Proceedings of the 2000 IEEE
International Conference on Robotics & Automation
San Francisco, CA • April 2000

Mobile Robot Navigation in Indoor Environments using Object and Character Recognition

Masahiro Tomono   Shin'ich Yuta

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

Abstract

Navigation in unknown environments requires the robot to obtain the destination positions without a map. The utilization of model-based object recognition would be a solution, where the robot can estimate the destination positions from geometric relationships between the recognized objects and the robot. This paper presents a robot system for this kind of navigation, in which the robot navigates itself to the room designated by room number. The robot has an environment model including a corridor and a door with a room number plate, and utilizes the model for the efficient recognition of the objects and the estimation of their positions.

1 Introduction

Navigation in unknown environments, where the robot has no exact geometric information in advance, requires the robot to obtain the destination positions by itself. The utilization of model-based object recognition would be a solution. Once the robot has recognized the position and pose of some object in the environment, a three-dimensional model of the object enable the robot to estimate the destination point using geometric relationships between the object and the robot.

This paper presents a system for this kind of navigation, in which the robot navigates itself to the room designated by room number. Object recognition technique is used to find a door, and character recognition is utilized to interpret the room number on the number plate near the door and to determine whether it is the destination or not. The robot has models of a corridor and a door with a room number plate, and utilizes geometric relationships between the objects.

The rest of the paper is organized as follows. Section 2 presents the framework of the proposed system and describes the target task. Section 3 illustrates the system architecture. Section 4 and Section 5 explain the methods for achieving the task and show the experimental results. Section 6 presents lessons learned obtained from the experiments, followed by the concluding remarks.

2 Recognition-based Navigation

2.1 Framework

Fig.1 depicts the framework of the proposed sytem. The robot recognizes objects in the environment through images taken with a camera, and determines its behavior based on the objects' positions. The Object Recognizer extracts features such as edges and colors from the images, and matches them with the model features provided by the Model Database. The Object Recognizer conducts feature grouping and filering in order to increase the recognition accuracy and efficiency [8, 6]. The recognized objects are stored in the Environment Model, where the 3D positions of the objects are estimated by the Position Estimater. Then, the Behavior Controller calculates the temporal destination using the geometric relationships between the objects in the Environment Model, and sends vehicle commands to the Vehicle Controller. The Vehicle Controller navigates the robot using the odometry information. The Behavior Controller also determines the camera orientations to take images of the target objects using the geometric relationships between the objects and the robot.

A merit of this method is that an object can be used as a cue to find the destination. A high-level human interface can be implemented by expressing the destination using a symbol assigned to an object such as a room number. Moreover, the knowledge about the recognized object will give the robot the ability of

0-7803-5886-4/00/$10.00© 2000 IEEE

high-level behavior control. For example, in the task that the robot goes to the room designated by room number, the robot can check whether it has passed by a room or not using the knowledge that room numbers are sequential.

The problem of this method is the difficulty of object recognition. Object recognition in cluttered environments is especially hard. A solution would be to utilize constraints about the environment. In the target task of this paper, the robot uses geometric constraints between a corridor and a door, such as the floor is horizontal, walls are perpendicular against the floor, and a door is modeled as a rectangle which are contained in a wall. These constraints make the position estimation easy and facilitate the feature filtering for object recognition.

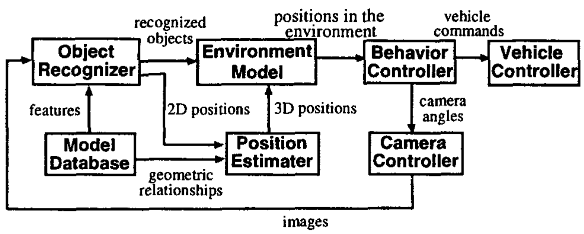


Figure 1: Framework

2.2 Target Task

In order to inspect the feasibility and effectiveness of the framework mentioned above, we have a target task in which the robot navigates itself using model-based recognition. The task to be given to the robot is navigation to the door which is designated by room number. The environment is a corridor in a building of University of Tsukuba (Fig. 2). The corridor includes doors, entrances to stairways, pillars, restroom entrances, and cabinets installed on the walls. The length of the corridor is about 30 meters. A room number plate is attached on the upper left side or upper right side of the door.

