



Title	Generation of stationary environmental map under unknown robot motion
Author(s)	Yagi, Y.; Hamada, H; Benson, N. et al.
Citation	IEEE International Conference on Intelligent Robots and Systems. 2000, 2, p. 1487-1492
Version Type	VoR
URL	<a href="https://hdl.handle.net/11094/14095">https://hdl.handle.net/11094/14095</a>
rights	c2000 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE..
Note	

*The University of Osaka Institutional Knowledge Archive : OUKA*

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

## Generation of Stationary Environmental Map under Unknown Robot Motion

Y. Yagi H. Hamada N. Benson M. Yachida

Department of Systems and Human Science  
Graduate School of Engineering Science, Osaka University  
Toyonaka, Osaka 560-8531, Japan  
y-yagi@sys.es.osaka-u.ac.jp

### Abstract

*Under the assumption of known motion of a robot, environmental maps of a real scene can be successfully generated by monitoring azimuth changes in an image. Several researchers have used this property for robot navigation. However, it is difficult to observe the exact motion parameters of the robot because of encoder measurement error. Therefore, observational errors in the generated environmental map accumulate in long movements of the robot. To generate a large environmental map, it is desirable not to assume known robot motion. In this paper, under the assumption of unknown motions of the robot, we propose a method to generate an environmental map and estimate the ego-motion of a robot, by using an omnidirectional image sensor.*

### 1 Introduction

Generation of stationary environmental maps is one of the important tasks for vision based robot navigation. For this purpose, a detailed analysis is not necessary but high speed and rough understanding of the environment around the robot is required. If considered from the standpoint of machine perception, autonomous navigation needs the field of view as wide as possible. Thus, a real-time omnidirectional camera, which can acquire an omnidirectional (360 degrees) field of view at video rate, is suitable for autonomous navigation. There have been several attempts to acquire omnidirectional images using a rotating camera, a fish-eye lens, a conic mirror and a spherical mirror. Over the past 15 years, researchers in computer vision, applied optics and robotics have investigated a number of papers related to omnidirectional cameras and their applications [1] [2] [3].

Under the assumption of known motion of the robot, environmental maps of real scenes are successfully generated by monitoring azimuth changes in the image. Yagi used this property for robot navigation with an

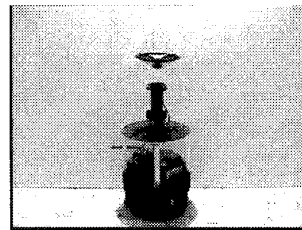


Figure 1: Robot System with HyperOmni Vision

omnidirectional image sensor [4]. Delahoche et al have proposed the incremental map building method based on the exploitation of the azimuth data given by omnidirectional vision and by an odometer [5]. The robot position estimation and map updating are based on the use of an Extended Kalman Filter. However, it is difficult to observe the exact motion parameters of the robot because of encoder measurement error. Observational errors in the generated environmental map accumulate in long movements of the robot. To generate a large environmental map, it is desirable not to assume known robot motion.

In this paper, under the assumption of unknown motions of the robot, we propose a method to generate a stationary environmental map and estimate the ego-motion of a robot, by using an omnidirectional image sensor. A relative relation between the environment map and robot location and orientation can be defined by a nonlinear observational equation. Our method can estimate all parameters in real-time by defining the nonlinear observational equation as a combination of two linear observational equations.

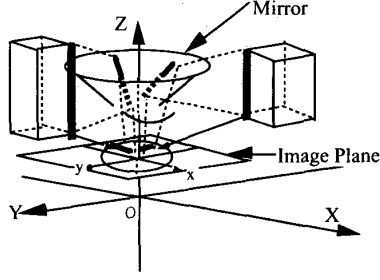


Figure 2: Hyperboloidal Projection

## 2 Robot System with HyperOmniVision

Figure 1 shows the robot system with an omnidirectional image sensor, HyperOmni Vision, using a TV camera with its optical axis aligned with the hyperboloidal mirror's axis [6]. Mounting a HyperOmniVision on a robot so that optical axis is vertical, we can acquire a 360-degree view around the robot. A hyperboloidal mirror yields the image of a point in space on a vertical plane through the point  $P$  and its axis. Thus, the point  $P$  at  $(X, Y, Z)$  is projected onto the image point  $p$  at  $(x, y)$  such that

