Iconic Memory-Based Omnidirectional Route Panorama Navigation

Yasushi Yagi, *Member*, *IEEE*, Kousuke Imai, Kentaro Tsuji, and Masahiko Yachida

Abstract—A route navigation method for a mobile robot with an omnidirectional image sensor is described. The route is memorized from a series of consecutive omnidirectional images of the horizon when the robot moves to its goal. While the robot is navigating to the goal point, input is matched against the memorized spatio-temporal route pattern by using dual active contour models and the exact robot position and orientation is estimated from the converged shape of the active contour models.

Index Terms—Omnidirectional vision, route panorama, localization, navigation, active contour model.

1 Introduction

A real-time omnidirectional camera that can acquire an omnidirectional (360 degrees) field of view at a video rate has applications in a variety of fields; one such area is that of autonomous navigation. Several researchers have investigated geometrically-based and iconic memory-based navigation methods using omnidirectional image sensors. [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12].

Iconic memory-based navigation is a common approach for visual navigation; its basic operation is a comparison between the present sensory input and previously memorized images. It is easy to relate a robot's action and sensory data without a geometrical model. Zheng and Tsuji's robot memorized the side of scene of a route from a panoramic view while it moved along the route [13]. Matsumoto et al.'s robot memorized the whole front view image at reference points along the route for visual navigation [14]. The correspondences between a present input image and previously memorized images were established using dynamic programming (DP) matching and correlation methods, respectively. Ishiguro and Tsuji have proposed a compact representation by expanding it into a Fourier series [15]. Each input image is memorized by the coefficients of the low-frequency components. KL transformation is another approach to compress memorized data [16], [17]. Ulrich and Nourbakhsh memorized utilizing a simple color histogram and navigated their mobile robot in both indoor and outdoor environments [18].

Most of these previous iconic memory-based approaches, in essence, select the image that corresponds to an input image from discretely memorized images. This means that it is impossible to estimate the position and orientation of the robot when it substantially deviates from a memorized path or the intervals between the memorized positions are too great. Therefore, the precision of position and orientation of a mobile robot depends on the spatial sampling density of a moving space. However, to precisely estimate a robot's position, these methods need large numbers of images to effectively memorize a moving space.

In this paper, we propose a new iconic memory-based navigation method that synthesizes a corresponding image pattern from an omnidirectional route panorama (ORP) that can be acquired by arranging points on the horizontal plane which passes through the virtual center of the lens and is taken by the robot moving along a route. The proposed method searches for a corresponding image pattern on the ORP. In reality, active contour models (ACMs) are used, and the position and orientation of the robot are estimated from the converged shape of the ACMs. The amount of memorized image data is small and the method is suitable for navigation over long distances. Furthermore, the robot position and orientation can be estimated, another definite positive advantage of the proposed method.

In a previous work, we used an ORP for iconic memory-based navigation [19]. However, the method transformed the window region, in a certain number of past frames of the ORP, which is a standard unit of a spatio-temporal representation, to a 2D Fourier power spectrum. The route was then memorized by a series of 2D Fourier power spectra. While the robot is navigating towards the goal point, it is controlled by comparisons with the pattern of the memorized Fourier power spectrum and its principal axis of inertia. The method can directly represent the temporal and spatial relations between environment and the robot, but it cannot estimate the exact robot position and orientation. In this earlier method, the robot was controlled by a visual-servo-like technique.

2 Omnidirectional Route Panorama

A robot moving along a route can observe objects in many directions, with a sensor being needed to view the environment around the robot so that it can navigate safely. We have proposed several omnidirectional image sensors such as COPIS, MISS, and HyperOmniVision for robot

- *Y. Yagi is with The Institute of Scientific and Industrial Research, Osaka University, 8-1 Mihogaoka, Ibaraki, Osaka, 567-0047 Japan. E-mail: yagi@am.sanken.osaka-u.ac.jp.*
- *K. Imai, K. Tsuji, and M. Yachida are with the Graduate School of Engineering Science, Osaka University, 1-3 Machikaneyama-cho, Toyonaka, Osaka, 560-8531 Japan. E-mail: k0usuke@rose.freemail.ne.jp, k-tsuji@yachi-lab.sys.es.osaka-u.ac.jp, yachida@sys.es.osaka-u.ac.jp.*

Manuscript received 17 Sept. 2002; revised 7 June 2004; accepted 7 June 2004. Recommended for acceptance by M. Irani.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 117392.

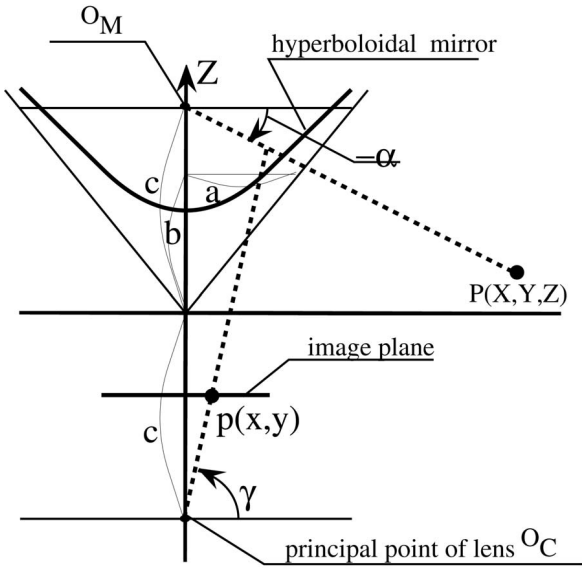


Fig. 1. Hyperboloidal projection has characteristics of a single center of projection.

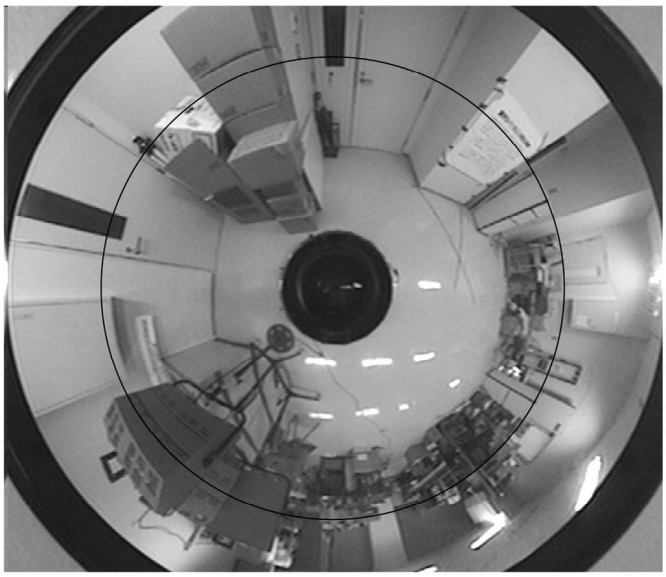


Fig. 2. Points on the horizontal plane that pass through the virtual center of the lens appear as a circle on an omnidirectional image.

navigation [6], [7], [5]. The method proposed here uses the image sequence of a horizontal part of the omnidirectional image, the ORP, while the robot continuously scans the view along the route.

