Proceedings of the 2001 IEEE
International Conference on Robotics & Automation
Seoul, Korea · May 21-26, 2001

Vision Based Navigation for Mobile Robots in Indoor Environment by Teaching and Playing-back Scheme

Lixin TANG, **Shin'ichi YUTA**

Intelligent Robot Laboratory
University of Tsukuba
Tsukuba, 305-8573 JAPAN
{tang, yuta}@roboken.esys.tsukuba.ac.jp

Abstract

This paper presents a vision based autonomous navigation system for mobile robots in an indoor environment by teaching and playing-back scheme. The system uses an omnidirectional image sensor to perceive the environment, and extracts vertical edges as feature lines. The system memorizes a sequence of environmental images and robot's poses during teaching stage. In the course of playback navigation, the system calculates the robot's position difference from the memorized and currently taken images, and then decides the trajectory to track the taught route. The detail algorithm and the effectiveness of this method with experiments are shown in this paper.

1 Introduction

While a mobile robot navigates along a designated route autonomously, it must recognize its position relative to the environment so as to follow the given route. Many approaches use environmental maps to give the path, and the robot estimates its position by matching sensor data with a map [1][2]. So, a detailed environmental map must be prepared before navigation. Such as, Ishiguro et al. builded T-Net for robot navigation[3].

In this paper, we propose a vision based navigation method in an indoor environment by teaching and playing-back. With this method, an operator controls a robot manually to teach a path in the first step, and then the robot navigates itself autonomously to track the trajectory given at teaching stage. While navigating autonomously, the robot adjusts its motion based on the position difference information gotten by matching a current image with one taken during teaching.

Some researchers have done similar studies. T. Ohno studied a method in which a robot position is gotten by a pre-learned relationship between an azimuth angle and position differences in two images[4]. In his method, a certain number of feature lines must be extracted from an image to acquire the differences correctly. However, since a usual camera with a view angle of 50 degrees was used in his system, it could not have enough feature lines. So, the method was not robust enough for a wide range of environment.

An omnidirectional image sensor, consisting of a special mirror and a CCD camera, can acquire a 360-degree view around a robot, so that it is very suitable for autonomous navigation. Now, many reserachers have used it for their navigation systems. For instance, Matsumoto proposed a visual navigation method using a memorized omnidirectional view sequence[5]. In his method, a robot navigates by matching a current view with a memorized view sequence using the templates consisting of the front and rear views. So the relative orientation and the lateral displacement are estiminated to realize a play-back navigation.

In this study, we use omnidirectional image information, and use natural vertical edges of the environment, such as doors and corners, as feature lines. We also use the recorded position by odometry at teaching stage. It makes the playback navigation very robust. The accumlation error of odmetory does not affect in this scheme.

In next section, we explain the navigation method in detail. Section 3 proposes the algorithm of position difference calculation. Section 4 presents the method for given trajectory tracking. The effectiveness of this navigation method is examined with experimental results in section 5, and the paper ends with a conclusion in section 6.

2 Vision based autonomous navigation by teaching and playing-back scheme

The proposed method of vision based navigation includes two stages: teaching route by operator and navigation by playing-back.

Stage 1. Teaching route by operator

A human controls a robot via radio or other aid, so that the robot moves along a designated route in an environment. While the robot is moving, it records a sequence of omnidirectional images of the environment observed from the robot and its positions estimated by odometry at every time interval Δt.

As the preparation for playback navigation, the system processes the image in real-time at each time interval. The process includes transforming the omnidirectional image into a panoramic image, extracting vertical edges as feature lines, searching the corresponding feature lines by matching this image with the one which was taken at one time interval before, and calculating the X-Y coordinates of these feature lines in the present robot frame.

The images taken at this stage are called reference image in our system.

The robot position estimated by odometory with the feature lines in the reference image and their X-Y coordinate positions, is memorized at each interval while teaching.

Stage 2. Navigation by playing-back

The robot starts moving along a taught route by following the memorized positions at the same speed with teaching in the environment based on odometory, and it takes an environmental image (called current image) at the same time interval Δt.

In each time interval, the system processes the current image to extract feature lines and to match them with ones of the reference image. And using the calculated positions of feature lines on X-Y coordinate, the robot can calculate the difference between the positions where the reference image and the current one were taken. In this step, many extracted feature lines and the least square calculation are used to estimate the accurate position difference. Then, the robot generates the trajectory to move itself to next recorded position and direction.

