Automatic Estimation of the Position and Orientation of Stairs to Be Reached and Climbed by a Disaster Response Robot by Analyzing 2D Image and 3D Point Cloud

Kazuya Miyakawa[1], Takuya Kanda[2], Jun Ohya[1], Hiroyuki Ogata[3], Kenji Hashimoto[4], and Atsuo Takanishi[1]
[1]Department of Modern Mechanical Engineering, Waseda University, Tokyo, Japan
[2]Department of Integrative Bioscience and Biomedical Engineering, Waseda University, Tokyo, Japan
[3]Department of System Design Engineering, Seikei Univeristy, Tokyo, Japan
[4]Department of Mechanical Engineering Informatics, Meiji University, Kanagawa, Japan
Email: alfred-kazu4869@asagi.waseda.jp

***Abstract*—In order to realize a disaster response robot that can reach and climb straight stairs within a certain range, this paper proposes a method for estimating the position and orientation of the stairs using 2D image and 3D point cloud. In this method, first, an object detection method is applied to an RGB image, and a 3D point cloud including stairs is extracted by combining the detection result and the 3D point cloud. Next, a 3D point cloud of a step candidate is extracted by applying plane estimation and region segmentation to the extracted 3D point cloud. The 3D point cloud of the step candidate is projected on a 2D plane, and the orientation of the stairs is estimated by detecting their contour and lines. In addition, the position of the stairs is estimated by searching for a combination of 3D point clouds of the step candidates located at equal intervals using the structural characteristics of the stairs. As a result of simulation using a disaster response robot WAREC-1, it was confirmed that the orientation of the stairs can be accurately estimated by the proposed method. It was also confirmed that the position could be accurately estimated under specific conditions.**

***Index Terms*—disaster response robot, reaching stairs, climbing stairs, object detection, 3D point cloud processing**

I. Introduction

Japan is one of the most disaster-prone countries where various disasters such as large-scale earthquakes and nuclear power plant accidents have occurred. When these disasters occur, investigating the damage situation and performing recovery work is strongly requested. However, disaster sites such as collapsed factories and high-radiation dose nuclear power plants are environments where there is a great danger for people to enter and work. Therefore, the demand for disaster-response robots that perform recovery work and rescue activities on behalf of humans at disaster sites has increased, from the viewpoint of preventing secondary disasters and reducing the work load at those dangerous sites.

In response to this growing demand, Waseda University has started to develop a four-limbed robot WAREC-1 [1] that aims at working in disaster sites. In order to perform various task there, disaster response robots are expected to have intelligent mobility including reaching or climbing stairs. In the DARPA Robotics Challenge (DRC) [2], a competition for rescue robots held in the United States in 2015, reaching and climbing straight stairs was set as one of the tasks. In DRC, the operator recognizes the straight stairs via an external sensor (RGB camera or Lidar) mounted on the robot. In addition, remote operation using a graphical operation interface supports reaching and climbing straight stairs. However, a disaster response robot, which is expected to operate in environments where the communications are unstable, requires autonomous operation based on sensing that does not rely on the operator. In this paper, we propose a method for estimating the position and orientation of the stairs based on 2D image and 3D point cloud analysis in order to realize an autonomous reaching and climbing stairs for a disaster response robot. In this paper, the straight stairs are targeted for reaching and climbing.

II. Proposed Method

In this section, we describe our proposed method for estimating the position and orientation of stairs. In this method, the inputs are an RGB image obtained by an RGB camera and a 3D point cloud obtained by a depth camera or a Lidar. Fig. 1 shows the processing flow of the proposed method. Steps 1 to 6 describe the processing common to position and orientation estimation. Steps 7 to 9 describe the processing unique to orientation estimation. Steps 10 to 12 describe the processing unique to position estimation. Steps 1 to 12 correspond to (1) to (12) in the figure.

Manuscript received September 5, 2019; revised August 5, 2020.

doi: 10.18178/ijmerr.9.9.1312-1321

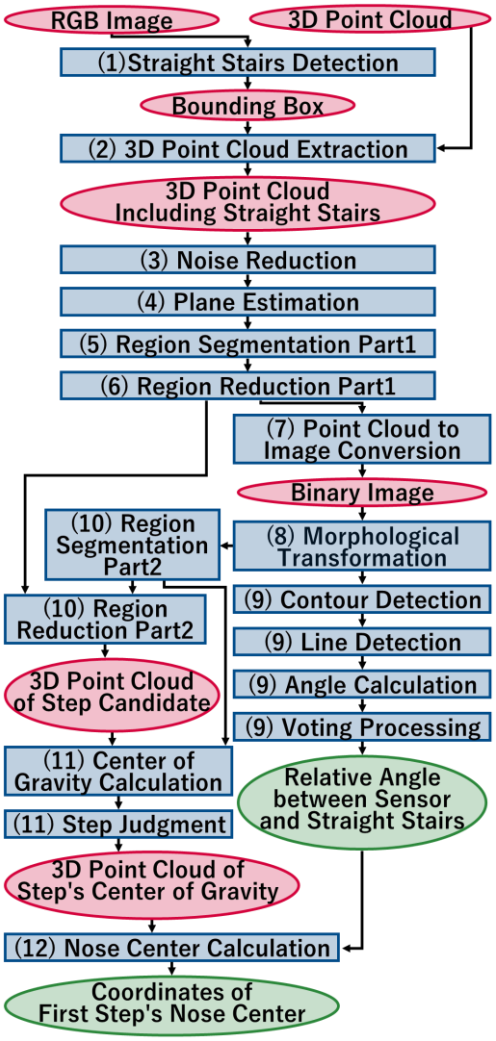


Figure 1. The processing flow of estimating the position and orientation of stairs.

Step 1) Stairs detection using single shot multibox detector

In this paper, the stairs are assumed to be in an indoor environment and its position, orientation and shape are unknown. As a related work on stairs detection, Shimakawa et al. [3] proposed a method in which line detection was applied to RGB and Depth images to detect stairs from lines corresponding to the edges of steps. However, in the environment assumed in this paper, there is a possibility that part of the edge of the steps may not be acquired, and it is difficult to detect stairs by their method. Therefore, in this research, in order to detect stairs of unknown positions, orientations and shapes from RGB images, object detection using a convolutional neural network (CNN) that learns general features of objects from a large amount of image data was used. In particular, we used the Single Shot MultiBox Detector (SSD) [4], which has high detection speed and accuracy. The original SSD is designed to be able to detect a total of 20 classes of objects, such as motorcycles, cats, and sofas, using PASCAL VOC [5]. As these 20 classes don't include stairs, we fine-tuned the original SSD using a new data set to enable detection of them.