2.1 Optical Relation of HyperOmni Vision

The hyperboloidal surfaces can be obtained by revolving hyperbola around the Z axis and having two focal points at $(0, 0+c)$ and $(0, 0, -c)$, as shown in Fig. 1. Using a world coordinates system (X, Y, Z), the hyperboloidal surface can be represented as:

$$\begin{aligned} &\frac{X^2+Y^2}{a^2} - \frac{Z^2}{b^2} = -1 \\ &c = \sqrt{a^2+b^2}, \end{aligned} \tag{1}$$

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

We use one of the hyperboloidal surfaces at $z > 0$ as a mirror. HyperOmni Vision consists of a CCD camera and a hyperboloidal mirror. The focal point of the hyperboloidal mirror O_M and the center of camera lens O_C are fixed at the focal points of the 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 is placed at a distance f (focal length of camera) from the camera lens center O_C and is parallel to the XY plane.

A brief description of how a point P in space is reflected by a hyperboloidal mirror and projected on an image plane follows. 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 another focal point O_C that 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, an omnidirectional image of the scene on the image plane can be obtained with a single center of projection O_M.

A hyperboloidal 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

$$\tan\theta = \frac{Y}{X} = \frac{y}{x}. \tag{2}$$

The angle in the image which can be easily calculated as y/x shows the azimuth angle θ of the point P in space. It is also easily understood that all points with the same azimuth in space appear on a radial line through the image center.

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:

$$\begin{aligned} Z &= \sqrt{X^2+Y^2}\tan\alpha + c \\ \alpha &= \tan^{-1}\frac{(b^2+c^2)\sin\gamma - 2bc}{(b^2-c^2)\cos\gamma} \\ \gamma &= \tan^{-1}\left(\frac{\sqrt{x^2+y^2}}{f}\right), \end{aligned} \tag{3}$$

where α 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. This method uses an image on a horizon and, so, the above equations are rewritten as follows:

$$\begin{aligned} (b^2+Z^2)\sin\gamma &= 2bZ \\ \gamma &= \tan^{-1}\left(\frac{\sqrt{x^2+y^2}}{f}\right). \end{aligned} \tag{4}$$

From (4), γ is independent of X and Y. This means that the patterns on the height Z invariably appear on the same radius in the omnidirectional image, as shown in Fig. 2. The black circle is the horizontal position in the omnidirectional image.

2.2 Definition of ORP

The robot begins to move and takes an image sequence. Each omnidirectional image is transformed into 2D polar

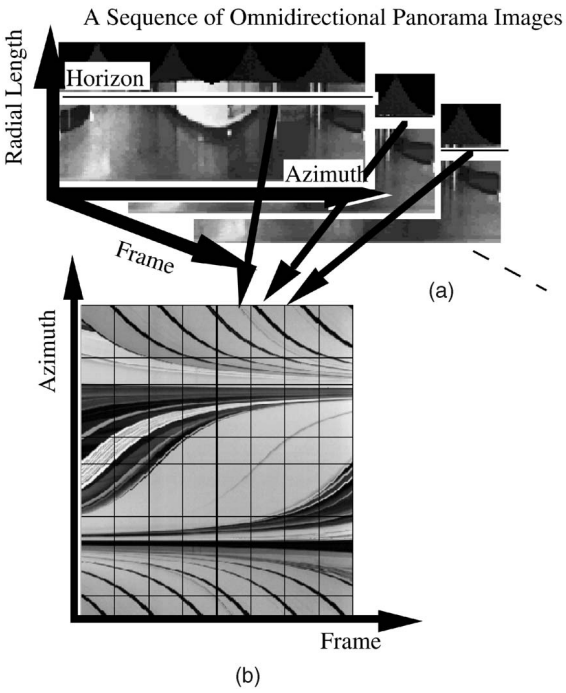


Fig. 3. Omnidirectional route panorama.

coordinates (r, θ) $(r = R(x^2 + y^2))$, called an omnidirectional panorama image. Points on the horizontal plane, which pass through the virtual center of the lens, appear as a straight line on the omnidirectional panorama image (drawn by the white line), as shown in Fig. 3a. This straight line is taken as a horizontal line. An ORP can be organized by arranging horizontal lines taken by the robot moving along the route, as shown in Fig. 3b. The ORP representation reduces the memory requirement for scene description. For instance, let's consider the case that the memory size of each horizontal line is 360 byte ($1byte/deg$). Since the scene is recorded every 10 cm and the robot moves 1 km, we need just 3.6 Mbyte for scene description.

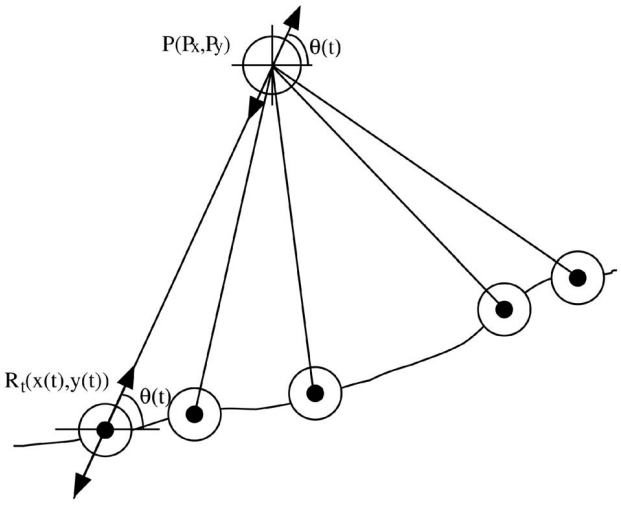


Fig. 4. Relation between memorized path and virtual viewpoint.

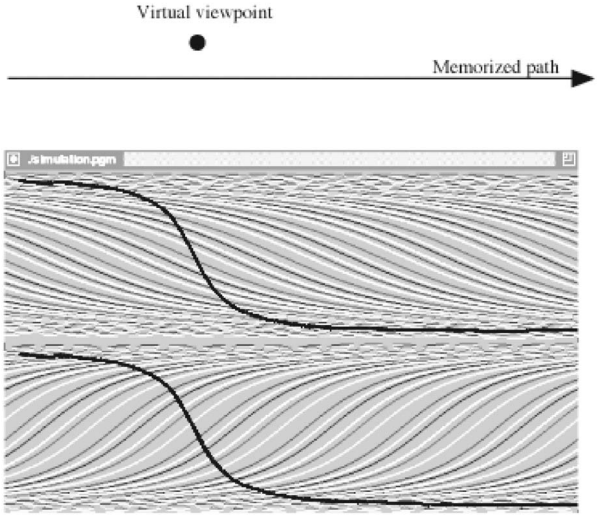


Fig. 5. Relation between iconic memory (memorized ORP) and virtual viewpoint.

