High-Speed Obstacle Avoidance and Self-Localization for Mobile Robots Based on Omni-directional Imaging of Floor Region

Daisuke Sekimori[1], Tomoya Usui[2], Yasuhiro Masutani[2], and Fumio Miyazaki[2]

[1] Akashi College of Technology, Akashi, Hyougo Japan
sekimori@akashi.ac.jp
[2] Osaka University, Toyonaka, Osaka Japan
{usui,masutani,miyazaki}@robotics.me.es.osaka-u.ac.jp
http://robotics.me.es.osaka-u.ac.jp/OMNI/

Abstract. In this paper, we propose a method of obstacle avoidance and a method of self-localization based on floor region provided by omni-directional imaging. With our methods, omni-directional imaging is used not for recognition of the three-dimensional environment but for detecting obstacles and landmarks in a wide area at high speed. Several experiments with a real robot according to the rules of the RoboCup Small-Size League was demonstrated, and proved the effectiveness of these methods.

1 Introduction

In obstacle avoidance and self-localization for autonomous mobile robots, information about the environment provided by an external sensor is indispensable.

Conventional studies on navigation, obstacle avoidance and self-localization of mobile robots using the omni-directional visual sensor propose methods to utilize vertical edges in the environment that appears radially from the image center[1][2]. However, these are based on the premise that vertical edges in the environment are stably extracted. Accordingly, these methods do not apply when the edges are blocked by moving objects or do not exist in the environment.

In this paper, we propose a method of avoiding obstacles around a robot and a method of self-localization based on omni-directional imaging of the floor region. We use an omni-directional visual sensor to detect obstacles and landmarks quickly and widely instead of a method to recognize the three-dimensions of the environment, in order to utilize the special features of an omni-directional visual sensor. Furthermore, we consider obstacle avoidance and self-localization in an environment where other obstacles exist.

In section 2, we explain omni-directional imaging of the floor region. In sections 3 and 4, we describe algorithms of obstacle avoidance and self-localization based on omni-directional imaging in detail. Finally, in section 5, we verify the effectiveness of these algorithms through several experiments with a real robot.

A. Birk, S. Coradeschi, and S. Tadokoro (Eds.): RoboCup 2001, LNAI 2377, pp. 204–213, 2002.

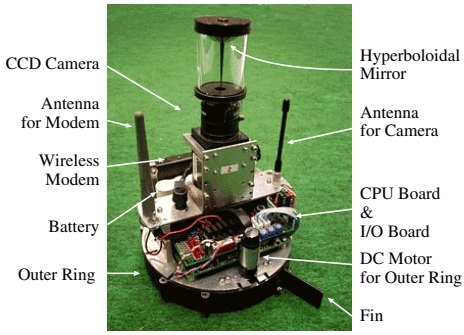


Fig. 1. Overview of the robot

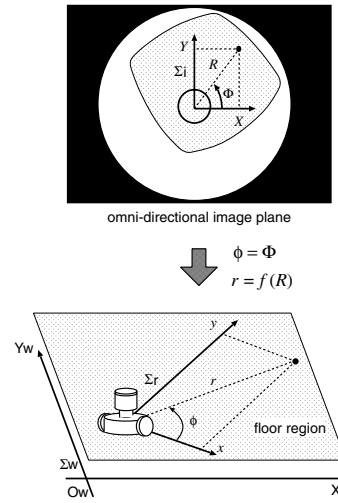


Fig. 2. Distance and direction of object

2 Omni-directional Imaging of Floor Regions

In a lot of indoor environments, it is possible to distinguish between the floor region and other objects by color. Accordingly, in this paper, we focus on an omni-directional, vertical-axis visual sensor model that is mounted on a mobile robot. The mobile robot that we have developed for the RoboCup Small-Size League is shown in **Fig.1**.

