Obstacle Avoidance Based on Plane Estimation from 3D Edge Points by Mobile Robot Equipped with Omni-directional Camera

Kazushi Watanabe, Ryosuke Kawanishi and Toru Kaneko

Dept. of Mechanical Engineering, Shizuoka University
3-5-1 Johoku, Naka-ku, Hamamatsu-shi, Shizuoka 432-8561, Japan
{f0130076, f5945015, tmtkane}@ipc.shizuoka.ac.jp

Atsushi Yamashita and Hajime Asama

Dept. 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

Abstract. In this paper, we propose a method for a mobile robot to avoid obstacles in its 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 avoid walls as obstacles. The method has the advantage that it can generate a 3D map in environments constructed by textureless planes. Experimental results show the effectiveness of the proposed method.

Keywords: navigation, environment recognition, omni-directional camera

1 Introduction

Localization and path planning are indispensable functions for an autonomous mobile robot. Since the robot does not have an environment map in unknown environment, the robot needs to generate a map by itself by measuring surrounding environment. In addition to map generation, path planning, movement, and localization are needed for the robot to generate the environment map over a wide area. Therefore, in order for the robot to work in unknown environment, the function of generating a map while autonomously navigating is required.

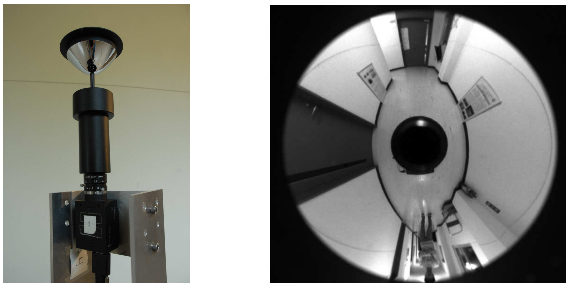
As a sensor for environment measurement in map generation, an omni-directional camera is sometimes used (Fig. 1(a)). An omni-directional camera is suitable for environment measurement [1], because the acquired image has a panoramic field of view (Fig. 1(b)).

There are methods for map generation and path planning in unknown environment of a mobile robot equipped with an omni-directional camera [2-4]. If we use multiple sensors for measurement, measurement result is more accurate and denser than using only an omni-directional camera. Then the robot can avoid the surface of a wall and obstacles, and it can run autonomously in unknown environment. However, this method needs other sensors in addition to a camera. Therefore this method has disadvantages in terms of cost and weight. If the robot can work in unknown environment using only an omni-directional camera, there are significant benefits in terms of cost and weight compared with using other sensors such as laser scanner, ultrasound sensor, etc.

Methods for generating map and path planning using a single omni-directional camera have been already proposed. However, generating a map in an environment with few textures is difficult by use of only a single omni-directional camera. Thus running in such environment is difficult for the robot. Therefore in [5][6], operation of robot is manual.

As a method for reconstructing an environment map denoted by a set of dense points, reference [7] is proposed. This method generates planes consistently fitting to 3-D edge points reconstructed from the image edge points. However, in [7], obstacle avoidance of robot using the result of plane reconstruction is not mentioned.

Therefore, in this paper, we propose a method for avoiding obstacles in an environment which has few feature points. 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. Since our method generates a dense 3-D map, the robot can run without a map generated beforehand.



(a) Omnidirectional camera. (b) Acquired image.

Fig. 1. Omnidirectional camera and acquired image.

2 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.

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

Next, the robot reconstructs planes around itself from the result of 3-D measurement. From the result of plane estimation, 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 behavior, the robot can move autonomously and generates an environment map.

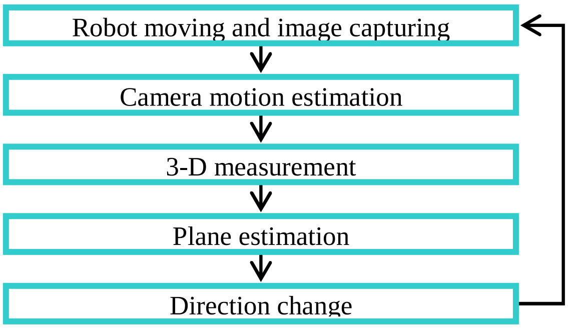


Fig. 2. Procedure of our proposed method.