3 Robot Localization by Synthesis

Under a precisely known robot motion, an arbitrary viewpoint image can be synthesized by stitching parts of consecutive images [20], [21]. Let us consider camera position $R_t(x(t), y(t))$, where the iconic-memory is organized, virtual view position $P(P_x, P_y)$, and orientation θ_p relative to the camera coordinates, as shown in Fig. 4. Since the direction of a vertical line at $P(P_x, P_y)$ is $\theta(t)$ and the robot orientation is θ_p, we can expand the vertical line toward a memorized path and calculate the intersection $R_t(x(t), y(t))$. Among $P(P_x, P_y)$, $\theta(t)$, θ_p, and $R_t(x(t), y(t))$, we have the following relation:

$$\theta(t) - \theta_p = \arctan \frac{P_y - y(t)}{P_x - x(t)}. \tag{5}$$

Then, we consider that the vertical image pattern along azimuth angle $\theta(t)$ at $R_t(x(t), y(t))$ is the same as that at $P(P_x, P_y)$. The panoramic image at the virtual viewpoint $P(P_x, P_y)$ is generated by stitching each vertical image pattern. In this case, the image pattern at the virtual viewpoint is estimated under given geometrical positions such as the memorized path, the virtual viewpoint, and the given iconic memory. From the principle of duality, if the image at a certain viewpoint, the iconic memory, and the memorized path are given, the position and orientation of the robot can be estimated. Our method is based on this concept. Fig. 5 shows an example of an ORP when the memorized path is straight and constantly sampled. In this case, the viewpoint is at the left side of the memorized path and the orientation of the robot is parallel to the path; then, the same image pattern lies on the tangent curve defined by (5). From these relations, we can estimate the robot position and orientation by searching tangent curves so that the image pattern is the same as the input. In practice, a position in the ORP where the image pattern is the same as the input is found by using an ACM.

Here, Kawasaki et al. [22] have proposed a camera position estimation method that matches the epipolar-plane image (EPI) obtained from the camera motion to the EPI generated from a CAD model by using DP matching. It is possible to find correspondence if the difference between the

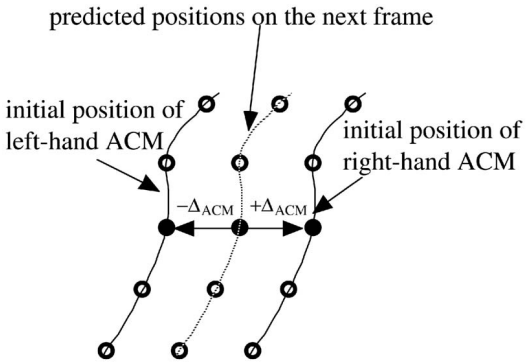


Fig. 6. Initial positions of ACMs.

two EPIs is small, but, in general, it is difficult to precisely model a complex scene. Furthermore, an environment in which a robot moves freely is not always stationary because of the appearance of unknown obstacles and moving objects.

The method we propose here does not need a 3D environmental model. By utilizing the global shape constraints of an active contour model, our proposed method can be applied to a dynamic environment where objects move around.

4 Active Contour Model (ACM) for Searching a Corresponding Image Pattern

An image pattern on an ORP that corresponds to an input is searched by an ACM. The advantage of using an ACM is that several different types of contour characteristics, such as image features, shape models, and smoothness of contours, can be defined using simple functions. Our proposed ACM actually consists of two ACMs. Corresponding control points placed on each of the ACMs are coupled and have a gravitational force. Then, the ACMs converge from both sides of a desired position in the ORP where the image pattern is the same as the input.

4.1 Outline

We assume that the rough initial position of the robot, the memorized path, and the iconic memory of the ORP during navigation are given and that the robot has an internal sensor for measuring its movement. Since the rough initial position of the robot and the iconic memory of the ORP are given, the robot can roughly predict the position of the image pattern on the iconic memory of the ORP from (5). However, the predicted position usually has observational and prediction errors; therefore, we define a certain size for the search region around the predicted position of the ACM. Actually, each ACM is shifted along a temporal axis from predicted positions of the ACMs at a certain margin Δ_{ACM}, as shown in Fig. 6. The desired positions are then estimated by converging both ACMs and we can estimate the robot's position and orientation by fitting the converged control points with the tangent function defined by (5). The tangent fitting is done using a least-square method. Once we can estimate the robot position, the next initial position of the robot can be predicted by adding the robot movement measured by the internal encoder to the estimated robot position. From our preliminary experiments, the margin Δ_{ACM} for shifting the ACMs was set as seven frames. From our robot system the sampling interval is $2\ cm/frame$, and corresponds to a $14\ cm$ error for the robot's position.

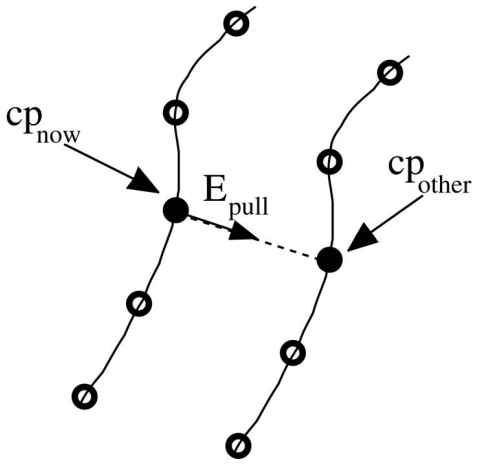


Fig. 7. Gravitational energy for converging coupled control points.

4.2 Definition of ACM

Our ACM is defined by the following equation:

$$E = \int_0^1 \big(E_{int}(v(s)) + E_{img}(v(s)) + E_{ext}(v(s))\big)ds, \quad (6)$$

where E, E_{int}, E_{img}, and E_{ext} represent total energy, internal energy representing smoothness and continuity, energy due to image features, and external energy for contour deformation, respectively. The internal energy is composed of a first-order term weighted by w_α and a second-order term weighted by w_β; here, $v(s)$ and $v_s(s)$ and $v_{ss}(s)$ are arc-length, curvature, and a change of curvature, respectively.

$$E_{int}(v(s)) = \Big(w_\alpha |v_s(s)|^2 + w_\beta |v_{ss}(s)|^2\Big)/2. \quad (7)$$

The image energy E_{img}, which attracts ACMs to edges with large intensity gradients, is defined by a first-order differential filter as follows:

$$E_{img}(v(s)) = w_{diff}\big|\nabla I(s) - \nabla I_{target}(\theta(s))\big|, \quad (8)$$

where $\nabla I_{target}(\theta(s))$ is a differential value along the horizontal line of the omnidirectional panorama image. $\nabla I(s)$ is a differential value on the memorized ORP. w_{diff} is a weighted coefficient. This energy tries to keep the ACMs on the edges.

