Proceedings of the 2003 IEEE
International Conference on Robotics & Automation
Taipei, Taiwan, September 14-19, 2003

Iconic Memory-based Omnidirectional Route Panorama Navigation

Yasushi Yagi Kousuke Imai Masahiko Yachida
Department of Systems and Human Science
Graduate School of Engineering Science, Osaka University
1-3 Machikaneyama, Toyonaka, 560-8531 Japan
TEL: +81-6-6850-6361 FAX: +81-6-6850-6341
email: y-yagi@sys.es.osaka-u.ac.jp

Abstract

A route navigation method for a mobile robot with an omni-directional image sensor is described. The route is memorized from a series of consecutive omni-directional images at the horizon when the robot moves to the goal. While the robot is navigating to the goal point, the 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 active contour models.

1 Introduction

A real-time omnidirectional camera that can acquire an omnidirectional (360 degrees) field of view at a video rate could be applied in a variety of fields such as autonomous navigation. Several researchers have investigated geometrical-based and iconic memory-based navigation methods using omnidirectional image sensors [1].

Iconic memory-based navigation is a common approach for visual navigation. The basic operation is a comparison between the present sensory input and previous memorized images. It is easy to relate the robot's action and sensory data without the geometrical model. Zheng's robot memorized the side of scene of a route from a panoramic view while it moved along the route [2]. Matsumoto et al.'s robot memorized the whole front view image at reference points along the route for visual navigation [3]. The correspondence between a present input image and previously memorized images was established by using dynamic programing (DP) matching and correlation methods, respectively. However, these methods need a large amount of memory for memorizing the route. Therefore, to reduce the data to be memorized, Ishiguro has proposed a compact representation by expanding it into a Fourier series [4]. Each input image is memorized by the coefficients of the low frequency components. KL transformation is another approach to compress a memorized data [5, 6]. Ulrich and Nourbakhsh memorized by a simple color histogram and navigated the mobile robot in both indoor and outdoor environments [7]. We have represented the relation between environment and robot behavior by performing 2-D Fourier transformations on an omnidirectional route panorama (ORP) that can be acquired by arranging points on the horizontal plane, and which passes through the virtual center of the lens, taken by the robot moving along the route [8]. The ORP in a certain number of past frames, which is a standard unit of spatio-temporal representation, is transformed to a 2D Fourier power spectrum. The route is 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 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.

However, most of these previous iconic memory-based approaches basically select the image that corresponds to an input image from discretely memorized images. This means the precision of position and orientation of the mobile robot depends on the spatial sampling density of a moving space. Especially, it is impossible to estimate the position and orientation of the robot when the robot deviates widely from a memorized path or the interval between memorized positions are too wide.

In this paper, we propose a new iconic memory-based navigation method that synthesizes a corresponding image pattern from an ORP. As the computational cost for synthesizing the corresponded image is high, our proposed method searches a corresponding image pattern in not only spatial but also spatio-temporal space by using active contour models (ACMs), and then estimates the position and orientation of the robot from the converged shape of the ACMs.

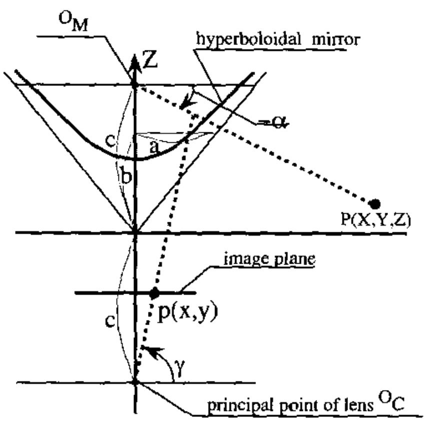


Figure 1. Hyperboloidal projection has characteristics of a single center of projection

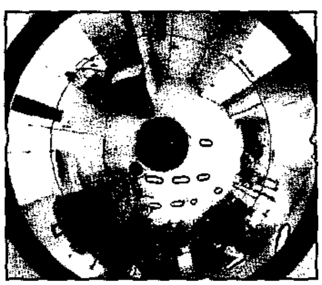


Figure 2. Points on the horizontal plane, which pass through the virtual center of the lens, appear as a circle on an omnidirectional image

2 Omnidirectional Route Panorama

A robot moving along a route can observe objects in many directions. A sensor is 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 navigation [9, 11, 10]. The method proposed here uses the image sequence of a horizontal part of the omnidirectional image, called the omnidirectional route panorama (ORP), while the robot continuously scans the view along the route.