Here we consider a conversion from a point of the image coordinate system Σ_i to the correspoinding point on the floor of the robot coordinate system Σ_r, as shown in **Fig.2**. We assume that the conversion from (X, Y) to (x, y) is expressed as follows:

$$R = \sqrt{X^2 + Y^2}, \quad \Phi = \tan^{-1} Y/X \tag{1}$$

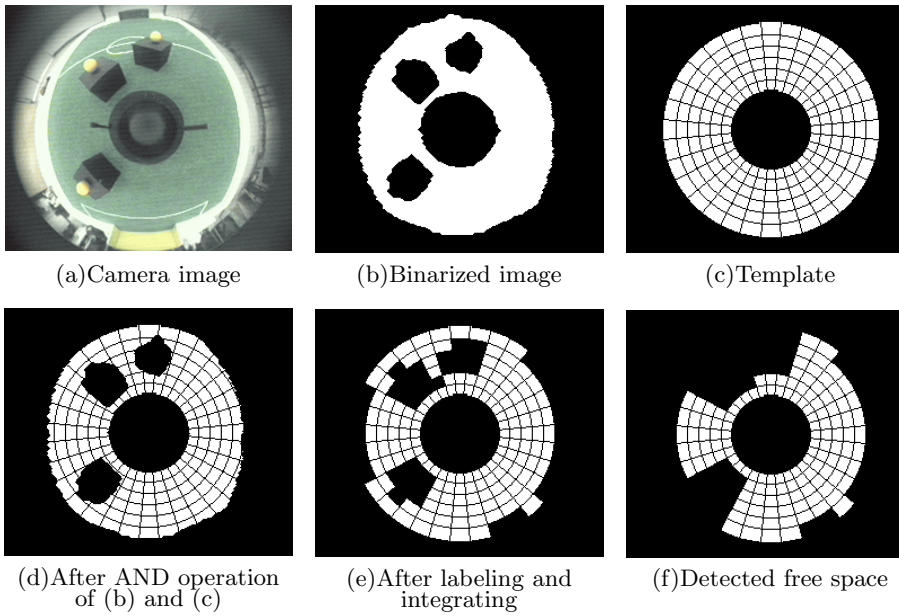
$$\phi = \Phi \tag{2}$$

$$r = f(R) \tag{3}$$

$$x = r\cos\phi, \quad y = r\sin\phi, \tag{4}$$

where $f(R)$ is a function determined from the shape of the omni-directional mirror.

Moreover, we make the following assumptions about the omni-directional image. (i) An omni-directional image with color is used, (ii) The floor region has a single color and other regions have other colors, (iii) White lines of constant width exist in the floor region.



(a)Camera image (b)Binarized image (c)Template

(d)After AND operation of (b) and (c) (e)After labeling and integrating (f)Detected free space

Fig. 3. Example of the method of detecting free space

3 Obstacle Avoidance Based on Omni-directional Imaging of Floor Regions

If the assumptions in the previous section are satisfied, the free space around the robot can be determined by observing the floor color in the robot's neighborhood. Therefore, in this section, we propose a method of high-speed obstacle avoidance based on omni-directional imaging of the floor region. In this method, the omni-directional image is first divided into many cells by superimposing a template image. Then, the free space around the robot is detected at high speed by using the labeling function, which is one of basic image processing functions. It is possible to modify resolution of the floor region totally or partially by changing the number of partitions in the template. In the following part of this section, first, we describe the algorithm for detecting free space around the robot. Second, as an example of obstacle avoidance based on the detected free space, we explain an obstacle avoidance method considering the mobile velocity of the robot.

3.1 Detection of Free Space

Step 1 (Binarizing)

The binarized image is taken from the omni-directional image (**Fig.3** (a)) by extracting floor color. Next, noise in the floor region is removed by using dilation and erosion processes a few times (**Fig.3** (b))

Step 2 (Dividing the floor region)
A template which is divided equally into I individual cells in a circumferential direction and J individual cells in a radial direction is prepared (**Fig.3** (c)). Let S_{ij} and A_{ij} $(i = 1, \cdots, I; j = 1, \cdots, J)$ denote a closed domain and the area of each cell, respectively. The floor region is divided into small areas of polar coordinates by using the AND operation between the binarized image and the template(**Fig.3** (d)).