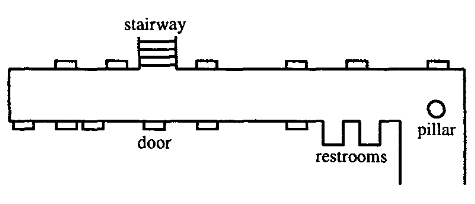


Figure 2: Environment

The robot has no map information at first, and is given the room number of the destination room. Then, the robot navigates itself to the door of the destination room by searching the door using the room number. For instance, given a command "go to Room L402", the robot runs along the corridor parallelly with a wall, and recognizes the room number on the plate attached near a door one after another to find Room L402.

Navigation by wall following and suitable actions at corners and crossings is coped with other mechanisms which are already studied at our laboratory [1], but it is beyond the scope of this paper.

A procedure to carry out the task is as follows.

(1) Navigation along the corridor
Given a room number, the robot sets the origin of the coordinate system to the current position, and starts to navigate. While it moves along the corridor, the robot keeps itself parallel with a wall by measuring the distance to the wall with its ultrasonic sensors.

(2) Finding of doors
While moving along the corridor, the robot takes images of the front scene. The robot recognizes doors based on the vertical edges in the image and the colors of the regions between two vertical edges. All the doors are supposed to be closed.

(3) Finding of a room number plate
The robot goes to the nearest one of the recognized doors. The robot estimates the position of the number plate, and moves to the appropriate location to take an image of the plate.

(4) Recognition of room number
The robot extracts the room number plate from the image, and recognizes the room number printed on the plate.

(5) Behavior decision
If the recognized room number is the destination, the robot moves to the front of the door and turns to face the door. Otherwise, it continues the task to search the destination.

Navigation along the corridor in step (1) has been implemented by our research group [1, 13]. Thus, we describe the methods for achieving steps (2) to (4) in Section 4 and Section 5.

3 System Architecture

3.1 Robot Architecture

The robot that carries out the task is based on the mobile-robot platform "YAMABICO", which has been developed by our research group. YAMABICO is a two-drivingwheeled self-contained robot with an odometer. It has four ultrasonic sensors to monitor obstacles in four directions (left, right, front, and back), and touch sensors on the bumper. Its control system consists of several boards including master board and vehicle control board, on which C programs are executed for vehicle control [3, 12].

The target robot in this paper will be built by installing a notebook PC and a video camera on a YAMABICO. The video camera is a Sony's EVI-D30, which has the degrees of freedom of pan and tilt, and which outputs 640*480 color images. The notebook PC is a Toshiba's Dynabook SS-3380, which has a 400MHz Pentium II MPU and 96 MB memory. The PC controls the camera orientation using the pan and tilt functions, captures an image, and analyzes the image to recognize objects in it. Based on the analyzed image, the PC determines what to do next and sends vehicle commands to the YAMABICO. The PC has two RS-232C ports, and communicates with the YAMABICO and the camera via each port.

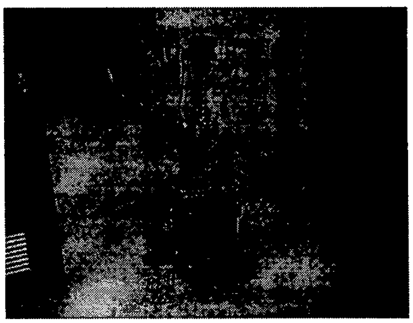


Figure 3: YAMABICO with a PC and a camera

3.2 Software Architecture

The software of the framework shown in Fig.1 is implemented in Java language (IBM's JDK1.1.8) on top of the notebook PC. The Object Recognizer has more than 20 image-processing operators including an edge detector, feature filters, and a character-template matcher. The Model Database contains a door model, a number plate model, and a corridor model. Each model is implemented as a Java object, which has basic attributes such as shape, size, and color. It also has geometric relationships with other objects. The Position Estimator has the rule for door-position estimation mentioned in Section 4, and a method for executing perspective transformation. The Environment Model consists of the instances of the models, whose positions are determined by the Position Estimator.