3 3-D measurement

Our method of 3-D measurement is based on reference[8]. In this paper, the camera motion is estimated based on the 8 point algorithm which uses correspondence relationship between the tracked feature points. We use the Lucas Kanade tracker algorithm for tracking feature points [9]. Simultaneously, the method of RANSAC (RANdom SAmple Consensus) [10] is applied to remove outliers. Finally, the reprojection errors are minimized by using bundle adjustment[11].

We employ a 3-D measurement method using estimated camera motion[12]. This method measures edge points which are tracked through the image sequences. Edge points are obtained from the captured image by the Canny edge detector [13].

3.1 Camera motion estimation

An omni-directional camera we use has a hyperboloid mirror in front of the camera lens. A ray vector **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 **r** is calculated by Eq. (1) from image coordinates $[u, v]^T$.

$$\mathbf{r} = \begin{bmatrix} su \\ sv \\ sf - 2c \end{bmatrix} \tag{1}$$

Here, 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.

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

The proposed method calculates a ray vector of image coordinates $[u, v]^{\mathrm{T}}$ at the arbitrary viewpoint.

Matrix **E** which satisfies Eq. (3) is called an essential matrix.

$$\mathbf{r}_i'^{\mathrm{T}}\mathbf{E}\mathbf{r}_i = 0 \tag{3}$$

where ray vectors $\mathbf{r}_i^{\mathrm{T}} = [x_i, y_i, z_i]^{\mathrm{T}}, \mathbf{r}_i'^{\mathrm{T}} = [x_i', y_i', z_i']^{\mathrm{T}}$ are those of the corresponding point in two images, respectively. By decomposing **E** into translational element and rotational element, camera motion is estimated.

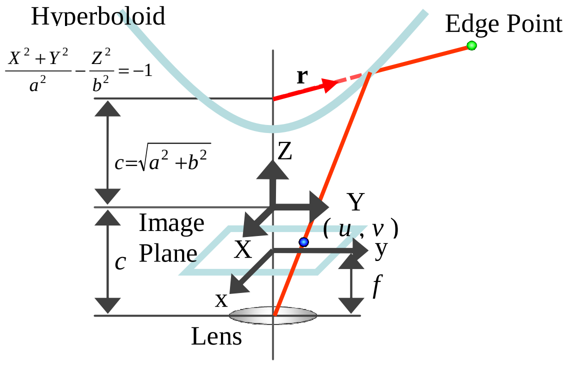


Fig. 3. Calculation of ray vector.

3.2 Outlier rejection

Not all feature points tracked in the image sequence correspond satisfactorily due to image noise, etc. Mistracked feature points should be rejected, by using the RANSAC (RANdom SAmple Consensus) algorithm [10].

In reference [8], determining essential matrix $\mathbf{E}_{rand}$ is repeated for a predetermined number of times, then we choose the case with the maximum number of feature points satisfying Eq. (4), and calculate essential matrix **E** again using remaining feature points.

$$\mathbf{r}_i'^{\mathrm{T}}\mathbf{E}_{rand}\mathbf{r}_i < q \tag{4}$$

where q is a given threshold.

3.3 Decision of feature point number

In [8], the optimum number of feature points to use in measurement is automatically decided by defining it as the number of feature points including the maximum number of outliers in a range in which camera movement is still estimated precisely enough. From the above, Eq. (5) and (6) are given as follows:

$$\zeta(k) = \left| \frac{\omega \sum_{i=k}^{k+\omega} i \cdot O_i - \sum_{i=k}^{k+\omega} i \sum_{i=k}^{k+\omega} O_i}{\omega \sum_{i=k}^{k+\omega} i^2 - \left(\sum_{i=k}^{k+\omega} i \right)^2} \right| \tag{5}$$

$$\zeta(k) < t \tag{6}$$

where O_i is the number of outliers and i is the number of extracted feature points. Equation (5) represents the slope of the straight line calculated by the least squares method, so $\zeta(k)$ shows the increase in the number of outliers when the number of the extracted feature points changes from k to $(\omega + k)$. $\zeta(k)$ is calculated stably by setting ω appropriately. In Eq. (6), t is a given threshold. The largest value of k which satisfies Eq. (6) is determined as the optimal number of extraction to camera motion estimation.

