A system is introduced that automatically collects three-dimensional data from each of the positions from which an environment is measured and automatically generates large-scale three-dimensional content. To scan an entire environment without data loss, these measurement positions must be dynamically determined on the basis of three-dimensional data measured by sensors. In this research, on the basis of actual data collected by using sensors, the optimal positions to which the robot should go next for measurement from the robot's current position are estimated by using a model trained by reinforcement learning.

It is extremely difficult for a robot to repeatedly collect measurement data through repeated trial and error by traveling all the way to estimated

Address correspondence to this article at the Graduate School of Informatics, Nagoya University, IB South Building 395, Furo-sho Chikusa-ku, Nagoya, Aichi, 464-8603, Japan; Tel/Fax: +81-52-789-5878; E-mail: nagao@i.nagoya-u.ac.jp

measurement positions in a real-world environment. Therefore, in this study, reinforcement learning is used to train a model. The model automatically selects the next optimal measurement position on the basis of sensor data by using a virtual environment constructed from 3D point cloud data of an existing building as the environment.

An experiment was conducted to automatically generate 3D content in the real world by implementing a learning model trained in a virtual environment into a robot system. The generated 3D content was applied to a VR disaster simulation system [14] and an automatic driving system for electric wheelchairs. The 3D content generated by the proposed method was confirmed to be useable as content for VR environments and as a map of an environment for autonomous driving.

2. RELATED WORK

Methods used in studies on reconstructing indoor 3D environments for use as 3D maps can be roughly divided into two categories. One consists of semi-automatic methods in which a human moves a sensor and generates a 3D scene [5,6,10,12,17,23,24]. The other consists of automatic methods in which an autonomous mobile robot senses an indoor area and generates a 3D scene [1,2,4,20]. When 3D reconstruction is performed by manually moving a sensor, data coverage guaranteed, so advanced methods can be performed while scanning. One such advanced method is 3D object recognition [22]. However, if the target to be scanned is large, scanning it manually becomes very difficult to do.

To automatically reconstruct 3D scenes, automatic scanning and automatic modeling must first be distinguished. In this paper, automatic scanning is focused on. The method of this paper can completely scan the interiors of buildings by using a 2D/3D laser scanner onboard a mobile robot.

A mobile robot with a 3D scanner was used in Adán *et al.*'s study [1], in which a strategy was proposed for automating the process of scanning the indoors with a mobile robot. Micro-aerial vehicles (MAVs) can even be used to gather 3D information on indoor environments, like in Shen *et al.*'s study [20]. They proposed an indoor mapping approach that uses an MAV equipped with a camera and laser range finder that gathers data at different heights in a room.

Bircher *et al.* [4] also proposed a 3D search method suitable for MAVs. This method voxelizes point clouds

collected from depth images to generate a 3D occupancy grid map. Next, it analyzes the map with a tree structure and estimates the positions at which data on parts that have not yet been mapped can be best collected. In this approach, search area boundaries are required.

Bai *et al.* [2] proposed a method that searches for an appropriate position and direction in a 2D map for mapping and exploration by using a deep neural network. Using a random dungeon generator, a virtual 2D map is generated and simulated. A small fragment of a generated 2D map is sampled and inputted to a neural network model. Then, the next measurement position is estimated on the basis of the partial 2D map. This method has not yet been applied to generating 3D maps.

3. MOBILE SCANNING ROBOT FOR RECONSTRUCTING BUILDING-SCALE INDOOR ENVIRONMENT

To reduce manual costs, a system for automatically collecting three-dimensional data of an entire environment is necessary. In this research, a scanning robot is designed in consideration of equipment costs, and it can run autonomously with low-cost sensors. The configuration of the robot is shown in Figure 1.

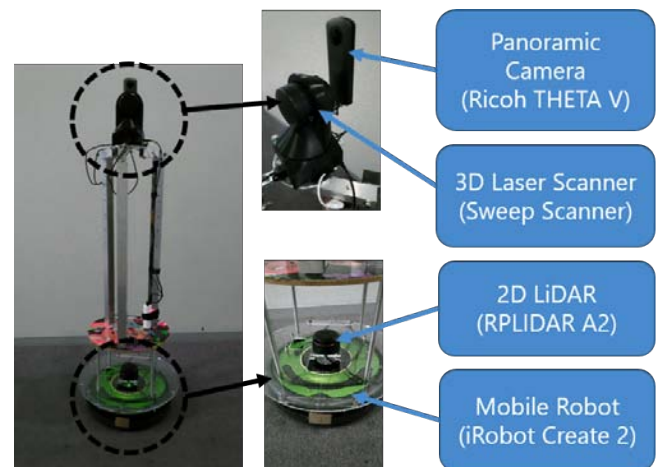


Figure 1: Indoor Scanning Robot.

The upper part of the robot is a scanning device that consists of a Sweep 3D Scanner that is capable of collecting 3D point clouds in 360 degrees and a full spherical camera, a Ricoh THETA V. This scanning device can collect 3D colored point clouds of the indoors. The lower part of the robot is a movement device that consists of a RPLIDAR A2 for 2D LiDAR and an iRobot Create 2. The position of the robot itself can be estimated by using the RPLIDAR A2, and the



Figure 2: 2D Map.

robot can be autonomously run in combination with the iRobot Create 2. Thus, a system has been created that automatically generates point cloud data in building environments with this robot.

3.1. Robot Navigation

To automatically reconstruct an indoor environment, first, determining the functionality of this robot that autonomously operates in general indoor environments is necessary. In this research, a general indoor environment is considered to have a flat floor, walls, and a ceiling. In addition, objects such as chairs and desks exist in the environment, and it is also assumed that humans are walking around. Thus, the robot is assumed to operate on a flat surface.

For autonomous driving, a map of the physical space to be traveled is necessary for the robot to understand its present location, generate a route to a destination, and then travel to it. Since the scanning robot operates on a flat plane, a 2D map is sufficient for the environment map. Therefore, in this research, both 2D range sensor data and odometry done with the iRobot Create 2 are integrated by using GMapping [8] to create a 2D occupancy grid map. An example of a created 2D map is shown in Figure 2.