So, in this method, the recorded position and direction information is only used to move to the next point, and the accumulating error of odometory does not affect the robot trajectory.

3 Algorithm of position difference calculation

The procedure of the image processing at teaching and playback stages, and position difference calculation are shown in figure 1. In this section, we explain the algorithm of each step in figure 1 with an example of image processing results.

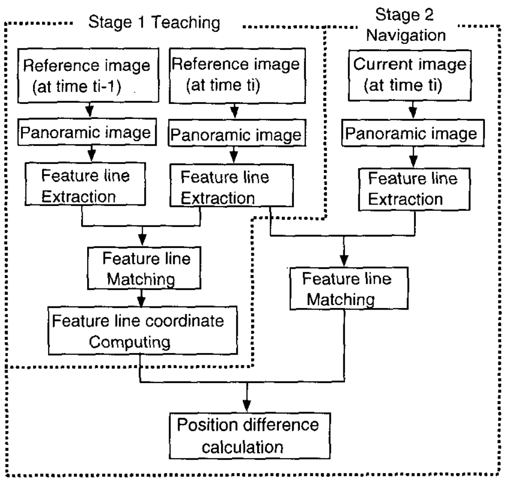

Figure 1: Image processing for position difference calculation

3.1 Feature line extraction

In our method, only vertical edges in an environment are used as feature lines. To extract them easily, we transform an omnidirectional image, as shown in figure 2(a), into a panoramic image as shown in figure 2(b). In order to reduce image processing time, only a ring area surrounded by two dotted line circles in figure 2(a), where vertical edges are centralized, is transformed. Then, the horizontal component of the differentials is acquired in the region. And the vertical edges where the differential values exceed a threshold are extracted as feature lines, as shown in figure 2(c).

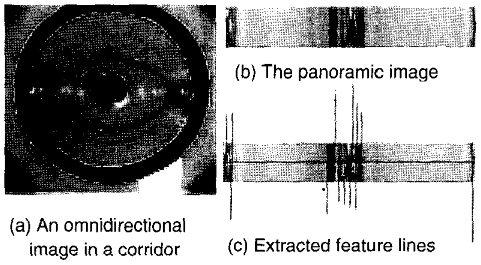
(a) An omnidirectional image in a corridor
(b) The panoramic image
(c) Extracted feature lines

Figure 2: Omnidirectional image processing

3.2 Feature line matching

Since the positions where images were acquired are not distant to each other, the feature lines, which can be matched between two images, should have the following features [4]: (1) the signs of differential values are same. (2) the horizontal differential values are similar. (3) the intervals between adjacent feature lines are similar in two images. (4) some feature lines have no any correspondents, but a feature line in an image can have only one correspondent in another image at most. (5) the order of correspondence does never change.

Based on above conditions, we define an evaluation function M to measure the corresponding relation of feature lines between two images.

$$M = \alpha \sum_{i=1}^{N_1} D_i + \beta \sum_{i=1}^{N_1} I_i + \gamma \sum_{j=0}^{N_2} P_j \tag{1}$$

where,

D_i : the absolute difference of horizontal differential values between feature line i in one image and its corresponding candidate in another image.

I_i : the absolute difference between two position intervals. One is the interval between feature line i and its adjacent feature line in an image. Another is the interval between the corresponding candidate of feature line i and the adjacent one of the candidate in another image.

P_j: the penalty value if one feature line has not a correspondent in another image.

α, β, γ : the weights of every term, and $\alpha + \beta + \gamma = 1$.

N_1: the number of feature lines whose candidates exist in another image.

N_2: the number of feature lines that have not any correspondents in another image.

The correspondence of feature lines between two images is assumed to be found out by searching to minimize the evaluation function M.

3.3 X-Y coordinate of feature line

As shown in figure 3, the coordinate (x_{lj}, y_{lj}) of feature line l_j in the robot frame at time t_i when the image is acquired, is calculated using two adjacent reference images taken at time t_i and time t_{i-1} during teaching.

Here, let R_{ri} and R_{ri-1} be the robot frames at time t_i and time t_{i-1} during teaching, and let the parameters: ϕ_{rji} and ϕ_{rji-1} denote the azimuth angles of feature line l_j in frame R_{ri} and frame R_{ri-1}, respectively. They can be acquired from the reference images taken at time t_i and time t_{i-1}.

