



Title	Real-time omnidirectional image sensor for mobile robot navigation
Author(s)	八木, 康史
Citation	IEEE International Symposium on Intelligent Control – Proceedings. 2002, p. 702-708
Version Type	VoR
URL	<a href="https://hdl.handle.net/11094/14091">https://hdl.handle.net/11094/14091</a>
rights	c2002 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

# Real-time Omnidirectional Image Sensor for Mobile Robot Navigation

Yasushi Yagi

Graduate School of Engineering Science, Osaka University  
1-3 Machikaneyama, Toyonaka, 560-8531 Japan

Email: y-yagi@sys.es.osaka-u.ac.jp

**Abstract**— A real-time omnidirectional camera which can acquire an omnidirectional (360 degrees) field of view at video rate could be applied in a variety of fields such as autonomous navigation, telepresence, virtual reality and remote monitoring. We have developed several omnidirectional image sensors. In this paper, we introduce the outlines and fundamental optics of our developed sensors and show examples of applications for robot navigation.

## I. INTRODUCTION

While there has been much work on mobile robots with vision systems which navigate in both unknown and known environments [1-4], most robots view things only in front of them and avoid static obstacles and as a result, they may collide against objects moving from the side or from behind. To overcome these drawbacks, a sensor is needed to view the environment around the robot so that it may navigate safely.

There have been several attempts to acquire omnidirectional images. 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].

From 1988, we have developed several real-time omnidirectional image sensors which consist of a CCD camera and a convex mirror placed in front of the camera which can observe panoramic image at once, and can be applied to robot navigation and human interaction problems [2], [?].

In this paper, we introduce our omnidirectional image sensors, named COPIS and HyperOmni Vision. COPIS which uses the conic mirror, is suitable for mobile robot navigation because its view field is side [2]. HyperOmni Vision, which consists of a CCD camera and a hyperboloidal mirror placed in front of the camera [3], has an important feature in that it has a focal point (center of projection).

## II. OMNIDIRECTIONAL IMAGE SENSORS

### A. COPIS and Conic Projection

The omnidirectional image sensor COPIS (Conic Projection Image Sensor) [2], [?] as mounted on a robot, consists of a conic mirror and a TV camera, with its optical axis aligned with the cone's axis, in a cylindrical glass tube, as shown in Figure 1(a). The key feature of COPIS is passive sensing of the omnidirectional environment in real-time (at the frame rate of a TV camera), using a conic mirror (See Figure 1(b)).

COPIS maps the scene onto the image plane through a conic mirror and a lens. We call this mapping "conic projection". Let us use the three dimensional coordinate system  $O-XYZ$ , aligned with the image coordinate system  $o-xy$  and the  $Z$ -axis pointed toward the cone's vertex (see Figure 2). We fix origin  $O$  at the camera center, thus the image plane is on a level of  $f$ , where  $f$  is the focal length of the camera. A conic mirror yields the image of a point in space on a vertical plane through the point and its axis. Thus, the point  $P$  at  $(X, Y, Z)$  is projected onto the image point  $p$  at  $(x, y)$  such that

$$\tan \theta = \frac{Y}{X} = \frac{y}{x} \quad (1)$$

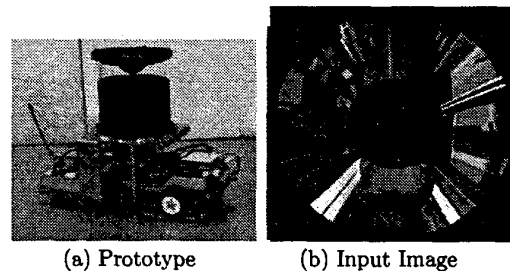


Fig. 1. COPIS

In other words, all points with a same azimuth in space appear on a radial line through the image center and its direction  $\theta$  indicates the azimuth. This means that the vertical line in the environment appears radially in the image plane. The vertical lines provide a useful cue in a man-made environment such as a room, containing many objects with vertical edges; for example, doors, desks and shelves. Many researchers use this characteristic for robot navigation. Details of our robot navigation methods are described later.