$$\tan \theta = Y/X \quad (1)$$

This means that the angle in the image, which can be easily calculated as  $y/x$  shows the azimuth angle  $\theta$  of the point  $P$  in space. Also, it can be easily understood that all points with the same azimuth in space appear on a radial line through the image center as shown in Figure 2. Therefore, with a hyperboloidal projection, the vertical edges in the environment appear radially in the image and azimuth angles are invariant to changes in distance and height. In this paper, we use the locus of azimuth angle of vertical edges while the robot is moving.

## 3 Assumption

The following properties of the environment and the mobile robot are assumed for image analysis.

The floor is almost flat and horizontal while walls and stationary objects such as desks or shelves have vertical planes. The robot moves in a man-made environment such as a corridor in a building or a road in down town. Motion parameters are two translational components ( $U, V$ ) and one rotational component  $\alpha$ . All parameters are unknown.

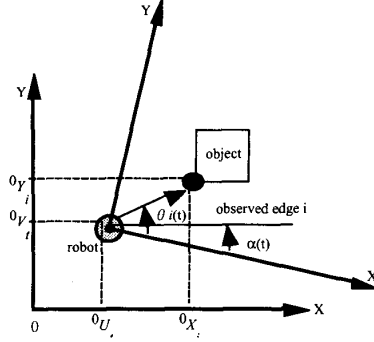


Figure 3: Coordinate System

## 4 Map Generation and Egomotion Estimation

### 4.1 Fundamental Principle

Let us denote the robot location and orientation at time  $t$  by  $({}^0U_t, {}^0V_t)$  and  $\alpha(t)$ , respectively. As shown in Figure 3, defining the position of the object  $i$  at time  $t = 0$  by  $({}^0X_i, {}^0Y_i)$ , the relation between the observed azimuth angle  $\theta_i(t)$  of object  $i$  at time  $t$  and the object location relative to the robot is obtained as follows,

$$\tan(\theta_i(t) - \alpha(t)) = \frac{{}^0Y_i - {}^0V_t}{{}^0X_i - {}^0U_t} \quad (2)$$

Here, unknown parameters are robot locations  $({}^0U_t, {}^0V_t)$ , orientation  $\alpha(t)$  and object locations  $({}^0X_i, {}^0Y_i)$ . The azimuth angle  $\theta_i(t)$  of the object  $i$  at time  $t$  can be obtained by the omnidirectional image sensor. Therefore, the total number of unknown parameters and the total number of observational equations are  $(2i + 3(t - 1) - 1)$  and  $i \times t$ , respectively. If the following relation is satisfied, the robot egomotion and object locations can be estimated at the same time.

$$i \times t \geq (2i + 3(t - 1) - 1) \quad (3)$$

Equivalently, location estimation can be done by observing three object points from five different robot positions or four object points from four different robot positions.

### 4.2 Alternative Real-time Map Generation and Egomotion Estimation

Map generation and robot egomotion estimation can be done by solving the aforementioned nonlinear equation 2. In general, we can solve a nonlinear observational equation by using iterative nonlinear estimation methods such as Levenberg-Marquardt [7]. However, one can consider that it takes a long computational