4 Plane reconstruction

We employ the method in [7] for reconstructing planes. In [7], planes fitting to edge points are reconstructed if they satisfy visibility constraints.

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 as follows.

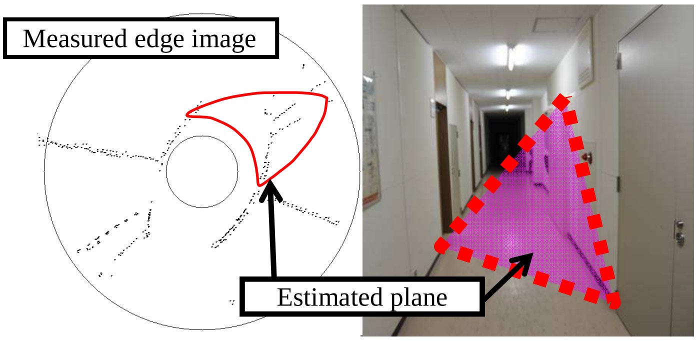


Fig. 4. Visibility constraint.

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 (Fig. 4). This plane is said to offence visibility constraint. Estimated planes which are inconsistent with the real environment are excluded by applying visibility constraint.

5 Obstacle avoidance

When the robot finishes plane reconstruction, it determines the direction which has no plane from its view point. In this paper, the area which has no plane is called an empty area. The robot runs toward the center of an empty area.

Since the purpose of our method is determining an advancing direction of the robot, it is enough 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 appear in the 2-D map (Fig. 5).

To detect empty areas, our method draw lines each of which connects the present point of the robot to each edge point in the 2-D map. Then it determines an area containing two adjacent lines as an empty area if the angle between the lines exceeds a threshold (Fig. 6). Because the method uses the result of plane reconstruction instead of 3-D measurement point data which sometimes include outliers, robust empty area detection is realized.

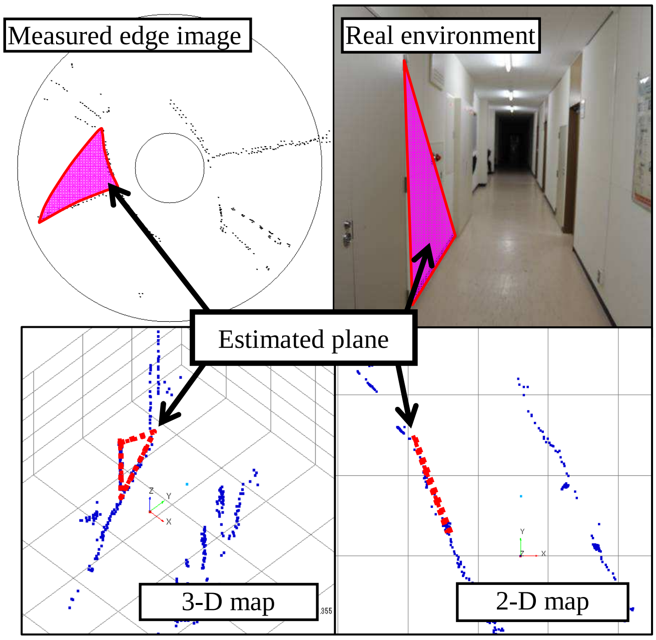


Fig. 5. Estimated plane.

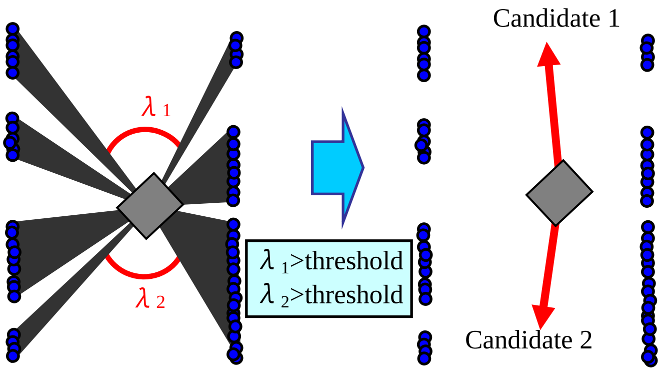


Fig. 6. Detection of empty areas.

