Obstacle Detection and Self-Localization without Camera Calibration Using Projective Invariants

Kyoung Sig Roh, Wang Heon Lee, In So Kweon
Dept. of Automation and Design Eng., KAIST
e-mail address : iskweon@cais.kaist.ac.kr

Abstract : *In this paper, we propose two new vision-based methods for indoor mobile robot navigation. One is a self-localization algorithm using projective invariant and the other is a method for obstacle detection by simple image difference and relative positioning. For a geometric model of corridor environment, we use natural features formed by floor, walls, and door frames. Using the cross-ratios of the features can be effective and robust in building and updating model-base, and matching.*

We predefine a risk zone without obstacles for a robot, and store the image of the risk zone, which will be used to detect obstacles inside the zone by comparing the stored image with the current image of a new risk zone. The position of the robot and obstacles are determined by relative positioning.

The robustness and feasibility of our algorithms have been demonstrated through experiments in corridor environments using an indoor mobile robot, KASIRI-II (KAist SImple Roving Intelligence).

1. INTRODUCTION

Self-localization and obstacle detection are two basic functions for a successful navigation of a mobile robot.

Among various approaches, vision-based approaches have been proven to be very effective and flexible. Matsumoto et al.[1] proposed a model of the route, the "View-Sequenced Route Representation (VSRR)", for autonomous navigation. A VSRR consists of a sequence of view images, which have necessary information for localization, steering angle determination and obstacle detection. The resolution of the localization was, however, about 0.5-1m and it is impractical in case that the rotation of camera is larger than about 15 degrees. Kosaka and Kak[2] implemented a system for a given environment using a CAD model based expectation map. This method constructed a complex database and required additional analysis for handling uncertainty.

In most vision-based approaches, databases for environments become complex because observed geometric properties are not invariant under the projective transformation. Thus, matching is also very complex and time consuming. Also most approaches require an exact calibration.

We present a new efficient self-localization algorithm, which does not need any calibration, using 1-D perspective invariant [3, 4, 5] and the relative positioning [9]. We detect obstacles by comparing the prestored risk zone with a current risk zone. The positions of the detected obstacles are also determined by relative positioning.

Our algorithm for navigation in corridors and similar indoor environments is based on two basic assumptions that the ground plane is flat and two parallel side-lines are formed by floor and two side walls. We also assume that an environmental map database is available for matching between the scene and the model. Intersection points between floor and the vertical lines of door frames are used as point features to compute cross ratios. As an off-line process, we construct a database consisting of the cross ratios of point features. Using the cross ratios in the constructed database, the correspondences between the model and scene features can be found. The corresponding point features in the database of a real environment and in the image are used to compute the positions of the mobile robot and obstacles inside the risk zone.

We demonstrate the robustness and feasibility of our algorithms through experiments in indoor environments using an indoor mobile robot.

The paper is organized as follows: In section 2, we explain the basic theory of cross ratio and an error model, and relative positioning. In section 3, we explain the

Proc. IROS 97 0-7803-4119-8/97/$10©1997 IEEE

model-base construction and hypotheses generation. In section 4, we present a self-localization algorithm. In section 5, we present a method for obstacle detection and pose determination. In section 6, we explain image processing algorithms for the extraction of point features and vanishing points. In section 7, we present experimental results for a typical corridor environment.

2. BASIC THEORY of PROJECTIVE INVARIANT

2.1 Projective invariant: Cross-Ratio

In this section, we review the cross-ratio and present a method to determine an error model of the cross-ratio.

Let (Q_1, Q_2, Q_3, Q4) and (P_1, P_2, P_3, P_4) denote points on a line in an object plane and the corresponding points on the image plane, respectively.

X_i is the coordinate of Q_i with respect to an object coordinate system and x_i is the corresponding coordinate of Pi in the camera coordinate system. Then we define the cross-ratio,1-D invariant, as follows:

$$C(x_1,x_2;x_3,x_4)=\frac{(x_1-x_2)(x_3-x_4)}{(x_1-x_3)(x_2-x_4)}=\frac{(X_1-X_2)(X_3-X_4)}{(X_1-X_3)(X_2-X_4)}. \quad (1)$$

