

# Mobile Robot Navigation in Textureless Unknown Environment Based on Plane Estimation by Using Single Omni-directional Camera\*

Kazushi Watanabe<sup>1</sup>, Ryosuke Kawanishi<sup>1</sup>, Atsushi Yamashita<sup>2</sup>, Yuichi Kobayashi<sup>1</sup>,  
Toru Kaneko<sup>1</sup> and Hajime Asama<sup>2</sup>

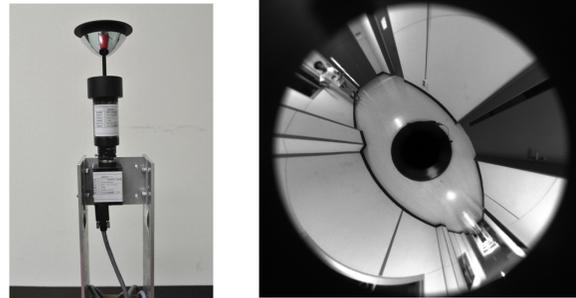
**Abstract**—In this paper, we propose a method for a mobile robot to navigation in textureless environment using an omni-directional camera. The method makes an environment map consisting of 3D edge points obtained from omni-directional camera images and estimates the locations of planes by analysing these 3D edge points so that the robot can autonomously travel while avoiding walls as obstacles. The method has the advantage that it can generate a 3D map in environments constructed by textureless planes. It is important to improve robustness of obstacle detection in order to realize navigation with a long distance. The proposed method was evaluated in more complex environment such as a long corridor and a corner.

## I. INTRODUCTION

For navigation of autonomous mobile robots, self-localization and path planning are basic functions [1]. In addition to those basic functions, an autonomous robot is required to build a map of its environment when it is located in an unknown environment. Including the requirement of map building, simultaneous localization and mapping (SLAM) has been actively investigated to adapt an autonomous environment to various kinds of unknown environment [2].

Among several instruments for sensing environment of a mobile robot, an omni-directional camera is one of widely-used sensors (see Fig. 1(a)). An advantage of omni-directional camera is its wide view range which does not require rotating camera to obtain 360-degree view range. This advantage is effective especially when robot is required to measure its environment online while moving [3]. An example of a panoramic view obtained by an omni-directional camera is depicted in Fig. 1(b).

Several researches have been conducted to realize map generation and path planning using omni-directional camera [4][5][6]. They use multiple kinds of sensors such as laser range finder and ultrasonic sensor and integrate their information for accurate sensing. On the other hand, using multiple sensors disadvantageous in the sense of cost and weight. If we can obtain sufficient sensing performance with



(a) Camera with a hyperboloid mirror. (b) Acquired image.

Fig. 1. Omnidirectional camera and acquired image.

a sole (omni-directional vision) sensor, it will bring great benefits of low cost, low weight and simplicity.

Coping with a sole omni-directional camera for robot navigation has been proposed by Morioka et al. [7] and Iwasa et al. [8]. One of the difficulties of recognizing environment with only omni-directional camera is that textures cannot be always detected in the environment. For example, indoor environment composed by corridors and walls (as can be seen in Fig. 1(b)) do not provide many texture information, which causes difficulty of texture-based recognition.

As a method for reconstructing an environment map denoted by a set of dense points, [9] was proposed. This method generated planes consistently fitting to 3-D edge points reconstructed from the image edge points. However, in [9], obstacle avoidance of robot using the result of plane reconstruction was not mentioned.

Therefore, we proposed a method for avoiding obstacles in an environment which has few feature points [10]. The proposed method reconstructs planes from a 3-D map generated by tracking the edge points in omni-directional image sequences. By interpolating the area which has few feature points, a dense 3-D map can be generated in environment with few textures. However, in [10], accuracy of plane reconstruction was low. Thus, it was difficult for the robot to avoid obstacles successively. It is important to improve robustness of obstacle detection in order to realize navigation for a long distance. Therefore, in this paper, we consider and propose a method about revisions of parameters and noise reduction for improvement of the robustness. Experiments are conducted in more complex environment such as a long corridor and a corner.