If there are several directions of empty area, the direction closest to the current direction is chosen as a new direction (Fig. 7). 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.

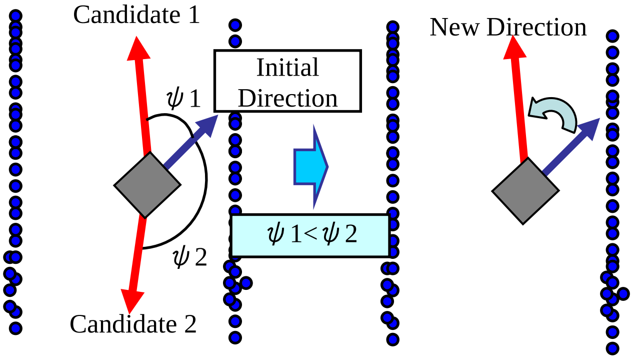


Fig. 7. Selection of a new direction from candidates.

6 Experiment

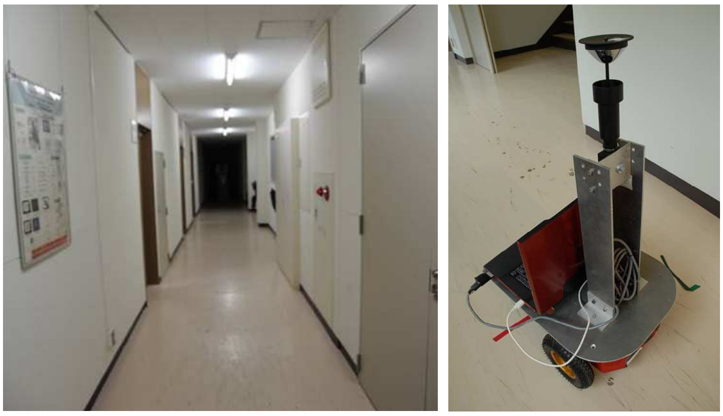
An experiment was carried out at a corridor (Fig. 8(a)). A mobile robot equipped with omni-directional camera for the experiment is shown in Fig. 8(b). The robot had a single omni-directional camera. The robot had two types of behavioral patterns, advance and rotation. Rotation was only used for change of direction when the robot detects obstacles existing in the forward direction.

A result of 3-D measurement in experimental environment is shown in Fig. 9(a), 10(a) and11(a). In these figures, the initial position of the robot is drawn as a triangle. A rectangle connected to the triangle by a line is the position of the robot after moving. The points existing around the triangle and the rectangle are measured edge points.

When we compare the experimental environment (Fig. 8(a)) with the experimental result (Fig. 9(a), 10(a) and 11(a)), the latter reconstructs only a part of the real environment.

The result of plane reconstruction using the result of 3-D measurement is shown in Fig. 9(b), 10(b) and 11(b). In these figures, the new direction of the robot determined from the result of plane reconstruction is represented by an arrow. Each plane is represented by 3 sides which are drawn by points in yellow.

From the result of plane reconstruction, a dense 3-D map is obtained. The viewing angle in Fig. 11(b) is almost same with that of Fig. 8(a). Reconstructed planes are in position of the plane in real environment. Since a dense 3-D map was obtained, the robot turned to the direction which had no plane. Herewith, the robot autonomously generated the map and avoided the wall while moving.



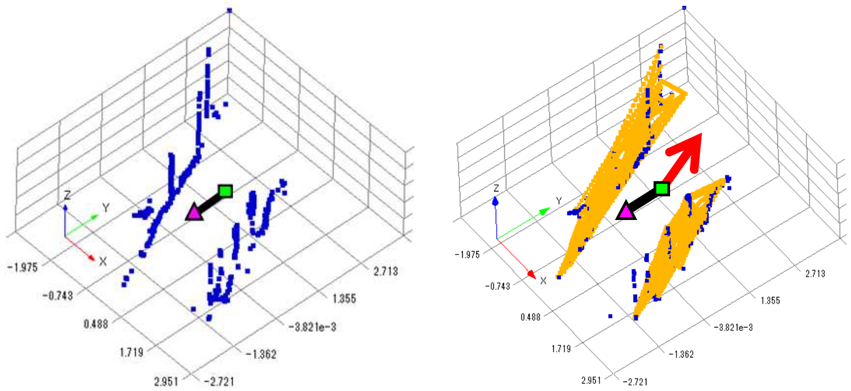
(a) Experimental environment. (b) Mobile robot.