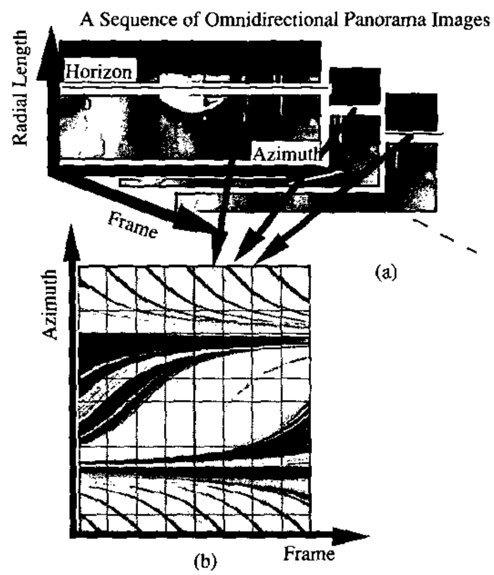


Figure 3. Omnidirectional route panorama

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 Figure 1. Using a world coordinates system (X, Y, Z) the hyperboloidal surface can be represented as:

$$\frac{X^2+Y^2}{a^2} - \frac{Z^2}{b^2} = -1 \qquad (1)$$
$$c = \sqrt{a^2+b^2}$$

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 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 the hyperboloidal mirror and projected on the 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 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, 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. Also, it can also be 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 threfore, 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 to 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 Figure 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 2-D polar 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 Figure 3 (a). 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 Figure 3 (b). The representation of ORP can save memory 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.

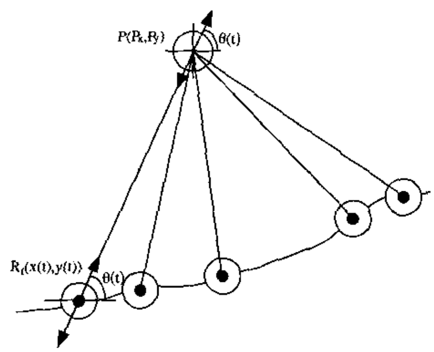


Figure 4. Relation between memorized path and virtual viewpoint

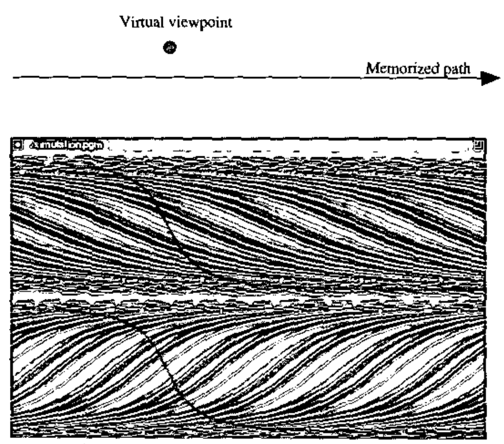


Figure 5. Relation between iconic memory (memorized ORP) and virtual viewpoint

3 Robot Localization by Synthesis

Under a precisely known robot motion, an arbitrary viewpoint image can be synthesized by stitching parts of consecutive images [12, 13]. Let us consider camera coordinates $\sum_{R_0}$, camera position $R_t(x(t), y(t))$ where iconic-memory is organized, virtual view position $P(P_x, P_y)$ and orientation θ_p relative to the camera coordinates as shown in Figure 4. Since the direction of a vertical line at $P(P_x, P_y)$ is $\theta(t)$ and the robot orientation 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 a 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 $P(P_x, P_y)$. Then 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 and the virtual viewpoint and the given iconic memory. From the principle of duality, if the image at the certain viewpoint and iconic memory and memorized path are given, the position and orientation of the robot can be estimated. Our method is based on this concept. Figure 5 shows an example of 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 that image pattern is same as the input. Actually, a position in ORP where the image pattern is the same as the input is found by using ACM.

Here, Kawasaki et al. [14] have proposed a camera position estimation method that matches the EPI obtained from the camera motion to the epipolar-plane image (EPI) generated from a CAD model by using DP matching. It is possible to find correspondence if the difference between the two EPIs is small, but in general, it is difficult to precisely model a complex scene. Furthermore, the environment is not always stationary because of the appearance of unknown obstacles and moving objects. As well, the robot moves freely in the environment.

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

4 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 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 during navigation, the memorized path and the iconic memory of the ORP are given, and that the robot has an internal sensor for measuring its movement. If 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). 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}. The desired positions are then estimated by converging both ACMs, and we can estimate the robot's position and orientation by fitting converged control points with the tangent function defined by (5) . The tangent fitting is done by 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. The certain margin Δ_{ACM} for shifting the ACMs was set as 7 frames from preliminary experiments. From our robot system the sampling interval is 2 $cm/frame$, and corresponded to a 14 cm error for the robot's position.