\*This work was in part supported by a MEXT KAKENHI Grant-in-Aid for Young Scientists (A), 22680017, and the Asahi-Glass Foundation.

<sup>1</sup>K. Watanabe, R. Kawanishi, Y. Kobayashi and T. Kaneko are with the Department of Mechanical Engineering, Shizuoka University, 3-5-1 Johoku, Naka-ku, Hamamatsu-shi, Shizuoka 432-8561, Japan f0130076, f5945015, tykobay, tmtkane@ipc.shizuoka.ac.jp

<sup>2</sup>A. Yamashita and H. Asama are with the Department of Precision Engineering, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan yamashita, asama@robot.t.u-tokyo.ac.jp

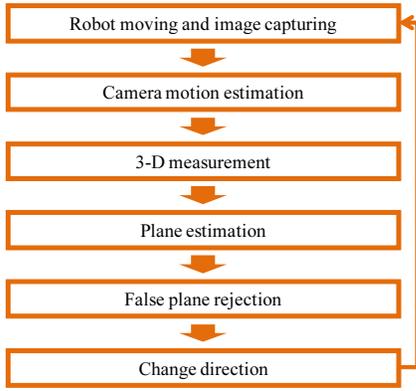


Fig. 2. Procedure of our proposed method.

## II. PROCESS OUTLINE

In the proposed method, we use a mobile robot equipped with an omni-directional camera as shown in Fig. 1(a). The camera attaches a hyperboloid mirror in front of its lens. An acquired image is shown in Fig. 1(b). Internal parameters of camera and distortion of lens are determined in advance.

Fig. 2 describes an overview of the method for 3-D measurement. An omni-directional camera is mounted on the mobile robot. The robot moves while capturing omni-directional images. When the robot moves along a fixed distance, it stops capturing images. Then the robot conducted the camera motion estimation and 3-D measurement.

Next, the robot reconstructs planes around itself by using the result of 3-D measurement. However, this result contains some false planes for the robot navigation. Therefore, we apply a false plane rejection method that uses the density of the points that construct the planes. Based on the obtained result the result of this, the robot determines the direction which has no plane from its view point. The robot turns to the determined direction and resumes moving and capturing images.

By repeating the above mentioned procedures, the robot can move autonomously and generates an environment map.

## III. 3-D MEASUREMENT

For the 3-D measurement, we followed measurement method described in [11]. In this paper, the camera motion is estimated based on the eight point algorithm which uses correspondence relationship between the tracked feature points. We use the Lucas Kanade tracker [12] algorithm for tracking feature points. In order to remove outliers obtained by the Lucas Kanade tracker algorithm, we applied the method of RANSAC (RANDOM SAMple Consensus) [13]. Finally, the projection errors are minimized by using bundle adjustment [14].

We employ a 3-D measurement method using estimated camera motion [15]. This method measures edge points which are tracked in the image sequence. Edge points are obtained from the captured image by the Canny edge detector [16].

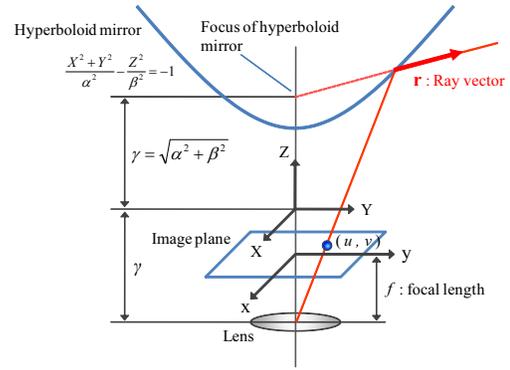


Fig. 3. Coordinate system of omnidirectional camera. Ray vector is defined as a unit vector which starts from focus of a hyperboloid mirror.