Fig. 8. Experimental environment and mobile robot.

7 Conclusion

In this paper, we proposed a method of avoiding obstacles for a mobile robot equipped with omni-directional camera. The proposed method makes the robot possible to autonomously move in unknown environment with few textures.

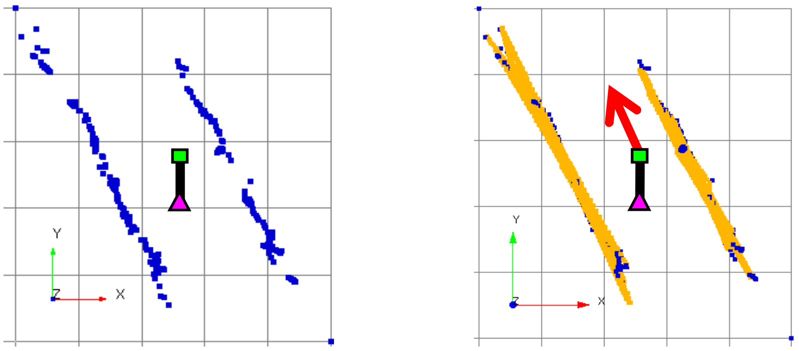
As future works, navigation in dynamic environment, and avoiding obstacles consisting of not only planes should be studied.



(a) Result of 3-D measurement.

(b) Result of plane estimation.

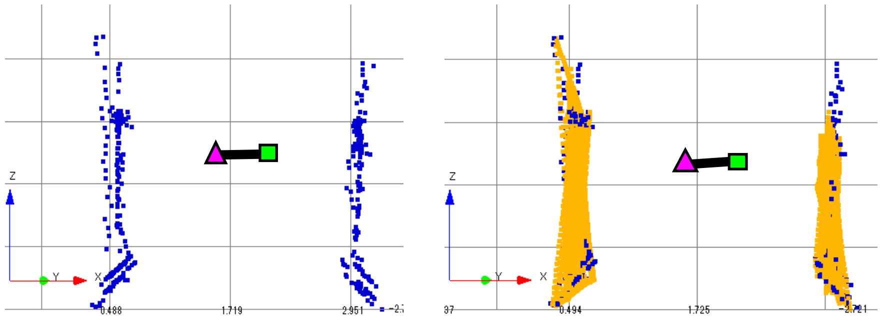
Fig. 9. Bird's eye view of a result of 3-D measurement and plane estimation.



(a) Result of 3-D measurement.

(b) Result of plane estimation.

Fig. 10. Top view of a result of 3-D measurement and plane estimation.



(a) Result of 3-D measurement.

(b) Result of plane estimation.

Fig. 11. Side view of a result of 3-D measurement and plane estimation.

Acknowledgements

This work was in part supported by MEXT KAKENHI, Grant-in-Aid for Young Scientist (A), 22680017.

References

1. C. Geyer and K. Daniilidis: “Omnidirectional Video”, *The Visual Computer*, Vol. 19, No. 6, pp. 405-416, 2003.
2. 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.
3. D. Kim, J. Sun, S. M. Rehg, A. F. Bobick: “Traversability Classification Using Unsupervised On-line Visual Learning for Outdoor Robot Navigation”, *Proceedings of the 2006 IEEE International Conference onRobotics and Automation, pp. 518-525, 2006*
4. K. Konolige, M. Agrawal, R. C. Bolles, C. Cowan, M. Fischler, and B. P. Gerkey, “Outdoor Mapping and Navigation Using Stereo Vision,” *Proceedings of the International Symposium on Experimental Robotics*, pp.1-12, 2006.
5. 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.
6. 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.
7. M. Tomono: “Plane Reconstruction with a Stereo Camera in Non-textured Envirionments,” *Proceeding of the 17th Robotics Symposia*, pp. 463-468, 2011. (in Japanese)
8. 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.
9. 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.
10. M. A. Fischler and R. C. Bolles: “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Carto-graphy”, *Communications of the ACM*, Vol. 24, No. 6, pp. 381-395, 1981.
11. 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.
12. 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.
13. 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.