Step 2) Extraction of 3D point cloud including stairs using frustum culling

As a related work on 3D object recognition, Fukai et al. [6] proposed a method for recognizing a target object by applying Iterative Closest Point (ICP) [7] to the 3D point cloud and the 3D CAD model prepared in advance. However, as described in Step 1, we target stairs with unknown shape, and it is impossible to use ICP that requires a 3D CAD model of the target object. Therefore, in this research, Frustum Culling [8] was used to extract the 3D point cloud that includes the stairs from the input 3D point cloud. In this method, first, the angle of view that includes the stairs is calculated based on the horizontal and vertical angle of view of the RGB camera and the coordinates of the bounding box estimated by the SSD. Next, based on the calculated angle of view, a frustum with the RGB camera as the origin is generated. Finally, by picking the points inside the frustum, the 3D point cloud that includes the stairs can be extracted.

Step 3) Noise removal using statistical outlier removal

The 3D point cloud normally contains noise, which has a significant impact on the estimation of the position and orientation of the stairs. Therefore, we used Statistical Outlier Removal [9] to remove such noise from the 3D point cloud including the stairs extracted by Frustum Culling. In this method, first, for each point of the input 3D point cloud, the average value and standard deviation of the distance from an arbitrary number of neighbor points are calculated. Next, the distance threshold is determined based on the calculated average value and standard deviation of the distance. Finally, the points whose distances are larger than the threshold are removed as noise.

Step 4) Plane estimation using random sample consensus

The stairs are composed of multiple steps whose planes are parallel to the horizontal plane, and the position and orientation of the stairs can be estimated from the 3D point cloud of the steps. Here, we used Random Sample Consensus (RANSAC) [10] to extract the 3D point cloud that is a step candidate from 3D point cloud obtained in Step 3. In this method, first, three points are randomly selected from the input 3D point cloud, and a temporary 3D plane is calculated. Next, the plane is applied to the input 3D point cloud, and the number of points that fit the plane is recorded. This process is repeated a specified number of times, and the plane that fits the most points is output as the most applicable 3D plane. Finally, by picking the points that fit the output plane, the 3D point cloud that is a step candidate can be extracted.

Step 5) Region segmentation using euclidean clustering

RANSAC outputs multiple regions on the same plane as one region due to the nature of the algorithm. When the 3D point cloud of the step and others are on the same plane, they need to be separated. Therefore, Euclidean Clustering was used to divide them into each region. In this method, the distance between each point of the input

3D point cloud is calculated and points less than the threshold of any distance are assigned to the same cluster. This process is applied to all points in the input 3D point cloud. By assigning all points to clusters, the 3D point cloud of step candidates is divided into regions.

Step 6) Region removal based on normal vectors

The normal vector of each region was used to remove the region of the inappropriate step candidate from the 3D point cloud obtained in Step 5. Here, we first focus on the fact that the normal direction of the step is almost vertical, and estimate the normal vector of each region. Next, the angle between the estimated normal direction and the vertical direction is calculated. Finally, the region where the calculated angle is equal to or larger than the threshold value of the arbitrary angle is removed from the input 3D point cloud as an inappropriate step candidate region.

Step 7) Generation of binary image of 3D point cloud using projection processing

As a related work on estimating orientation of the stairs, Patill et al. [11] proposed a method of applying line detection to RGB image. In their paper, the midpoint of the line corresponding to the edge of the step is calculated for each step, and the orientation of the stairs is estimated from the line passing through the midpoint of the edges of the plurality of steps. However, as described in Step 1, in the environment assumed in this paper, it may not be possible to acquire a part of the edge of the step, and it is not appropriate to use their method here. Therefore, the orientation of the stairs was estimated from the slope of the edge of the step instead. However, the slope on an RGB image that does not include depth information does not represent the true slope of the step. So, we estimated the orientation from the 3D point cloud of the step candidate. In order to estimate the orientation of the stairs from the 3D point cloud obtained in Step 6, the 3D point cloud is converted into a binary image here. We focused on the fact that the steps of the stairs are almost horizontal. Since the orientation estimation doesn't use the height of the steps and the relationship between steps, the subsequent processing can be performed not only from 3D point cloud processing but also from 2D image processing. In this method, first, the input 3D point cloud is projected on a horizontal plane for each region. Next, a binary image is generated by plotting each point of the projected 3D point cloud with pixel values of 255 on an image initialized with all pixel values of 0. The white pixel region in the binary image generated by the above processing represents the shape of the step of the stairs when viewed from directly above. And, the angle between the edge of the step and the horizontal direction when viewed from directly above represents the orientation of the stairs. Therefore, the orientation of the stairs can be estimated by detecting the line of the edge of the step from the generated binary image and calculating the angle formed with the horizontal direction.

Step 8) Noise removal using morphological transformation

In order to remove noise from the generated binary image, morphological transformations such as closing and opening were performed. Closing is a process in which erosion is performed after dilation, and removes noise in the white pixel region. Meanwhile, opening is a process in which dilation is performed after erosion, and removes noise in the black pixel region. Here, after applying the closing to the binary image, the opening is applied to remove the noise.

Step 9) Orientation estimation using line detection

In order to estimate the orientation of the stairs from the binary image cleaned in Step 8, processing such as contour detection, line detection, angle calculation, and voting was performed. Here, we focus on the line of the edge of the step, and in particular, use the horizontal line longer than the depth direction and easier to detect for the orientation estimation. First, the contour of the white pixel region in the input binary image is detected. Next, a line that fits the detected contour is extracted using stochastic Hough transform. Then, the elevation angle of the line is calculated, and is voted in a ballot box provided for each 1.0[degree]. Finally, the average value of the angles stored in the ballot box with the largest number of votes is output as the orientation of the stairs.

Step 10) Region removal based on comparison between binary image and 3D point cloud

In order to remove the region of the inappropriate step candidate from the 3D point cloud obtained in Step 6, the 3D point cloud of the step candidates and the binary image derived in Step 8 are used. Here, we focus on the fact that the *xy* components of the 3D point cloud of the step candidate obtained in Step 6 correspond to the *xy* components of the binary image generated in Step 7. Here, we first apply labeling to remove the effect of the slight noise remaining in the input binary image. Here, the label of the region with the largest area is determined as the target label. Next, based on the *xy* components of the pixels in the region to which the target label is assigned on the binary image, a 3D point cloud corresponding to the target label region is extracted from the input 3D point cloud. This process has the same effect as applying the noise removal using the morphological transformation to the input 3D point cloud.