The Vehicle Controller is a Java object to control the YAMABICO. It hides the details of the communication protocol between the PC and the YAMABICO. In the current implementation, the Vehicle Controller has just "go forward/backward" and "turn left/right" commands, although the YAMABICO has various vehicle functions. The Camera Controller is a Java object to control the camera. It has "pan left/right", "tilt up/down", "zoom in/out", and "capture an image" commands.

The Behavior Controller is a Java object, which is responsible for coordinating object/character recognition, position estimation, vehicle control, and camera control. Object/character recognition and vehicle control sometimes fail because of various reasons including noises, obstacles, and estimation errors. In order for the robot to be robust, the Behavior Controller tries to recover the failures by redoing the actions. An action here is also a Java object, which has a goal to be achieved. If the goal of an action is not achieved, the Behavior Controller redoes the action, sometimes with different parameter values. Object/character recognition and vehicle control are implemented as an action of this sort.

4 Door Recognition

4.1 Door Model

A three-dimensional door model is used for door recognition. Fig. 4 illustrates the model. The door consists of two door plates and a room number plate. Each door plate is modeled as a rectangle which contains a small rectange for a window and a circle for a knob (the window and the knob are not used in this paper). The number plate is modeled as a small rectangle which is attached on a side edge of the door. The top edge of the plate is collinear with that of the door. The colors of the door and the number plate are light green and white respectively, while the walls are white.

The following geometric constraints are supposed between the floor, walls, and doors. (1) the floor is modeled as a horizontal plane, (2) walls are planes perpendicular with the floor, (3) two walls facing each other are parallel, (4) a door is contained in a wall, (5) the top edges of the doors are parallel with the floor, (6) the bottom edges of the doors are included in the floor.

The door model is implemented as a Java object, and is stored in the Model Database as mentioned in Section 3.

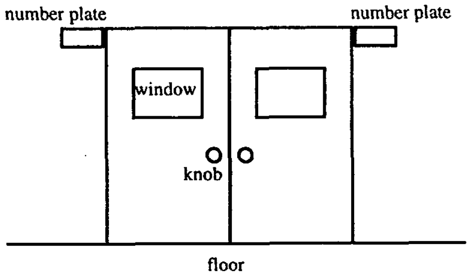


Figure 4: Door model

4.2 Door Recognition

(1) Detection of Door Regions

Doors are recognized using shape and color. In order to avoid occlusion problems, we use not the whole shape but only the side edges of a door. In the condition that the optical axis of the camera is parallel with the floor, the side edges of the doors are vertical in the image. Thus, the side edges of the doors must be included in the vertical edges extracted from the image.

The color of the door is utilized to find the door regions between the extracted vertical edges. If a region between two vertical edges includes a certain amount of pixels whose values are the same with the color value of the door model, the robot regards it as a door region. We adopt the following color value based on the YIQ basis in order to make the color value invariable against lightness [10].

$$H = \begin{cases} tan^{-1}(I/Q) & \text{for } \sqrt{I^2+Q^2} \geq th1 \\ undefined & \text{otherwise} \end{cases}$$

$$\begin{aligned} I &= 0.6R - 0.28G - 0.32B \\ Q &= 0.21R - 0.52G + 0.31B \end{aligned}$$

Here, R, G, and B are the red, green, and blue values respectively in the RGB space. Note that the color value H is unstable for monochrome, and that $th1$ is the threshold to determine whether the color value makes sense or not.

Although the utilization of both vertical edges and color increases the accuracy of recognition, there can be recognition errors. Other shape features including the top and bottom edges, the window, and the knob could help improve the accuracy.

(2) Estimation of Door Position

Once a door is recognized in the image, the position of the door in the real environment can be estimated using geometric relationships. Fig. 5 shows the geometric relationships between the camera and a door. The parameters are described below, where (x_w, y_w, z_w) denotes a point in the world coordinate system, and (x_v, y_v, z_v) denotes a point in the camera coordinate system.

L : distance in the x_w direction from the camera to the door's edge.
D : distance from the camera to the wall.
θ : angle between the optical axis and the x_w axis.
α : angle between the optical axis and the line from the focal point to the door's edge.
D_2 : distance between the optical axis and the door's edge in the image coordinate system.
kD_2 : distance between the optical axis and the door's edge in the camera coordinate system.
f : focal length of the camera.