In Eq.(1), the cross ratio is a function of four variables :

$$C=C(x_1,x_2,x_3,x_4). \quad (2)$$

For noisy observations, we define a noisy invariant and its variation:

$$\tilde{C}=\tilde{C}(\tilde{x}_1,\tilde{x}_2,\tilde{x}_3,\tilde{x}_4) \quad (3)$$

$$V(\tilde{C})=E[(\tilde{C}-C)^2]=E\left[\left(\sum_{i=1}^{5}[\,\xi_i\frac{\partial\tilde{C}}{\partial\tilde{x}_i}]\right)^2\right]=\sigma_0^2\sum_{i=1}^{5}\left[\left(\frac{\partial\tilde{C}}{\partial\tilde{x}_i}\right)^2\right] \quad (4)$$

where ξ_i denotes an independently distributed noise term with the mean "0" and the variance σ_o^2.

2.2 Relative Positioning of Detected Obstacles

Two plane projective invariants on a plane are defined as follows:

$$\mathbf{I}_1=\frac{\det(\mathbf{p}_5\mathbf{p}_2\mathbf{p}_3)\det(\mathbf{p}_4\mathbf{p}_1\mathbf{p}_2)}{\det(\mathbf{p}_4\mathbf{p}_2\mathbf{p}_3)\det(\mathbf{p}_5\mathbf{p}_1\mathbf{p}_2)}=\frac{\det(\mathbf{P}_5\mathbf{P}_2\mathbf{P}_3)\det(\mathbf{P}_4\mathbf{P}_1\mathbf{P}_2)}{\det(\mathbf{P}_4\mathbf{P}_2\mathbf{P}_3)\det(\mathbf{P}_5\mathbf{P}_1\mathbf{P}_2)} \quad (5)$$

$$\mathbf{I}_2=\frac{\det(\mathbf{p}_5\mathbf{p}_3\mathbf{p}_1)\det(\mathbf{p}_4\mathbf{p}_1\mathbf{p}_2)}{\det(\mathbf{p}_4\mathbf{p}_3\mathbf{p}_1)\det(\mathbf{p}_5\mathbf{p}_1\mathbf{p}_2)}=\frac{\det(\mathbf{P}_5\mathbf{P}_3\mathbf{P}_1)\det(\mathbf{P}_4\mathbf{P}_1\mathbf{P}_2)}{\det(\mathbf{P}_4\mathbf{P}_3\mathbf{P}_1)\det(\mathbf{P}_5\mathbf{P}_1\mathbf{P}_2)}$$

where $\mathbf{p}_i$ and $\mathbf{P}_i$, i=1~5, represent the coordinates of points on the image plane and the corresponding points on an object plane, respectively.

We rewrite Eq. (5) as follows:

$$\mathbf{I}_1\frac{\det(\mathbf{P}_4\mathbf{P}_2\mathbf{P}_3)}{\det(\mathbf{P}_4\mathbf{P}_1\mathbf{P}_2)}=\frac{\det(\mathbf{P}_5\mathbf{P}_2\mathbf{P}_3)}{\det(\mathbf{P}_5\mathbf{P}_1\mathbf{P}_2)}=\frac{X_5(Y_2-Y_3)-Y_5(X_2-X_3)+(X_2Y_3-X_3Y_2)}{X_5(Y_1-Y_2)-Y_5(X_1-X_2)+(X_1Y_2-X_2Y_1)}$$

$$\mathbf{I}_2\frac{\det(\mathbf{P}_4\mathbf{P}_3\mathbf{P}_1)}{\det(\mathbf{P}_4\mathbf{P}_1\mathbf{P}_2)}=\frac{\det(\mathbf{P}_5\mathbf{P}_3\mathbf{P}_1)}{\det(\mathbf{P}_5\mathbf{P}_1\mathbf{P}_2)}=\frac{X_5(Y_3-Y_1)-Y_5(X_3-X_1)+(X_3Y_1-X_1Y_3)}{X_5(Y_1-Y_2)-Y_5(X_1-X_2)+(X_1Y_2-X_2Y_1)} \quad (6)$$

or

$$\begin{aligned}AX_5-BY_5&=-C\\DX_5-EY_5&=-F\end{aligned} \quad (7)$$

where

$$A=\mathbf{I}_1\frac{\det(\mathbf{P}_4\mathbf{P}_2\mathbf{P}_3)}{\det(\mathbf{P}_4\mathbf{P}_1\mathbf{P}_2)}(Y_1-Y_2)-(Y_2-Y_3)$$