Step 3 (Labeling and determination of the presence of obstacles)
N labeled areas are derived from the divided image. We define area A_k and the center of gravity $\boldsymbol{p}_k = (x_k, y_k)^T$ of each area $(k = 1, \cdots, N)$. Then the area ratio r_{ij} of the floor color is computed for every cell, using the following equation:

$$r_{ij} = \frac{1}{A_{ij}} \sum_{\substack{k \\ \boldsymbol{p}_k \in S_{ij}}} A_k \tag{5}$$

Next, the presence of an obstacle is judged for each cell by using the following equation (**Fig.3** (e)):

$$q_{ij} = \begin{cases} 0 \ (r_{ij} < r_{th}) \\ 1 \ (r_{ij} \geq r_{th}) \end{cases}, \tag{6}$$

where $q_{ij} = 0/1$ signifies the presence/absence of obstacles, and r_{th} denotes a threshold which determines the presence of an obstacle.

Step 4 (Detecting free space)
The distance to the closest obstacle in each direction around the robot is derived as discrete one-dimensional data a_j by using the following equation:

$$a_j = \min_{q_{nj}=0}(n-1) \qquad n = 1, \cdots, I+1, \tag{7}$$

where $q_{(I+1)j} = 0$ for convenience (**Fig.3** (f)).

3.2 Obstacle Avoidance Based on Mobile Velocity

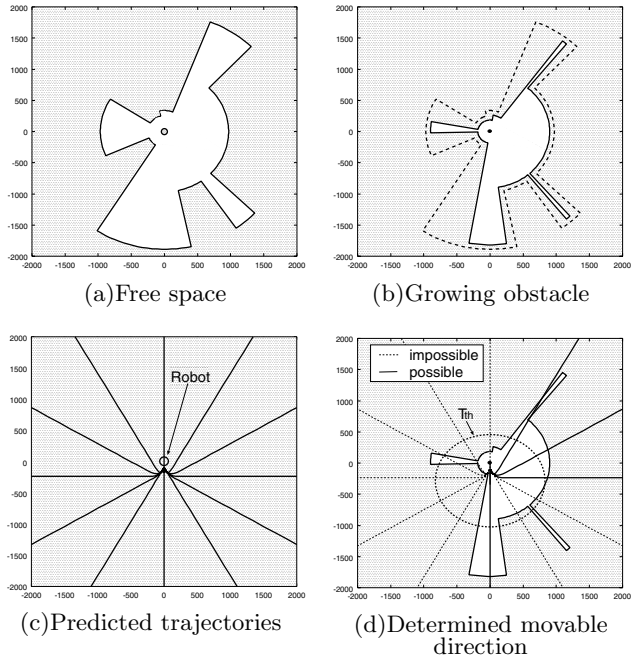
If free space around the robot is detected, we can propose various algorithms for obstacle avoidance for various purposes. We have developed an omni-directional mobile robot equipped with an omni-directional visual sensor as shown in **Fig.1**[3]. For this robot, we proposed a simple obstacle avoidance utilizing the free space detection mentioned above which considers only a goal direction[4]. In this section, we propose a method also considering the mobile velocity of the robot.

Although the robot we developed has an omni-directional mobile mechanism, the magnitude of the acceleration vector is limited in order to avoid slip and shock during sudden changes in direction. Actually, the modified command velocity $\boldsymbol{v}_r(t)$ is determined in the following equation:

$$\boldsymbol{v_r}(t) = \begin{cases} \boldsymbol{v_a} + \frac{\Delta \boldsymbol{v}}{|\Delta \boldsymbol{v}|}\alpha \cdot t \ (t \leq t_a) \\ \boldsymbol{v_b} \qquad\qquad\quad (t > t_a) \end{cases} \quad t_a = \frac{|\Delta \boldsymbol{v}|}{\alpha}, \ \ \Delta \boldsymbol{v} = \boldsymbol{v_b} - \boldsymbol{v_a}, \tag{8}$$

where the original command velocity(desired velocity) is $\boldsymbol{v}_b$. The time is 0 when the original command velocity is received, and the velocity of the robot at that time is $\boldsymbol{v}_a$. α is the upper limit value of acceleration.