D can be measured using ultrasonic sensors. D_2 is calculated from the x coordinate value of the recognized vertical edge of the door in the image. $\frac{k}{f}$, which is camera's intrinsic parameter, is measured in advance. θ is a crucial factor for accuracy, and is mentioned in the next section. Using these parameters, L and α are calculated as follows.

$$L = \frac{D}{tan(\theta+\alpha)}, \quad \alpha = tan^{-1}(\frac{k \cdot D_2}{f})$$

(L, D, z_w) is the relative position of the door edge from the camera in the world coordinate system.

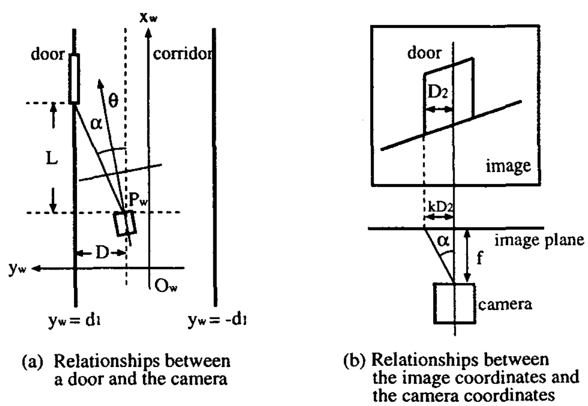


Figure 5: Geometric relationships between a door and the camera

(3) Error Analysis

The estimation of door position is affected by factors such as errors in the camera orientation (θ) and errors in the ultrasonic sensors (D), and the slant of the camera's vertical axis. The most critical factor is the camera orientation. Fig.6 shows the relationships between errors in θ and errors in L in the case that D is 100cm. The error ΔL becomes larger as L becomes larger. For example, when L is 400 cm, the error $\Delta\theta$ of two degrees causes the error ΔL of 70cm.

There could be two approaches to cope with this problem. One is to obtain the accurate value of θ. θ can be detected by the differentiation of a sequence of D measured by a ultrasonic sensor, but it is difficult to obtain sufficient accuracy. The vanishing point that is estimated using two parallel segments such as boundaries between the floor and the walls in the image can be used to calculate θ, but this is also difficult if the boundaries are not clear. The other approach is to reduce the expected ΔL by repeating the estimation of door position while the robot is moving toward the door. When the robot comes near the door, ΔL will be small (within 50cm). We employ the latter approach in the current system.

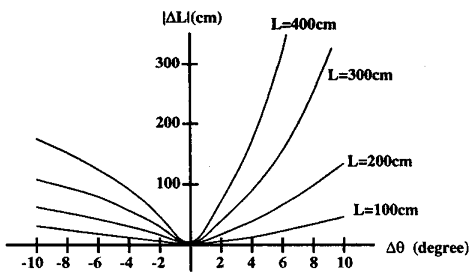


Figure 6: Errors in door position estimation

4.3 Experimental Results

We have conducted an experiment of door recognition. Table 1 shows the result. In the experiment, the robot took 30 images of a corridor while moving along the corridor. The robot moved 15 meters, and inspected six different doors. The distance between the robot and the wall was 120cm. In Table 1, *total* denotes the total number of the doors which appeared in the images. *success* is the number of the doors which were recognized successfully. *failure*1 is the number of the doors which were not recognized. *failure*2 is the number of the different objects which were recognized as a door.

The success rate is 74%. When the distance between the robot and the door is 300cm or less, the success rate is more than 90%. This indicates that the possibility of missing doors can be very low by taking images at short intervals while the robot is moving.

The experiment was conducted under fluorescent lamps at night. Although the lightness varied with the relative positions of fluorescent lamps and doors, the variation did not affect the recognition. This shows the color value mentioned in Section 4.2 is stable against lightness under the stable light source such as a fluorescent lamp. However, fluctuation was observed in the daytime experiment when strong sunlight came into the corridor. The color constancy in the daytime is a crucial problem to improve the robustness.

Table 1: Success rate of door recognition

total	*success*	*failure*1	*failure*2	*success rate*
58	43	15	13	0.74%

Fig.7 shows an example of detected door edges. Three pairs of vertical edges are extracted successfully. The leftmost pair of vertical edges, which looks like a bold line, is a recognition error of type *failure*2.