$$D=\mathbf{I}_2\frac{\det(\mathbf{P}_4\mathbf{P}_3\mathbf{P}_1)}{\det(\mathbf{P}_4\mathbf{P}_1\mathbf{P}_2)}(Y_1-Y_2)-(Y_3-Y_1)$$

$$B=\mathbf{I}_1\frac{\det(\mathbf{P}_4\mathbf{P}_2\mathbf{P}_3)}{\det(\mathbf{P}_4\mathbf{P}_1\mathbf{P}_2)}(X_1-X_2)-(X_2-X_3)$$

$$E=\mathbf{I}_2\frac{\det(\mathbf{P}_4\mathbf{P}_3\mathbf{P}_1)}{\det(\mathbf{P}_4\mathbf{P}_1\mathbf{P}_2)}(X_1-X_2)-(X_3-X_1)$$

$$C=\mathbf{I}_1\frac{\det(\mathbf{P}_4\mathbf{P}_2\mathbf{P}_3)}{\det(\mathbf{P}_4\mathbf{P}_1\mathbf{P}_2)}(X_1Y_2-X_2Y_1)-(X_2Y_3-X_3Y_2)$$

$$F=\mathbf{I}_2\frac{\det(\mathbf{P}_4\mathbf{P}_3\mathbf{P}_1)}{\det(\mathbf{P}_4\mathbf{P}_1\mathbf{P}_2)}(X_1Y_2-X_2Y_1)-(X_3Y_1-X_1Y_3).$$

Assume that we want to know the relative position (X_5, Y_5) of an object point with respect to known four object points (X_1, Y_1), (X_2, Y_2), (X_3, Y_3), and (X_4, Y_4). If the image coordinates of the five points are found, two plane projective invariants I_1 and I_2 can be computed by Eq.(5). Therefore, from Eq.(7), the relative position (X_5, Y_5) is uniquely determined.

3. MODEL-BASE CONSTRUCTION and HYPOTHESES GENERATION

In this section, we explain how to construct a database using the cross ratio. Fig.1 shows a top-down view of a typical corridor scene. Point features on the left and the right wall, which are the intersection points between floor and door frames, are represented by L_K and R_K, respectively.

We compute the cross-ratio $\tilde{C}$ and its variance $\mathbf{V}(\tilde{C})$ for a set of point features, (L^i_K, L^i_{K+1}, L^i_{K+2}, L^i_{K+3}) or (R^i_K, R^i_{K+1}, R^i_{K+2}, R^i_{K+3}) in a corridor **i**. We store {i, LEFT or RIGHT, K} into the entries of a hash table, whose range is determined by $\tilde{C}\pm\mathbf{V}(\tilde{C})$.

Given an image, we extract a set of point features and compute the cross-ratio for the set. Then, the correspondences between the model and the scene features are hypothesized by indexing the entries of the hash table with a similar cross-ratio.

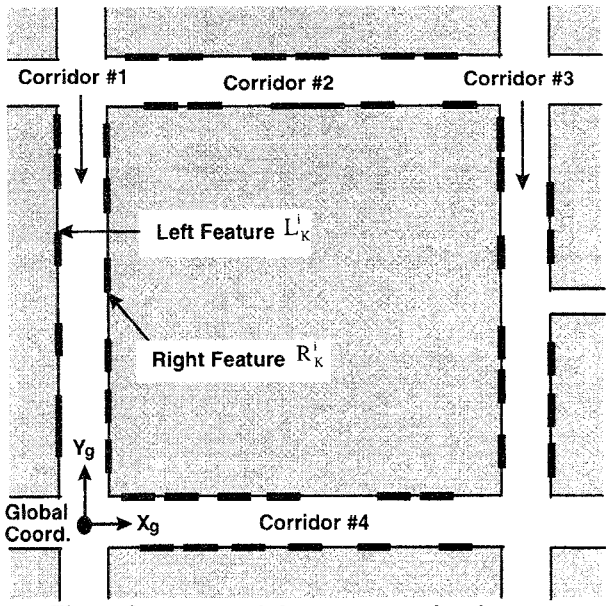


Fig.1. Scene model to setup a database.

Fig.2 shows an image and an extracted vanishing point, and point features. Table 1 presents the image coordinates of point features and a cross-ratio computed from these points and the hypotheses generated by the cross-ratio value.

In Table 1, for example, (1, RIGHT, 5) means that an image feature, denoted as 1 in Fig.2, corresponds to the fifth door frame on the RIGHT hand side of corridor 1.