4.2 Definition of ACM

Our ACM is defined by a following equation.

$$E = \int_0^1 \left(E_{int}\left(v(s)\right) + E_{img}\left(v(s)\right) + E_{ext}\left(v(s)\right)\right) ds \tag{6}$$

where E, E_{int}, E_{img} and E_{ext} represent the total energy, the internal energy representing smoothness and continuity, the energy due to image features, and the external energy for contour deformation, respectively. The internal energy is composed of a 1st-order term weighted by w_α and a 2nd-order term weighted by w_β;

$$E_{int}\left(v(s)\right) = \left(w_\alpha \left|v_s(s)\right|^2 + w_\beta \left|v_{ss}(s)\right|^2\right)/2 \tag{7}$$

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

$$E_{img}\left(v\left(s\right)\right) = w_{diff}\left|\nabla I\left(s\right) - \nabla I_{target}\left(\theta\left(s\right)\right)\right| \tag{8}$$

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

$$E_{ext}\left(v\left(s\right)\right) = E_{pull}\left(v\left(s\right)\right) + E_{const}\left(v\left(s\right)\right) \tag{9}$$

E_{pull} which is an energy for drawing coupled control points to each other is defined by

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

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 $E_{const}\left(v\left(s\right)\right)$ for a constant interval by spring models as show in Figure 6.

$$E_{const}\left(v\left(s\right)\right) = w_{const}\left[\begin{array}{l} \left(\left(d^{i}_{routepanorama} - d^{i}_{input}\right)\right. \\ \left.- \left(d^{i-1}_{routepanorama} - d^{i-1}_{input}\right)\right) \end{array}\right] \tag{11}$$

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

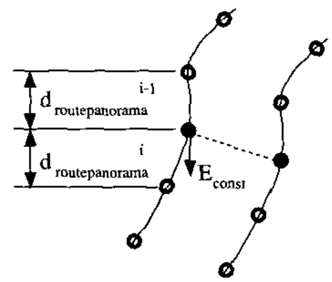
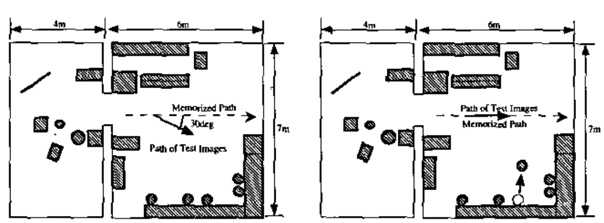


Figure 6. Interval energy for keeping intervals between control points on ACMs

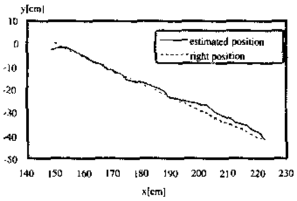
(a) (b)

Figure 7. Experimental environment and mobile robot with HyperOmni Vision

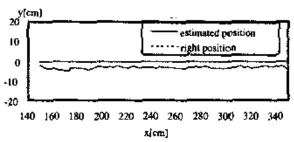


(a) (b)

Figure 8. Experimental layout for evaluating effectivity (a) The robot moved toward a 30-degree direction (b) Object in the room moved closer



(a)The robot moved toward a 30-degree direction



(b) Object in the room moved closer

Figure 9. Results of the estimation of the robot's position

5 Experiments

Several experiments were done in our laboratory. In Section 5.1, we evaluated our iconic memory-based navigation method by using an off-line experimental system. We show two results in this paper. In Section 5.2, we show the experimental result of autonomous navigation. We used the commercial mobile robot B12 (Real World Interface, Inc) in both experiments. Figure 7 (a) and (b) show the experimental environment and the mobile robot with the omnidirectional image sensor HyperOmni Vision.

5.1 Results of Evaluation of Effectiveness

The robot moved straight 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 moving direction of the robot. In this experiment, we recorded a horizontal line in the omnidirectional panorama image and the robot encoder data while the robot moved. The robot movement was given by the operator. Recorded test images were matched with memorized ORP by our proposed method. Then, we evaluated the precision of the localization of the robot under several different moving conditions. Figure 8 (a) is the case where the robot moves toward a direction 30 degrees from the x-axis. Figure 9(b) shows the result of the estimated position of the robot. The estimated moving trajectory of the robot is drawn by a black line. The dotted line is the correct trajectory of the robot. The average error and standard deviation of the position estimation were 1.98 cm and 1.00 cm, respectively.

In the real world, objects are sometimes moved by someone. That means that the navigation system should adapt to changes in parts of the scene. Figure 9 (b) is the result of the estimated robot position when an object in the environment was moved to a different position as shown in Figure 8(b). In this case the ratio of the scene change was approximately 10 %. The average error and standard deviation of the estimated position were 3.46 cm and 0.56 cm, respectively. The estimation error in the second experiment was bigger than the other. However, we could consider that the correlation at the region, where the scene changed, was continuously low, therefore the robot can precisely estimate its positions by masking such a region.