The field viewed by COPIS is limited by the vertex angle of the conic mirror and the visual angle of the camera lens. Therefore, the main field viewed by COPIS is a side view and the resolution along a radial direction of the conic mirror is sufficient to extract vertical edges. As shown in Figure 2(b), the vertical section of COPIS is a flat mirror and the ray from the point  $P$  changes its optical direction using it. This means that the vertical angular resolution

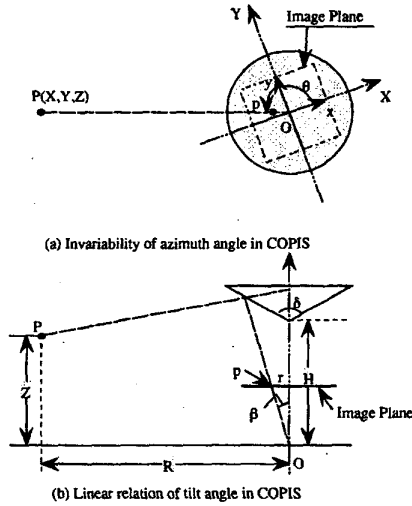


Fig. 2. Conic Projection

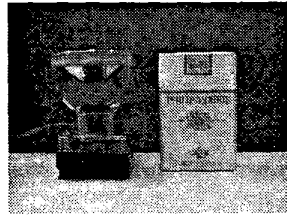


Fig. 3. HyperOmni Vision

of COPIS is linear in the input image. We represent the image and the space by two polar coordinate systems  $(r, \theta)$  and  $(R, \theta, Z)$ , respectively. As shown in Figure 2, simple geometrical analysis of the vertical section through  $P$  and the  $Z$ -axis yields the following equations of the conic projection :

$$\begin{aligned} r &= f \tan \beta \\ \beta &= \delta - \tan^{-1} \left( \frac{R + H \sin \delta}{Z - H(1 - \cos \delta)} \right) \\ (0 \leq \beta \leq \frac{\pi}{2}, \beta < \delta) \end{aligned} \quad (2)$$

where  $H$  is the distance between lens center and vertex of the cone. If distance  $H$  and the focal length  $f$  are given, the object height  $Z$  can be calculated by substituting the observed vertex angle  $\beta$  into 2.

From another standpoint, the virtual lens center  $O'$  corresponded to the lens center  $O$  is defined by

$$\begin{bmatrix} R \\ Z \end{bmatrix} = \begin{bmatrix} H \sin \frac{\delta}{2} \\ H(1 + \cos \frac{\delta}{2}) \end{bmatrix} \quad (3)$$

The virtual lens center does not cross at a single point but appears on a circle defined by 3. A serious drawback of COPIS is that it does not have a focal point (a center of a viewpoint). Therefore, it is impossible to generate a distort-

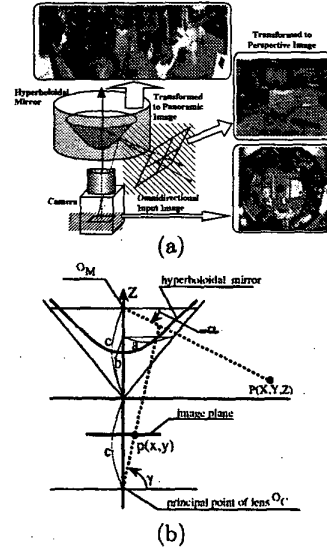


Fig. 4. Hyperboloidal Projection

tion free perspective image from an input omnidirectional image.

#### B. HyperOmniVision

We have developed a new omnidirectional image sensor named HyperOmni Vision which consists of a CCD camera and a hyperboloidal mirror placed in front of the camera (See Figure 3) [3]. The hyperboloidal mirror has a focal point which makes possible easy generation of any desired image projected on any designated image plane, such as a perspective image or a panoramic image, from an omnidirectional input image deformed by the hyperboloidal mirror (See Figure 4(a)). This easy generation of perspective or the panoramic images allows the robot vision to use existing image processing methods for autonomous navigation and manipulation. It also allows a human user to see familiar perspective images or panoramic images, instead of an unfamiliar omnidirectional input image deformed by the hyperboloidal mirror.