Step 11) Judgment of steps based on structural characteristics of stairs

In order to extract the 3D point cloud of the step from the 3D point cloud obtained in Step 10, the step was judged based on the structural characteristics of the stairs. The structural characteristic used here is that the steps of the stairs are arranged at equal intervals in the directions of the xyz axes. First, the center of gravity α of the region of interest label acquired in Step 10 and the center of gravity β of each region of the input 3D point cloud are calculated. Then, the 3D coordinates obtained by combining the *xy* component of α and the *z* component of β are recorded as the representative points of the step

candidates. A depth camera and Lidar get different 3D point cloud densities depending on the distance to the target object. When the density of the 3D point cloud of the step is biased, the *xy* component of β deviates from the center of gravity of the step in the direction of higher density. On the other hand, the density of white pixels in the region of the target label acquired in Step 10 is almost constant because of the passing of the closing in Step 8. Therefore, we consider that the *xy* component of α is less affected by the density bias than the *xy* component of β. Thus, we adopted the method of calculating the representative point combining α and β. Next, search for combinations that are arranged at equal intervals in the directions of the *xyz* axes from all the recorded representative points. Finally, the 3D point cloud to which each representative point of the search result belongs is output as the 3D point cloud of the step.

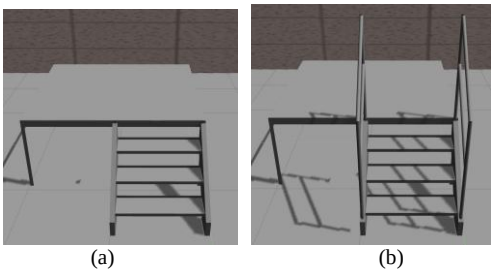
Step 12) Position estimation based on center of gravity of step and orientation of stairs

In order to estimate the position of the stairs from the center of gravity of each step and the 3D point cloud of each step obtained in Step 11, the orientation of the stairs estimated in Step 9 was used. Here, we focus on the fact that the orientation of the stairs represents the angle of the line of the edge of the step. First, a line that passes through the center of gravity and is orthogonal to the line of the edge of the step is obtained from the center of gravity of the first step and the orientation of the stairs. Next, the calculated line is applied to the 3D point cloud of the first step, and the points on the line are picked. Finally, from the picked points, the point closest to the sensor is output as the position of the stairs.

III. Experiments and Discussion

This section describes the experiments performed to verify the effectiveness of the proposed method, followed by its results and considerations. Section A explains the experimental environment. Section B illustrates an example of applying the processing of the proposed method. Section C denotes the results of the orientation and position estimation of the stairs. In Section D, we discuss the experimental results.

A. Experimental Environment



(a) (b)

Figure 2. Stairs used for experiments. (a) Without handrails (b) With handrails.

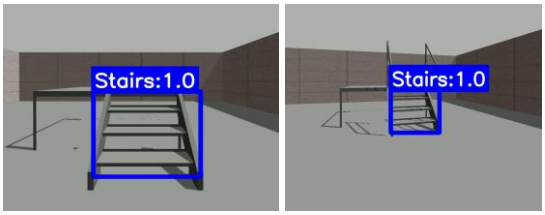
The experiment was performed in a virtual environment developed in the GAZABO simulator [12]. It takes the RGB image and the 3D point cloud as inputs acquired by Multisense SL, which is a sensor equipped with an RGB camera and Lidar. Two types of straight stairs were prepared for the experiment. The difference of the two stairs is the presence or the absence of handrails. The appearances of these stairs are shown in Fig. 2(a) and (b).

B. Example of Applying the Processing of the Proposed Method

Step 1) Stairs detection using SSD

The SSD was fine-tuned using an original data set to enable detection of stairs. Here, the VGG16 model [13] trained by ImageNet [14] is used as the feature extractor. The data set was created by collecting 3523 RGB images of stairs from ImageNet and Open Images Dataset v4 [15] and manually annotating them. 80% of the data set was used as training data and the rest as test data.

In this experiment, the stairs were successfully detected in all 32 patterns. Fig. 3(a) and (b) show examples of detection using SSD. The value following the word "Stairs" in Fig. 3 indicates the *confidence* that signifies how convinced the SSD judged the object in the bounding box to be stairs. In this experiment, we regard the detection as successful when the confidence is over 0.9.

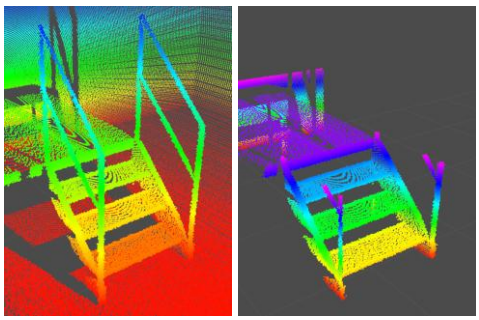


(a) (b)

Figure 3. Stairs detection using SSD. (a) Stairs without handrails / Position1 / Orientation0 (b) Stairs with handrails / Position2 / Orientation15.

Step 2) Extraction of 3D point cloud including stairs using frustum culling

Fig. 4 shows an extraction example of 3D point cloud including stairs using frustum culling. Fig. 4(a) represents a 3D point cloud acquired by Lidar. Fig. 4(b) is the extraction result of a 3D point cloud that includes the stairs by Frustum Culling. In Fig. 4(a) and (b), the color is changed based on the value of the z component of the 3D point cloud.

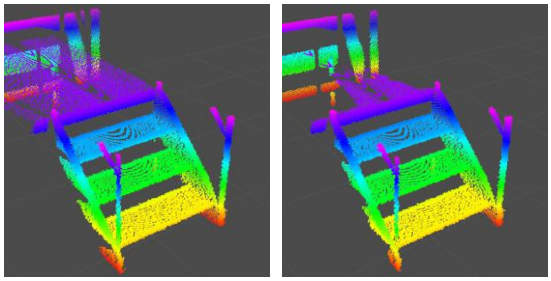


(a) (b)

Figure 4. Extraction of 3D point cloud including stairs using frustum culling. (a) A 3D point cloud acquired by Lidar. (b) A 3D point cloud that includes the stairs extracted by Frustum Culling.

Step 3: Noise removal using Statistical Outlier Removal

Fig. 5 shows an example of removing noise using Statistical Outlier Removal. Fig. 5(a), identical to Fig. 4(b), is shown again for convenience. Fig. 5(b) is the result of removing noise.

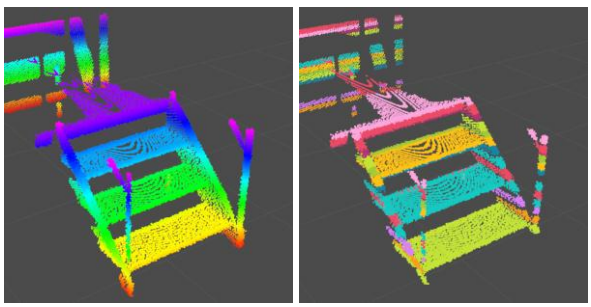


(a) (b)

Figure 5. Noise removal using Statistical Outlier Removal. (a) A 3D point cloud that includes the stairs extracted by Frustum Culling. (b) A 3D point cloud from which noise has been removed by Statistical Outlier Removal.