With the created map, the robot must estimate its current position and plan a route in order to autonomously travel to a next position from which to perform measurement. In this study, an autonomous running system for scanning robots was implemented by using the navigation stack of the robot operating system (ROS) [13], enabling self-position estimation, route planning, and generation of control commands sent to iRobot Create 2.

The robot must be controlled manually when creating a 2D map. However, once the 2D map is created, the robot enters autonomous mode, autonomously travels to a next measurement position, and automatically collects point cloud data around the position.

3.2. 3D Colored Point Cloud Generation

When the robot arrives at a measurement position, it starts collecting three-dimensional point cloud data on the surrounding area. As mentioned, the Sweep 3D Scanner, which is capable of collecting 3D point clouds, is used in combination with a Ricoh THETA V whole spherical camera to collect data on the environment at each measurement position as 3D colored point cloud data.

When the robot arrives at a measurement position, there is the possibility that the scanner, which is at the top, will sway for a while. This can lead to noise being contained in the point cloud data collected by the scanner, so information is used from the inertial measurement unit (IMU) of Sweep 3D Scanner to start collecting point clouds on the surrounding environment after the scanner has stabilized. Preliminary processing is necessary because there is a possibility that useless data and outliers may exist in the collected raw data. Since Sweep 3D Scanner uses Sweep, which is a 2D laser range sensor, the laser reflection intensity can be obtained together with coordinates at the time of measurement. On the basis of the intensity, some noise is deleted by deleting points whose values are remarkably low. Also, VoxelGridFilter, included in the point cloud library (PCL) [18], is used to unify the density of point clouds, and outliers are excluded by using StatisticalOutlierRemoval.

Ricoh's THETA V camera is used in combination with the scanner, and the color information of an environment is acquired in 360 degrees around the robot as a panoramic image. Shooting with a full celestial camera is fast, but there are frequent opportunities for an image to include the figure of a person passing by. When a human enters a panoramic image, their figure becomes noise that colors the point clouds, which causes trouble when matching between the point cloud and the image. Therefore, in this research, RetinaNet is used [11]. It is a deep learning model for detecting existing objects in panoramic images to determine whether there is a human in the



Figure 3: Generation of Colored Point Cloud.

images. If it is recognized that a person has entered a panoramic image, the scanner pauses to wait 20 seconds. Then, the image is shot again, and this continues every 20 seconds until no human appears in the image. Thus, this prevents people from appearing in panoramic images.

Since the panoramic images and point cloud data taken with the all-sky camera contain 360-degree data on the surroundings measured from the same origin, correspondence is established between each point of the point cloud and the pixels of an image. To match each pixel of a collected panoramic image to each point of a three-dimensional point cloud, it is necessary to match each other's coordinate systems. In this study, both are converted to the same spherical coordinate system, and points are matched with pixels by using a neighborhood method based on angle. Then, a colored point cloud is generated. A collected raw point cloud, panoramic image, and generated colored point cloud are shown in Figure 3.

When measurement at one place is completed, the generated three-dimensional colored point cloud data and the data measured so far are automatically synthesized. The processing of this part has two steps. First, coordinate transformation is performed on coordinates to each point of the point cloud measured by using the self-position of the robot, and a point is added to the previous point cloud data. However, it is difficult for the robot to autonomously travel accurately without there being a gap of 1 mm between the robot itself and the destination position and angle specified by the robot, so there is an error in the locational accuracy at a measurement position. Thus, there is a deviation between the point clouds due to the error. Therefore, fine matching is performed by using PCL's iterative closest point (ICP) algorithm.

The robot autonomously can then travel to a measurement position, collect three-dimensional shape information and color information on the surroundings, and generate a three-dimensional colored point cloud. Then, the generated point cloud is matched with the

previous point cloud. This process is repeated to synthesize point clouds measured at multiple measurement points into one large-scale point cloud.

4. DEEP REINFORCEMENT LEARNING TO FIND NEXT-BEST-VIEW POSITIONS

The robot autonomously travels to a measurement point, gathers data on the surroundings after the scanner stabilizes, and then travels to the next measurement point. Data are collected with this procedure. Therefore, measurement positions suitable for the target environment must be estimated in advance.

In this study, the next measurement positions are estimated sequentially by using a model trained by reinforcement learning [21] with actual data collected by using sensors. In reinforcement learning, instead of considering the optimal movement of the target robot to be a correct answer, a reward is given to each action of the robot. Robots can learn strategies for taking optimal actions by comparing how they act and how many rewards are likely to be obtained and then choosing behaviors that are likely to receive the maximum rewards.

Instead of using data actually collected by sensors, the measurement position can be calculated by using a rule-based method with the floor plan of a building. Various objects such as desks and chairs appear in the real world, and thus, a point cloud could contain parts of the environment that are occluded by other objects, and there may be parts that cannot be acquired. To reconstruct a whole environment without loss, measurement positions need to be dynamically determined on the basis of 3D point clouds measured by sensors. However, since there is variety in environments that affects such clouds, designing a rule for estimating the next optimum measurement position is very difficult.

By using reinforcement learning, it is thought that a robot can learn strategies to select positions appropriate for sensors to measure three-dimensional

point clouds, that is, the various situations the robot in this research faces.

4.1. Problem Definition

Research on reinforcement learning mainly operates in a game environment or a simulation environment, and there is little analysis on complex environments in the real world. Also, in many cases, training is conducted in the training environments published by OpenAI gym [15], tested, and compared with existing methods. Therefore, the gap with the subject of this research is large. In this research, the next measurement position is estimated dynamically on the basis of data actually measured by a 3D scanner. The state of the environment is represented by a point cloud consisting of about 200,000 points, and taking into consideration the coordinates of the three axes, x, y, and z, the total number of values is 600,000.

Also, since all spaces in which the robot can operate need to be considered, the dimension of the output of the reinforcement learning-trained model is high. In other words, the problem faced in this research shares a gap with existing research topics on reinforcement learning, so a solution specialized for complicated tasks is necessary.