The relation between frame R_{ri} and frame R_{ri-1} is given by

$$^{R_{ri}}_{R_{ri-1}}T = \begin{bmatrix} \cos\theta_{ri-1} & -\sin\theta_{ri-1} & x_{ri-1} \\ \sin\theta_{ri-1} & \cos\theta_{ri-1} & y_{ri-1} \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

where, $(x_{ri-1}, y_{ri-1}, \theta_{ri-1})$ is the pose of frame R_{ri-1} referred to frame R_{ri}, and was recorded during teaching.

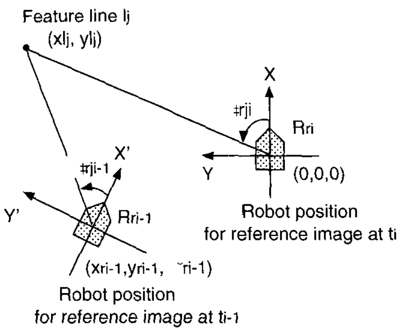

Figure 3: Location of a feature line and the robot for reference images at t_i and t_{i-1}

If let (x'_{lj}, y'_{lj}) denote the coordinate of feature line l_j in frame R_{ri-1}, the relation between (x'_{lj}, y'_{lj}) and (x_{lj}, y_{lj}) can be represented by the following equation.

$$^{R_{ri-1}}P_{lj} = ^{R_{ri}}_{R_{ri-1}}T^{-1} * ^{R_{ri}}P_{lj} \tag{3}$$

where, $^{R_{ri-1}}P_{lj} = [x'_{lj}, y'_{lj}, 1]^T$, and $^{R_{ri}}P_{lj} = [x_{lj}, y_{lj}, 1]^T$.

So, we can calculate (x'_{lj}, y'_{lj}) as

$$x'_{lj} = (x_{lj} - x_{ri-1})cos\theta_{ri-1} + (y_{lj} - y_{ri-1})sin\theta_{ri-1} \tag{4}$$

$$y'_{lj} = (-x_{lj} + x_{ri-1})sin\theta_{ri-1} + (y_{lj} - y_{ri-1})cos\theta_{ri-1} \tag{5}$$

In addition, the relation between the azimuth angle ϕ_{rji} and the coordinate (x_{lj}, y_{lj}) of feature line l_j in frame R_{ri} is given by

$$y_{lj} = x_{lj} * tan\phi_{rji} \tag{6}$$

We can also acquire the similar expression in frame R_{ri-1} as

$$y'_{lj} = x'_{lj} * tan\phi_{rji-1} \tag{7}$$

Therefore, the coordinate (x_{lj}, y_{lj}) of feature line l_j in frame R_{ri} is obtained by

$$x_{lj} = \frac{a * y_{ri-1} - b * x_{ri-1}}{c * \cos\theta_{ri-1} - d * \sin\theta_{ri-1}} \tag{8}$$

$$y_{lj} = x_{lj} * \tan\phi_{rji} \tag{9}$$

where,

$a = \cos\theta_{ri-1} - \sin\theta_{ri-1}\tan\phi_{rji-1}$,
$b = \sin\theta_{ri-1} + \cos\theta_{ri-1}\tan\phi_{rji-1}$,
$c = \tan\phi_{rji} - \tan\phi_{rji-1}$,
$d = 1 + \tan\phi_{rji}\tan\phi_{rji-1}$.

3.4 Position difference calculation

During playback navigation, the pose (x, y, θ) of the robot at time t_i will be calculated in frame R_{ri} by the least square method.

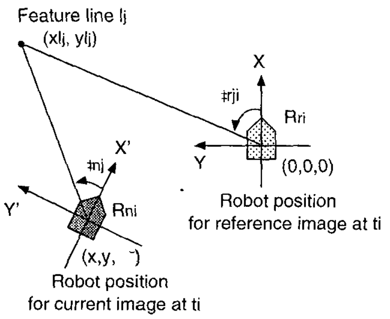

Figure 4: Location of a feature line and the robot for a reference image and a current one at t_i

As shown in figure 4, if let R_{ni} denote the robot frame at time ti during navigation, the coordinate (x'_j, y'_j) of feature line l_j in frame R_{ni} can be calculated as

$$x'_j = (x_{lj} - x)cos\theta + (y_{lj} - y)sin\theta \tag{10}$$

$$y'_j = (-x_{lj} + x)sin\theta + (y_{lj} - y)cos\theta \tag{11}$$

where, (x_{lj}, y_{lj}) is the coordinate of feature line l_j referred to frame R_{ri}, which was calculated in section 3.3.