The hyperboloidal surfaces can be obtained by revolving hyperbola around the  $Z$  axis and have two focal points at  $(0, 0 + c)$  and  $(0, 0, -c)$  as shown in Figure 4(b). Using the world coordinates system  $(X, Y, Z)$  the hyperboloidal surface can be represented as:

$$\begin{aligned} \frac{X^2 + Y^2}{a^2} - \frac{Z^2}{b^2} &= -1 \\ c &= \sqrt{a^2 + b^2} \end{aligned} \quad (4)$$

where  $a$  and  $b$  define the shape of a hyperboloidal surface.

We use one of the hyperboloidal surfaces at  $Z > 0$  as the mirror. Figure 4(b) shows the geometry of HyperOmni Vision which consists of a CCD camera and a hyperboloidal mirror. It should be noted that the focal point of the hyperboloidal mirror  $O_M$  and the lens center of camera  $O_C$  are

fixed at the focal points of hyperboloidal surfaces  $(0, 0, c)$  and  $(0, 0, -c)$ , respectively and the axis of the camera is aligned with that of the hyperboloidal mirror. The image plane should also be placed at a distance  $f$  (focal length of camera) from the lens center of camera  $O_C$  and be parallel to the  $XY$  plane.

A hyperboloidal mirror yields an omnidirectional image of the scene around its axis. We can observe the image using a TV camera with its optical axis aligned with that of the hyperboloidal mirror. HyperOmni Vision maps the scene onto the image plane through a hyperboloidal mirror and a lens. We call this mapping "hyperboloidal projection".

We will briefly describe how a point  $P$  in space is reflected by the hyperboloidal mirror and projected on the image plane. A ray going from the point  $P(X, Y, Z)$  in space toward the focal point of the mirror  $O_M$  is reflected by the mirror and passes through the other focal point  $O_C$  which intersects the image plane at a point  $p(x, y)$ . Any point in space in the field of view (360 degrees around the  $Z$  axis) of the hyperboloidal projection satisfies this relation. Therefore, we can obtain an omnidirectional image of the scene on the image plane with a single center of projection  $O_M$ .

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. This good feature is the same as a conic projection as shown in 1. Therefore, with a hyperboloidal projection, the vertical edges in the environment appear radially in the image and the azimuth angles are invariant to changes in distance and height.

By simple geometrical analysis, equations relating the point in space  $P(X, Y, Z)$  and its image point on the image plane  $p(x, y)$  can be derived as follows.

$$\begin{aligned} Z &= \sqrt{X^2 + Y^2} \tan \alpha + c \\ \alpha &= \tan^{-1} \frac{(b^2 + c^2) \sin \gamma - 2bc}{(b^2 - c^2) \cos \gamma} \\ \gamma &= \tan^{-1} \left( \frac{\sqrt{x^2 + y^2}}{f} \right) \end{aligned} \quad (5)$$

where  $\alpha$  denotes the tilt angle of the point  $P$  from the horizontal plane,  $f$  is the focal length of camera lens, and  $a$ ,  $b$  and  $c$  are parameters defining the shape of the hyperboloidal mirror.

From equations 1, 4 and 5, the azimuth angle  $\theta$  and the tilt angle  $\alpha$  of the line connecting the focal point of the mirror  $O_M$  and the point in space  $P$  can be obtained from the position of the image point  $p(x, y)$ . This means that the equation of the line connecting  $O_M$  and  $P$  can be determined uniquely from the coordinates of image point  $p(x, y)$ , regardless of the location of the point  $P$  in space.

### III. OMNIDIRECTIONAL SENSING FOR AUTONOMOUS ROBOT

#### A. Egomotion and posture estimation

A general technique for estimating egomotion is presented by Nelson and Aloimonous [6]. To estimate egomotion with 6 DOF, this technique requires optical flows on three orthogonal great circles on spherical coordinates. Egomotion is determined by iterative calculation, so the computational cost is high. Gluckman and Nayer have estimated 6 DOF of the egomotion of the camera by using their omnidirectional camera, however, estimation error is not so small [7].