Because of the acceleration limit, the robot can not turn immediately in a mobile direction at the same rate as the original command velocity in case that the mobile velocity is high. Therefore, it is not enough to detect obstacles only in the command direction. In this method, first, the position of an obstacle is extracted from the detected free space. Next, the trajectories of the robot are predicted based on current mobile velocity and projected mobile velocity in every direction. Finally, the mobile direction for obstacle avoidance can be determined beforehand. If this is impossible, the robot stops immediately and keeps searching until it finds a free space. This process is demonstrated in **Fig.4**. Details are omitted due to lack of space.



(a)Free space (b)Growing obstacle

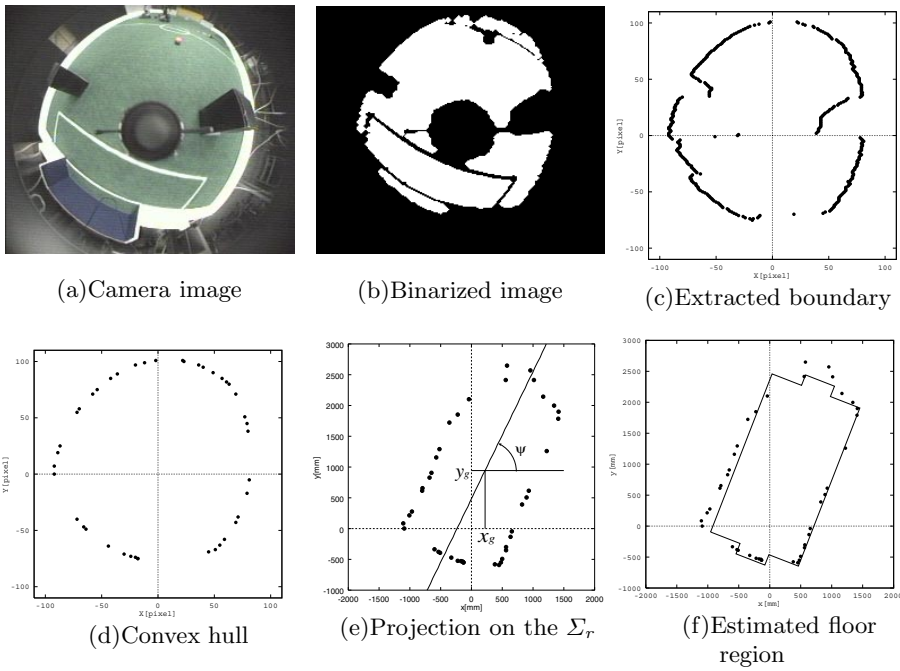
(c)Predicted trajectories (d)Determined movable direction

Fig. 4. Example of the method for determining movable direction

4 Self-Localization Based on Omni-directional Imaging of Floor Region

It is important for mobile robots to estimate self-location based on landmarks in the environment. However, if a specified point is regarded as a landmark, self-localization can not be estimated when the landmark is hidden.

On the other hand, a whole image of a known floor region can be acquired at one time with omni-directional imaging. Therefore, we propose a method of self-localization that regards the whole floor region as a landmark. In this method, even if part of the floor region is hidden by obstacles, the hidden part can be compensated by computing the convex hull of the set of boundary points of the floor region in the omni-directional imaging. After that, using the least squares method, considering the property of omni-directional imaging, a set of boundary points after restoration is applied to the already-known shape of the floor region. Although the problem is non-linear, by utilizing good approximate values provided from the geometric features (e.g. center of gravity, principle axis of second moment) of the region surrounded by the set of points, estimated values are calculated by solving the linearized equation. The algorithm of self-localization using the field in the RoboCup Small-Size League is explained as follows:



(a)Camera image (b)Binarized image (c)Extracted boundary

(d)Convex hull (e)Projection on the Σ_r (f)Estimated floor region

Fig. 5. Example of the method of self-localization

Step 1 (Extracting the floor region)
The binarized image is provided by extracting floor color (green in the case of the RoboCup) from the omni-directional image(**Fig.5** (a)). Noise is removed by using dilation and erosion processes several times (**Fig.5** (b))

Step 2 (Extracting the boundaries of the floor region)
Minimum $X_{min,i}$ and maximum $X_{max,i}$ of the X coordinates of the region

corresponding to each Y coordinate Y_i are detected by extracting the boundaries of the floor region.