### A. Camera Motion Estimation

An omni-directional camera we use has a hyperboloid mirror in front of the camera lens. A ray vector  $\mathbf{r} = [x, y, z]^T$  is defined as a unit vector heading to the reflection point on the mirror surface from the focus of the hyperboloid mirror as shown in Fig. 3. A ray vector  $\mathbf{r}$  is calculated by Eq. (1) from image coordinates  $[u, v]^T$  as

$$\mathbf{r} = \begin{bmatrix} su \\ sv \\ sf - 2c \end{bmatrix}, \quad (1)$$

where  $s$  is defined as Eq. (2). In Eq. (2),  $a$ ,  $b$  and  $c$  are the hyperboloid parameters and  $f$  is the focal length of the lens.

$$s = \frac{a^2(f\sqrt{a^2 + b^2} + f\sqrt{u^2 + v^2 + f^2})}{a^2f^2 - b^2(u^2 + v^2)} \quad (2)$$

The proposed method calculates a ray vector of image coordinates  $[u, v]^T$  at an arbitrary viewpoint.

Matrix  $\mathbf{E}$  is called an essential matrix and satisfies

$$\mathbf{r}'_i{}^T \mathbf{E} \mathbf{r}_i = 0, \quad (3)$$

where ray vectors  $\mathbf{r}_i = [x_i, y_i, z_i]^T$  and  $\mathbf{r}'_i = [x'_i, y'_i, z'_i]^T$  correspond to two images captured before and after motion, respectively. The essential matrix contains information of relative position and orientation differences between two observation points. Our method calculates a camera rotation matrix  $\mathbf{R}$  and a translation vector  $\mathbf{t}$  from the essential matrix  $\mathbf{E}$  by singular value decomposition. Our method measures 3D coordinates of object points by using the estimated camera motion. Since the measurement precision depends on the precision of the camera motion estimation, it is very important to estimate the essential matrix  $\mathbf{E}$  precisely.

Here, Eq.(3) is transformed into

$$\mathbf{u}^T \mathbf{e} = 0, \quad (4)$$

where

$$\mathbf{u} = [x_i x'_i, y_i x'_i, z_i x'_i, x_i y'_i, y_i y'_i, z_i y'_i, x_i z'_i, y_i z'_i, z_i z'_i]^T$$

$$\mathbf{e} = [e_{11}, e_{12}, e_{13}, e_{21}, e_{22}, e_{23}, e_{31}, e_{32}, e_{33}]^T,$$

and  $e_{jk}$  is the  $j$ -th row and  $k$ -th column element of matrix  $\mathbf{E}$ . The essential matrix  $\mathbf{E}$  is obtained by solving the system

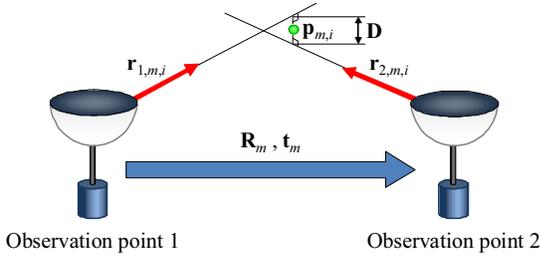


Fig. 4. Calculation of 3D Coordinates

of equations for more than eight pairs of corresponding ray vectors. This means that we solve

$$\min_{\|\mathbf{e}\|=1} \|\mathbf{U}\mathbf{e}\|^2, \quad (5)$$

where  $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n]^T$ . Vector  $\mathbf{e}$  is given as the eigenvector of the smallest eigenvalue of  $\mathbf{U}^T\mathbf{U}$  and then the essential matrix  $\mathbf{E}$  is obtained. There,  $\mathbf{E}$  is represented by the rotation matrix  $\mathbf{R}$  and the translation vector  $\mathbf{t} = [t_x, t_y, t_z]^T$ .  $\mathbf{R}$  and  $\mathbf{T}$  are calculated from the essential matrix  $\mathbf{E}$  by singular value decomposition.

$$\mathbf{E} = \mathbf{R}\mathbf{T}, \quad (6)$$

where  $\mathbf{T}$  is a matrix given as follows.

$$\mathbf{T} = \begin{bmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{bmatrix}$$

### B. 3D Measurement

3D coordinates of an object point which is projected as an edge point in the image are given based on triangulation with two cameras set in the geometrical relation given by rotation matrix  $\mathbf{R}_m$  and translation vector  $\mathbf{t}_m$ , where  $m$  is the number of measurement. We calculate 3D coordinates of  $\mathbf{p}_{m,i}$  ( $i$ -th feature point) by using  $\mathbf{R}_m$  and  $\mathbf{t}_m$  (Fig.4).