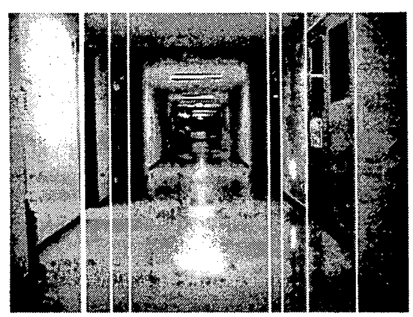


Figure 7: Detected door edges

Table 2 shows an experimental result of position estimation. In the experiment, the robot was placed at the points 400 to 700cm distant from a door, and after moving forward 200cm from the points, the robot took an image of the corridor. The purpose of moving the robot is to include the errors which are expected to be caused by the robot movement. The table lists the averages and standard deviations of the estimated door positions according to the distance between the robot and the doors (L = 200 to 500cm). The number of sampled data was 20 for each L. The distances between the robot and the wall were 100 to 120cm. The deviation is large when L is large as analyzed in the previous section.

Table 2: Result of door position estimation

L [cm]	200	300	400	500
average [cm]	202	304	416	544
deviation [cm]	26	20	34	64

5 Room Number Recognition

5.1 Number Plate Recognition

(1) Estimation of Number Plate Position

After estimating the door postion, the robot goes to a side edge of the door and stops there. The room number plate is attatched on the side edge or the other side edge. Even though the robot estimates the door position at the near point from the door, the estimation error can be around 50cm as shown in the previous section. This error can cause a failure in taking a number plate image. Thus, the robot pans the camera to take images of the door edge, and moves itself until the door's side edge comes within an appropriate region in the images. Then, the robot tilts up the camera toward the direction in which the number plate must be located, and takes an image. The tilt angle is calculated easily with the distance between the robot and the wall, and the height of the number plate from the camera.

(2) Number Plate Recognition

After taking an image of the number plate, the robot extracts the plate region from the image. The extraction based on position estimation using the door model is theoretically possible, but this is risky because of errors in the robot position and the camera orientation. Thus, we employ object recognition technique to extract the plate from the image. The robot extracts the plate using feature filtering/grouping on the basis of collinearity, parallelism, and proximity between the edges [8, 6]

The procedure of the plate recognition is as follows. (1) extracting edges: Edges are detected from the image. A curved edge is approximated with a chain of line segments. The line segments having the same slope and close end points are connected to be handled as one segment. (2) filtering segments by slope: Only the segments which are parallel or perpendicular with the floor are selected as candidates of the plate's edges. The slopes of these segments can be calculated through perspective projection based on the camera angles. (3) filtering segments by region: Only the segments which exists outside the door region are selected. (4) grouping segments into a L-junction: Two segments are grouped as a L-junction, if they are approximately perpendicular and a pair of their end points are close. (5) grouping L-junctions into a quadrangle: Two L-junctions are grouped as a a quadrangle, if two pairs of their end points are close and their corners are far. (6) filtering quadrangles by size: The quadrangles beyond the estimated plate size are rejected.

Steps (2) and (3) are examples of the feature filtering based on the geometric relationships between the camera and the plate. This filtering increases the recognition efficiency by reducing the number of candidate segments which comprise L-junctions in step (4).

5.2 Character Recognition

Before recognizing the room number, the character shapes are required to be rectified since the shapes are distorted by perspective projection. Several rectification methods have been proposed [5, 9]. In our current implementation, we suppose that the angle between the camera and the number plate is not very acute, and employ the weak perspective camera model. Under this assumption, a rectangluar plate is projected on the image as a parallelogram, and so the robot approximates quadrangles extracted from images to parallerograms. The robot calculates the affine transformation from the parallerogram extracted from the image to the rectangle whose size is stored in the plate model. Using this transformation matrix, the robot rectifies the character shapes.

The characters for room numbers consist of alphabet letters (A to Z, a to z) and figures (0 to 9). Character recognition is conducted using template matching. The template of a character is represented by a dot pattern on the 50*50 grid. Input characters are also normalized to a dot pattern on the same grid. The robot selects the best matched template according to the matching of both the character part (black dots) and the background part (white dots) of the grid.

5.3 Experimental Results

We have conducted experiments in order to evaluate the above-mentioned methods for plate recognition and room number recognition. The experiments were conducted under fluorescent lamps at night.