The problem is one of dynamically estimating the next optimum measurement position on the basis of already measured data. This problem is called the "next best view (NBV) problem." The NBV problem is the problem of choosing an optimal viewpoint for measurement in consideration of the shape of an object [3,4,7,9,16]. Although the definition of the optimal viewpoint also varies according to the actual task, it is mainly studied in the field of 3D object reconstruction, and the NBV problem is to solve the task of estimating the position and orientation of a 3D scanner to complete the model of a 3D object.

Although there are existing studies targeting the reconstruction of 3D scenes, most methods predict the next position that can be used to measure the most points, on a rule basis, with a partial 3D point cloud already constructed as an environment. However, various circumstances exist that affect information related to environments, and even if a measurement position is slightly different in the same environment, the measured data also differ, so it is difficult to finely design a rule for estimating the optimum positions one by one. In this research, the next optimum position is estimated by using a reinforcement learning-trained model based on data actually collected with sensors.

However, it is difficult to directly input three-dimensional point group data into a learning model as is and to determine the coordinates of the next measurement position directly as the output of the learning model. This is because the amount of three-dimensional point cloud data is huge and the amount of computation involved in training processing is very large. In addition, to make the learned model understand which areas are scanned and which areas are not, it is necessary to input a point cloud that combines all point clouds so far, not the point cloud data of only one measurement. Therefore, the more input there is, the larger the amount of input data becomes. Since the processing becomes more and more heavy and the amount of useless data included in the input data also increases, it becomes difficult for the learning model to learn features from the enormous amount of data, and its training is difficult to converge.

In actual three-dimensional reconstruction, the cause of missing parts in a point cloud can be considered to be due to the intervals between measurement positions and occlusion caused by obstacles. If the intervals are too large, the data between them cannot be measured, and if they are too close, the efficiency worsens, and the measurement time becomes longer. In many cases, missing parts due to intervals can be determined from the floor portion of the point cloud. In addition, since the scanner of the robot in this research is fixed at a point higher than common indoor obstacles, most of the parts occluded by obstacles are on the floor and can be seen from the lower part of the point cloud. Furthermore, since the motion of the scanning robot is limited to two dimensions, it is possible to significantly reduce the computation amount by compressing measured point groups into two dimensions in order to estimate measurement positions.

Therefore, in this study, the following processing was performed on point clouds in order to compress measured point clouds into two dimensions and input them into the model.

- Deleted the part of the ceiling that is 1 m or higher than the scanner.
- Moved the point cloud down along the Z axis (direction of gravity).
- Converted the projected two-dimensional point cloud into an image with a resolution of 1 pixel = 0.05 cm.

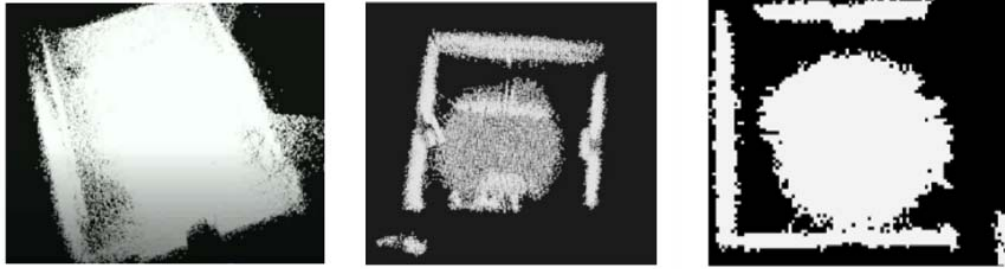


Figure 4: Compression of Point Cloud Data.

An example is shown in Figure 4. The left side of the figure is the original point cloud, the middle is the point cloud with the upper part deleted, and the right is the transformed two-dimensional projection image.

If only the data measured at one place were input, the information on the previous measurement positions would be lost, and the learning model would not be able to distinguish between a part measured in the current environment and a part not measured. In this research, a 2D map of an environment is generated in advance, a modified form of the map is input into the learning model as global information on an environment. As shown in Figure 5, this is a combination of a 2D map and a 2D projection image of data measured at each measurement position by using the coordinates of the measurement positions.

The synthesized image is distinguished into obstacles, dangerous areas that are close to obstacles, scanned areas, unscanned areas, and exterior areas outside a building by using pixel values, and this is called a “progress image.” This image is input into the learning model. Using the image, the robot can see how much of the entire area was scanned up to the current step and how many missing parts there are.

Also, if learning is performed with the positions of all areas as candidate positions, the number of action choices becomes considerably large, and the learning model does not converge. Furthermore, the number of candidate positions varies depending on the environment, so there is a problem in that the trained model cannot be used in different environments.

Therefore, in this research, considering the measurement range of the scanner, an area of -5 m to 5 m from the current measurement position is discretized every 0.5 m and used as a candidate position for the next measurement position. In addition, the closer the scanner is to a point, the more accurately the point can be measured, so there are few situations in which the data of an area close to a measurement position are lost, the data of an area far from a position become coarse, or missing parts appear. Since the distance between measurement positions is too small, there arises a problem of duplicate measurement.

Therefore, the dimension of output can be further reduced by excluding positions within a range of -3 m to 3 m from a measurement position from among candidate positions. Discretizing positions within a range of -5 m to 5 m into 0.5 m units will result in 400 candidate positions, and further excluding positions from -3 m to 3 m will result in 256 candidate positions. 256 candidate positions can be listed from 0 to 255 from left to right and from top to bottom. Thus, each element of a 256×1 vector output by the learning model can correspond to each candidate position, and selecting a candidate position will select the position corresponding to the maximum value of the vector.

As input data in addition to the progress image, two feature vectors are prepared: a candidate position vector and a residual area vector that can be calculated from a progress image. The candidate position vector is a 256×1 vector representing information on obstacles at 256 candidate positions. As shown in Figure 6, it indicates whether each value corresponds



Figure 5: Progress Image.

to an obstacle at the corresponding candidate position. Using this feature vector, the learning model can learn the distinction between measurement positions that the robot can go to and those to which the robot cannot go.