## IV. PLANE RECONSTRUCTION

We employ the method proposed by Tomono [9] for reconstructing planes. In [9], planes fitting to edge points are reconstructed if they satisfy visibility constraints. Consideration of visibility constraints means that we remove planes which should not be visible if we assume that other detected planes (located between the plane and the camera) exist in reality.

A triangle is generated by choosing arbitrary combination of three edge points. The planes which satisfy visibility constraint and fitting to edge points are chosen as correctly estimated planes. Visibility constraint is considered as follows.

If an estimated plane exists in real environment, objects behind the plane are invisible. Contrarily, if there exists an object visible through an estimated plane, the plane is determined as a plane which is inconsistent with the real environment. This plane is said to offence visibility constraint. Estimated planes which are inconsistent with the real environment are excluded.

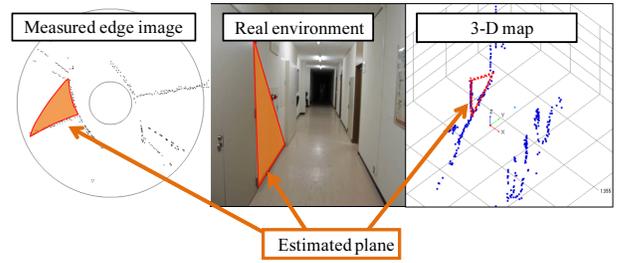


Fig. 5. Estimated plane.

## V. NAVIGATION METHOD

For the robot navigation, we follow navigation method described in [10]. In this method, the absolute scale between the robot and the wall is not measured. Therefore, the robot avoids the obstacle by using only information of the direction of the plane from the robot. When the robot finishes plane reconstruction, it determines the direction which has no plane from its view point. In the method, the area which has no plane is called an empty area. The robot runs toward the center of the empty area. Since the purpose of the method is determining an advancing direction of the robot, it is sufficient to search for empty areas on the horizontal 2-D plane where the height of point is not considered. A plane outputted in the 3-D map is a triangle whose sides consist of edge points. Many edge points are on each side in the 3-D map, and a straight line segment consisting of these edge points appears in the 2-D map (Fig. 5).

However, in [10], accuracy of plane reconstruction was low. Thus, it is difficult for the robot to avoid obstacles successively. Therefore, in this section (B), we consider and propose a method about revisions of parameters and noise reduction for improvement of the robustness.

### A. Searching for Empty Areas

To detect empty areas, our method divides the whole orientation around the robot into small sectors, and examines whether each sector has edge points of estimated planes in the 2-D map. A sector which has no edge point is an empty sector, however, false planes are occasionally estimated to produce some outliers in the sector. Therefore, the method regards a sector which does not have edge points more than a threshold as an empty sector (Fig. 6). Here, edge points is counted within a certain range from the robot. The method determines a set of consecutive empty sectors which has an angle larger than a threshold as an empty area.

The robot determines the center of the empty area as a new direction (Fig. 7 (Point1)). If there are multiple directions of empty area, the direction closest to the current direction is chosen as a new direction (Fig. 7 (Point2)). By considering a new direction every time the robot stops at a fixed distance, the robot can generate a map of unknown environment while avoiding obstacles.