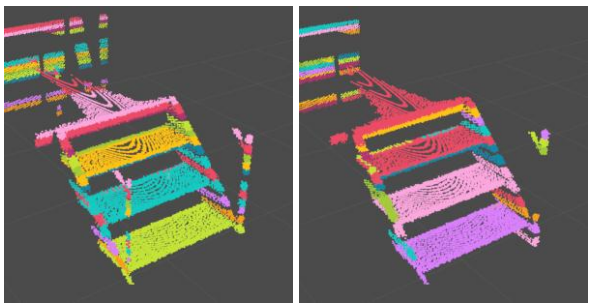
Step 4) Plane estimation using RANSAC

In this experiment, in the random sampling, considering that the stair steps are almost horizontal, we picked three points where the error of the elevation angle of the temporary plane was within 2.5[degree]. In addition, we regard the points whose distance from the plane is within 3.0[cm] to be fitted to the plane, taking into account the measurement error of the Lidar. Fig. 6(a) shows the 3D point cloud identical to Fig. 5(b). Fig. 6(b) displays a 3D point cloud of the step candidate extracted by RANSAC. Planes of the extracted step candidate are represented by different colors.



(a) (b)

Figure 6. Plane estimation using RANSAC. (a) A 3D point cloud from which noise has been removed by Statistical Outlier Removal. (b) A 3D point cloud of the step candidate extracted by RANSAC.



(a) (b)

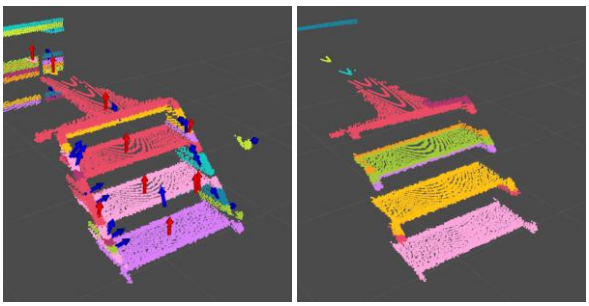
Figure 7. Region segmentation using Euclidean Clustering. (a) A 3D point cloud of the step candidate extracted by RANSAC. (b) A 3D point cloud of the step candidate divided by Euclidean Clustering.

Step 5) Region segmentation using Euclidean Clustering

Here, the distance threshold was also set to 3.0[cm] in consideration of the Lidar measurement error. Fig. 7(a) is the same as Fig. 6(b), and Fig. 7(b) shows a 3D point cloud of the step candidate in Fig. 7(a) divided by Euclidean Clustering. Generated clusters are again represented by different colors.

Step 6) Region removal based on normal vectors

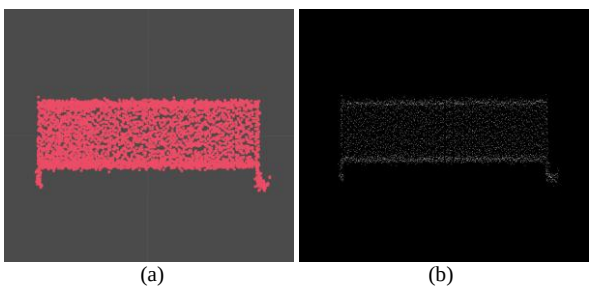
The threshold of the angle was set to 15.0[degree] to determine the regions of the inappropriate step candidate. Fig. 8(a) and Fig. 8(b) show the region removal based on normal vectors. Fig. 8(a) is the clustering result in Fig. 7(b). Red arrows are added to indicate normal lines of the regions where the angle between the normal direction and the vertical direction is less than the threshold value. Blue arrows are also added to indicate the normal lines of the regions where the above angle is equal to or more than the threshold value. Here, the points belonging to the regions with the blue arrow normals are removed as inappropriate step candidates. Fig. 8(b) shows a 3D point cloud obtained by removing these regions.



(a) (b)

Figure 8. Region removal based on normal vectors. (a) A 3D point cloud of the step candidate divided by Euclidean Clustering. (b) A 3D point cloud obtained by removing the region based on the normal vector. Red arrows: Normal lines of the regions where the angle between the normal direction and the vertical direction is less than the threshold value. Blue arrows: Normal lines of the regions where the above angle is equal to or more than the threshold value. The points belonging to the regions with the blue arrow normal are removed as inappropriate step candidates.

Step 7) Generation of binary image of 3D point cloud using projection processing



(a) (b)

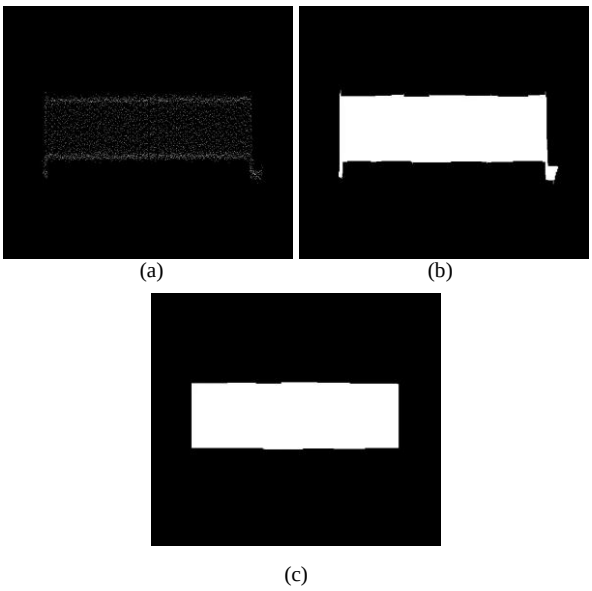
Figure 9. Generation of binary image of 3D point cloud using projection processing. (a) A 3D point cloud of the step candidate projected on a horizontal plane. (b) A binary image generated by plotting (a).

In generating a binary image, the projected 3D point cloud was plotted on the image on a scale of 2[mm/pixel]. This means that a binary image with a horizontal

dimension of 500[pixel] is generated from a 3D point cloud of a step with a width of 1.0[m] in the horizontal direction. The smaller the scale is, the higher is the resolution of the binary image. However, if the angular resolution of the Lidar is coarse, the white pixels inside the step region become sparse, and the number of times the closing in Step 8 is applied increases. Fig. 9(a) shows a 3D point cloud of the step candidate projected on the horizontal plane. Fig. 9(b) shows a binary image generated by plotting (a).

Step 8) Noise removal using morphological transformation

Fig. 10(a) to (c) show the noise removal using morphological transformation. Here, the closing and opening were performed 25 times each. Fig. 10(a) is the generated binary image in Fig. 9(b). Fig. 10(b) and (c) are the results of noise removal by closing and opening, respectively.