In the case of mobile robot navigation, the egomotion of the robot is usually caused by juggling due to unevenness in the ground plane. Under the assumption of the ground plane motion of the robot, we have proposed the method for robustly estimating the rolling and swaying motions of the robot from omnidirectional optical flows by using special characteristics of an omnidirectional camera with a single center of projection [9].

The optical flow for a projected point on the image plane can be decomposed into perpendicular radial and circumferential components with respect to the center of projection. The circumferential component is perpendicular to the radial one. From 1 and 5, the radial and the circumferential component can be defined by  $\frac{\partial \theta(t)}{\partial t}$ ,  $\frac{\partial \alpha(t)}{\partial t}$  respectively. From 5, the input image taken by HyperOmni Vision can be easily transformed to the coordinates where the origin is fixed at the focal point  $O_M$ . Therefore, it is equivalent to evaluate  $\frac{\partial \alpha(t)}{\partial t}$  in place of the radial component of optical flow  $\frac{\partial \theta(t)}{\partial t}$ . Now, we examine how both the flow components,  $\frac{\partial \alpha(t)}{\partial t}$  and  $\frac{\partial \theta(t)}{\partial t}$ , arise.

Supposing that the robot moves horizontally in a static environment. From 1 and 5, generally,  $\frac{\partial \alpha(t)}{\partial t}$  and  $\frac{\partial \theta(t)}{\partial t}$  are caused by translational motions of the robot. However, in the case of a point on the horizontal plane which passes through the focal point  $O_M$ , from 5, the tilt angle holds zero constantly because  $Z$  is equal to  $c$ . This means that the radial component of optical flow at any point on the horizontal plane which passes through the focal point  $O_M$  is independent of the translational motion of the robot. The radial component of optical flow at a point on the horizontal plane is effected by the rolling motion as shown in Figure 5. Here, illustrations use the cylindrical coordinate system. The rotation is referred to as "rolling motion" and the translation is referred to as "vertical motion". We call the radial component of optical flow at a point on the horizontal plane in the cylindrical coordinates as "radial flow".

On the other hand, the circumferential component of optical flow  $\frac{\partial \theta(t)}{\partial t}$  at a point on the horizontal plane (called circumferential flow) is independent of the rolling motion. The circumferential flow is caused by the sway motion and the translational motion of the robot as shown in Figure 6.

As a result, the radial and the circumferential flows in the cylindrical coordinates can be represented by a general sine function. As shown in the next equation, in case

of the radial flow, the amplitude, phase shift and offset of sine function corresponded to the roll angle  $\delta(t)$ , direction (angle  $\gamma(t)$ ) of rolling axis and deviation  $\alpha_t$  of vertical position, respectively.

$$y_{cyl}(\theta_{it}) = \tan \delta_t \sin \theta_{it} + \gamma_t + \frac{\alpha_t}{R_{it}} f \quad (6)$$

The circumferential flow can also be represented by the following function.

$$y_{cyl}(\theta_{it}) = \frac{M_{Ht}}{R_{it}} f \sin(\theta_{it} + \xi_t) + \tan \psi_t \quad (7)$$

Here,  $R_{cyl}$  is the radius of the projected cylindrical image plane.  $f$  is the focal length of camera.  $y_{cyl}(\theta_{it})$  is the magnitude of the radial flow at azimuth angle  $\theta_{it}$  in the cylindrical coordinates.  $R_{it}$  and  $\theta_{it}$  are the distance and azimuth angle of the feature point  $i$  at time  $t$ .  $M_{Ht}$  is the horizontal motion,  $R_{it}$  is the distance to the feature point at azimuth angle,  $f$  is the focal length of the camera,  $\xi_t$  is the direction of horizontal translation, and  $\epsilon_t$  is the rotation about the  $Z$  axis.