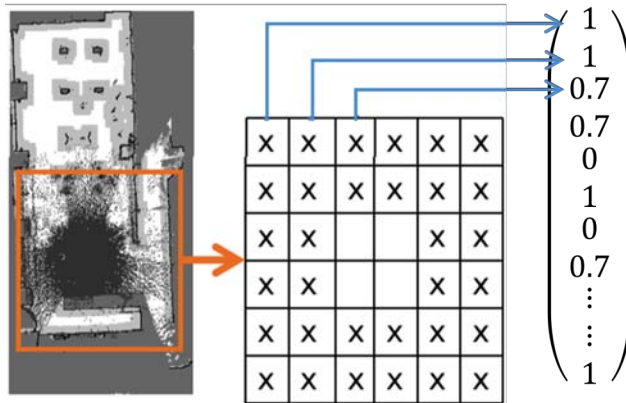


Figure 6: Candidate Position Vector.

The residual area vector represents the size of the area not yet scanned around each candidate position. Like the candidate position vector, this is a 256×1 vector, and as shown in Figure 7, for all candidate positions, an area within a range of 5 m is cut out as an observation area. Then, the learning system counts the number of unscanned pixels in the observation area and normalizes the number. By using this feature vector, the learning model can consider the circumstances of each candidate position.

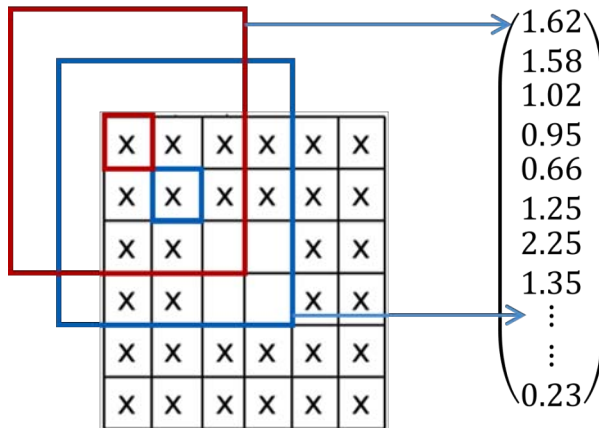


Figure 7: Residual Area Vector.

Thus, the problem of efficiently scanning an area specified on the basis of a measured three-dimensional point cloud without loss is regarded as a problem of filling all 2D maps in a short amount of time.

4.2. Training Learning Model in Virtual Environment

To train a model by reinforcement learning, a large amount of sampling data is necessary, and it is

necessary to repeat trial and error in environments in which a robot operates. However, collecting measurement data by moving a robot to all estimated measurement positions in a physical environment is extremely difficult and very costly. To solve this problem, it is common practice to train a model with reinforcement learning in a virtual environment in order to efficiently perform learning and training. In this research, 3D point cloud data of existing buildings were set as virtual environments for reinforcement learning, and a simulation system that combines 3D scanning and 2D map generation was developed.

To implement the virtual environment of the task of this research, data on the environment are first necessary. Considering that a learning model trained in a virtual environment can be used for operation in a real-world environment, it is thought that the environment data need to faithfully reproduce the real-world environment as much as possible. For this reason, a virtual environment was constructed by using real-world precise three-dimensional point cloud data as environment data, which was created by hand and with highly accurate measuring instruments.

For the virtual environment to be used to output the aforementioned progress image according to the behavior of the robot, a 2D map must be generated from the point cloud of the environment data. Since three-dimensional point cloud data representing the layout of an environment are ground truth data of a virtual environment, it is possible to extract the floor and obstacles from the point cloud according to the coordinate values in the direction of gravity and generate a 2D map.

Also, in a virtual environment, measurement needs to be simulated at arbitrary measurement positions. Considering that a large amount of trial and error is necessary for learning, this must be done at high speed. The essence of a three-dimensional scanning simulation is to extract points that can be measured from a certain viewpoint in the 3D point cloud data of an environment. The problem at that time is that, at points of the same angle, points far from the viewpoint are occluded by obstacles at closer points. The coordinates of the nearest neighbor points are difficult to calculate in an orthogonal coordinate system but can be easily calculated by converting point cloud data to a spherical coordinate system.

The point cloud of an environment is downsampled, and it is moved so that the coordinates of the origin

coincide with the measurement position with respect to all points of a point group. Then, points within 6 m from the origin are extracted as observation areas, these points are converted into a spherical coordinate system, and points closest to the origin are extracted from the set of points at the same angle. As a result, measurement done by a virtual sensor is successfully simulated.

In a virtual environment, it is necessary to simulate a sensor and not only output an observation signal but also give both reward and end signals to the learning model. Compensation as designed in this research consists of reward R_s for progress, reward R_k for number, and reward R_i for position.

Progress Reward R_s

The value of R_s depends on the area scanned effectively by a certain action of the robot (determination of measurement position). In a virtual environment, progress images of each step are recorded, and the scanned areas of two consecutive progress images are compared. Then, as the effective progress obtained with this measurement, a positive reward is given according to the number of pixels that has increased since the last time.

Number Reward R_k

A negative R_k is given for each step so that increasing the measurement time is not a wasteful strategy.

Position Reward R_i

If a robot cannot arrive at a measurement position or a position that is too close to an obstacle is predicted, a negative reward R_i is given.

The learning model selects a series of measurement positions through several steps, and

when an end signal is given, it terminates and initializes the internal state. The flow from initialization to termination is defined as one episode. In this study, the three following patterns end one episode.

Termination Due to Limit Number

Since the amount of time that a real-world robot can operate is limited, the upper limit of the number of times must be considered. In the training experiment in this study, the upper limit is set to 100, and if it exceeds 100 steps for 1 episode, it ends.

Termination Due to Completion

When all areas are scanned, the episode ends. The robot cannot necessarily scan perfectly against an actual environment, and there are parts that cannot be measured due to the structure of the robot. Therefore, when parts of areas not scanned become less than a certain level, the measurement is terminated.