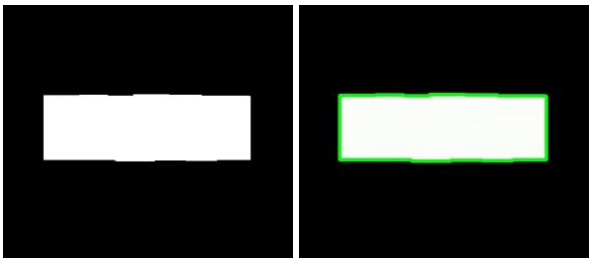
(a) (b)

(c)

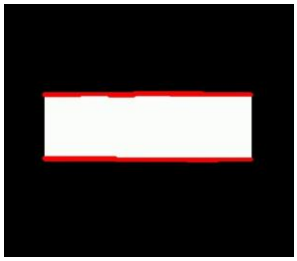
Figure 10. Noise removal using morphological transformation. (a) A generated binary image. (b) Noise removal by closing. (c) Noise removal by opening.

Step 9) Orientation estimation using line detection

Fig. 11(a) to 11(c) show the result of estimating orientation using line detection. Fig. 11(a) is the output binary image in Step 8. Green lines in Fig. 11(b) represent the contour of the region of the step candidate. Red lines in Fig. 11(c) represent those in the horizontal direction among the contour lines.



(a) (b)

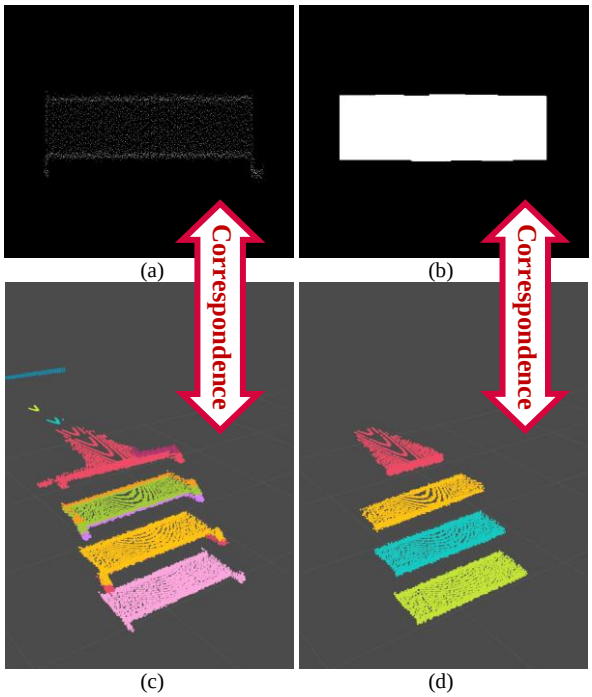


(c)

Figure 11. Orientation estimation using line detection. (a) A binary image from which noise has been removed by morphological transformation. (b) An RGB image in which the detected contour is superimposed on (a). (c) An RGB image in which the detected line is superimposed on (a).

Step 10) Region removal based on comparison between binary image and 3D point cloud

Fig. 12(a) to (d) show the region removal based on comparison between binary image and 3D point cloud. Images in Fig. 12(a) and (b) are outputs of Steps 7 and 8, respectively. Fig. 12(c) shows a 3D point cloud obtained by removing the region based on the normal vector. Fig. 12(d) illustrates a 3D point cloud obtained by removing the region based on the comparison between the binary image and the 3D point cloud. In Fig. 12(d), it can be observed that the protrusions at both ends of the step that existed in Fig. 12(c) were removed. This is the result of extracting the 3D point cloud corresponding to Fig. 12(b) from the 3D point cloud of Fig. 12(c).



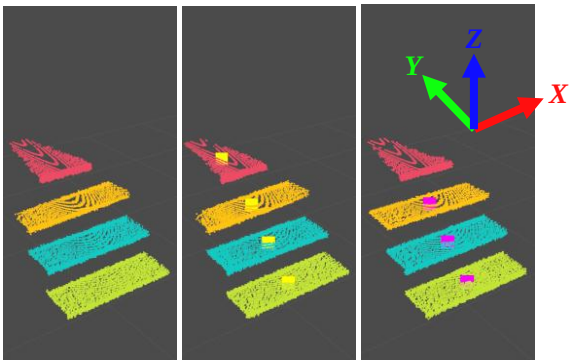
(a) (b)

(c) (d)

Figure 12. Region removal based on comparison between binary image and 3D point cloud. (a) A generated binary image. (b) A binary image from which noise has been removed by opening. (c) A 3D point cloud obtained by removing the region based on the normal vector. (d) A 3D point cloud obtained by removing the region based on the comparison between the binary image and the 3D point cloud. The white pixels in (a) and (b) correspond to the 3D point clouds in (c) and (d).

Step 11) Judgment of steps based on structural characteristics of stairs

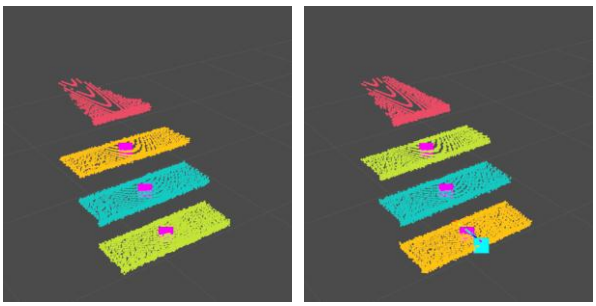
The combinations of representative points are searched by limiting the range of each axis. We referred to the dimensions of the stairs specified in the Japanese Building Standard Law Enforcement Ordinance, Section 3, Article 23 [16] to determine the range. According to literature [16], the depth (*y*-axis) of the step should be15[cm] or more, and the interval in the height of the step (*z*-axis) should be 23[cm] or less. Fig. 13(a) to (c) show the judgment of steps based on structural characteristics of stairs. Fig. 13(a) is the output 3D point cloud of Step 10. Yellow points in Fig. 13(b) are the representative points of the 3D point cloud of the step candidate. Magenta points in Fig. 13(c) are the representative points of the 3D point cloud of the step.



(a) (b) (c)

Figure 13. Judgment of steps based on structural characteristics of stairs. (a) A 3D point cloud obtained by removing a region based on a comparison between a binary image and a 3D point cloud. (b) A 3D point cloud in which the representative points of the 3D point cloud of the step candidate are superimposed on (a). (c) A 3D point cloud in which representative points of the 3D point cloud of the step are superimposed on (a). Yellow points: The representative points of the 3D point cloud of the step candidate. Magenta points: The representative points of the 3D point cloud of the step.