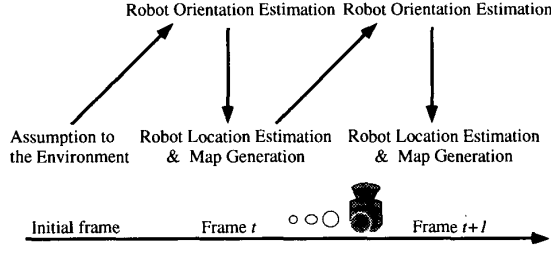


Figure 4: Process Sequence

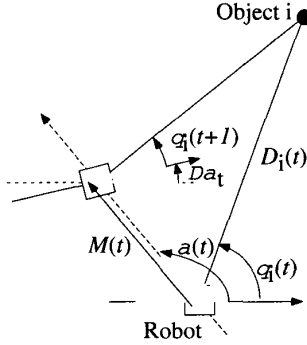


Figure 5: Geometrical Relation between Azimuth Change and Robot Movement

time for converging an evaluation function of the iterative nonlinear estimation method. Therefore, in this paper, we solve the observational equation in real-time by redefining as combination of two linear observational equations. Actually, orientation and location parameters are alternatively estimated while the robot is moving. Figure 4 shows a sequence of the proposed method. First, we assume that object distance is far enough or roughly constant from the robot. An azimuth change of an object is represented by a sine function. Therefore, under this assumption, the robot orientation is estimated by fitting a sine function. If the robot orientation is given, the equation 2 is modified to a linear function, and the location estimation can be done by observing three object points from three different robot positions. When once the location estimation is done, estimated location data is used for azimuth change sine fitting in the next frame. The computational cost of each process is absolutely low. All estimation can be done in real-time while the robot is moving. Details are described in next two sections.

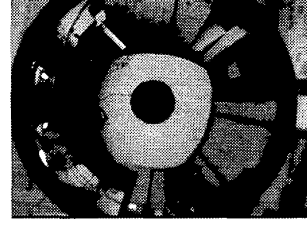


Figure 6: Input Image

#### 4.2.1 Robot Orientation Estimation

In case of a point on the horizontal plane which passes through the focal point  $O_m$ , the circumferential component of optical flow at a point on the horizontal plane (call circumferential flow) is independent of the rolling motion, and is caused by the sway motion and the translational motion of the robot. We have used this characteristic for estimating swaying motion of the robot. This characteristic also appears on azimuth changes of objects. Actually, the azimuth changes can be represented by

$$\Delta\theta_i = \frac{M(t)}{D_i(t)} \sin(\theta_i(t) - \alpha(t)) - \Delta\alpha_t \quad (4)$$

As shown in Figure 5,  $M(t)$ ,  $D_i(t)$ ,  $\alpha(t)$  and  $\Delta\alpha_t$  are the robot movement, distance between the object  $i$  and the robot, the robot orientation at time  $t$  and the azimuth change between time  $t$  and  $t+1$ , respectively. From the equation 4, under a swaying motion, the magnitude of azimuth changes is constant regardless of observed azimuths. Under a translational motion, the magnitude of azimuth changes depends on  $M(t)/D_i(t)$ . Thus, it is difficult to fit observed data with the equation because object distances  $D_i(t)$  are different from each other. In our previous work, we focused on the sign of this equation that corresponds with that of the numerator because of the positive denominator [8]. The azimuth changes have the opposite sign with respect to the direction of robot's translational motion. Therefore, the swaying motion was estimated by evaluating the sign of azimuth changes. However, the sign of the azimuth changes was sensitive to observational noise.

In this paper, we do not evaluate the sign but fit observed azimuth changes to the equation 4 directly. As mentioned before, the magnitude of obtained azimuth changes depends on  $D_i(t)$ . If the distance  $D_i(t)$  is known, unknown parameters are  $M(t)$ ,  $\alpha(t)$  and  $\Delta\alpha_t$ , and we can fit the sine function to equation 4. Equivalently, robot orientation estimation can be done by observing three object points from two different robot positions. Therefore, we normalize the obtained az-

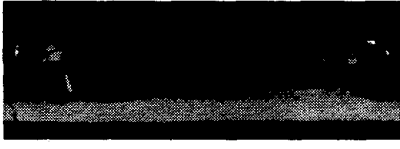


Figure 7: Polar Coordinate Transformation



Figure 8: Edge Image

imuth changes by distance  $D_i(t)$ . Actually, distance  $D_i(t)$  is calculated from the generated map in the prior frame. In general, obtained azimuth angles have observational error. Thus, by using linear least squares method, we estimate the swaying motion (robot orientation) from all obtained azimuth change data whose objects distances have been already estimated in the prior frame.