As shown in 6, the offset  $\frac{\alpha_t}{R_{it}} f$  actually depends on the distance from the robot to the observed feature point. Therefore, in the case of a large difference in distances among observing features, it is difficult to estimate the precise rolling motion of the robot. However, in general, the distance  $R_{it}$  is sufficiently shorter than the magnitude of the deviation  $\alpha_t$  of the vertical motion, and most of the observed features are at a similar distance from the robot. Therefore, one can consider that  $\frac{f}{R_{it}}$  is approximately constant. Under a swaying motion, the magnitude of circumferential flows are constant regardless of observed azimuths (See Figure 6(a)). Under a translational motion, the magnitude of obtained flow depends on  $\frac{M_{Ht}}{R_{it}}$ . Thus, it is difficult to fit observed data with the equation because  $M_{Ht}$  is not enough smaller than  $R_{it}$ . However, the sign of this equation corresponds with the one of the numerator because of the plus denominator. The circumferential flows have the opposite sign with respect to the direction of translational motion of the robot. Therefore, the swaying motion was estimated by evaluating the sign of the circumferential flow. Figure 7 shows an example of estimated result of radial flow model.

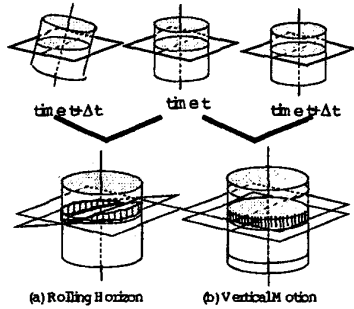


Fig. 5. Radial Flow in HyperOmni Vision

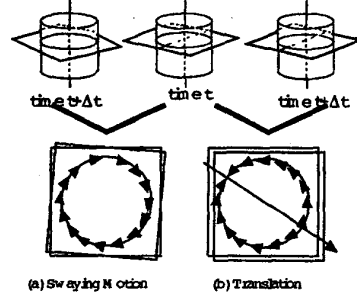


Fig. 6. Circumferential Flow in HyperOmni Vision

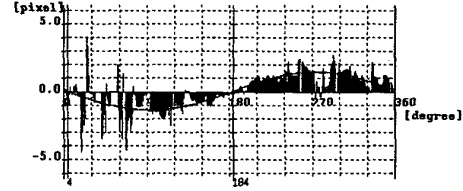


Fig. 7. An Estimated Result of Radial Flow Model

## B. Geometrical Map generation

Generation of a stationary environmental map 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 has been much work on mobile robots with vision systems which navigate in both unknown and known environments. We also propose the methods for guiding navigation of a mobile robot and generating environmental map by monitoring azimuth changes of vertical edges in the image while the robot is moving.

Let us denote the robot location and orientation at time  $t$  by  $({}^0X_t, {}^0Y_t)$  and  $\alpha(t)$ , respectively. As shown in Figure 8, 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,

$$\tan(\theta_i(t) - \alpha(t)) = \frac{{}^0Y_i - {}^0Y_t}{{}^0X_i - {}^0X_t} \quad (8)$$

If the robot location and orientation at time  $t$  are given by a robot internal encoder, unknown parameters in 8 are only the position  $({}^0X_i, {}^0Y_i)$  of the object  $i$  at time  $t = 0$ . Under the assumption of known motion of the robot, we could estimate locations of objects around the robot by triangulation [11].

If robot orientation  $\alpha(t)$  is given, unknown param-

ters are robot locations  $({}^0X_t, {}^0Y_t)$  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. Therefore, the location estimation can be done by observing three object points from three different robot positions [12]. The fast and robust estimation can be done because the equation 8 becomes linear.

Under the assumption of unknown motions of the robot, 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.

$$(i-3)(t-1) \geq 2 \quad (9)$$

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 [13].

Map generation and robot egomotion estimation can be done by solving aforementioned nonlinear equation 8. In general, a nonlinear observational equation can be solved by the iterative nonlinear estimation method such as Levenberg-Marquardt. However, one can consider that it takes a long computational time for converging an evaluation function of the iterative nonlinear estimation method. Therefore, we solve observational equation in real-time by redefining as combination of two linear observational equations [13]. Actually, orientation and location parameters are alternately estimated while the robot is moving.