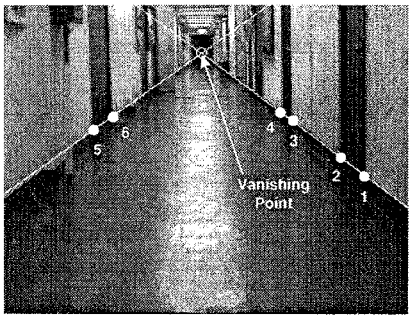


Fig.2. An image and extracted point features.

Table1. Result of a hypotheses generation.

Pt.	Image (Pixel)	Cross Ratio	Generated Hypotheses
1	(568,264)	0.046	**(1, RIGHT, 5)** (3, RIGHT, 1)
2	(530,238)		
3	(452,180)		
4	(436,168)		

4. SELF-LOCALIZATION

Assume that the virtual robot center is defined by the vertical projection of the optical center onto floor.

In Fig.3, for example, the perspective projection of the virtual robot center produces an image point (X_m, Y_m) on the extended virtual image plane. Perspective projection of the virtual robot center provides a unique robot attached landmark, which will be used for self-localization by the relative positioning.

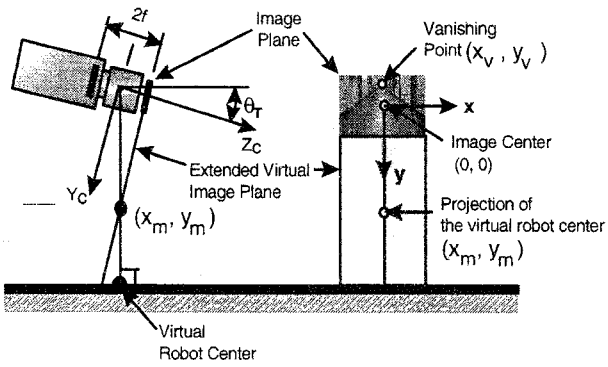


Fig.3 The image coordinate of the lens center.

From Fig.3, we can derive the relationship between the projected point of the virtual robot center (x_m, y_m) and a vanishing point (x_v, y_v) as:

$$f : s_y y_v = s_y y_m : f \tag{8}$$

where s_y is a scale factor which transforms an image coordinate to a metric coordinate.

Thus, the coordinate of the projection of the virtual robot center on the image plane is

$$y_m = \left(\frac{f}{s_y}\right)^2 \frac{1}{y_v} = f_y^2 \frac{1}{y_v}. \tag{9}$$

Now, we can compute the position of the robot by the relative positioning, given by Eq. (7).

We test the self-localization algorithm with an image shown in Fig.2. We assume that the correspondences are given from the matching process explained in section 3. Table 2 shows the image coordinates of the extracted vanishing point and point features, and the corresponding model coordinates.

Table2. Maching Result

Pt.	Image (Pixel)	Global Coord.(cm)	Pt.	Image (Pixel)	Global Coord.(cm)
1	(568,264)	(103.5, 843)	4	(436,168)	(103.5,1353)
2	(530,238)	(103.5,943)	5	(138,200)	(-103.5,1151)
3	(452,180)	(103.5,1258)	6	(168,178)	(-103.5,1318)
Vanishing Pt.	(x_v, y_v)=(312, 74)				

For the experiment, we used a camera with f=16 mm and s_y=0.0013mm and obtained (x_m, y_m)=(0, 1070950.5).

Using the extracted four image points 1, 2, 5, 6 and (x_m, y_m) with the corresponding object points, we can estimate the object coordinates of (x_m, y_m) that corresponds to the position of the robot in the global coordinate system by relative positioning. Table 3 presents the estimation result by the proposed method.

Table3. Result of a self-localization

Computed Value	True Value
(-14.5, 437.2) cm	(-5.0, 439.0) cm
(Xm, Ym)=(0, 1070950.5)	

In Eq.(9), if the optical axis of camera is parallel with the floor, Y_v becomes zero. Thus, $Y_m=\infty$. In practice, we can assume $Y_m=\infty$ with a very small error because f_y is very large.

