Autonomous Navigation for Mobile Robots Referring Pre-recorded Image Sequence

Takayuki OHNO   Akihisa OHYA   Shin'ichi YUTA

Intelligent Robot Laboratory
Institute of Information Sciences and Electronics
University of Tsukuba
Tsukuba, Ibaraki 305, JAPAN
phone: +81–298–53–5155   fax: +81–298–53–6471
email: yuta@is.tsukuba.ac.jp
url: http://www.roboken.esys.tsukuba.ac.jp/

Abstract

The purpose of this study is to realize autonomous navigation of a mobile robot with referring pre-recorded images, without any environment maps. The strategy of navigation in this research is that the differences of position and orientation of robot between reference and present one are continuously estimated from two images, and are used in the locomotion control commands. In this paper, we use only vertical lines in the image as extracted features, and propose the method to calculate the estimated differences of position and orientation from two images. We also show some results of simulations and experiments using an actual mobile robot, and examine its reliability.

1 Introduction

When a mobile robot navigates along a designated path autonomously, in many cases the robot follows the path by recognizing its position in the environment. In the case of a wheeled-type mobile robot, the robot can estimate its position to some extent with the odometry system. But the error of this position estimation is increased with the motion of the robot. In this case, the robot should adjust its position by using landmarks in the environment. Such navigation methods combining landmark information with an odometry system are very popular subjects in the field of mobile robotics [1][2]. Our group has also done similar research about in- and out-door navigation [3].

For navigation using landmarks, an environment map which describes positions of landmarks is needed. Such an environment map is commonly made by a human's actual measurement. It is easy to imagine that to make an environment map takes a lot of time and labor. Also, special landmarks, for example, laser beacons, should be set up if they are no natural objects.

In this paper, we propose an autonomous navigation method for a mobile robot in an indoor environment by obtaining information about the designated path in the form of an image sequence, without any environment maps nor any specially prepared landmarks.

As a navigation method for a mobile robot, Matsumoto et al. have described realtime navigation using memorized whole images of the front view and found correspondence with correlation [4]. However, this method needs much memory to memorize whole images and an expensive high-speed image processing system.

In our method, we use only vertical lines in the images as feature lines, estimate position and orientation from the difference between reference and current images, and achieve autonomous navigation. First, a human controls a robot and records images on videotape from the robot camera in teaching mode. While autonomously navigating, the differences of position and orientation between the recorded image and current robot camera image are calculated, and the locomotion system is controlled to decrease the differences. This method doesn't need to set up any artificial landmarks, and it is easy to record the reference images as path information.

In section 2, we explain the basic strategy of autonomous navigation, and propose the method to calculate the differences of position and orientation of the robot from two images in section 3. The availability of this method is examined with simulation results in section 4, and experimental results with an actual mobile robot are presented in section 5. Based on these results, we examine this navigation method in section 6. Finally the conclusions are presented in section 7.

2 Autonomous Navigation Using Pre-recorded Images

The strategy of this navigation method is shown in figure 1. First, an a teaching mode, a human controls a robot by radio or other interface and records images on videotape from the robot camera as reference images. During autonomous navigation, the locomotion system is controlled to decrease the difference between the recorded reference image and the current camera image.