Considering the observation error of the azimuth angle of feature line l_j, we obtain the following expression in frame R_{ni}.

$$arctan(y'_j, x'_j) = \phi_{nj} + \Delta_j \tag{12}$$

where, ϕ_{nj} is the azimuth angle of feature line l_j in frame R_{ni}, which can be acquired from the current image taken at time t_i. Δ_j is the observation error of ϕ_{nj}.

If the squared sum $E(x,y,\theta)$ of error Δ_j with respect to all feature lines is defined as expression 13, we may calculate the robot's pose (x,y,θ) to minimize $E(x,y,\theta)$.

$$E(x,y,\theta) = \sum_{j=1}^{N} \Delta_j^2 = \sum_{j=1}^{N} [arctan(y'_j, x'_j) - \phi_{nj}]^2 \tag{13}$$

where, N is the number of corresponding feature lines.

Let $F_j(x,y,\theta)$ denote the left term of expression 12,

$$\begin{aligned} F_j \quad & (x,y,\theta) = arctan(y'_j, x'_j) \\ = \quad & arctan(\frac{(-x_{lj}+x)sin\theta + (y_{lj}-y)cos\theta}{(x_{lj}-x)cos\theta + (y_{lj}-y)sin\theta}) \end{aligned} \tag{14}$$

and suppose that $x = x_0 + \Delta x, y = y_0 + \Delta y, \theta = \theta_0 + \Delta\theta$. Then, we expand $F_j(x,y,\theta)$ at (x_0, y_0, θ_0) as

$$\begin{aligned} F_j \quad & (x_0 + \Delta x, y_0 + \Delta y, \theta_0 + \Delta\theta) \\ = \quad & F_j(x_0, y_0, \theta_0) + \frac{\partial F_j}{\partial x}\Delta x + \frac{\partial F_j}{\partial y}\Delta y + \frac{\partial F_j}{\partial \theta}\Delta\theta \end{aligned} \tag{15}$$

$E(x,y,\theta)$ can be expressed as follows, too.

$$\begin{aligned} E \quad & (x_0 + \Delta x, y_0 + \Delta y, \theta_0 + \Delta\theta) \\ = \quad & \sum_{j=1}^{N} [F_j(x_0, y_0, \theta_0) - \phi_{nj} + \frac{\partial F_j}{\partial x}\Delta x + \frac{\partial F_j}{\partial y}\Delta y \\ + \quad & \frac{\partial F_j}{\partial \theta}\Delta\theta]^2 \end{aligned} \tag{16}$$

To minimize $E(\Delta x, \Delta y, \Delta\theta)$, we must let partial differentials $\frac{\partial E}{\partial \Delta x}, \frac{\partial E}{\partial \Delta y}, \frac{\partial E}{\Delta \theta}$ be equal to 0. So, $(\Delta x, \Delta y, \Delta\theta)$ is calculated as

$$\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta\theta \end{bmatrix} = \begin{bmatrix} \sum a_j^2 & \sum a_j b_j & \sum a_j c_j \\ \sum a_j b_j & \sum b_j^2 & \sum b_j c_j \\ \sum a_j c_j & \sum b_j c_j & \sum c_j^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum a_j d_j \\ \sum b_j d_j \\ \sum c_j d_j \end{bmatrix} \tag{17}$$

where,

$a_j = \frac{\partial F_j}{\partial x} = \frac{y_{lj} - y_0}{(x_{lj}-x_0)^2 + (y_{lj}-y_0)^2}$,

$b_j = \frac{\partial F_j}{\partial y} = -\frac{x_{lj} - x_0}{(x_{lj}-x_0)^2 + (y_{lj}-y_0)^2}$,

$c_j = \frac{\partial F_j}{\partial \theta} = -1$,

$d_j = \phi_{nj} - F_j(x_0, y_0, \theta_0)$,

$F_j(x_0, y_0, \theta_0) = arctan(\frac{(-x_{lj}+x_0)sin\theta_0 + (y_{lj}-y_0)cos\theta_0}{(x_{lj}-x_0)cos\theta_0 + (y_{lj}-y_0)sin\theta_0})$.