In the case that $Y_m=\infty$, plane projective invariants I_1 and I_2 become:

$$\mathbf{I}_1 = \frac{\det(\mathbf{p}_4\mathbf{p}_1\mathbf{p}_2)}{\det(\mathbf{p}_4\mathbf{p}_2\mathbf{p}_3)} \frac{0(x_2 - x_3) - \infty(y_2 - y_3) + (x_2y_3 - x_3y_2)}{0(x_1 - x_2) - \infty(y_1 - y_2) + (x_1y_2 - x_1y_2)}$$

$$\mathbf{I}_2 = \frac{\det(\mathbf{p}_4\mathbf{p}_1\mathbf{p}_2)}{\det(\mathbf{p}_4\mathbf{p}_3\mathbf{p}_1)} \frac{0(x_3 - x_1) - \infty(y_3 - y_1) + (x_3y_1 - x_1y_3)}{0(x_1 - x_2) - \infty(y_1 - y_2) + (x_1y_2 - x_1y_2)}$$

or

$$\mathbf{I}_1 \approx \frac{\det(\mathbf{p}_4\mathbf{p}_1\mathbf{p}_2)}{\det(\mathbf{p}_4\mathbf{p}_2\mathbf{p}_3)} \frac{(y_2 - y_3)}{(y_1 - y_2)}. \quad (10)$$

$$\mathbf{I}_2 \approx \frac{\det(\mathbf{p}_4\mathbf{p}_1\mathbf{p}_2)}{\det(\mathbf{p}_4\mathbf{p}_3\mathbf{p}_1)} \frac{(y_3 - y_1)}{(y_1 - y_2)}$$

Table 4 presents the result of self-localization when $Y_m=\infty$.

Table4. Result of a self-localization when $Y_m=\infty$

Computed Value	True Value
(-14.5, 437.2) cm	(-5.0, 439.0) cm
(Xm, Ym)=(0, ∞)	

Therefore, we can compute the position of robot without calibration using a single image.

5. OBSTACLE DETECTION

In this section, we present methods for detecting obstacles based on image difference between a stored reference image and the current input image and estimating the position of the detected obstacle with respect to extracted features by relative positioning.

5.1 Detection of Risk Zone

Fig.4 shows a risk zone of the robot.

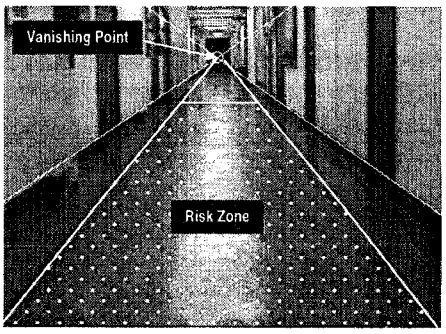
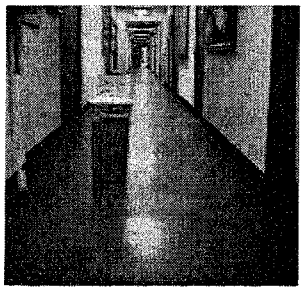


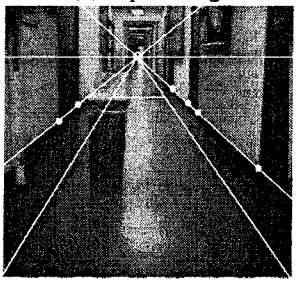
Fig.4 The Configuration of Risk Zone.

Fig.5 shows the procedures of an obstacle detection algorithm. Fig.5(a)~(d) show the result of risk zone extraction. Fig.5(e) shows the image difference between the current risk zone with the risk zone of non-obstacle, which is stored previously. Fig.5(f) shows a region of an extracted obstacle, bounded by white lines.

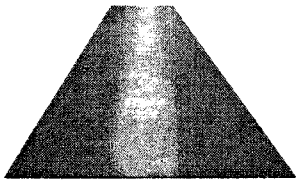


(a) Input image

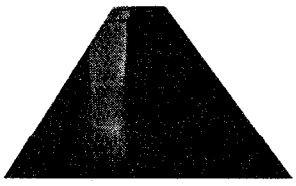
(b) Vanishing point



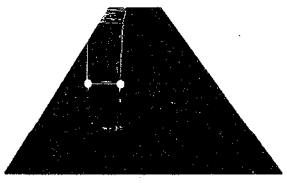
(c) Risk zone and point features



(d) Reference risk zone



(e) Difference



(f) Corner points

Fig.6 shows the extracted point features and the corner points of the detected obstacle. Thus, we can obtain positions of the obstacle from Eq.(7).