Step 12) Position estimation based on center of gravity of step and orientation of stairs



(a) (b)

Figure 14. Position estimation based on center of gravity of step and orientation of stairs. (a) A 3D point cloud in which the representative points of the 3D point cloud of the step are superimposed and displayed on the 3D point cloud obtained by removing the region based on the comparison between the binary image and the 3D point cloud. (b) A 3D point cloud in which the position of the stairs by estimating the position based on the center of gravity of the first step and the orientation of the stairs is superimposed on (a). Rainbow line: A 3D point cloud on a line that passes through the center of gravity of the first step and is orthogonal to the edge of the step. Cyan point: The coordinates of the point that is closest to the sensor in the 3D point cloud on the rainbow line, and represents the position of the stairs.

According to the representative points of the step obtained in Step 10 (displayed again as Fig. 14(a)), the position and the orientation of the stairs are estimated. Fig. 14(b) shows the result of the estimation. The rainbow line in Fig. 14(b) represents a 3D point cloud on a line that passes through the center of gravity of the first step and is orthogonal to the edge of the step. The cyan point in Fig. 14(b) is the coordinates of the point that is closest to the sensor in the 3D point cloud on the rainbow line, and represents the position of the stairs.

C. Experiment of Estimating the Position and Orientation of Stairs

To roughly approach to the stairs and bring it into sight, the disaster response robot WAREC-1 first moves along the path generated from the map information of the floor plan. As the motion of WAREC-1 is not precise enough, there will be normally a gap between the actual and the ideal goals of the robot. In the experiment, we assume that the maximum distance between WAREC-1 and stairs is 3.0[m] and the maximum orientation gap is 45[degree]. From this assumption, a total of 16 combinations of position and orientation within the range were applied as initial state of the robot to each of the two types of stairs in Fig. 3, and the relative position and orientation of the stairs were estimated. The Multisense SL was set at a height of 1.0[m] from the ground, which is the same height as the sensor actually installed on the body of WAREC-1. In this experiment, the center coordinates of the nose (the tip of the step) of the first step of the stairs are defined as "stair position", and the relative angle between Multisense SL and the stairs is defined as "stair orientation". Fig. 15 shows the pattern of the position and orientation of the stairs. The green point in Fig. 15 indicates the position of Multisense SL, the red point indicates the position of the stairs, and the blue range indicates the orientation of the stairs. Here, the positions of Multisense SL are labeled as Position1-Position4 as shown in Fig. 15. The orientations of the stairs are also labeled as Orientation0-Orientation45.

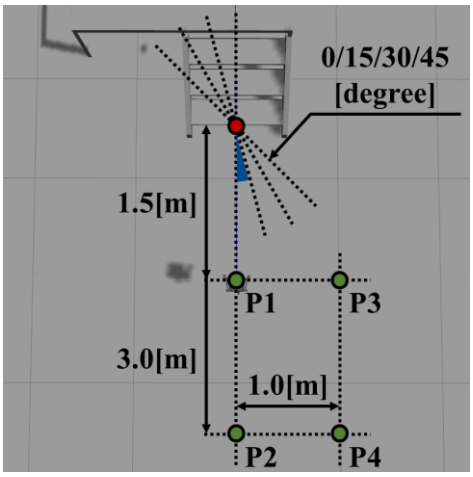


Figure 15. The pattern of the position and orientation of the stairs. Red point: The position of the stairs. Green points: The positions of Multisense SL. (P1-P4: Position1-Position4). Blue ranges: The orientations of the stairs. (0-45: Orientation0-Orientaion45).

①Result of estimating the orientation of stairs

In this experiment, the orientations of the stairs were successfully estimated in all the 32 patterns. Table I summarizes the errors in estimating the orientation of the stairs. These errors are the absolute value of the difference between the true value in the orientation of the stairs and the estimated value, and the unit of the error is [degree]. Table I shows that the orientation of the stairs can be estimated with an error of less than 0.3[degree].

TABLE I. THE ERROR [DEGREE] IN ESTIMATION OF THE ORIENTATION

Position pattern		Orientation pattern			
		0	15	30	45
Stairs without handrail	Position1	0.04660	0.05240	0.02847	0.008616
	Position2	0.01511	0.1977	0.07508	0.005113
	Position3	0.1516	0.1575	0.1766	0.02176
	Position4	0.1117	0.2320	0.08965	0.005148
Stairs with handrail	Position1	0.03320	0.1172	0.02649	0.003610
	Position2	0.06696	0.2024	0.1651	0.06099
	Position3	0.06618	0.2280	0.2245	0.003496
	Position4	0.2323	0.2803	0.08936	0.01087

②Result of estimating the position of stairs

The position of stairs was successfully estimated in 19 out of the 32 patterns. Table II summarizes the errors in estimating the position of the stairs. These errors derive from the Euclidean distance between the true value of the position of the stairs and the estimated value, and the unit is [mm]. Table II shows that the position estimation tends to fail in Position2 and Position4. Moreover, the errors in Position3 and Orientations15,30,45 are prone to be large.

TABLE II. THE ERROR [MM] IN ESTIMATION OF THE POSITION

Position pattern		Orientation pattern			
		0	15	30	45
Stairs without handrail	Position1	20.45	19.56	22.83	46.68
	Position2	19.73	Failure	Failure	Failure
	Position3	19.93	73.68	107.9	127.4
	Position4	Failure	10.31	Failure	Failure
Stairs with handrail	Position1	16.23	16.29	33.49	Failure
	Position2	16.88	Failure	Failure	Failure
	Position3	12.94	73.26	109.2	129.1
	Position4	Failure	20.36	Failure	Failure

D. Considerations on the Results of Estimating the Position of Stairs

Here, we will consider the causes of errors from three viewpoints

①Vanishing of 3D point cloud on steps out of field by Frustum Culling

In this experiment, Frustum Culling was used to extract the 3D point cloud including the stairs from the 3D point cloud obtained by a Lidar. Fig. 16(a) to (d) show an example of estimating the position of the stairs with handrails partially out of view (Position3 and Orientation45). (a) is the stairs detected using SSD, (b) is the 3D point cloud acquired by Lidar, (c) is the stairs point cloud extracted using Frustum Culling, and (d) is a 3D point cloud in which the position of the stairs (dotted in cyan) and the representative point of the step (dotted in magenta) are superimposed on (b). Here, the generated Frustum does not include the whole stairs. Unlike the orientation estimation using the edge of the step, the position estimation uses the center of gravity of the step. Therefore, if a part of the 3D point cloud of the step is missing, the calculated center of gravity will deviate from the center of gravity of the step, which may lead to an increase in error or failure in position estimation (Fig. 16(d)). This phenomenon is observed in Position3 / Orientation15・30・45. Table II shows that the position estimation error is relatively larger than other patterns.