### B. Consideration and Solution of Recognition Error

One of the problems in the proposed method is that plane detection can fail even when the robot is close to a wall. One reason for this wrong detection of planes is that the number

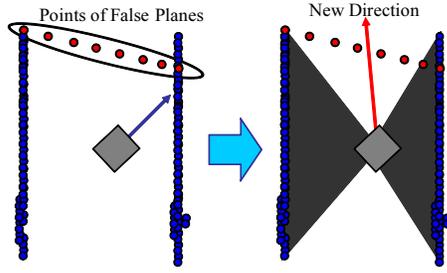


Fig. 6. Ignoring false plane area.

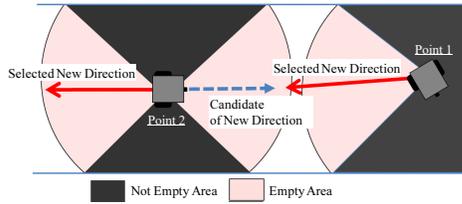


Fig. 7. Detection of empty areas.

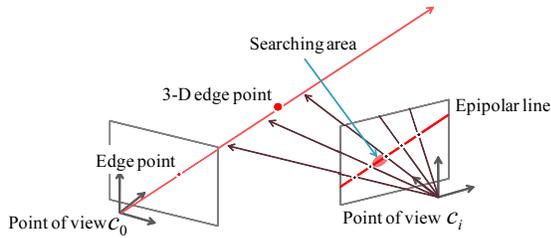


Fig. 8. Epipolar line.

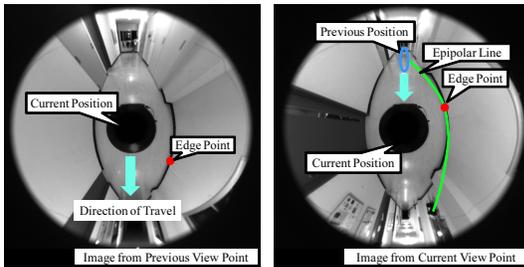


Fig. 9. Epipolar line on boundary between floor and wall.

of detected points on the boundary between the floor and the wall can decrease when the robot is running parallel to the wall. This is caused by the principle of edge reconstruction method and can be explained by the followings: In the edge reconstruction method, correspondence of two points in the two image frames (before and after robot's motion) is found by template matching. In the proposed method, template matching (scanning) is conducted along an epipolar line, which is a line of sight from one viewpoint projected to the other image frame (see Fig. 8). When the robot is running parallel to the boundary between the floor and the wall, the epipolar line coincides with the boundary (see Fig. 9). This means that too many candidates appears in the template matching process. Consequently, corresponding points cannot be obtained correctly.

In this paper, we evaluated the relationship between the angle between the direction of the robot and the boundary

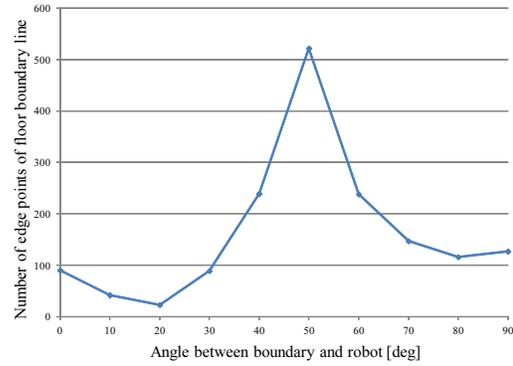


Fig. 10. Number of the edge points on line between floor boundary and walls.

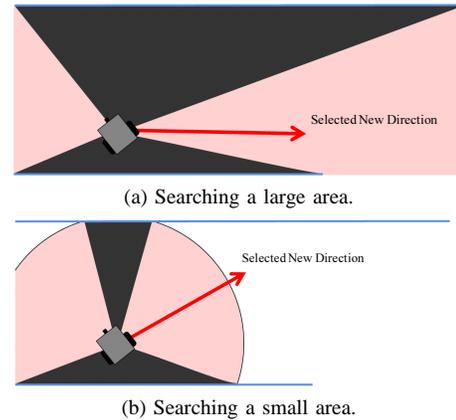


Fig. 11. Difference between direction due to size of searching area.