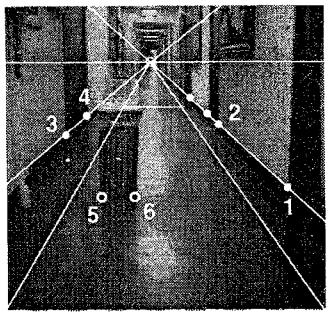


Fig.6 Extracted point features (1,2,3,4) and two corner points of the obstacle(5,6)

Table 5 presents the image and the corresponding global coordinates of the point features obtained by the self-localization algorithm, and the image coordinates of the corner points of the detected obstacle. Then, we can compute the global coordinates of the obstacle from Eq.(7). The computed and exact coordinates are given in Table 5.

Table 5. Result of relative positioning of a detected obstacle.

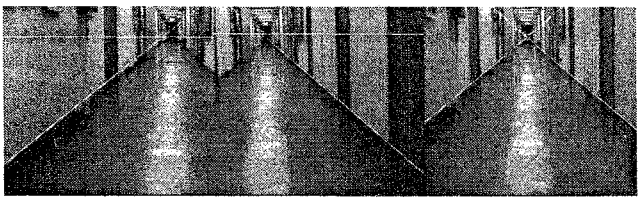
Pt.	Image .(Pixal)	Global Coord.(cm)	Computed (cm)	Exact (cm)
1	(451,316)	(103.5, 943)		
2	(342,198)	(103.5,1258)		
3	(95,221)	(-103.5,1151)		
4	(131,187)	(-103.5,1358)		
5	(151,301)	?	(-26.3,937.5)	(-30.0,945.0)
6	(210,303)	?	(3.4,940.3)	(-2.0,945.0)

6. IMAGE PROCESSING

We must extract point features to obtain the correspondences between the model and the scene using the cross ratio. In this paper, we select point features that are the intersections between floor and door frames which are distinguished landmarks in any corridor environment.

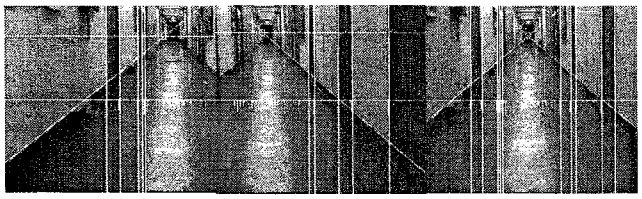
For detecting the vanishing point, we use the Hough transformation method. In order to reduce the computing time, we limit the range of the Hough space. We develop a method to detect the vanishing point by using the fact that the vanishing point is always projected on a constant horizontal line on the image plane, although one of two parallel lines is only detected.

Fig.7 shows the extracted vanishing point with left(a), right(b), and both(c) of two parallel lines. Fig.8 shows the extracted vertical lines for each scene in Fig.7.



(a) From left line (b) From right line (c) From both lines

Fig.7 Vanishing point extraction

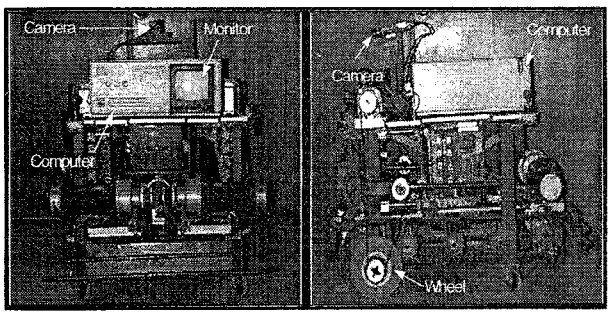


(a) From left line (b) From right line (c) From both lines

Fig.8 Vertical lines extraction

7. EXPERIMENTS

Experiments have been carried out in an indoor corridor environment using a mobile robot KASIRI-II. The mechanism of KASIRI-II consists of wheels for conventional running and an infinite path wheel for running on unflat floor such as stairs. We use a 585 pentium as the master controller. It also includes a motion control board, a vision processing board(MVB-02), IR sensors, servo motors and drivers, and sonars. Fig.9 shows the photograph of KASIRI-II.



(a) Front view (b) Side view

Fig.9 Photograph of KASIRI-II.

7.1 Self-Localization

Fig.10 shows an experiment scenario for testing the accuracy of the self-localization and obstacle detection algorithms. The robot is commanded to navigate along the center of corridor #1 from Y_g=358cm to Y_g=1558cm. An obstacle, shown in Fig.5, is located at (0, 1380).