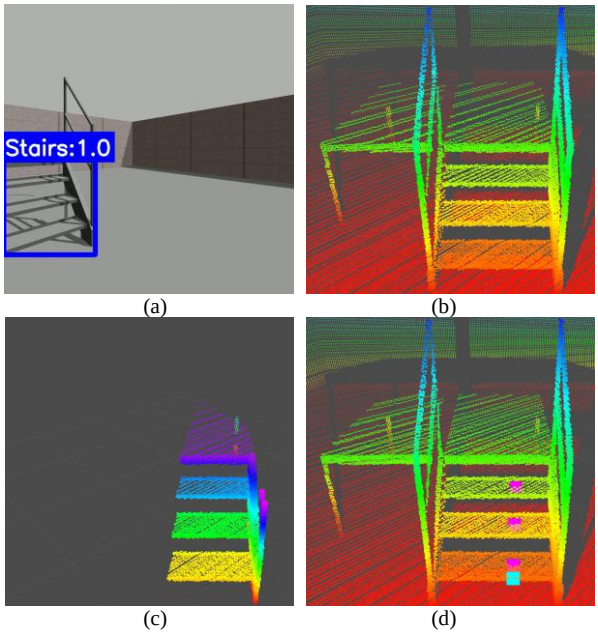


Figure 16. The estimation of the position of the stairs in "Stairs with handrails/Position3/Orientation45". (a) Detection using SSD. (b) A 3D point cloud acquired by Lidar. (c) A 3D point cloud that includes the stairs extracted by Frustum Culling. (d) A 3D point cloud in which the position of the stairs and the representative point of the step are superimposed on (b).

②Vanishing of 3D point cloud on steps with high density difference by Statistical Outlier Removal

In this experiment, Statistical Outlier Removal was used to remove noise from the 3D point cloud including the stairs extracted by Frustum Culling. Fig. 17(a) to (d) show the noise removal using Statistical Outlier Removal. Fig. 17(a) and (b) are an example of applying to stairs without handrails at Position2 and Orientation30, while Fig. 17(c) and (d) are another example of applying at Position2 and Orientation0. (b) and (d) show 3D point clouds from which noise has been removed by Statistical Outlier Removal from (a) and (c). The yellow points in (b) and (d) indicate the representative points of the step candidate. In this method, when the density of the cloud of the step differs a lot as in (a) and (b), the cloud of the step existing in the low-density region is removed as

noise (Fig. 17(b)). As a result, a part of the 3D point cloud of the step is lost, as in ①, which may lead to an increase in errors and a failure in position estimation. This phenomenon can be observed at Position2 and Position4 in Table II where the position estimation failed except for Position2 / Orientation0 and Position4 / Orientation15. On the other hand, when the cloud density of the step is even, as in the example for stairs without handrails at Position2 and Orientation0 (Fig. 17(c) and (d)), the position can be successfully estimated.

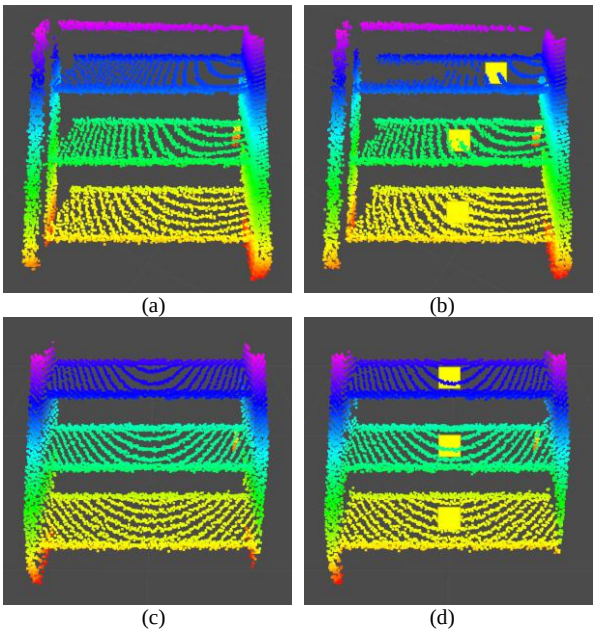


Figure 17. Noise removal using Statistical Outlier Removal. (a)(b) An example of applying to stairs without handrails at Position2 and Orientation30. (c)(d) An example of applying to stairs without handrails at Position2 and Orientation0. (a)(c) A 3D point cloud including the stairs extracted by Frustum Culling. (b)(d) A 3D point cloud from which noise has been removed by Statistical Outlier Removal.

Next, we would like to discuss the relationship between the density of the step point cloud and cleaning noise by Statistical Outlier Removal. Fig. 18(a) to (d) show the noise removal to the 3D point cloud of the third step, which affected the result of position estimation. (a) and (b) are an example of applying to stairs without handrails at Position2 and Orientation30, while (c) and (d) are an example at Position2 and Orientation0. (a) and (c) are point clouds of the third steps extracted by Frustum Culling. (b) and (d) are results of removing noise by Statistical Outlier Removal. In these figures, the step point cloud is divided into five areas (shown in different colors) whose width is 15[cm] in the horizontal direction. The five areas are numbered Area 1 to 5 from left to right as shown in Fig. 18(a). Table III summarizes the number of points of each area before and after the noise removal, and the standard deviation of the number of points before the removal. Two facts can be observed from this table. One is that the larger the standard deviation is, the more the amount of points is removed. The other is that more points are removed in areas with fewer points. Summarizing the above two points, it can be seen that when Statistical Outlier Removal is applied to the area where the density difference of the 3D point cloud is high, the points in the area where the density is low tend to be removed.

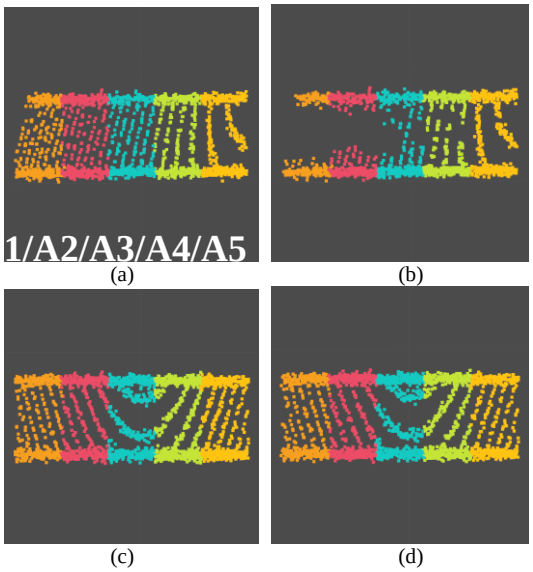


Figure 18. Noise removal using Statistical Outlier Removal to the 3D point cloud of the third step. (a)(b) An example of applying to stairs without handrails at Position2 and Orientation30. (c)(d) An example of applying to stairs without handrails at Position2 and Orientation0. (a)(c) A 3D point cloud including the stairs extracted by Frustum Culling. (b)(d) A 3D point cloud from which noise has been removed by Statistical Outlier Removal.