Step 3 (Computing the convex hull)
In order to compensate for the boundaries, the convex hull [5] of a set M of points $\{(X_1, Y_1), \cdots, (X_M, Y_M)\}$ is derived from the set of boundary points $\{(X_{min,1}, Y_1), \cdots, (X_{min,N}, Y_N), (X_{max,1}, Y_1), \cdots, (X_{max,N}, Y_N)\}$(**Fig.5** (d)).

Step 4 (Converting)
The convex hull of a set of points is converted into a floor coordinate system from the image coordinate system, and a set of points $\{(x_1, y_1), \cdots, (x_M, y_M)\}$ is provided by Eq. (1)∼Eq. (4).

Step 5 (Geometric features)
The closed domain surrounded by the set of points and the infinitesimal area of the region are S and da, respectively. And we derive the following equation:

$$m_{pq} = \int_S x^p y^q da \tag{9}$$

The center of gravity (x_g, y_g) and the directions ψ_1, ψ_2 of the principal axes are calculated by the following equations:

$$x_g = \frac{m_{10}}{m_{00}}, \;\; y_g = \frac{m_{01}}{m_{00}} \tag{10}$$

$$\psi_1 = \frac{1}{2}\tan^{-1}\frac{2(m_{11} - x_g \cdot y_g \cdot m_{00})}{m_{20} - m_{02} - (x_g^2 - y_g^2)m_{00}}, \;\; \psi_2 = \psi_1 + \frac{\pi}{2} \tag{11}$$

Each m_{pq} is derived easily from the set of points by dividing the region into triangular segments. Moreover, we calculate the second moments I_1 and I_2 about the principle axes ψ_1 and ψ_2 respectively, and select the direction ψ as follows (**Fig.5** (e)):

$$\psi = \begin{cases} \psi_1 \; (I_1 \leq I_2) \\ \psi_2 \; (I_1 > I_2) \end{cases} \tag{12}$$

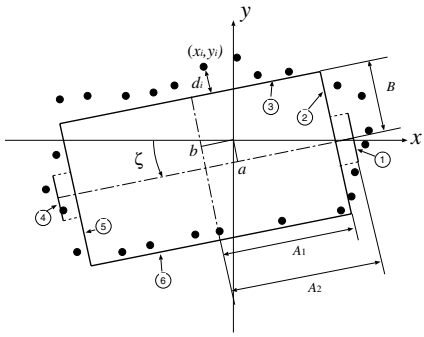
Step 6 (Matching with the model)
The set of points $\{(x_1, y_1), \cdots, (x_M, y_M)\}$ is converted into the set of points $\{(\tilde{x}_1, \tilde{y}_1), \cdots, (\tilde{x}_M, \tilde{y}_M)\}$ by the following equations:

$$\tilde{x}_i = (x_i - x_g)\cos\psi + (y_i - y_g)\sin\psi \tag{13}$$

$$\tilde{y}_i = -(x_i - x_g)\sin\psi + (y_i - y_g)\cos\psi \tag{14}$$

It is assumed that the boundary of the floor region consists of some segments of a line. Here, we assume the field of the RoboCup, which consists of six straight lines, as shown in **Fig.6**. Based on relational position, the line each point $(\tilde{x}_i, \tilde{y}_i)$ belongs to is determined, and the distance d_i between the point and the corresponding line is calculated. Next, we calculate parameters a, b, ζ that minimize the following evaluation function J:

$$J(a, b, \zeta) = \sum_{i=1}^{M} w_i {d_i}^2, \tag{15}$$



$$x\cos\zeta + y\sin\zeta - (A_2 - a) = 0 \cdots ①$$
$$x\cos\zeta + y\sin\zeta - (A_1 - a) = 0 \cdots ②$$
$$-x\sin\zeta + y\cos\zeta - (B - b) = 0 \cdots ③$$
$$-x\cos\zeta - y\sin\zeta - (A_2 + a) = 0 \cdots ④$$
$$-x\cos\zeta - y\sin\zeta - (A_1 + a) = 0 \cdots ⑤$$
$$x\sin\zeta - y\cos\zeta - (B + b) = 0 \cdots ⑥$$