Here, if let (x_0, y_0, θ_0) be $(0,0,0)$, as shown in figure 4, $(\Delta x, \Delta y, \Delta\theta)$ is the robot's pose (x,y,θ), which we want to obtain, at time ti in frame R_{ri}.

4 Method for given trajectory tracking

Here, we consider the case that the images and the robot's poses were recorded in point A, B and C at time t_{i-1}, t_i and t_{i+1} during teaching, as shown in figure 5 . So, the taught route from time t_{i-1} to time t_{i+1} was replaced by line AB and line BC. And let's assume that the robot comes to point B' at time t_i of the playback stage. The trajectory in next time interval should be determined. Using the position difference calculation method given in section 3, the robot's pose at point B' will be estimated in the robot frame at point B, and then the recorded poses of the robot at point B and C can be transformed into the poses with respect to the robot frame at point B', and the equation of line BC in the robot frame at point B' will be obtained. This is the line which the robot will track after B", where the system finishes the calculation of the equation of line BC. Between B' and B", the navigation system carries out image processing, position difference estimating, and equation of line calculating, and the robot continues to move along the previous tracking line. And then, the robot will start tracking line BC from point B".

In this method, the straight line to follow and the relative robot posture to this line are gotten at each time interval.

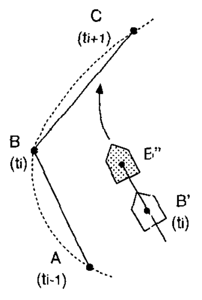

Figure 5: Trajectory tracking

Figure 6: Mobile robot system

So, the control input for robot's angular velocity $\omega(t)$ or steering angle to track this line can be calculated by the feedback of these relative posture values[6].

5 Experiments

5.1 Mobile robot system

The mobile robot, named "YAMABICO"(in figure 6), which has been developed by our research group, is used in our experiments. YAMABICO is a two driving wheeled autonomous mobile robot. It is controlled by several function modules, in which Master Module controls the entire system, and Locomation Module provides the straight line following algorithm[6] and controls the motor drive system of the robot.

To carry out our autonomous navigation, a notebook PC and an omnidirectional image sensor are mounted on YAMABICO. The PC communicates with YAMABICO via a RS232C serial line.

The used notebook PC is a Panasonic Let's Note CF-B5V, with a Celeron processor clocked at 500MHz. It controls image capturing, processes images, and calculates the robot's position difference in our navigation system. The notebook PC connects the image sensor with an REX-9590 video capture card (RATOC Systems International, Inc.). Using this card, a 320×240 image is captured in our system.

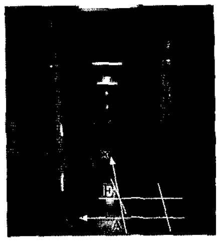

Figure 7: Experimental environment

(x, y : mm θ: degree)

		Real value	Calculated value	Difference
C	X	500	548	48
	Y	-500	-442	58
	θ	0	0.6	0.6
D	X	500	485	15
	Y	0	31	31
	θ	0	-0.2	0.2
E	X	500	428	72
	Y	0	25	25
	θ	5	5.06	0.06

Table 1: The results of position calculation

The omnidirectional image sensor consists of a hyperboloidal mirror and a CCD camera (Sony EVI-330).

5.2 Results on position difference calculation and navigation

To evaluate the accuracy of position difference calculation and the effectiveness of this navigation method, we made two experiments using our mobile robot system in the corridor in front of our laboratory.

(1) Position difference calculation

The experimental environment is shown in figure 7. The robot was located at point A, B, C, D and E, and took images of the environment, named I_A, I_B, I_C, I_D and I_E, respectively. The real poses of the robot at point C, D, and E are shown in table 1. We used the images I_A, I_B as reference images, and the images I_C, I_D, I_E as current images, for calculating the poses of the robot at Point C, D, and E, with respect to frame R_{XAY} attached to the robot at point A. The robot's pose at point B, referred to frame R_{XAY}, is $(0, -500, 0)$. The results of calculated poses and the differences between calculated values and real values are shown in table 1.

The experimental results show that the error of θ is small, and the errors of x and y are confined inside 10cm. It shows that it can provide good information to use the method in the navigation system.