Termination Due to Bad Position

When the predicted measurement position is outside the scan area, the robot cannot arrive at it, so this episode ends in a measurement failure.

The learning model of this study was trained by using this virtual environment. The learning model was a model based on the Proximal Policy Optimization (PPO) method [19] proposed by OpenAI. The PPO method outputs actions against an input state, and it utilizes a neural network Actor that selects actions to obtain the maximum reward and a neural network Critic that evaluates the performance of the Actor.

The architecture of the Actor used in this research is shown in Figure 8. When a normalized progress image is inputted, high-level features are extracted while being downsampled via a series of convolutional layers. Then, the candidate position vector and the

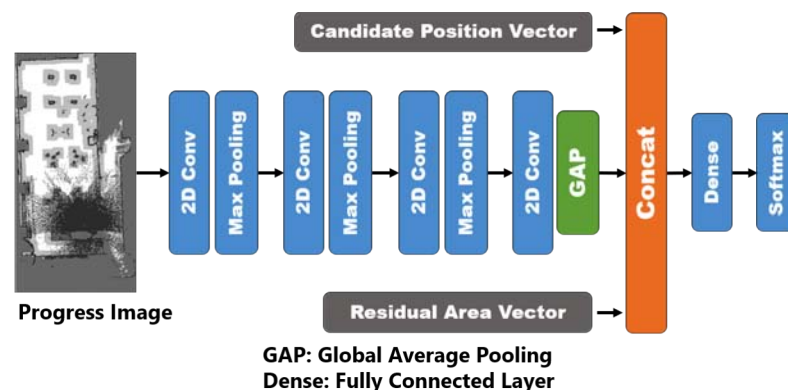


Figure 8: Neural Network Architecture of Actor.

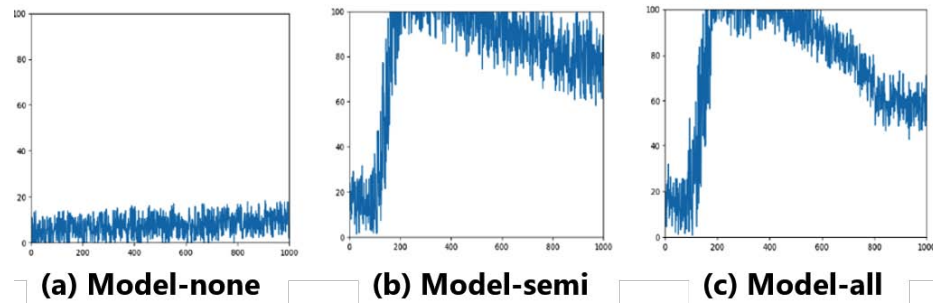


Figure 9: Three Cases of Learning History.

residual area vector are integrated, and the probability of selecting each candidate position with a softmax function via the entire connecting layer is outputted. In the Critic's architecture, the feature extraction mechanism is almost the same as that of the Actor. Features are passed through all of the connection layers in the latter half, and one value is output as an advantage.

Using the virtual environment, the following three models were individually trained with 1000 episodes: model-none, which inputs only a progress image, model-semi, which inputs a progress image and candidate position vectors, and model-all, which inputs a progress image and two feature vectors. A history of the number of episodes and the number of measurements is shown in Figure 9.

As shown in Figure 9a, model-none had few steps in each episode. The predicted measurement positions greatly departed from the scan area, and the episode ended. That is, the measurement positions estimated by model-none were often located where the robot could not go, so learning failed.

In comparison, as shown in Figure 9b, when input included the candidate position vector, the distinction between arrivable and non-arrivable positions was efficiently learned. The position information was first learned, the number of steps increased to 100 at once, and the number temporarily stopped changing. The reason is that, since the maximum number of steps was set at 100, an area that was not measured even if it was measured in 100 steps remained. To maximize the reward, the percentage of the area measured in 100 steps increased. After that, the number of steps dropped, and the robot began to measure the area efficiently.

The trend in Figure 9c is similar to Figure 9b, but by using the residual area vector, the speed of learning increased.

5. EVALUATION

In this research, a learned model was implemented in the robot system described in Section 3, and on-site experiments were performed on automatically generating real-world 3D content on one floor of a building at Nagoya University as shown in Figure 10.

The start position was set to the east at the right of the map, the robot was manually controlled, and a 2D map was created. Once the map was created, the robot automatically entered autonomous driving mode. It autonomously traveled to the start position, collected the first 3D point cloud data and panoramic image with the start position as the initial position, and transferred this information to the server PC together along with its own current position. Each point of the point cloud data and each pixel of the panoramic image were matched with each other on a server PC to create a 3D colored point cloud. In addition, the server PC compressed the point cloud data into two dimensions and created the first progress image by combining it with the 2D map. The progress image was input to the learned model, the output measurement position was transmitted to the robot, the robot autonomously traveled to that position, and the data were measured and combined with the previous data. This process was repeated to automatically reconstruct indoor 3D environments.

Moreover, to verify the effectiveness of the method for estimating measurement positions proposed in this research, a rule-based method was implemented for estimating measurement positions [1] as a comparison. Using the robot with this method implemented, an experiment was conducted in the same area as before.