and the number of the measured edge points by running the robot in a corridor that constructed with two parallel walls and a floor. 10 times 3-D measurements were performed in increments of 10 degrees from the parallel direction (0 deg) to the boundary to the vertical (90 deg). The result of these measurement is shown in Fig. 10. In Fig. 10, the horizontal axis indicates the angle between the boundary and the direction of the robot, and the vertical indicates the amount of measured edge points on the boundary. Edge points were obtained the most when the direction was 50 degree, and the edge points tend to decrease when the robot runs parallel to the wall. This result indicates that the more the robot runs parallel to the wall, the less measured edge points are detected.

Under the consideration of the above-mentioned issues on the false plane detection, we propose to apply parameter setting for robust navigation as follows: The range of searching empty area is set 2 m, which is shorter than the range that covers most of the detected points by the proposed method. The effect of shortening the range is illustrated in Fig. 11. In Fig. 11(a), all of the detected points are considered for the navigation, and the robot's direction will be set almost parallel to the wall. On the other hand, when we shorten the searching range as depicted in Fig. 11(b), the robot will take a new direction further from the parallel direction. This will decrease the risk of false plane detection. The robot will be still possible to avoid the other wall when it approaches to it.

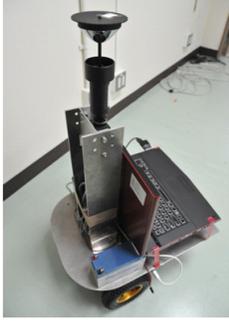


Fig. 12. Mobile robot.



Fig. 13. Experiment environments (Straight path).



Fig. 14. Experiment environments (Corner).

## VI. EXPERIMENT

Two experiments were conducted at a corridor of an indoor environment. A mobile robot equipped with omni-directional camera for the experiment is shown in Fig. 12. The robot had a single omni-directional camera. The robot had two types of behavioral patterns, advance and rotation. It followed the navigation strategy described in the previous section. First experiment was conducted at a long straight path (Fig. 13). Second experiment was conducted at a corner (Fig. 14).

### A. Navigation in a Long Distance Straight Line

An experiment was conducted at a corridor with length of 20 m (see Fig. 13). The result of navigation is shown in Fig. 15. In the figure, the result of 3-D measurement is denoted by blue dots. The result of plane reconstruction is denoted by orange dots. Intersections of green and red line segments depict points where measurement by omni-directional camera was conducted. Current direction of the robot at each measurement points is denoted by green line segment. New direction for navigation calculated by the proposed method at each measurement point is denoted by a red line segment. A triangle in the bottom of the figure denotes starting location and a rectangular in the top denotes final location of the robot. It can be seen that the robot moved along the direction of red line segment and changed its direction at the next intersection of red and green line segments. The robot finally reached the final location without colliding with the walls.

In this experiment, 3-D measurement and plane estimation were performed 16 times. The robot proceeded to the unknown direction while avoiding the walls using the result of plane estimation. The robot sometimes approached close to the walls. In such case, the robot detected a wall and avoided the wall by changing the moving direction to an empty space. It can be seen that the two straight walls along the corridor were not precisely measured, though the detected planes construct approximate lines corresponding to the walls. This is caused by error in plane construction. Especially, some false planes were estimated in the moving direction of the robot. However, these planes were ignored by using the method presented in section 5(B).

The number of the edge points of the floor boundary is shown in Fig. 16. In Fig. 16, the number described in

the horizontal axis indicates the each position where plane estimations were conducted. The position is described as intersections of green and red line in Fig. 15. In Fig. 16, the number is almost over 200. This is sufficient number to reconstruct planes for avoiding obstacles. In the experiment, the robot rarely ran parallel to the walls. When the robot ran parallel to the walls, the number decreased to under 100 at the measurement point of the robot 1, 12 and 14. However, after passing each of these points, the robot ran away from the wall and the number increased owing to the parameter setting presented in section 5(B). Herewith, we confirmed that our method can be conducted in a long distance.