#### 4.2.2 Map Generation and Robot Location Estimation

If robot orientation  $\alpha(t)$  is given, unknown parameters are robot locations  $({}^0U_t, {}^0V_t)$  and object locations  $({}^0X_i, {}^0Y_i)$ . The total number of unknown parameters and total number of observational equations are  $(2i+2(t-1)-1)$  and  $i \times t$ , respectively. The location estimation can be done by observing three object points from three different robot positions [9]. The fast and robust estimation can be done because the equation 2 becomes linear. In the proposed method, robot orientation is calculated by the normalized azimuth change fitting method described in the prior section.

The environmental map and robot location have estimated error due to observation error of azimuth angle of vertical edges. Therefore, we estimate the more precise location using consecutive measurements by the least squares method. Mathematically, if we define the squared error  $f$ , as shown in equation 5, of the least squares method using the equation 2, values of location  $({}^0X_i, {}^0Y_i)$  can be found by solving the following partial differential equations.

$$f = \sum_i \sum_t \left\{ \begin{array}{l} ({}^0X_i - {}^0U_t) \sin(\theta_i(t) - \alpha(t)) \\ - ({}^0Y_i - {}^0V_t) \cos(\theta_i(t) - \alpha(t)) \end{array} \right\}^2 \quad (5)$$

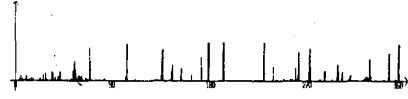


Figure 9: 1-D Projection Data

$$\begin{bmatrix} \frac{\partial f}{\partial {}^0X_k} \\ \frac{\partial f}{\partial {}^0Y_k} \\ \frac{\partial f}{\partial {}^0U_j} \\ \frac{\partial f}{\partial {}^0V_j} \end{bmatrix} = \begin{bmatrix} \sum_i \sum_t A(i, t) \sin(\theta_i(t) - \alpha(t)) \\ - \sum_i \sum_t A(i, t) \cos(\theta_i(t) - \alpha(t)) \\ - \sum_i \sum_t A(i, t) \sin(\theta_i(t) - \alpha(t)) \\ \sum_i \sum_t A(i, t) \cos(\theta_i(t) - \alpha(t)) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$A(i, t) = \begin{Bmatrix} ({}^0X_i - {}^0U_t) \sin(\theta_i(t) - \alpha(t)) \\ - ({}^0Y_i - {}^0V_t) \cos(\theta_i(t) - \alpha(t)) \end{Bmatrix} \quad (6)$$

$$(k = 1, \dots, i, j = 1, \dots, t)$$

## 5 Algorithm for Map Generation

### 5.1 Estimation of loci of azimuths

As shown in Figure 7, an input omnidirectional image (See Figure 6) is transformed into 2D polar coordinates. Next, as shown in Figure 8, we apply a 3x3 Sobel operator to the 2D polar image and project it onto the horizontal axis to get a 1-D projection as shown in Figure 9. To estimate the loci of azimuths of vertical edges, the correspondence of edge length between edges in the 1-D projection of consecutive images is established by using a correlation method in the restricted search field. As consecutive images are sampled densely, one can consider that the azimuth angle of the vertical edge in next frame is in the neighborhood of the azimuth angle in the current frame. Therefore, a certain margin of search field in the next frame is set around the current azimuth angle of the obtained vertical edge. After matching a few frames, the search region can be limited to a narrow one by calculating the locus of each edge from the equation 2. We then evaluate the conformation of neighboring relations.