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 by 7. If the robot orientation is given, the equation 8 is modified to a linear function, and the location estimation can be done by observing three object points from three different robot positions. Once the location estimation is done, estimated location data are 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. Figure 9 shows the experimental result of map generation.

### C. Robust map generation by sensor fusion

Due to the cost and limitation of various sensors and stability of sensing, a single sensor is generally not sufficient to provide a satisfying result. Thus sensor fusion or integration has been widely used to enhance the map precision. Among them, the ultrasonic sensor is often used with the vision sensor. This is due to the fact that the ultrasonic sensor is fast at acquisition of the environment model, though it is poor at angle resolution and error-prone as specular reflection occurs. On the other hand, the vision sensor is good in resolution, though its slow at image

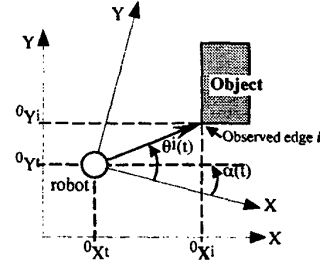


Fig. 8. Coordinate System

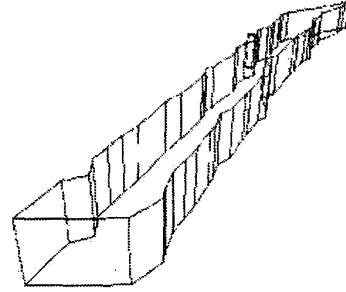


Fig. 9. Generated 3D Model (Corridor)

processing, and suffers from varying lighting conditions in the environment. In map building, by combining the use of these two sensors, it allows the exploitation of the visions area or edge information with the readily available ultrasonic range information.

To build the floor map, we adopt the fusion of an omnidirectional vision sensor with the ultrasonic sensor [14]. The omnidirectional vision sensor we use can get a less distorted and faster scan of the surrounding floor than traditional vision sensors. Also, the range filtering method we use on the ultrasonic range data can provide a more reliable initial estimation of the free space. To facilitate the fusion of heterogeneous sensor input at all times, we use the grid based representation as our fusion basis. A safety index defined on the grid representation is first derived from the filtered range data, and then revised according to the color and edge information extracted from the inverse perspective transformed floor image. The final obtained free space in the floor map can then be used for either obstacle avoidance or robot relocation by the methods mentioned above.

The system consists of two major modules. The image module gets input of an omnidirectional image and corrects the distortion by inverse perspective transformation. The distortion corrected image is then fed into the edge extraction and color clustering modules to obtain the edge map, and the color map, respectively. These two maps reflect the connectivity information about the robot's surrounding environment. The sonar module gets input of 16 sonar ring readings, and produces a sonar map. The sonar map reflects the range information about the robot's environment.

From the sonar map, an initial area of the surrounding free space is also estimated and used by the color clustering module to give an initial estimation of the surrounding floor color. The local map module does the main fusion process for the three separate maps previously obtained. The fusion result represents the environment model of the robot. When the robot moves, the maps are shifted with encoder information to keep in validity. Other modules like plan primitive and command modules can be added for robot control. In terms of fusion, there are three levels of sensor fusion involved in these modules. The sonar module does the fusion of 16 ultrasonic sensors to produce the echo map; the color clustering modules fuse an initial free space information from the echo map to produce a uniform color map; and finally the local map module fuses the echo, color and edge map from different sensors to produce the final free space around the robot. Figure 10 (a) and (b) show the input floor map after inverse perspective transformation and the final estimated free space, respectively.

#### D. Route navigation

##### D.1 Reactive navigation

Low-level action commands such as road following and collision avoidance need fast reaction capability. An artificial potential field is a common approach to realize a reactive collision avoidance behavior. We describe a new method for reactive visual navigation based on omnidirectional sensing [15]. The robot is projected at the center of the input image by the omnidirectional image sensor HyperOmni Vision. Therefore, rough free space around the robot can be extracted by the active contour model (See Figure 11). The method produces low-level commands that keep the robot in the middle of the free space and avoid collision by balancing the shape of extracted active contours.