$A_1 = 1370[\text{mm}], A_2 = 1550[\text{mm}],$
$B = 762.5[\text{mm}]$

Fig. 6. Boundary lines of the floor region (RoboCup field)

where w_i is the weighting factor and set in inverse proportion to its estimated variance. Since the error in the image is considered uniformly distributed, the weighting factor is given as $w_i = 1/\{\frac{df}{dR}(R_i)\}^2$. Although this minimization problem is non-linear, because the set of points is converted into a good approximation from the geometric features, the following approximation can be introduced, $\zeta \approx 0, \sin\zeta \approx \zeta, \cos\zeta \approx 1$. Then a,b and ζ which satisfies $\partial J/\partial a = 0, \partial J/\partial b = 0, \partial J/\partial \zeta = 0$ are solved. Using the solution, new x_g, y_g and ψ values are given in the following equations:

$$x_g^{'} = x_g - a\cos\zeta + b\sin\zeta \tag{16}$$
$$y_g^{'} = y_g - a\sin\zeta - b\cos\zeta \tag{17}$$
$$\psi^{'} = \psi + \zeta \tag{18}$$

In order to decrease the error caused by the approximation, if conditions: $|a| > a_{th}, |b| > b_{th}, |\zeta| > \zeta_{th}$ ($*_{th}$ means each threshold) are satisfied, x_g, y_g, ψ is replaced by $x_g^{'}, y_g^{'}, \psi^{'}$ and we return to the beginning of this step.

Step 7 (Converting to self-location)
The position and orientation of the robot in the field coordinate system ${}^w x, {}^w y, {}^w \theta$ are computed in the following equations:

$${}^w\theta = \psi^{'}, \quad \pi - \psi^{'} \tag{19}$$
$${}^w x = -x_g^{'}\cos({}^w\theta) + y_g^{'}\sin({}^w\theta) \tag{20}$$
$${}^w y = -x_g^{'}\sin({}^w\theta) - y_g^{'}\cos({}^w\theta) \tag{21}$$

For the two solutions of ${}^w\theta$, either is selected according to other landmarks(e.g. goal position) or the time series. The estimated floor region is finally shown in **Fig.5** (f). By weighting various factors, an estimation that attaches great importance to nearby points is realized.

5 Experiments

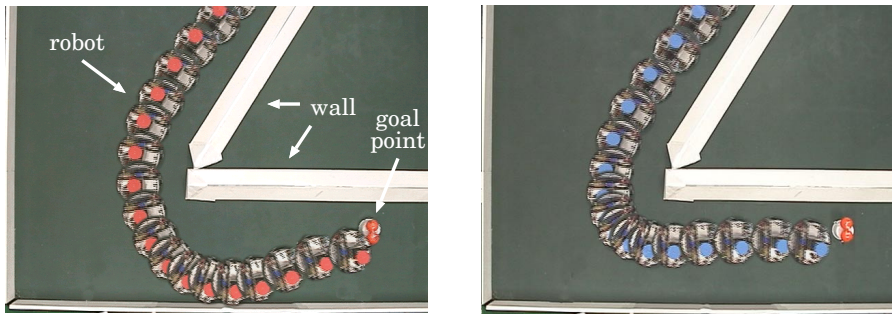
We equipped the robot with omni-directional vision by mounting a wireless CCD camera (RF System, PRO-5) with a hyperboloidal mirror (Accowle)(**Fig.1**), and applied the proposed method of omni-directional imaging, which is provided at the lens center 165[mm] above the floor. By using model[6] for conversion from distance R[pixel] in the image to distance r[mm] on the floor region, the following relation is derived: $f(R) = R/(A - B\sqrt{C + R^2})$ where $A = 10.12, B = 2.054 \times 10^{-2}, C = 2.298 \times 10^5$. IP5005(Hitachi) was used for image processing, with a resolution of 256×220[pixel^2].