### 5.2 Global map building by combining local maps

A global map is generated by combining local maps at each position of the robot. Generally, the error of the measurement by triangulation is inversely proportional to the trigonometric parallax and the distance between view positions. A large estimated error of local map occurs when the azimuth angle of the vertical edges does not change significantly. Therefore, combination of local maps is done by selecting the object location estimated from the azimuth locus with

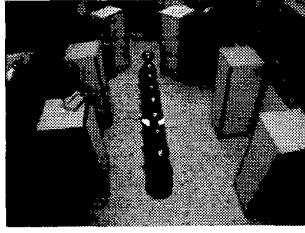


Figure 10: Experimental Scene

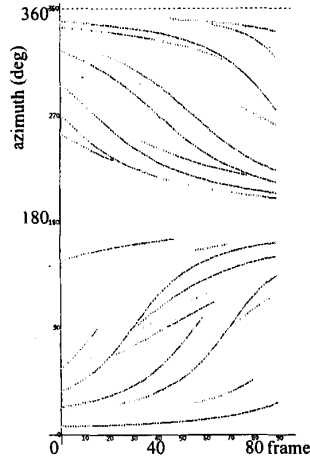


Figure 11: Locus Map of Vertical Edges

large azimuth change and wide standard deviation of azimuth. Because a least squares method give the precise measurement when observational azimuth angles of vertical edges distribute randomly. Actually, we use the following evaluation function  $V_i$ .

$$V_i = |\theta_i(t_2) - \theta_i(t_1)| + \sqrt{\frac{1}{t_2 - t_1} \sum_{k=t_1}^{t_2} \left( \theta_i(k) - \frac{\sum \theta_i(k)}{t_2 - t_1} \right)^2} \quad (7)$$

The first term means the magnitude of azimuth change between time  $t_1$  and time  $t_2$ , and the second term means the distribution of azimuth within interval between time  $t_1$  and time  $t_2$ .

## 6 Experimental Results

Two experiments were conducted for evaluating accuracy of measurements and effectiveness in our com-

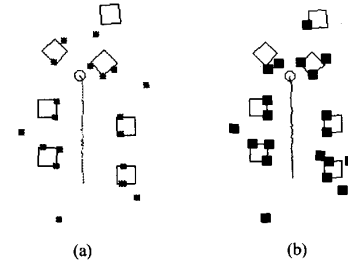


Figure 12: Experimental Results — The robot moved in a straight line — (a) Proposed method (b) In case of given robot orientation



Figure 13: Experimental Scene

puter room.

In the first experiment, the robot moves in a straight line. As shown in Figure 10, the robot moves toward the top. Figure 11 shows the locus map of azimuth angles of vertical edges. The vertical edges drawn with thick lines were used for map generation and location estimation of the robot. Figure 12 are results of map generation and location estimation of the robot. Figure 12 (a) shows a result of map generation and robot location estimation. A black cluttered line shows the estimated trajectory of the robot and the black squares show the estimated map (location of the vertical edges). Figure 12 (b) shows our previous result. In this case, robot orientation was given by the internal sensor and the previous method only generated an environmental map and estimated robot location under assumption of known robot orientation [9]. As shown in Figure 12 (b), a black cluttered line shows the estimated trajectory of the robot and the black squares show the estimated map (location of the vertical edges). The proposed method measures with as high precision as the previous method. Actually, average errors of the location measurement of the robot and stationary environmental map were approximately 3.3 cm and 18cm, respectively.

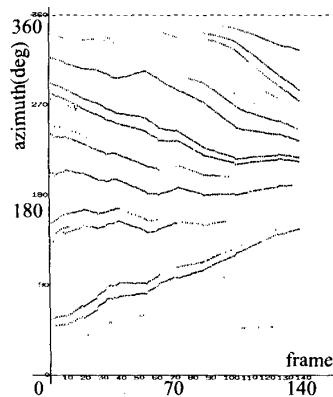


Figure 14: Locus Map of Vertical Edges

A second experiment was done in a more real situation. As shown in 13, the robot snaked through the computer room. Figure 14 shows the locus map of azimuth angles of vertical edges. Figure 15 (a) shows the result of a generated environmental map and estimated trajectory of robot movement. Figure 15 (c) shows our previous result. In this case, robot orientation was given by the internal sensor. In case of the proposed method, the average error of the location measurement of the robot was approximately 10 cm. Figure 15 (b) shows the result when we skipped the process of azimuth change normalization by object distance. In this case, a large error occurred. It means that the normalization is effective for map generation by the proposed method [9].