The basic idea of following a path and avoiding collision is similar to a combination of Santos-Victor's method [16] and Holenstein's method [17], respectively. Santos-Victor is based on optical flows from two cameras directed opposite each other and just following the corridor. It may collide against objects moving from the side or behind because of only lateral observation. For collision avoidance, Holenstein measured omnidirectional range data by setting 24 ultrasonic sensors around the robot. We can generate both behaviors at the same time by making use of the omnidirectional image sensor, then the robot can avoid obstacles and move along the corridor by tracking the close-looped curve with an active contour model. Furthermore, the method can directly represent the spatial relations between the environment and the robot on the image coordinate. Thus the method can control the robot without geometrical 3D reconstruction.

A boundary of free space around the robot can be represented by a close-looped line in the input image because the robot is projected to the center of the input image by HyperOmni Vision. First, a rough free space area around the robot is detected by using a contour extraction model, which is a small influence of environmental changes, ba-

sically based on an active contour model. Then, by evaluating a shape of extracted free space area, the method produces low-level motor commands that allow it to avoid the obstacle and keep the robot in the center of the free space.

By evaluating three features calculated from a shape of extracted free space area (active contours), the method produces low-level motor commands that avoid the obstacle and keep the robot in the middle of the free space. To be more specific, the robot action (moving direction) is determined by combining three features; the principal axis of inertia, the center of gravity of the extracted free space region and repulsive forces from the environment (obstacle and wall).

As shown in Figure 12 (a), the possibility of collision with the unknown object became evident, and the robot changed its direction and moved around the arc toward the left side. Next, as shown in Figure 12 (b), the robot changed its direction and moved around the arc toward the right side, and finally came back to the middle of the corridor.

##### D.2 Geometrical map-based navigation

Since the environmental map is given, the location and the motion of the robot can be estimated by detecting the azimuth of each object, in the omnidirectional image [11]. We assume that the robot is initially parked at a standard position, driven around a room and a corridor of the building via a given route and then rough movement of the robot can be measured using an internal sensor. There are, however, measurement errors caused by a swaying motion of the robot. The sway motion means unobservable errors are made in orientation when tires slip on slightly rough ground. A rough location of the robot can be calculated from the starting position and from movement. Thus, the azimuth angles of vertical edges are estimated using the given map and rough location. Using azimuth information from both the input image and the environmental map, one can estimate location and motion of the robot, even if all edges are not extracted correctly from the omnidirectional image, because the omnidirectional image sensor can observe a 360 degree view around the robot. After matching the observed edges with the environmental map, the undetectable vertical edges are recognized as unknown obstacles and their locations are estimated. When there is an obstacle along a path toward a given goal position, the robot has to change its path. By evaluating the distance between the robot and the goal position, the robot plans a new minimum length path without colliding against the obstacle and moves toward the goal position.

## IV. CONCLUSIONS

In this paper, we describe our omnidirectional image sensors, named COPIS and HyperOmni Vision. COPIS which uses the conic mirror, is suitable for mobile robot navigation because its main field of view is a side view. HyperOmni Vision has an important feature in that it has a focal point (center of projection) and an input image can easily be

transformed to any desired image projected on any designated image plane, such as a pure perspective image or a panoramic image.

We also propose several methods that apply an omnidirectional image sensor for robot navigation. Advantages of omnidirectional sensing is not only its wide view angle but also special properties such as invariability of azimuths, circumferential continuity, periodic characteristic, symmetric characteristic and rotational invariant. Proposed methods use these characteristics of omnidirectional image sensors for image processing. The fundamental research on the omnidirectional image processing and sensor designing that use inherent optical characteristics of the omnidirectional image sensor continues to be studied.

#### ACKNOWLEDGEMENT

This work was supported by a Grant-in-Aid for Scientific Research from the Ministry of Education, Science and Culture, Japanese Government, and Japan Society for the Promotion of Science under grant JSPS-RFTF 99P01404.