The first experiment measures the success rate of taking number plate images. In the experiment, the robot is located at the points of 50cm, 75cm, 100cm, 125cm, and 150cm far from a door's side edge. After moving forward 100cm from each point, the robot detected the door's side edge, and stops an appropriate position to take a number plate image. This experiment simulates the situation in which there are errors of -50cm to 50cm in the estimation of door position, and evaluates the robustness of the navigation to a door's side edge. The distance between the robot and the wall is 100cm, and eight doors were used.

The result is shown in Table 3. In Table 3, *total* denotes the total number of trials. *success* is the number of the successful trials, in which the image contaied the whole region of the number plate. *failure*1 is the number of the failures in which the robot stopped at the door's side edge but a portion of the number plate was outside the image. *failure*2 is the number of the failures in which the robot did not stop at the door's side edge. The failures of type *failure*1 are considered to have been caused by errors in the robot position and camera orientation. The failures of type *failure*2 are considered to have been caused by the failures of the door recognition.

Table 3: Success rate of taking number-plate images

total	*success*	*failure*1	*failure*2	*success rate*
40	30	4	6	0.75

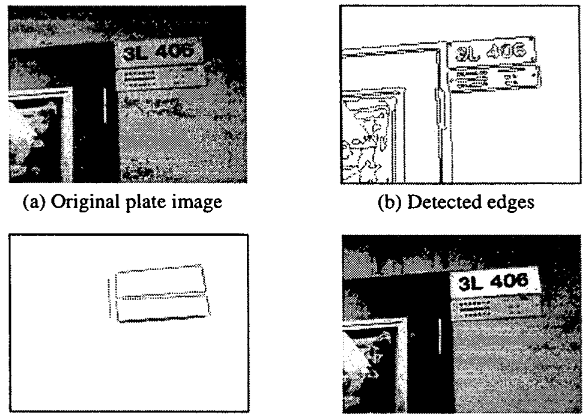
The second experiment measures the success rate of the room number recognition. In the experiment, the robot detects the number plate region from the image, and extracts the characters written in the region to recognize the room number. The samples are the images which were taken successfully in the first experiment.

Table 4 shows the result. *total* denotes the total number of the sample images. *success* is the number of the images in which the room numbers were recognized correctly. *failure*1 is the number of the character recognition failures, where the number plates were extracted correctly. *failure*2 is the number of the plate recognition failures. Three of the four plate recognition failures were caused by dark images, where the edge detection of the number plates failed. These dark images were brought by the backlight poured through the door windows. Since both the plate color and the wall color are white, the lightness is crucial for the plate recognition.

Fig.8 shows an example of the detected number plate and the extracted room numbers.

Table 4: Success rate of room number recognition

total	*success*	*failure*1	*failure*2	*success rate*
30	24	2	4	0.80



(a) Original plate image (b) Detected edges

(c) Plate regions extracted by feature filtering (d) Binarized plate region

Figure 8: Detected room number plate

6 Discussions

(1) Lessons Learned

Although the models used in the proposed system are rather simple, we have found that the models are useful in object recognition and position estimation. The followings are lessons learned which we obtained through the experiments.

Position estimation based on models

The geometric relationships between 3D models and the robot are useful to estimate the position of the target object. In our system, the door position is estimated using the relationships between the robot, the wall, and the door. The plate position is also estimated using the relative position from a door edge. A major problem here is how to select the reliable data. There are a lot of errors in the robot position, the camera orientation, and the recognition results. As mentioned above, errors in the door position could be very large, and so it is very risky to use the estimated data without any evaluation. The criteria to evaluate the reliablity of the estimated data are needed. A desirable scenario is that the robot has plural methods to achieve the goal, and that selects the best method based on the most reliable data in the situation.

Feature filtering based on models

Feature filtering is an important technique to enhance the efficiency and accuracy of object recognition. Filtering parameters such as the slope and length of a segment are obtained from the mapping between 3D models and 2D images. The error analysis, however, is also crucial here. The criteria to evaluate the reliablity of the estimated data would play an essential role.

Behavior Strategy

Even though the criteria to evaluate errors mentioned above are provided, the robot can fail its task because of various factors. It is crucial for the robot to select an alternative action to achieve the task when the previous action failed. Two aspects would be needed to consider. One is to develop the criteria to determine whether an action has succeeded or failed. The other is a programming framework with which programmers can implement robot's behavior strategies systematically.