The external energy, which pushes or pulls control points perpendicular to the curvature of the active contours, controls the direction of movement of the ACMs. We define this energy by

$$E_{ext}(v(s)) = E_{pull}(v(s)) + E_{const}(v(s)). \quad (9)$$

As shown in Fig. 7, E_{pull}, which is the energy for drawing coupled control points to each other, is defined by

$$\begin{aligned} \frac{\partial}{\partial\theta} E_{pull}(v(s)) &= w_{pull}\big(\theta_{cp_{other}}(s) - \theta_{cp_{now}}(s)\big) \\ \frac{\partial}{\partial t} E_{pull}(v(s)) &= w_{pull}\big(t_{cp_{other}}(s) - t_{cp_{now}}(s)\big), \end{aligned} \quad (10)$$

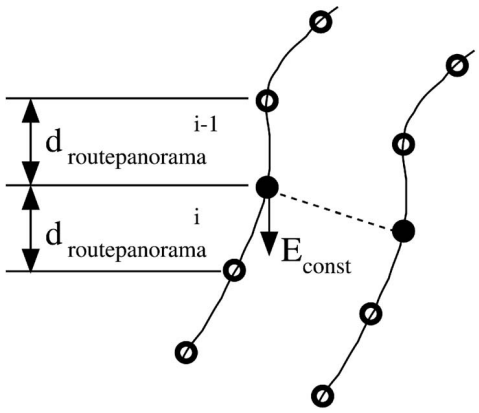


Fig. 8. Interval energy for keeping intervals between control points on ACMs.

where w_{pull} is a weighted coefficient. cp_{now} and cp_{other} are coupled control points. In the case of a straight memorized path, the interval of the azimuth angle θ between neighboring control points is constant. Therefore, we defined the energy $E_{const}(v(s))$ for a constant interval by spring models, as shown in Fig. 8,

$$E_{const}(v(s)) = w_{const} \begin{bmatrix} \left((d^i_{routepanorama} - d^i_{input} \right) \\ -\left(d^{i-1}_{routepanorama} - d^{i-1}_{input} \right) \end{bmatrix}, \qquad (11)$$

where w_{const} and d are a weighted coefficient and the interval of the azimuth angle between neighboring control points, respectively. The subscript on the right shoulder is the number of the control point.

5 Experiments

We undertook several experiments in our laboratory. In Sections 5.1, 5.2, 5.3, 5.4, and 5.5, we evaluated our iconic memory-based navigation method by using an offline experimental system. The robot moved straight for approximately $5\ m$ and the interval for the sampling images was $2\ cm$. Here, we define the orthogonal coordinates o-xy along the memorized path, in which the x axis is parallel to the direction the robot is moving. In this experiment, we recorded a horizontal line in the omnidirectional panorama image and the robot encoded data while it moved. The robot's movement was given by the operator. Start and stop positions were measured by hand and the measured position and odometer data were used for establishing ground truths. Recorded test images were matched with memorized ORP by our proposed method. In Section 5.6, we show the experimental results of autonomous navigation. We used a commercial mobile robot B12 (Real World Interface, Inc) in both experiments. Figs. 9a and 9b show the experimental environment and the mobile robot with the HyperOmni Vision omnidirectional image sensor.

(a) (b)

Fig. 9. Experimental environment and mobile robot with HyperOmni Vision.

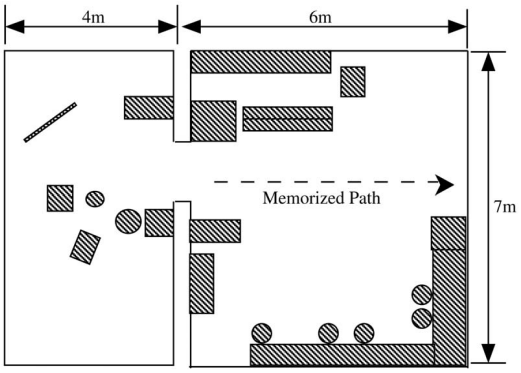


Fig. 10. Memorized path of the robot.

5.1 Accuracy of the Estimation of Location and Rotation

We evaluated the precision of the localization of the robot under several different moving conditions. Fig. 10 shows the experimental environment and memorized path. Fig. 11a is the case where the robot moves along the memorized path. Fig. 12a shows the result of the estimated robot position. The estimated moving trajectory of the robot is drawn as a black line. The dotted line is the correct trajectory of the robot. The average error and the standard deviation of the estimated position were $1.25\ cm$ and $0.64\ cm$, respectively. Fig. 11b is the case where the robot moves in a direction 30 degrees from the x-axis. Fig. 12b shows the results of the estimated position of the robot. The average error and the standard deviation of the position estimation were $1.98\ cm$ and $1.00\ cm$, respectively.

Next, seven different types of curved paths were prepared for evaluating the accuracy of position estimation and the robot moved along the curved paths. The average error and the standard deviation of the position estimation were $2.16\ cm$ and $1.46\ cm$, respectively. The average errors of the curved paths are similar to the average errors of the straight paths. Fig. 13 shows one of the results of an estimated position of the robot while the robot moved along a curved path.

It can be seen from these experiments that the method robustly estimates a robot's position along not only a straight path, but also along curved paths.