### B. Navigation in a Corner

An experiment was conducted at a corner (Fig. 14). The result of navigation is shown in Fig. 17. In the figure, Indications of blue points, orange points, red and green line segments, triangle and rectangular are the same as the case in Fig. 15.

In this experiment, 3-D measurement and plane estimation were performed 4 times. At the initial steps, the robot moved toward the front wall. Then, the robot detected the wall in front and left side of the robot. The robot turned the corner by avoiding these walls. Occasionally, plane estimation was failed. In that case, some planes estimated in the moving direction of the robot. However, these planes were ignored by using the method presented in section 5(B). This result indicates that the proposed method is effective to navigate a robot in an environment including a corner.

## VII. CONCLUSIONS AND FUTURE WORKS

### A. Conclusions

In this paper, we proposed a method of navigation method for a mobile robot equipped with an omni-directional camera. To travel a long distance, we considered revisions of parameters. The proposed method enabled the robot to autonomously move in unknown environment with few textures over long distance. Experiments showed the effectiveness of the proposed method.

### B. Future Works

As future works, navigation in dynamic environment, and avoiding obstacles consisting of not only planes should be studied.

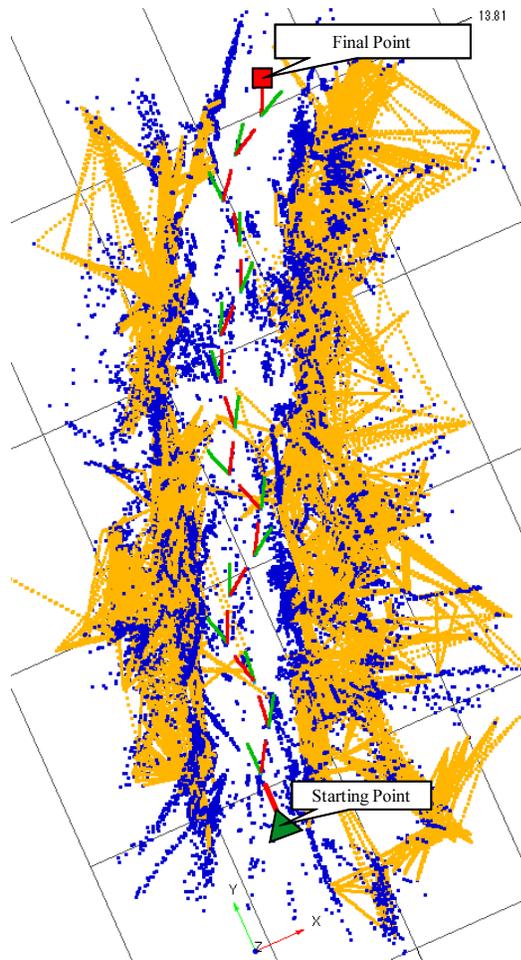


Fig. 15. Result of navigation in a long straight path (Top view.).

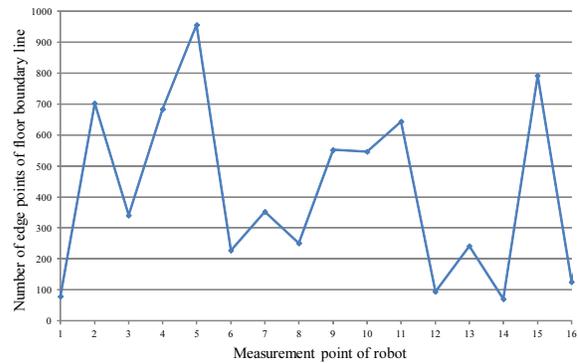


Fig. 16. Number of edge points on floor boundary in a long straight path.