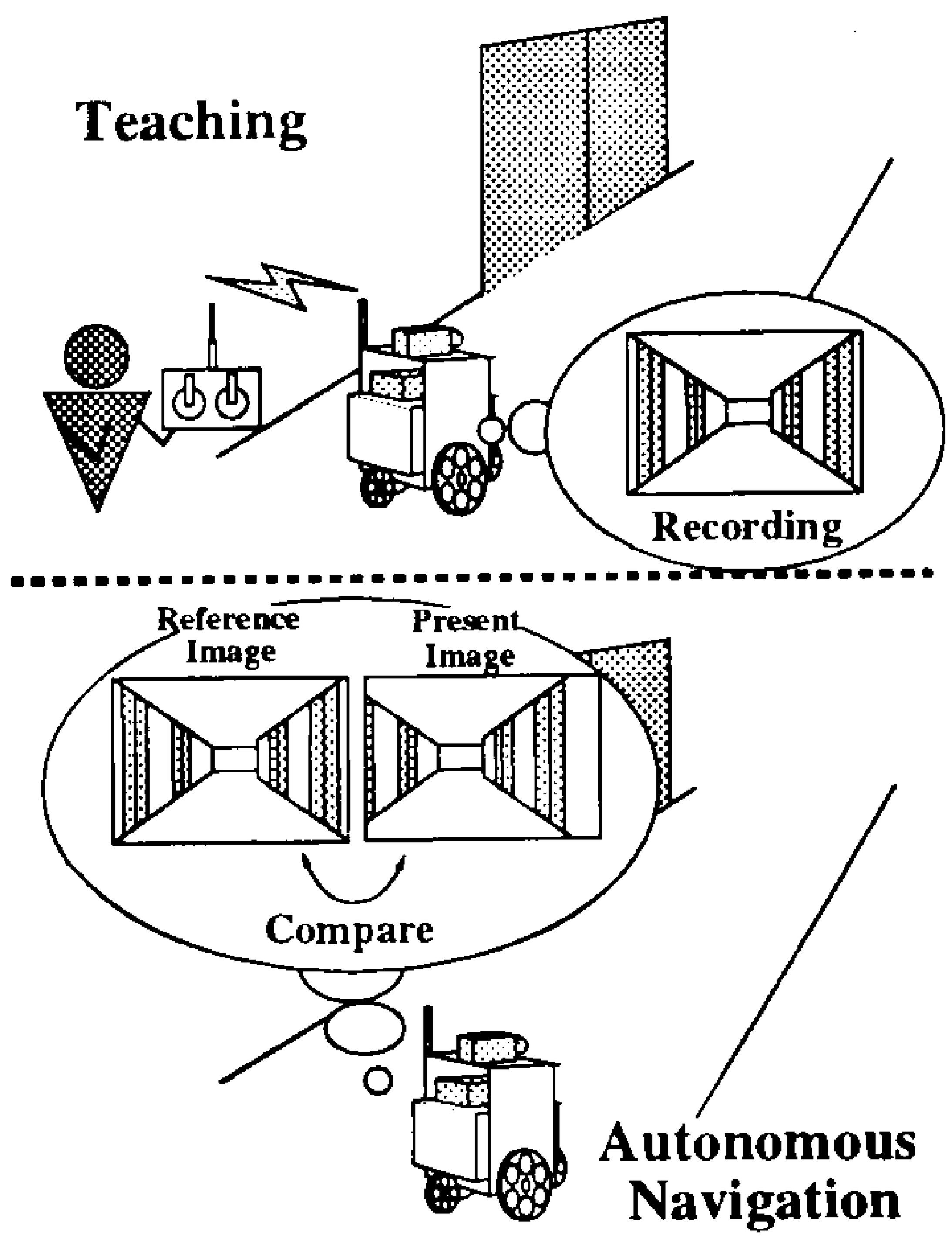


Figure 1: The basic strategy of this navigation method.