TABLE III. THE NUMBER OF POINTS PER AREA USING STATISTICAL OUTLIER REMOVAL ON STAIRS WITHOUT HANDRAIL

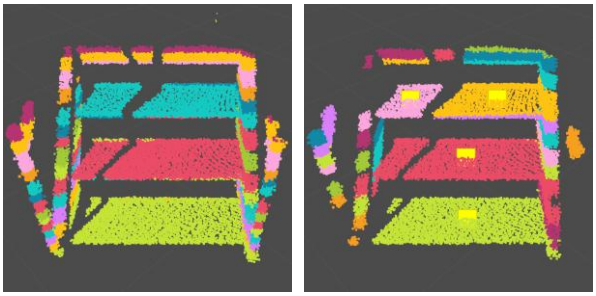
Position/Orientation pattern		Area					Standard Deviation
		1	2	3	4	5	
Position2/ Orientation30	Before	257	318	352	376	395	48
	After	200	263	313	351	395	
	Delta	57	55	39	24	0	
Position2/ Orientation0	Before	364	413	439	425	379	28
	After	364	413	439	425	379	
	Delta	0	0	0	0	0	

③Division of 3D point cloud of steps by occlusion

In this experiment, Euclidean Clustering was used to divide multiple regions existing on the same plane from the 3D point cloud of the step candidate extracted by RANSAC. Fig. 19(a) and (b) show the result of segmentation applying to the stairs with handrails at Position1 and Orientation45. (a) shows a 3D point cloud of the step candidate extracted by RANSAC, and (b) shows a 3D point cloud of the step candidate divided by Euclidean Clustering. In (a), the color of the 3D point cloud is changed for each plane of the extracted step candidate. In (b), the color of the 3D point cloud is changed for each cluster. The yellow point in (b) indicates the representative points of the step candidate. In this method, when stairs with handrails are recognized diagonally (e.g. at Position1 and Orientation45), the occlusion generated by handrails may divide the 3D point cloud of the step, and one step may be assigned to two

clusters by Euclidean Clustering (Fig. 19(b)). This can lead to an increase in errors and a failure in position estimation. This phenomenon was only observed on stairs with handrails at Position1 and Orientation45. Table II shows that the position estimation failed.

Consequently, in order to estimate the position of stairs more accurately using the sensor installed on the disaster response robot, it is necessary to correct the position and orientation of the robot using the detection results and the orientation of the stairs.



(a) (b)

Figure 19. Segmentation of the region using Euclidean Clustering to "the stairs with handrails / Position1 / Orientation45". (a) A 3D point cloud of the step candidate extracted by RANSAC. (b) A 3D point cloud of the step candidate divided by Euclidean Clustering.

IV. Conclusion

In this paper, we propose a method for estimating the position and orientation of stairs based on 2D image and 3D point cloud analysis, so that a disaster response robot can autonomously reach and climb the stairs. First, 3D point cloud including the stairs is extracted by combining the estimated bounding box of the stairs on the RGB image with the 3D point cloud obtained by Lidar. Next, the 3D point cloud of the step candidate is extracted. Then the orientation estimation is performed using the detected lines, and the position estimation is performed by searching for those located at equal intervals of the extracted step candidates. In the experiments using the GAZEBO simulator, it was confirmed that stairs detection and orientation estimation were performed accurately in all combinations of positions and orientations. However, it was confirmed that position estimation was not possible in some patterns. To improve the success rate and accuracy of estimation, the position and orientation of the disaster response robot are corrected based on the results of stair detection and orientation estimation, and the distance between the sensor and the stairs needs to be short and the relative angle needs to be small.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

K.M. carried out the experiment and wrote the paper with support from J.O. and H.O. J.O. and H.O. supervised the research and the paper. T.K. contributed implementation of the research and the technical details. K.H. and A.T. designed and directed the project.

Acknowledgement

The authors of this paper acknowledge to the support of Reina Yoshizaki, Keisuke Namura and Takashi Matsuzawa of Waseda University.

References

[1] K. Hashimoto et al., "WAREC-1 - a Four-limbed robot having high locomotion ability with versatility in locomotion styles," in *Proc. 2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*, pp. 172-178, 2017.

[2] A. Norton et al., "Analysis of human–robot interaction at the DARPA robotics challenge finals," *The International Journal of Robotics Research*, vol.36, (5-7), pp. 483-513, 2017.

[3] M. Shimakawa, S. Murakami, K. Kiyota, "Study on stairs detection using RGB-Depth images," *31st Fuzzy System Symposium*, pp. 699-702, 2015.

[4] W. Liu et al., "SSD: Single shot multibox detector," *European Conference on Computer Vision*, pp. 21-37, 2016.

[5] The PASCAL Visual Object Classes Homepage (last access 2nd January 2020)/ [Online]. Available: http://host.robots.ox.ac.uk/pascal/VOC/

[6] H. Fukai, G. Xu, "Robust and fast 3D object recognition using exhaustive search and ICP algorithm," in *Proc. 37th Annual Conference of the IEEE Industrial Electronics* Society, pp. 4526-4531, 2011.

[7] S. Rusinkiewicz, M. Levoy, "Efficient variants of the ICP algorithm," *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, pp. 145-152, 2001.

[8] U. Assarsson, T. Moller, "Optimized view frustum culling algorithms for bounding boxes," *Journal of Graphics, GPU, & Game Tools*, pp. 9-22, 2000.

[9] Statistical Outlier Removal (last access 2nd January 2020). [Online]. Available: http://pointclouds.org/documentation/tutorials /statistical_outlier.php

[10] R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J. Frahm, "Usac: A universal framework for random sample consensus," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 2022-2038, 2013.

[11] U. Patil et al., "Deep learning based stair detection and statistical image filtering for autonomous stair climbing," in *Proc. 2019 Third IEEE International Conference on Robotic Computing*, pp. 159-166, 2019.

[12] GAZEBO (last access 2nd January 2020) [Online]. Available: http://gazebosim.org/

[13] K. Simonyan, A. Zisserman, "very deep convolutional networks for large-scale image recognition," *International Conference on Learning Representations*, 2014.

[14] ImageNet (last access 2nd January 2020). [Online]. Available: http://www.image-net.org/

[15] Open Images Dataset V4(last access 2nd January 2020). [Online]. Available: https://storage.googleapis.com/openimages /web/download_v4.html

[16] Building Standard Law Enforcement Ordinance (last access 2nd January 2020) [Online]. Available: https://elaws.e-gov.go.jp /search/elawsSearch/elaws_search/lsg0500/detail?lawId=325CO0 000000338#V

Kazuya Miyakawa received B.S. degree from the Department of Modern Mechanical Engineering, the School of Creative Science and Engineering, Waseda University, Japan, in 2018. Currently, he belongs to the at the Department of Modern Mechanical Engineering, Waseda University as an M.S. student. His research areas include image processing, robot vision, etc.