5.2 Evaluation of Sampling Interval along a Memorized Path

To ascertain how accuracy is related to the sampling interval along the memorized path, we evaluate the position estimation error relative to the image sampling interval. Fig. 14 shows the relationships among azimuth angular resolution, sampling interval, and position estimation error. The azimuth

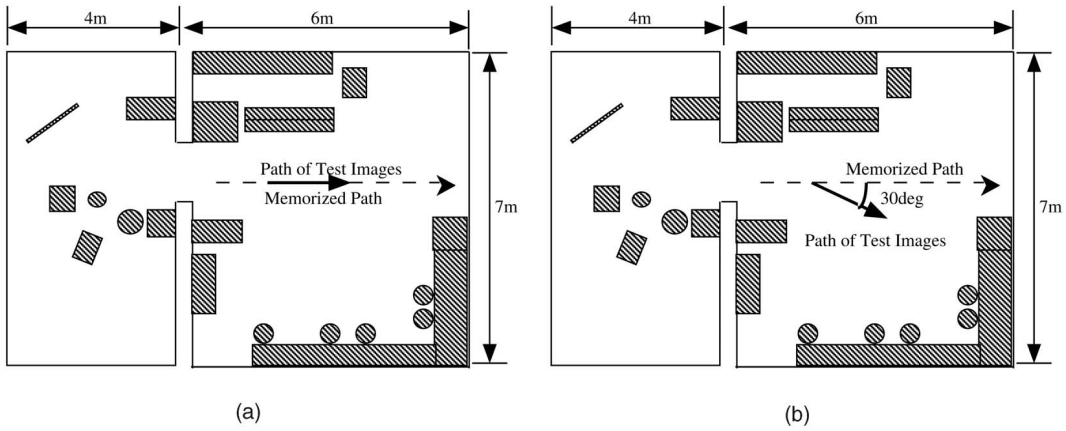


Fig. 11. Experimental layout for evaluating effectiveness. (a) The robot moved along the memorized path. (b) The robot moved in a 30-degree direction.

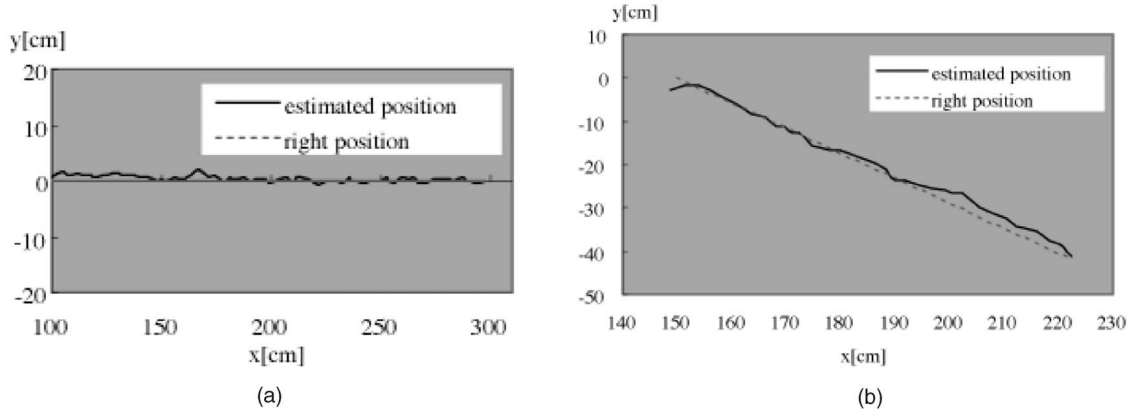


Fig. 12. Results of the estimation of the robot's position. (a) Along the memorized path. (b) Toward a 30-degree direction.

angular resolution is defined by the number of pixels on the horizon. The ordinate axis is the position estimation error. The abscissas axis shows the interval of the image sampling when the ORP is memorized. Figs. 14a and 14b show the position estimation errors when the robot moves along the memorized path and moves in a 30 degree direction, respectively. In the case where a sampling interval is dense, such as a few centimeters, the accuracy of position estimation is in proportion to the azimuth angular resolution. Here, the width of the experimental environment is about $6\ m \times 7\ m$. The average distance between the camera and the environment can be assumed as approximately $3\ m$ and the azimuth change while the robot moves in such an environment is about 0.2 degree.

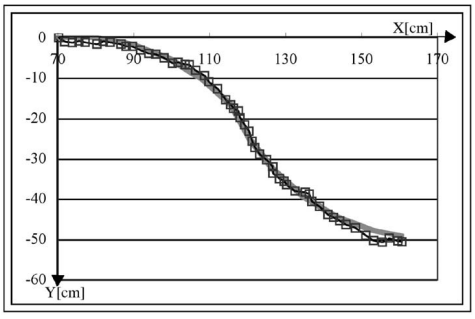


Fig. 13. The robot moves along a curved path.

This angular resolution is lower than the input image resolution (0.33 degree = 360 deg / 1,080 pixel). This gives the limitation for the accuracy of position estimation. In the case of a wide sampling interval, the temporal resolution of ORP is lower than angular resolution of an input image. As shown in the figures, this kind of imbalance increases the position estimation error.

5.3 Influence of Small Perturbations in the Ground Plane and in Calibration Errors of the Horizon Line

In the case of mobile robot navigation, the egomotion of the robot is usually caused by jiggling due to some unevenness in the ground plane. In this section, we evaluate the influence on our method of small perturbations in the ground plane.

HyperOmni Vision has the good characteristics of having a single center of projection. By using this characteristic, an input image can be transformed to an image from an arbitrarily slanting camera. Therefore, we can easily synthesize the slanting image from the input image and evaluate the accuracy of the localization of the robot.

Fig. 15 shows the relation between perturbation and position estimation error. The ordinate axis is the position estimation error. The abscissas axis shows the maximum size of a given camera slant. The size of the slant is randomly given. From this experiment, we find that the robot can

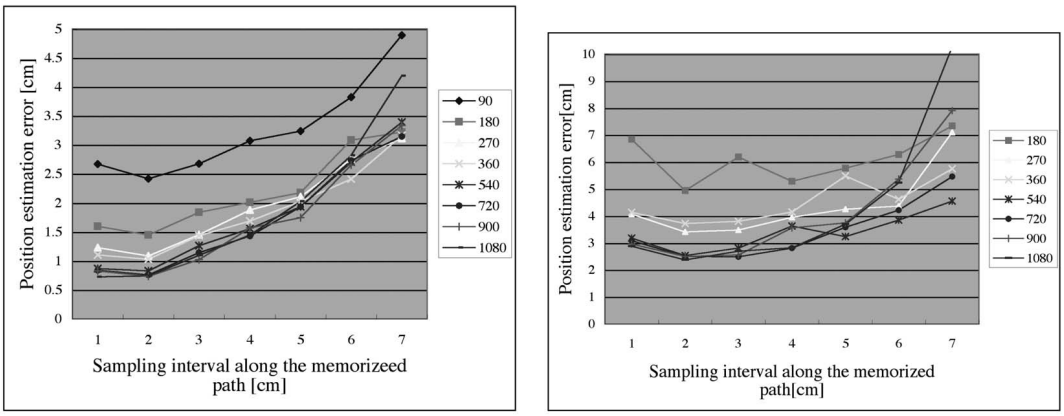


Fig. 14. Evaluation of sampling interval. (a) Along the memorized path. (b) Toward a 30-degree direction.

estimate its position within a few cm error when the slant of the camera is lower than 4 degree. In the case of an outdoor environment, large slants can sometimes occur, but such a large slant can be measured by using a gyroscope and a clinometer. Usually, the angular resolution of such sensors is better than 0.1 degrees; therefore, 4 degrees is sufficient for practical use in both indoor and outdoor environments.