5.1 Obstacle Avoidance

As an example of obstacle avoidance, we undertook an experiment in which the robot turned at an acute angle corner of about 55 deg. and aimed at a goal point.

We experimented on two types of obstacle avoidance. One took no account of the robot's velocity and the other took the robot's velocity into account under the following conditions: number of cells of template $I = 6$,$J = 32$, maximum velocity of the robot v=400mm/s, upper limit of acceleration α=300mm/s^2.

As a result, both turns were executed without colliding with the wall and the robot reached the goal point. However, because the method that takes the robot's velocity into acount allows for early directional changes, it facilitated turning corners smoothly. And, the mean cycle of the total process was about 40ms when the robot's velocity was taken into account.



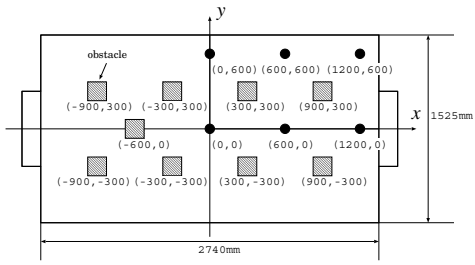
(a)Taking no account of the robot's velocity (b)Taking account of the robot's velocity

Fig. 7. Experimental results of obstacle avoidance (top view)

5.2 Self-Localization

In the field of the RoboCup Small-Size League, cubic obstacles with sides of 130[mm] were put as shown in **Fig.8**. The robot stopped at six places and

estimated self-position 10 times at each spot. The RMS of error is shown in **Fig.8**. Results showed errors at each place amounted to dozens of mm, precise enough for the RoboCup. And, the mean cycle of the total process for the robot, including estimation of self-position, was about 45ms.



Robot position		Error(RMS)		
x [mm]	y [mm]	Δx [mm]	Δy [mm]	$\Delta\theta$ [deg]
0	0	84	19	1.6
600	0	41	16	2.6
1200	0	11	30	2.2
0	600	37	7	0.5
600	600	73	20	3.3
1200	600	21	10	5.2

Fig. 8. Observing points and errors in estimation

6 Conclusion

In this paper, we propose a method of obstacle avoidance and a method of self-localization based on floor region provided by omni-directional imaging by the robot. In addition, high-speed processing is realized by observing a large area of the floor region quickly instead of observing a large area of the floor region in detail. We are planning to demonstrate the effectiveness of our methods in an actual competition of the RoboCup.

References

1. Y.Yagi, S. Kawato and S. Tsuji : "Collision Avoidance for Mobile Robot with Omni-directional Image Sensor (COPIS)", Proceedings IEEE the International Conference on Robotics and Automation, Vol.1, pp. 910-915, 1991
2. A.Clerentin, L.Delahoche, E.Brassart: "Cooperation between two omnidirectional perception systems for mobile robot localization", Proceedings of the 2000 IEEE/RSJ Intemational Conference on Intelligent Robots and Systems, pp. 1499-1504, 2000
3. N.Mori et al.: "Realization of Soccer Game with Small Size Robots which have Omni-directional Mobile Mechanism and Omni-directional Vision — Strategies of Team OMNI —"(in Japanese), Proceedings of the 6th SIG-Challenge of Japanese Society for Artificial Intelligence, pp. 42-47, 2000
4. D.Sekimori et al.: "High-speed Obstacle Avoidance of Mobile Robot Based on Labeling of Omni-directional Image"(in Japanese), Proceedings of the 18th Annual Conference of the Robotics Society of Japan, Vol.3,pp. 995-996,2000
5. Mark de Berg et al.: "Computational Geometry: Algorithms and Applications", Springer-Verlag, 2000
6. Y.Yagi: "Real-time Omnidirectional Image Sensor"(in Japanese), Journal of Robotics Society of Japan, Vol.13, No.3, pp. 347-350, 1995