As a result, 3D content was generated through measurement done 26 times with the reinforcement learning-trained model and 24 times with the rule-based method. A comparison is shown in Figures 11 and 12. As can be seen, the measurement positions of the learning model method increased the number of

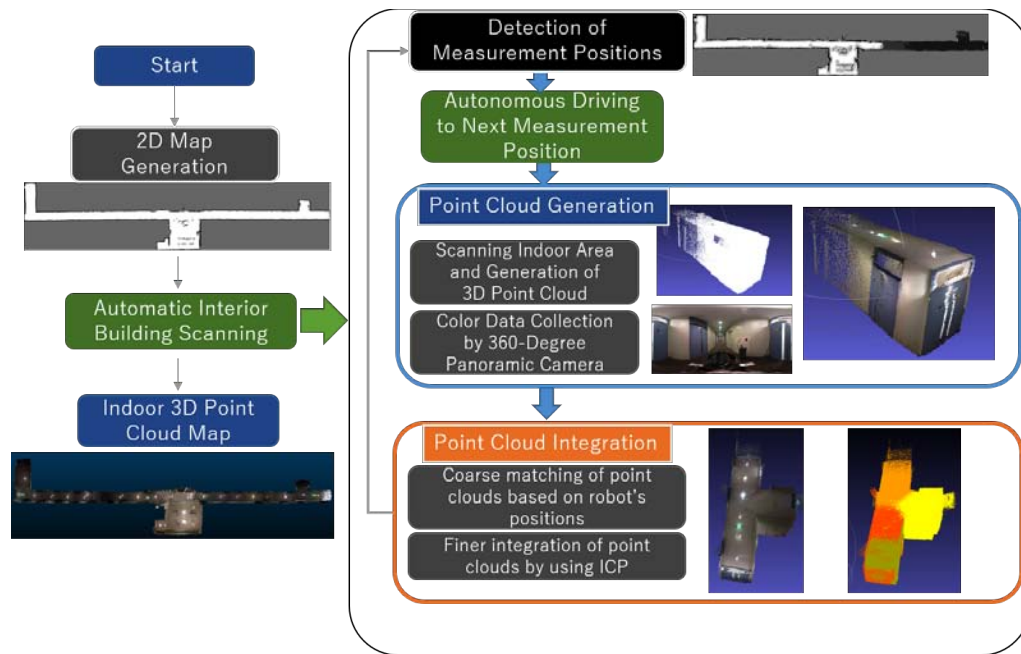


Figure 10: Overall Process of Building Scanning.

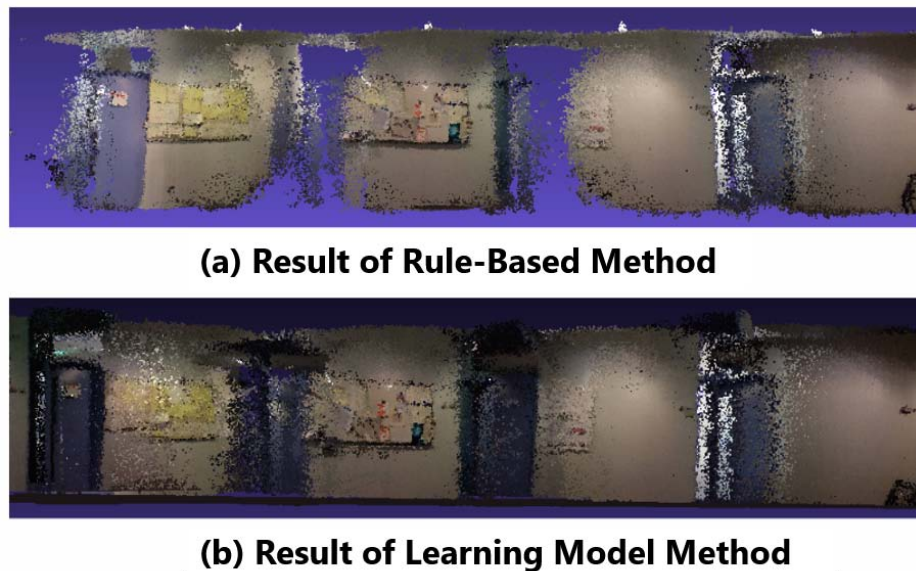


Figure 11: Comparison in Corridor Area.

measurements only by two more than the measurement positions estimated by the rule-based method, but the quality greatly improved.

To numerically compare the results of the proposed method and the conventional rule-based method, the following two indicators were used. One is the number of points included in the generated point cloud data, and the other is the average point density calculated by accumulating the number of points per unit volume (point density) with respect to all measurement positions.

Before comparing, both types of data were downsampled with `VoxelGridFilter` in 0.01 m units to exclude duplicate points. As a result, there were 6,526,922 points in the 3D content with the rule-based method and 10,776,687 points in the 3D content with the learning model method. For each measurement position, the number of neighboring points within the 0.1 m surrounding radius was counted, and the density was calculated over the volume in that range. Then, the average value of the point density was calculated. In the case of 3D content with the rule-based method, it

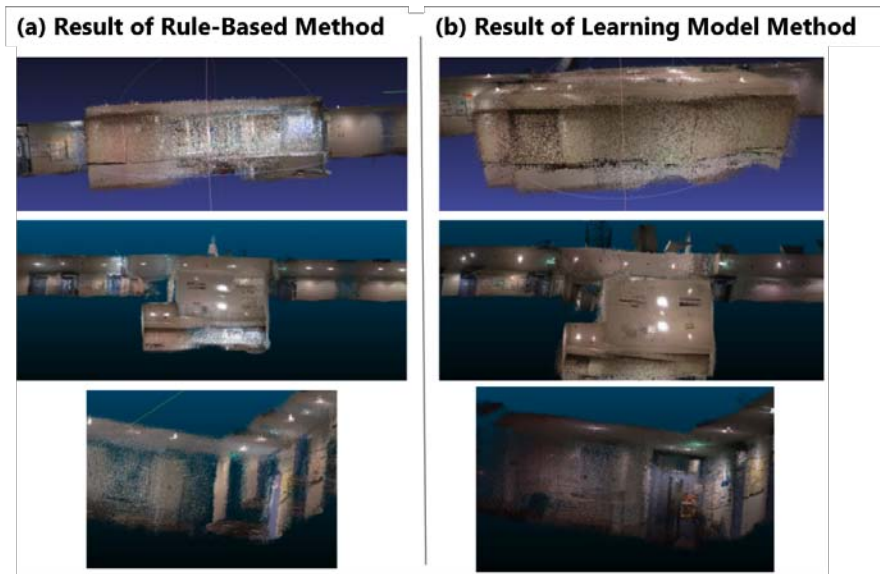


Figure 12: Comparison in Other Areas.

was 63,235.2, and in case of 3D content with the learning model method, it was 69,625.2.