Next, we changed the height of the robot and evaluated calibration errors of the horizon line. The camera position is actually $1\ cm$ higher than the memorized path; we prepared six different paths for this experiment. In total, the robot moved 12 times. The average error in the position estimation was $1.86\ cm$; therefore, we feel that small calibration errors do not present a serious problem for robot navigation.

5.4 Influence of Unknown Obstacles and the Effectiveness of Obstacle Masking

In the real world, objects are sometimes moved by an outsider, so a navigation system needs to be able to adapt to changes in parts of the scene. Fig. 16a is the result of the estimated robot position when an object in the environment is moved to a different position. Dark curves show the estimated trajectories of the robot's movements. Light gray curves are the results of applying automatic masking to an obstacle region. Details of automatic masking are described below. Fig. 16b shows the average and standard deviation of an estimated position of the robot.

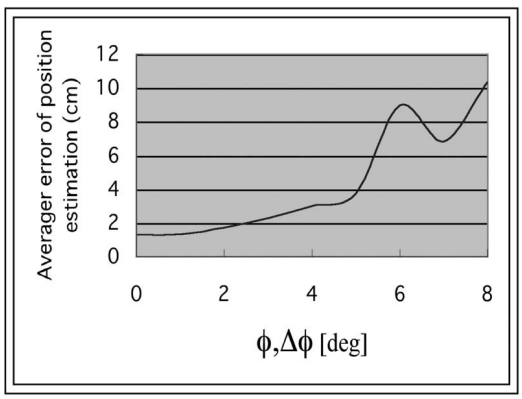


Fig. 15. Influence of small perturbations in the ground plane.

In this experiment, we changed the unknown obstacle position relative to the robot and changed the texture on the unknown obstacle. The ordinate axis is the position estimation error. The abscissa axis shows the complexity of the texture on the obstacle relative to a background environment. The complexity of the texture is defined by the number of edges on the unknown obstacle and the background. Basically, estimated error increases, but, especially, the position error increases when the percentage of the edges on the obstacle is bigger than 25 percent, as shown in Fig. 16b with a dark curve. However, it should be considered that the correlation value of the ACM at the obstacle region, where the scene changed, always remained low; therefore, the robot could precisely estimate its various positions by masking such a region. Indeed, as drawn by the light gray curve, we applied automatic masking to an obstacle region and estimated the robot's position. An algorithm of automatic masking is very simple. First, we calculate by subtraction between inputs and the ORP along an ACM. Next, we mask all of the control points in a period of azimuth that the subtraction is higher than a certain threshold and its pereiod is wider than a certain threshold. Figs. 16c and 16d show the position estimation error relative to obstacle direction. Ninety deg corresponds to the side view. Zero and $180\ deg$ correspond to the front and back. In the case that the robot moved along the memorized path, the shape of the converged ACM was almost a vertical line on the ORP. The position estimation error was independent of the obstacle direction, as shown in Fig. 16c. As a result, the effectiveness of masking is small. In the case that the robot moved in a 30 degree direction, the shape of the converged ACM was a tangent curved. Both sides of the tangent curve are more sensitive to position estimation as shown in Fig. 16d. Anyway, Figs. 16b and 16d show that the masking enabled robust and precise position estimation.

5.5 Influence of Illumination Change

Though the above experiments were done in daytime, the illumination conditions can be changed by sunshine coming through windows and by indoor lighting or a blackout.

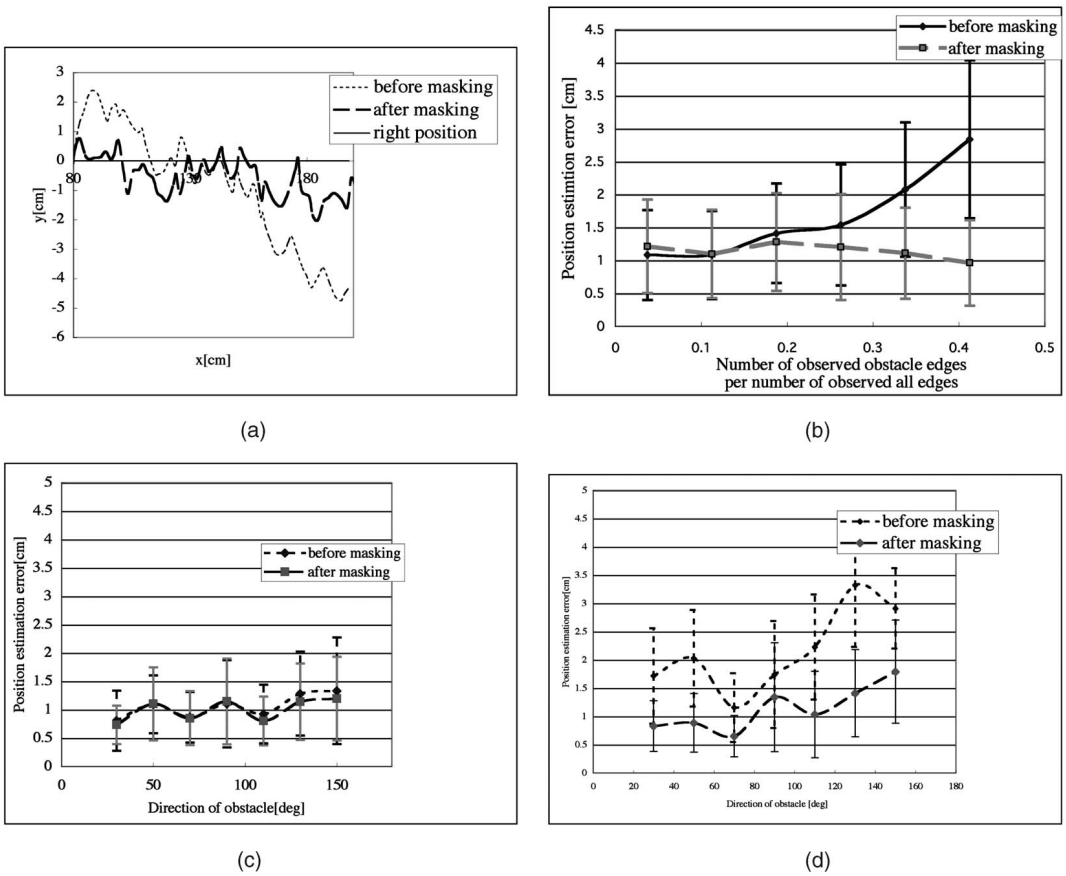


Fig. 16. Influence of unknown obstacle and effectiveness of obstacle masking. (a) Trajectory of estimated position of the robot. (b) Evaluation of influence of the complexity of the texture on the obstacle. (c) Position estimation error relative to observed azimuth of the obstacle—In the case that the moves along the memorized path. (d) Position estimation error relative to observed azimuth of the obstacle—In the case that the robot moves in a 30 degree direction.