## 7 Conclusions

In this paper, under the assumption of unknown motions of the robot, we proposed a method to generate a stationary environmental map and estimate the egomotion of a robot, by using an omnidirectional image sensor. We are currently trying to navigate a long route in both an indoor and outdoor environment.

## References

- [1] Saraclik, K. B., "Characterizing an Indoor Environment with a Mobile Robot and Uncalibrated Stereo," *Proc. of IEEE ICRA*, 984-989, 1989.
- [2] Zheng, J. Y. and Tsuji, S., "Panoramic Representation of Scenes for Route Understanding," *Proc. of ICPR*, 161-167, 1990.
- [3] Peri, V. and Nayar S. K., "Omnidirectional Video System," *Proc. of U.S-Japan Graduate Student Forum in Robotics*, 28-31, 1996.

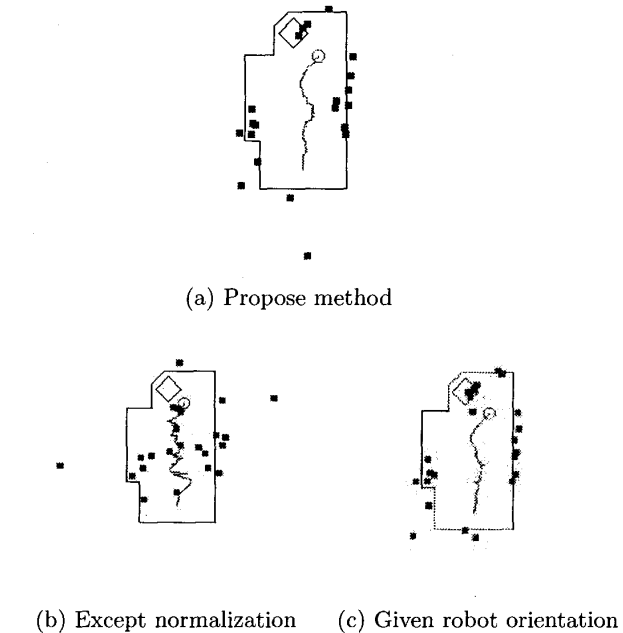


Figure 15: Experimental Results of Map Generation and Location Estimation of the Robot

- [4] Yagi, Y., Nishizawa, Y. and Yachida, M., "Map-based Navigation for a Mobile Robot with Omnidirectional Image Sensor COPIS," *IEEE Trans. Robotics and Automation*, 11, 5, 634-648, 1995.
- [5] Delahoche, L., Pegard, C., Mouaddib, E. M. and Vasseur, P., "Incremental Map Building for Mobile Robot Navigation in an Indoor Environment," *Proc. of IEEE ICRA*, 2560-2565, 1998.
- [6] Yamazawa, K., Yagi, Y. and Yachida, M., "Omnidirectional Imaging with Hyperboloidal Projection," *Proc. of IEEE IROS*, 2, 1029-1034, 1993.
- [7] Etoh, M., Aoki, T. and Hata, K., "Estimation of Structure and Motion Parameters for a Roaming Robot that Scans the Space," *Proc. of Int. Conf. Computer Vision*, vol. 1, 579-584 1999.
- [8] Yagi, Y., Nishii, W., Yamazawa, K., Yachida, M., "Stabilization for Mobile Robot by using Omnidirectional Optical Flow," *IROS*, 618-625, 1996.
- [9] Yagi, Y., Shouya, K., and Yachida, M., "Environmental Map Generation and Egomotion Estimation in a Dynamic Environment for an Omnidirectional Image Sensor," *IEEE Proc. of ICRA*, 2000.