5.2 A Result of Autonomous Navigation

The autonomous navigation was achieved. The robot was controlled by a standard proportional control method. Figure 10 shows the results of autonomous navigation. In this case, the operator gave the initial position (100 cm, 0 cm), the sub goal position (200 cm, -50 cm) and the goal position (300 cm, 0 cm), respectively. Then the robot automatically navigated to the goal position. The black line was the estimated robot position and the red line was the trajectory of the robot movement. Until the sub goal 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 sub goal position, estimation errors increased a little, but the robot almost followed the correct path. In this case, the location error at the goal position is approximately 4 cm.

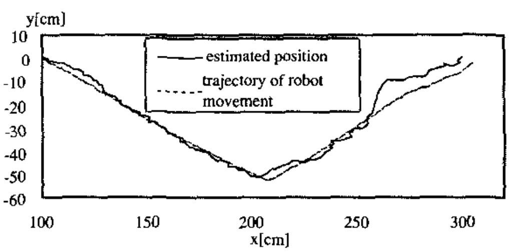


Figure 10. Results of autonomous navigation

6 Conclusions

In this paper, we proposed the iconic memory-based navigation. A robot's position was navigated by comparing the input with memorized omnidirectional route panorama. The omnidirectional route panorama can save iconic-memory, and thus the method does not have great memory requirement even if the robot memorizes a long distance and over a long time period. The processing time of the current system was 200 $ms/frame$. While this is fast enough for indoor navigation, however, we are presently trying to optimize the navigation algorithm to make a faster system. In this paper, we did not mask the region where the scene changed. As we consider that masking such a region makes for a the robust system, we will also focus on this problem in the future.

Acknowledgement

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] Yagi, Y., "Omnidirectional Sensing and Its Applications," *IEICE Trans. Information and Systems*, Vol.E82-D, No.3, pp.568-579, 1999.

[2] Zheng, J. Y. and Tsuji, S., "Panoramic Representation for Route Recognition by a Mobile Robot," *Int. J. Computer Vision,* 9, 1, pp.55-76, 1992.

[3] Matsumoto, Y., Inaba, M. and Inoue, H., "Memory-based Navigation using Omni-View Sequence," *Proc. of Int. Conf. Field and Service Robotics,* pp.184-191, 1997.

[4] Ishiguro, H. and Tsuji, S., "Image-Based Memory of Environment," *Proc. of Int. Conf. Intelligent Robots and Systems,* pp.634-639, 1996.

[5] Aihara, N., Iwasa, H., Yokoya, N. and Takemura, H., "Memory-based Self-localization Using Omnidirectional Images," *Proc. of Int. Conf. Pattern Recognition,* pp.1799-1803, 1998.

[6] Jogan, M., Leonardis, A., "Robust Localization Using Panoramic View-based Recognition," *Proc. IAPR Int. Conf. on Pattern Recognition,* Vol.IV, pp.136-139, 2000.

[7] Ulrich, I., and Nourbakhsh, I., "Appearance-Based Place Recognition for Topological Location," *Proc. the 2000, IEEE Int. Conf. Robotics and Automation,* pp.1023-1029, 2000.

[8] Yagi, Y., Fujimura, S. and Yachida, M., "Route Representation for Mobile Robot Navigation by Omnidirectional Route Panorama Fourier Transformation," *Proc. IEEE Int. Conf. Robotics and Automation*, pp.1250-1255, 1998.

[9] 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.

[10] 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.

[11] Yagi, Y., Okumura H. and Yachida, M., "Multiple Visual Sensing System for Mobile Robot," *Proc. IEEE Int. Conf. on Robotics and Automation*,vol.2, pp.1679-1684, 1994.

[12] Yamaguchi, K. , Yamazawa, K. , Takemura, H. and Yokoya, N. , "Real-time generation and presentation of view-dependent binocular stereo images using a sequence of omnidirectional images," *Proc. 15th IAPR Int. Conf. on Pattern Recognition* , Vol. IV, pp. 589-593, Sep. 2000.

[13] Takahashi, T., Kawasaki, H., Ikeuchi, K. and Sakauchi, M., "Expanding Possible View Point of Virtual Environment Using Panoramic Images," Proc. Int. Conf. Pattern Recognition, pp.468-471, 2000.

[14] Kawasaki,H., Ikeuchi, K. and Sakauchi, M., "EPI Analysis of Omni-Camera Image, Proc. Int. Conf. Pattern Recognition," pp.379-383, 2000.