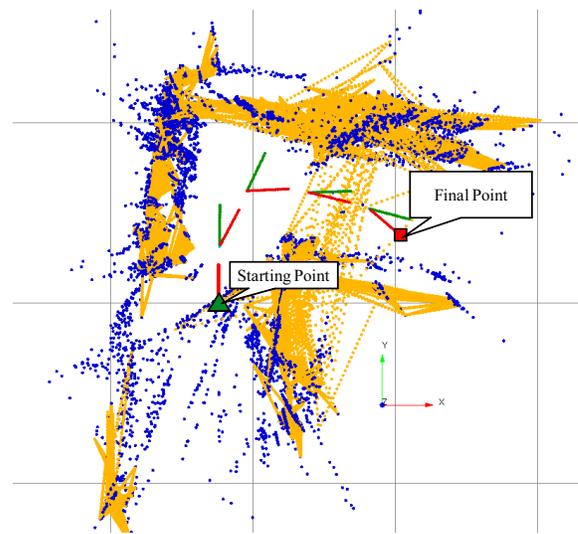


Fig. 17. Result of navigation in a corner (Top view.).

## REFERENCES

- [1] J. C. Latombe: "Robot Motion Planning", *Kluwer Academic Publishers*, Norwell, Mass, 1991.
- [2] D. Nister: "Reconstruction From Uncalibrated Sequences with a Hierarchy of Trifocal Tensors", *Proceedings of the 6th European Conference on Computer Vision*, Vol. 1, pp. 649-663, 2000.
- [3] C. Geyer and K. Daniilidis: "Omnidirectional Video", *The Visual Computer*, Vol. 19, No. 6, pp. 405-416, 2003.
- [4] C. Wang and C. Thorpe: "Simultaneous Localization and Mapping with Detection and Tracking of Moving Objects", *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, pp. 2918-3004, 2002.
- [5] J. Miura, Y. Negishi, and Y. Shirai: "Mobile robot map generation by integrating omnidirectional stereo and laser range finder", *Proceeding of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 250-255, 2002.
- [6] J. Meguro and Y. Amano: "Development of an Autonomous Mobile Surveillance System Using a Network-based RTK-GPS", *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 3107-3112, 2005.
- [7] H. Morioka, Y. Sangkyu, O. Hasegawa: "Vision-based Mobile Robot's SLAM and Navigation in Crowded Environments", *Proceeding of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1-4, 2010.
- [8] H. Iwasa, N. Aihara, N. Yokoya and H. Takemura: "Memory-based Self-Localization Using Omnidirectional Images", *Systems and Computers in Japan*, Vol. 34, No. 5, pp.56-68, 2003.
- [9] M. Tomono: "Plane Reconstruction with a Stereo Camera in Non-textured Environments," *Proceeding of the 17th Robotics Symposia*, pp. 463-468, 2011. (in Japanese)
- [10] K. Watanabe, R. Kawanishi, T. Kaneko, A. Yamashita and H. Asama: "Obstacle Avoidance Based on Plane Estimation from 3-D Edge Points by Mobile Robot Equipped with Omni-directional Camera", *Advances in Intelligent and Soft Computing (Proceedings of the 12th International Conference on Intelligent Autonomous Systems)*, Springer, 2012, to appear.
- [11] R. Kawanishi, A. Yamashita and T. Kaneko: "Three-Dimensional Environment Model Construction from an Omnidirectional Image Sequence", *Journal of Robotics and Mechatronics*, Vol. 21, No. 5, pp. 574-582, 2009.
- [12] J. Shi and C. Tomasi: "Good Features to Track", *Proceedings of the 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 593-600, 1994.
- [13] M. A. Fischler and R. C. Bolles: "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography", *Communications of the ACM*, Vol. 24, No. 6, pp. 381-395, 1981.
- [14] B. Triggs, P. McLauchlan, R. Hartley and A. Fitzgibbon: "Bundle Adjustment - A Modern Synthesis", *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, Springer-Verlag LNCS 1883, pp.298-372, 1999.
- [15] M. Tomono: "Dense Object Modeling for 3-D Map Building Using Segment-based Surface Interpolation", *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, pp. 2609-2614, 2006.
- [16] J. F. Canny: "A Computational Approach to Edge Detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, No. 6, pp. 679-698, 1986.