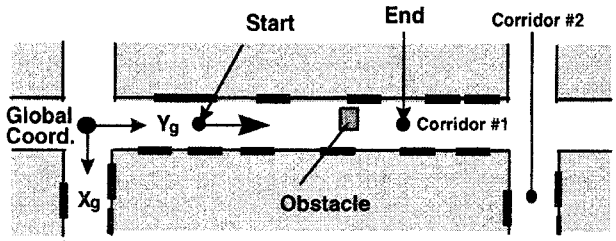


Fig.10 The run region of a mobile robot

Fig.11 shows the results of the self-localization of the mobile robot.

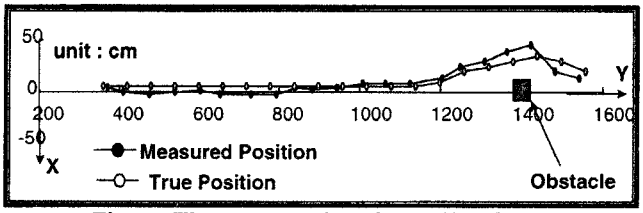


Fig.11 The result of self-localization

Fig.12 shows the error between the true positions, measured by a tape measure, and the extracted positions of the robot.

The proposed self-localization algorithm has successfully computed the robot positions with a maximum error smaller than 20cm.

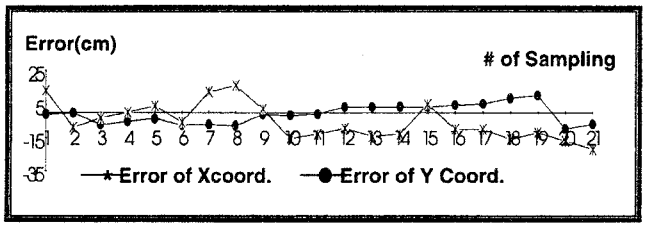


Fig.12 The error of self-localization

Table 6 shows the result of obstacle detection. The error is within 15cm.

Table 6. Results of obstacle detection

	Measured	*True*
1	*(-2.5,1373.2)*	*(0,1380)*
2	*(-30.3,1376.0)*	*(-30,1380)*

8. CONCLUSION

In this paper, we proposed new vision-based approaches for indoor mobile-robot navigation. For self-localization, we presented a method using projective invariant to search for correspondences. We also presented a method to detect obstacles and to determine the relative position of the detected obstacles using relative positioning.

Thus, we could solve the basic problems for navigation of a mobile robot in corridor environment by only vision system.

The average computing time in self-localization and obstacle detection are 2.08 sec and 1.49 sec on a Pentium 90MHz, respectively.

The error of self-localization and the error by relative positioning is within 20cm.

REFERENCE

[1] A.Kosaka and A.C.Kak, "Fast Vision-Guided Mobile Robot Navigation Using Model-Based Reasoning and Prediction of Uncertainties," *CVGIP:Image Under.* Vol. 56, No.3, November, pp. 271~329, 1992

[2] Y.Matsumoto, M.Inaba and H.Inoue, "Visual Navigation using View-Sequenced Route Representation," *Proc. of ICRA'96*, Minneapolis, Minnesota, April 83-88, 1996

[3] J.L.Mundy and A.Zisserman, "Geometric Invariance in Computer Vision," *MIT Press*, 1992

[4] R.O.Duda and P.E.Hart, "Pattern Classification and Scene Analysis," *John Wiley & Sons*, 1973

[5] J.G.Semple, "Algebraic Projective Geometry," *Oxford Press*, 1952

[6] I.S.Kweon et al., "Vision-based Behaviors for Indoor Mobile Robots," *IEICE Technical Report*, Vol. 91, no. 437, 1992

[7] R.Y.TSAI, "A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses," *IEEE J. of Robotics and Auto.*, Vol. RA-3, No. 4, August, 1987

[8] S.S.Jung., "Robust Self-Localization of Indoor Mobile Robots Using a Single CCD Image," *KAIST MS. Thesis*, 1996.

[9] J.R.Mohr,L.Morin and E.Grosso, "Relative Positioning with Uncalibrated Cameras," Geometric Invariance in Computer Vision, edited by L.Mundy and A.Zisserman, *MIT Press*, 1992, pp.440-460.