(2) Example on navigation

In this experiment, we let the robot navigate along a taught route in the same corridor (figure 7 and figure 8). The route is a continuous line with the sections of a 1[m] straight line, two half-circles of 400 [mm] in radius, and another 2.4[m] straight line. First, The robot was controlled to move along the route at the speed of $30cm/s$ and recorded the images and the robot's poses. And as playing back, the robot started moving from the same point at

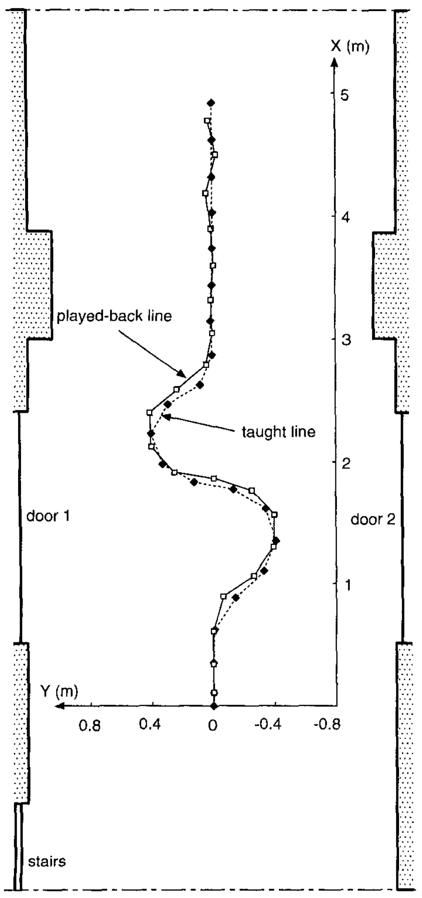

Figure 8: The corridor environment and the navigation result

same speed. In this experiment, the system took images at every one second interval. Figure 8 shows the result of this experiment. In this figure, the dot curve is the recorded teaching trajectory of the robot, and the solid one indicates the actual trajectory of the robot during navigation relative to the taught trajectory.

Figure 8 shows obviously that the robot tracked the taught route successfully.

6 Conclusions

In this study, we proposed an approach to a vision based navigation for mobile robots by teaching and playing-back.

In our method, a robot records a sequence of environmental images as reference ones during teaching stage. The corresponding relation of feature lines is found by matching among three images (a current image and two reference images). The robot's position is accurately calculated using the position relations of corresponding feature lines in the three images. These processes can be carried out in real-time. Experimental results confirmed this method is effective to realize a precise control of navigation.

The important thing is that the result of the calculated position has no accumulating error with the taught trajectory. So, in this method, without an accurate odometry system, it can realize a robust navigation system.

The important point of this method is the reliable feature line matching, we are planning to use color images to make it more robust, as next problem.

References

[1] Y. Yagi, K.Shouya, and M.Yachida, "Environmental Map Generation and Egomotion Estimation in a Dynamic Environment for an Omnidirectional Image Sensor", IEEE Int. Conf. on Robotics and Automation, San Francisco, USA, pp. 3493-3498, April, 2000.

[2] Arnaud Clerentin, Laurent Delahoche, Claude Pegard, and Eric Brassart, "A localization method based on two omnidirectional perception systems cooperation", IEEE Int. Conf. on Robotics and Automation, San Francisco, USA, pp.1219-1224, April, 2000.

[3] H. Ishiguro, T. Miyashita, and S. Tsuji, "T-Net for Navigating a Vision-Guided Root in a Real World", IEEE Int. Conf. on Robotics and Automation, Nagoya, Japan, pp.1068-1073, May, 1995.

[4] T. Ohno, A. Ohya, and S. Yuta "Autonomous Navigation for Mobile Robots Referring Pre-recorded Image Sequence", Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and System, Osaka, Japan, pp.672-679, 1996.

[5] Y. Matsumoto, K. Ikeda, M. Inaba and H. Inoue, "Visual Navigation using Omnidirectional View Sequence", IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Kyongju, Korea, pp.317-322, October, 1999.

[6] S. Iida, and S. Yuta, "Vehicle Command System and Trajectory Control for Autonomous Mobile Robots", Proc. IEEE/RSJ Int. Workshop on Intelligent Robots and System, Osaka, Japan, pp.212-217, 1991.