6. APPLICATIONS

The purpose of this research is to automatically generate large-scale 3D content at low cost. Using a disaster simulation system for building-scale VR [14] created by the laboratory to which the authors belong, the effectiveness of the content generated by the method proposed in this research was verified.

Using the game engine Unity developed by Unity Technologies, the 3D content created in this research was used as a simulation environment. The results are shown in Figure 13. It was confirmed that 3D content automatically generated at low cost with the method proposed in this research can be used as VR content.

In addition to using real-world 3D content for VR and AR, the content can have another important role, that is, as an environmental map for the automatic movement of intelligent robots, such as autonomous

mobile robots and drones. Using a self-driving electric wheelchair system developed by the laboratory to which the authors belong, it was verified that the 3D content generated in this study can be used as a 3D map for automatic traveling.

Many mechanisms for moving automatically to specified destinations have been studied as basic functions of mobile robots. The basic functions include generating environment maps and estimating self-positions. 3D content generated in this research was downsampled, and whether it could be used as a three-dimensional map for estimating the self-position of wheelchairs indoors was verified. As shown in Figure 14, an electric wheelchair equipped with 3D LiDAR was able to properly estimate its own position.

7. CONCLUDING REMARKS

In this paper, a method was proposed for realizing a system that can automatically convert a large-scale indoor environment into 3D data so that it can be used in VR and for automatic driving. Specifically, the

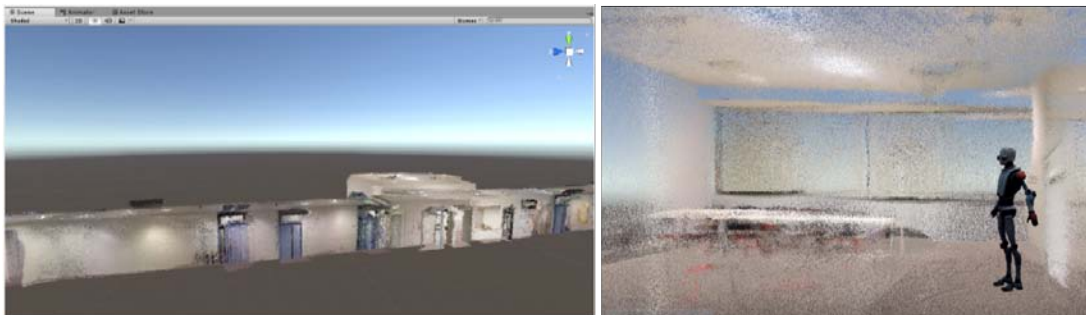


Figure 13: Building-Scale Virtual Reality.

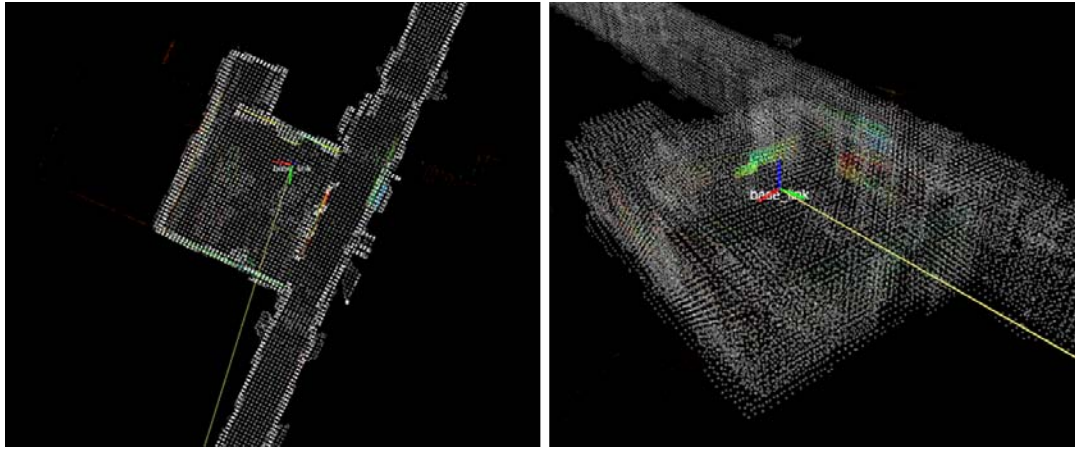


Figure 14: 3D Map for Self-Driving Wheelchair.

problem of determining the optimal observation position, called the “next best view problem,” is solved by a more general method that combines deep learning and reinforcement learning.

The problem of selecting the optimal position in a complex indoor environment was difficult to generalize, but it became a problem that could be solved by introducing a model trained by reinforcement learning.

3D data on a building was collected by using the proposed method and compared with the conventional method. As a result, a clear improvement was observed.

Although 3D measurement of an entire indoor area is very expensive, the robot used in this study can be easily manufactured at low cost.

In addition, the simulator used for training in reinforcement learning is based on a 3D point cloud that has already been collected. By improving the accuracy of the proposed system of this research, a more advanced simulator can be created. This would create a positive cycle and contribute to the generation of many high-quality indoor 3D maps.

The 3D point cloud generated in this research can also be used for constructing VR spaces and estimating the current location of autonomous driving vehicles.

In the near future, real-world VR will be promoted and linked to various services. This research is basic technology for that purpose.

In addition, not only automatic driving outside of buildings but also inside of buildings will be realized to make fully automatic logistics possible. To that end, it is

necessary to create 3D maps of various indoor facilities by using the proposed method. Technology for automatically generating such basic data that can be used for many purposes will become even more important.

To make the results of reinforcement learning more universal, incorporating more diverse indoor environments into the simulator is necessary. This is a project that would complement the creation of 3D maps for large cities on a global scale. To that end, it is necessary to make the results of this research available to the public so that a large number of scanning robots can be manufactured. As mentioned above, this will contribute significantly to VR services and fully automated logistics, so the ripple effect could be quite large.