Thus, we undertook an experiment at night. Fig. 17 shows ORPs in daytime and at night, respectively. From this figure, the center and the bottom regions in the ORP show different patterns which was mainly caused by sunshine from windows. Fig. 18 shows the results of the estimated position of the robot. The average error and the standard deviation of the estimated position were $4.67\,cm$ and $1.95\,cm$, respectively. Even if such a serious difference occurred, the robot could still estimate its position by the characteristics of an ACM such as a global shape constraint.

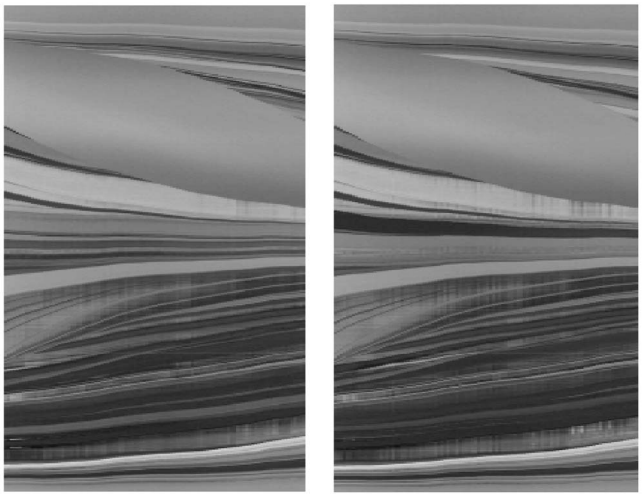


Fig. 17. Comparison of ORPs under different illumination conditions. (a) Daytime. (b) Night.

5.6 Results of Autonomous Navigation

Autonomous navigation was achieved in our experiments, and the robot was controlled by a standard proportional control method. Fig. 19 shows the results of autonomous navigation. In this case, the operator gave the initial position $(100\,cm, 0\,cm)$, the subgoal position $(200\,cm, -50\,cm)$ and the goal position $(300\,cm, 0\,cm)$, respectively; the robot then automatically navigated to the goal position. The black line is the estimated robot position and the red line shows the actual trajectory of the robot movement. Until the subgoal position, the robot can estimate precisely, so the trajectory of the robot is almost the same as the estimated position. After the robot reached the subgoal position, estimation errors increased a little. The location error at the goal position was approximately $4\,cm$. From Fig. 14, the accuracy limitation of the current robot system was a few centimeters. Further, as the ACM is basically an energy minimization technique, the

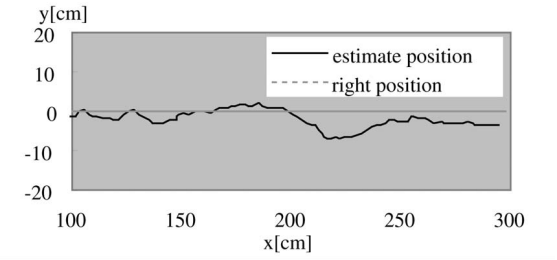


Fig. 18. Trajectory of the estimated robot position when the robot moved under different lighting conditions.

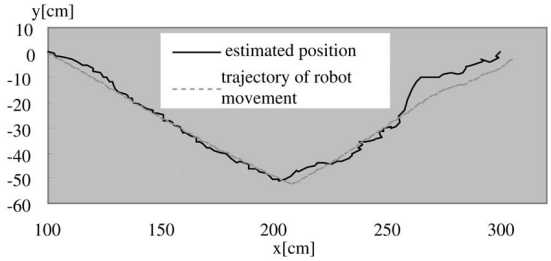


Fig. 19. Results of autonomous navigation.

converged position of the ACM depends on the initial position at each frame. If the image features of the control points are weak, it is difficult to converge these control points to the correct positions. Generally, the response of a standard proportional robot method is not so fast. Indeed, we consider that the location error at the goal position occurs for these reasons.

6 Conclusions

In this paper, we proposed an iconic memory-based navigation method. A robot's position was navigated by comparing the input with the memorized omnidirectional route panorama. An advantage of this method is that robot localization can be established without knowledge of the scene geometry. The omnidirectional route panorama saves on iconic-memory capacity and, thus, the method has a low memory requirement, even if the robot has to memorize a long distance path and moves over a long time period. In this paper, we demostrated our iconic memory-based approach by using an ACM. The basic iconic memory-based idea does not limit the model for fitting a tangent curve; we can use other deformable models. The processing time of the current system was 200 $ms/frame$; while this is fast enough for indoor navigation, we are presently trying to optimize the navigation algorithm to make the system faster.We evaluated the precision of the localization of the robot under several different moving conditions and environments. From these experiments, we found that the method could robustly estimate the robot position at an accuracy sufficient for robot navigation. We are planning to apply our method for long-distance autonomous navigation in both indoor and outdoor environments and to develop a practical robot navigation system. To apply our method in such environments, we will use our masking technique to find and ignore occluded regions.

Acknowledgments

Part of this work was supported by a Grant-in-Aid for Scientific Research from the Ministry of Education, Science and Culture, Japanese Government.