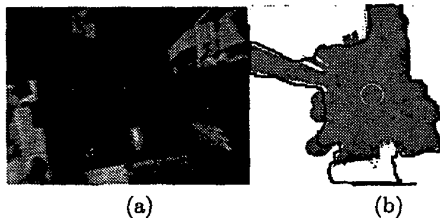


Fig. 10. The floor map after inverse perspective transformation and the final estimated result of free space

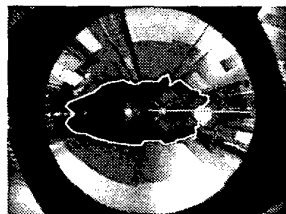


Fig. 11. Result of Extracted Free Space and Principal Axis of Inertia around the Center of Gravity

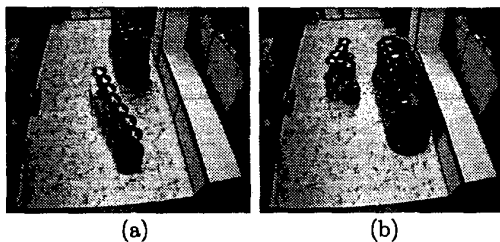


Fig. 12. A Result of Collision Avoidance against an Unknown Moving Obstacle

#### REFERENCES

- [1] Yagi, Y.: Omnidirectional Sensing and Its Applications, *IEICE Trans. Information and Systems*, Vol.E82-D, No.3, pp.568-579 (1999).
- [2] Yagi, Y. and Kawato, S.: Panorama Scene Analysis with Conic Projection, *Proc. IEEE/RSJ Int. Workshop on Intelligent Robots and Systems*, pp.181-187 (1990).
- [3] Yamazawa, K., Yagi, Y. and Yachida, M.: Omnidirectional Imaging with Hyperboloidal Projection, *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, No.2, pp.1029-1034 (1993).
- [4] Yagi, Y., Kawato, S. and Tsuji, S.: Real-time Omnidirectional Image Sensor (COPIS) for Vision-guided Navigation, *IEEE Trans. Robotics and Automation*, Vol.10, No.1, pp.11-22 (1994).
- [5] Nelson, R.C. and Aloinomos, J.: "Finding Motion Parameters from Spherical Motion Fields," *Biological Cybernetics*, 58, pp.261-273 (1988).
- [6] Gluczman, J. and Nayer, S. K.: Ego-motion and Omnidirectional Cameras, *Proc. IEEE Int. Conf. on Computer Vision*, pp.999-1005 (1998).
- [7] Yagi, Y., Nishii, W., Yamazawa, K. and Yachida, M.: Stabilization for Mobile Robot by using Omnidirectional Optical Flow, *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp.618-625 (1996).
- [8] Yagi, Y., Nishizawa, Y. and Yachida, M.: Map-Based Navigation for a Mobile Robot with Omnidirectional Image Sensor COPIS, *IEEE Trans. Robotics and Automation*, Vol.11, No.5, pp.634-648 (1995).
- [9] Yagi, Y., Shouya, K. and Yachida, M.: Environmental Map Generation and Egomotion Estimation in a Dynamic Environment for an Omnidirectional Image Sensor, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp.3493-3498 (2000).
- [10] Yagi, Y., Hamada, H., Benson, N. and Yachida, M.: Generation of Stationary Environmental Map under Unknown Robot Motion, *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp.1487-1492 (2000).
- [11] Wei, S., Yagi, Y. and Yachida, M.: Building a Local Floor Map by Use of Ultrasonic and Omnidirectional Vision Sensors, *Advanced Robotics*, Vol.12, No.4, pp.433-453 (1998).
- [12] Yagi, Y., Nagai, H., Yamazawa, K. and Yachida, M.: Reactive Visual Navigation based on Omnidirectional Sensing - Path Following and Collision Avoidance-, *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Vol.1, pp.58-63 (1999).
- [13] Santos-Victor, J., Sandini, G., Curotto, F. and Garibaldi, S.: Divergent Stereo in Autonomous Navigation: From Bees to Robots, *Int. J. of Computer Vision*, 14, pp.159-177 (1995).
- [14] Holenstein, A. and Badreddin, E.: Collision Avoidance in a Behavior-based Mobile Robot Design, *Proc. IEEE Int. Conf. on Robotics and Automation*, Vol.1, pp.898-903 (1991).