REFERENCES

- [1] Adán A, Quintana B, Vázquez AS, Olivares A, Parra E, Prieto S. Towards the Automatic Scanning of Indoors with Robots. *Sensors* 2015; 15: 11551-74. <https://doi.org/10.3390/s150511551>
- [2] Bai S, Chen F, Englot B. Toward Autonomous Mapping and Exploration for Mobile Robots through Deep Supervised Learning. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* 2017; pp.2379-84. <https://doi.org/10.1109/IROS.2017.8206050>
- [3] Banta JE, Wong LR, Dumont C, Abidi MA. A Next-Best-View System for Autonomous 3-D Object Reconstruction. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 2000; 30(5): 589-98. <https://doi.org/10.1109/3468.867866>
- [4] Bircher A, Kamel M, Alexis K, Oleynikova H, Siegwart R. Receding Horizon ‘Next-Best-View’ Planner for 3D Exploration. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)* 2016; pp.1462-8. <https://doi.org/10.1109/ICRA.2016.7487281>
- [5] Charrow B, Kahn G, Patil S, Liu S, Goldberg K, Abbeel P, Michael N, Kumar V. Information-theoretic Planning with Trajectory Optimization for Dense 3D Mapping. *Robotics: Science and Systems (RSS)* 2015; Vol.11. <https://doi.org/10.15607/RSS.2015.XI.003>

- [6] Concha A, Civera J. RGBDTAM: A Cost-effective and Accurate RGB-D Tracking and Mapping System. In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2017; pp.6756-63.
<https://doi.org/10.1109/IROS.2017.8206593>
- [7] Foissotte T, Stasse O, Escande A, Wieber P-B, Kheddar A. A Two-Steps Next-Best-View Algorithm for Autonomous 3D Object Modeling by a Humanoid Robot. In Proc. of IEEE International Conference on Robotics and Automation (ICRA) 2009; pp.1159-64.
<https://doi.org/10.1109/ROBOT.2009.5152350>
- [8] Grisetti G, Stachniss C, Burgard W. Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling. In Proc. of the IEEE International Conference on Robotics and Automation (ICRA) 2005; pp.2432-7.
- [9] Krainin M, Curless B, Fox D. Autonomous Generation of Complete 3D Object Models using Next Best View Manipulation Planning. In Proc. of IEEE International Conference on Robotics and Automation (ICRA) 2011; pp.5031-7.
<https://doi.org/10.1109/ICRA.2011.5980429>
- [10] Laidlow T, Bloesch M, Li W, Leutenegger S. Dense RGB-D-Inertial SLAM with Map Deformations. In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2017; pp.6741-8.
<https://doi.org/10.1109/IROS.2017.8206591>
- [11] Lin T-Y, Goyal P, Girshick R, He K, Dollár P. Focal Loss for Dense Object Detection. In Proc. of the IEEE International Conference on Computer Vision (ICCV) 2017; pp. 2980-8.
<https://doi.org/10.1109/ICCV.2017.324>
- [12] Millane A, Taylor Z, Oleynikova H, Nieto J, Siegwart R, Cadena C. C-blox: A Scalable and Consistent TSDF-based Dense Mapping Approach. In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2018; pp.995-1002.
<https://doi.org/10.1109/IROS.2018.8593427>
- [13] Move base. http://wiki.ros.org/move_base. (accessed September 29, 2019).
- [14] Nagao K, Yang M, Miyakawa Y. Building-Scale Virtual Reality: Reconstruction and Modification of Building Interior Extends Real World. International Journal of Multimedia Data Engineering and Management 2019; 10(1): 1-21.
<https://doi.org/10.4018/IJMDEM.2019010101>
- [15] OpenAI. <https://gym.openai.com/envs/>. (accessed September 29, 2019).
- [16] Potthast C, Sukhatme GS. A Probabilistic Framework for Next Best View Estimation in a Cluttered Environment. Journal of Visual Communication and Image Representation 2014; 25(1): 148-64.
<https://doi.org/10.1016/j.jvcir.2013.07.006>
- [17] Romanoni A, Fiorenti D, Matteucci M. Mesh-based 3D Textured Urban Mapping. In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2017; pp.3460-6.
<https://doi.org/10.1109/IROS.2017.8206186>
- [18] Rusu RB, Cousins S. 3D is Here: Point Cloud Library (PCL). In Proc. of the IEEE International Conference on Robotics and Automation (ICRA) 2011; pp.1-4.
<https://doi.org/10.1109/ICRA.2011.5980567>
- [19] Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. Proximal Policy Optimization Algorithms. arXiv preprint arXiv: 1707.06347 2017.
- [20] Shen S, Michael N, Kumar V. Obtaining Liftoff Indoors: Autonomous Navigation in Confined Indoor Environments. IEEE Robotics & Automation Magazine 2013; 20(4): 40-8.
<https://doi.org/10.1109/MRA.2013.2253172>
- [21] Sutton RS, McAllester DA, Singh SP, Mansour Y. Policy Gradient Methods for Reinforcement Learning with Function Approximation. In Proc. of Advances in Neural Information Processing Systems 2000; pp.1057-63.
- [22] Tateno K, Tombari F, Navab N. When 2.5D is not enough: Simultaneous Reconstruction, Segmentation and Recognition on Dense SLAM. In Proc. of IEEE International Conference on Robotics and Automation (ICRA) 2016; pp.2295-302.
<https://doi.org/10.1109/ICRA.2016.7487378>
- [23] Wang K, Ding W, Shen S. Quadtree-Accelerated Real-Time Monocular Dense Mapping. In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2018; pp.1-9.
<https://doi.org/10.1109/IROS.2018.8594101>
- [24] Yang S, Zhu X, Nian X, Feng L, Qu X, Mal T. A Robust Pose Graph Approach for City Scale LiDAR Mapping. In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2018; pp.1175-82.
<https://doi.org/10.1109/IROS.2018.8593754>

Received on 24-09-2019

Accepted on 15-10-2019

Published on 15-11-2019

[DOI: https://doi.org/10.31875/2409-9694.2019.06.2](https://doi.org/10.31875/2409-9694.2019.06.2)

© 2019 Yang and Nagao; Zeal Press

This is an open access article licensed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted, non-commercial use, distribution and reproduction in any medium, provided the work is properly cited.