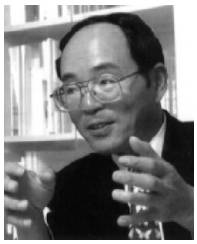
References

[1] Z.L. Cao, S.J. Oh, and E.L. Hall, "Dynamic Omnidirectional Vision for Mobile Robots," *J. Robotic Systems,* vol. 3, no. 1, pp. 5-17, 1986.
[2] P. Greguss, "The Tube Peeper: A New Concept in Endoscopy," *Optics and Laser Technology,* pp. 41-45, 1985.
[3] Y. Yagi and S. Kawato, "Panorama Scene Analysis with Conic Projection," *Proc. IEEE/RSJ Int'l Workshop Intelligent Robots and Systems,* pp. 181-187, 1990.
[4] J. Hong, X. Tan, B. Pinette, R. Weiss, and E.M. Riseman, "Image-Based Homing," *Proc. IEEE Int'l Conf. Robotics and Automation,* pp. 620-625, 1991.
[5] K. Yamazawa, Y. Yagi, and M. Yachida, "Omnidirectional Imaging with Hyperboloidal Projection," *Proc. IEEE/RSJ Int'l Conf. Intelligent Robots and Systems,* no. 2, pp. 1029-1034, 1993.
[6] Y. Yagi, S. Kawato, and S. Tsuji, "Real-Time Omnidirectional Image Sensor (COPIS) for Vision-Guided Navigation," *IEEE Trans. Robotics and Automation,* vol. 10, no. 1, pp. 11-22, 1994.
[7] Y. Yagi, H. Okumura, and M. Yachida, "Multiple Visual Sensing System for Mobile Robot," *Proc. IEEE Int'l Conf. Robotics and Automation,* vol. 2, pp. 1679-1684, 1994.
[8] V. Peri and S.K. Nayar, "Omnidirectional Video System," *Proc. US-Japan Graduate Student Forum in Robotics,* pp. 28-31, 1996.
[9] C. Pegard and E.M. Mouaddib, "Mobile Robot Using a Panoramic View," *Proc. IEEE Int'l Conf. Robotics and Automation,* vol. 1, pp. 89-94, 1996.
[10] J.S. Chahl and M.V. Srinivasan, "Reflective Surfaces for Panoramic Imaging," *Applied Optics,* vol. 36, no. 31, pp. 8275-8285, 1997.
[11] S. Baker and S.K. Nayer, "A Theory of Single-Viewpoint Catadioptric Image Forma-tion," *Int'l J. Computer Vision,* vol. 35, no. 2, pp. 1-22, 1999.
[12] Y. Yagi, "Omnidirectional Sensing and Its Applications," *IEICE Trans. Information and Systems,* vol. E82-D, no. 3, pp. 568-579, 1999.
[13] J.Y. Zheng and S. Tsuji, "Panoramic Representation for Route Recognition by a Mobile Robot," *Int'l J. Computer Vision,* vol. 9, no. 1, pp. 55-76, 1992.
[14] Y. Matsumoto, M. Inaba, and H. Inoue, "Memory-Based Navigation Using Omni-View Sequence," *Proc. Int'l Conf. Field and Service Robotics,* pp. 184-191, 1997.
[15] H. Ishiguro and S. Tsuji, "Image-Based Memory of Environment," *Proc. Int'l Conf. Intelligent Robots and Systems,* pp. 634-639, 1996.
[16] N. Aihara, H. Iwasa, N. Yokoya, and H. Takemura, "Memory-Based Self-Localization Using Omnidirectional Images," *Proc. Int'l Conf. Pattern Recognition,* pp. 1799-1803, 1998.
[17] M. Jogan and A. Leonardis, "Robust Localization Using Panoramic View-Based Recognition," *Proc. IAPR Int'l Conf. Pattern Recognition,* vol. IV, pp. 136-139, 2000.
[18] I. Ulrich and I. Nourbakhsh, "Appearance-Based Place Recognition for Topological Location," *Proc. 2000 IEEE Int'l Conf. Robotics and Automation,* pp. 1023-1029, 2000.
[19] Y. Yagi, S. Fujimura, and M. Yachida, "Route Representation for Mobile Robot Navigation by Omnidirectional Route Panorama Fourier Transformation," *Proc. IEEE Int'l Conf. Robotics and Automation,* pp. 1250-1255, 1998.
[20] K. Yamaguchi, K. Yamazawa, H. Takemura, and N. Yokoya, "Real-Time Generation and Presentation of View-Dependent Binocular Stereo Images Using a Sequence Of Omnidirectional Images," *Proc. 15th IAPR Int'l Conf. Pattern Recognition,* vol. IV, pp. 589-593, Sept. 2000.
[21] T. Takahashi, H. Kawasaki, K. Ikeuchi, and M. Sakauchi, "Expanding Possible View Point of Virtual Environment Using Panoramic Images," *Proc. Int'l Conf. Pattern Recognition,* pp. 468-471, 2000.
[22] H. Kawasaki, K. Ikeuchi, and M. Sakauchi, "EPI Analysis of Omni-Camera Image," *Proc. Int'l Conf. Pattern Recognition,* pp. 379-383, 2000.

Yasushi Yagi received the BE and ME degrees in control engineering in 1983 and 1985, respectively, and the PhD degree in 1991, from Osaka University. He is a professor at the Institute of Scientific and Industrial Research, Osaka University, Ibaraki, Japan. In 1985, he joined the Product Development Laboratory, Mitsubishi Electric Corporation, where he was working on robotics and inspections. In 1990, he was a research associate of information and computer science, Faculty of Engineering Science, Osaka University. From 1993 to 1996, he was a lecturer of systems engineering, Faculty of Engineering Science, Osaka University. From 1996 to 2003, he was an associate professor of systems and human science, Graduate School of Engineering Science, Osaka University. From June 1995 to March 1996, he was an academic visitor of Department of Engineering Science, University of Oxford. Computer vision and robotics are his research subjects. He is a member of the IEEE and the IEEE Computer Society.

Kousuke Imai received the BE and ME degrees in system engineering in 2000 and 2002, respectively. In 2002, he joined Toshiba. His research interests include computer vision and robot navigation.

Kentaro Tsuji received the BE degree in system engineering in 2003. He is currently a master's course student in the Graduate School of Engineering Science at Osaka University. His research interests include computer vision and robot navigation.



Masahiko Yachida received the BE and MSc degrees in electrical engineering and the PhD degree in control engineering, all from Osaka University, Osaka, Japan, in 1969, 1971, and 1976, respectively. He joined the Department of Control Engineering, Faculty of Engineering Science, Osaka University in 1971 as a research associate and became an associate professor at the same department. He then moved to the Department of Information and Computer Science of the same university as a professor in 1990. In 1993, he became a professor at the Department of Systems Engineering of the same university. Since 1997, he has been a professor of systems and human science, Graduate School of Engineering Science, Osaka University. He was a research student of the research laboratory Riso, Danish Atomic Energy Commission in 1967, joined the Intelligent Industrial Robot Project at the Electrotechnical Laboratory of the Japanese government from 1969 to 1970, was a research associate at the Coordinated Science Laboratory, University of Illinois, Urbana, from 1973 to 1974, a research fellow at the Fachbereich Informatik, Hamburg University from 1981 to 1982, and a CDC professor in the Department of Computer Science, University of Minnesota in 1983. He is an author of *Robot Vision* (Shoukoudou, received Ohkawa Publishing Prize), coauthor of *Pattern Information Processing* (Ohm-sha), and an editor of *Computer Vision* (Maruzen). He was a chairman of the Technical Committee on Computer Vision, Information Processing Society of Japan, and a chairman of the Technical Committee on Pattern Recognition and Media Understanding, Institute of Electronics, Information, and Communication Engineers, Japan. His interests are in the fields of computer vision, image processing, mobile robots, and artificial intelligence.