(2) Related Work

There has been a lot of researches of recognizing corridors and doors [11, 10, 4, 2, 7]. In these researches, the recognized objects are used as a landmark for self-localization, and the objects are registered in a map in advance (manually or by learning). Our method uses the recognized objects to estimate destination points, and needs no maps made in advance. Rather, we can think the robot makes local maps based on object models temporally during navigation.

Room number recognition is utilized to detect a landmark in [9]. Room numbers are used as a landmark for the robot to obtain the absolute position in the building. The methods for plate recognition and character shape correction are different from ours. The method for detecting character regions whose positions cannot be predicted in advance is proposed in [5], which employs features such as spatial frequecy and grey-level contrast. This method would be useful also in our target task, for example, in order to detect a guideboard attached on a wall.

7 Conclusions

We have developed a robot system that achieves the task in which the robot navigates itself to the room designated by room number. The experiments of door recognition, plate recognition, and room number recognition show the feasibility of our method. Future work includes the improvement of the recognition accuracy, the increase of object models, and the development of the framework for behavior selection.

References

[1] Y. Ando, T. Tsubouchi, and S. Yuta, Development of Ultra-Wide-Angle Laser Range Sensor and Navigation of a Mobile Robot in a Corridor Environment, *Journal of Robotics and Mechatronics*, Vol. 11, No. 1, pp. 25-32, 1999.

[2] I. Horswill, The Polly System, In *Artificial Intelligence and Robile Robots*, pp. 125-139, AAAI Press/MIT Press, 1998.

[3] S. Iida and S. Yuta, Vehicle Command System and Trajectory Control for Autonomous Mobile Robots, *Proc. of IROS'91*, pp. 212-217, 1991.

[4] T. Kanbara, J. Miura, and Y. Shirai, Selection of Efficient Landmarks for an Autonomous Vehicle, *Proc. of IROS'93*, pp. 1332-1338, 1993.

[5] Y. Liu, T. Yamamura, T. Tanaka, and N. Ohnishi, Character-Based Mobile Robot Navigation, *Proc. of IROS'99*, pp.610-616, 1999.

[6] W. B. Mann and T. O. Binford, SUCCESSOR: Interpretation Overview and Constraint System, *Proc. of 1996 Image Understanding Workshop*, pp. 1505-1518, 1996.

[7] A. Ohya, A. Kosaka, and A. Kak, Vision-Based Navigation of Mobile Robot with Obstacle Avoidance by Single Camera Vision and Ultrasonic Sensing, *Proc. of IROS'97*, pp. 704- 711, 1997.

[8] S. Sarkar and K. L. Boyer, Integration, Inference, and Management of Spatial Information Using Bayesian Networks: Perceptual Organization, *IEEE Trans. PAMI*, Vol. 15, No. 3, pp. 256-274, (Mar. 1993).

[9] T. Tanaka, Y. Yamazaki, H. Watanabe, T. Katae, and K. Yamafuji, Self-Position Detection System Using a Visual-Sensor for Mobile Robots, *Journal of Robotics and Mechatronics*, Vol. 11, No. 1, pp. 17-24, 1999.

[10] T. Tsubouchi and S. Yuta, Map Assisted Vision System of Mobile Robots for Reckoning in a Building Environment, *Proc. of ICRA'87*, pp. 1978-1984, 1987.

[11] M. Yachida, T. Ichinose, and S. Tsuji, Model-Guided Monitoring of a Building Environment by a Mobile Robot, *Proc. of IJICAI'83*, pp. 1125-1127, 1983.

[12] S. Yuta, S. Suzuki, and S. Iida, Implementation of A Small Size Self- Contained Autonomous Robot - Sensor, Vehicle Control, and Description of Sensor Based Behavior -, R. Chatila et al. Eds, *Experimental Robotics II*, pp. 344-359, Springer-Verlag, 1993.

[13] S. Yuta, Experimental Research on the Autonomous Mobile Robot which Navigate in the Real Environment, *Proc. of the Second International Conference on Mechatronics and Machine Vision in Practice*, pp. 59-66, (Sep. 1995).