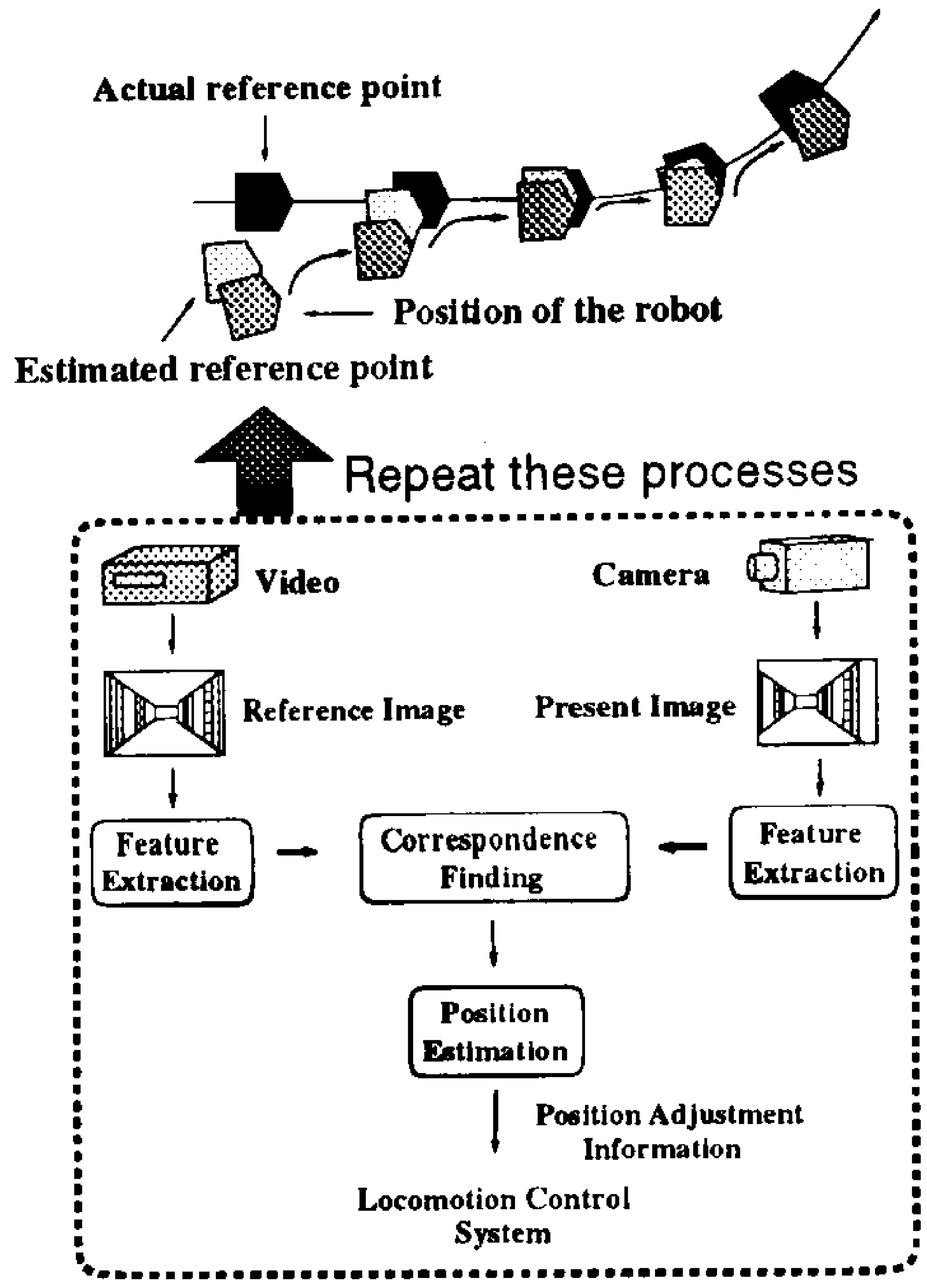


Figure 2: The algorithm of this navigation method.

As features in an image, we use vertical lines because there are many such features in a typical indoor environment, and it is easy to extract them. In order to extract them, the horizontal component of the derivative of an image is obtained in a region around vertical center, and points where the differential value exceeds a threshold level are found. It is easiest to find correspondence of feature lines between two images if there are not large differences between the two images.

The algorithm used in this navigation method is shown in figure 2. Vertical lines used as features are extracted in both a reference image from the videotape and the current camera image, and the correspondence between the features is found. Then, the differences of position and orientation of the robot are estimated using the positions of these feature lines, and these differences are used in the locomotion control system as the position adjustment information.

3 Position Estimation from Two Images

In this section, we explain the method used to estimate the differences of position and orientation of the robot by using only the position difference of feature lines between reference and current images. Because it is difficult to analyze exactly the relationship between the differences of robot position and the obtained image, we derived the method based on experimental results.

3.1 Correspondence Finding

To find the correspondence of feature lines between two images, we use the following rules [5].

- The signs of differential values at two corresponding feature lines are the same.
- The horizontal differential values at corresponding feature lines are similar.
- The intervals between adjacent feature lines are similar in two images.
- The order of correspondence never crosses.
- Some feature lines which have no correspondence can exist.

Based on these rules, after the extraction of the vertical lines by image differentiation for horizontal component, the corresponding relationship of feature lines of both images is searched to minimize the evaluation function M as follows.

M = (penalty value) × (number of feature lines in images which have no correspondence)
+ $\sum$ (absolute difference value of intensities of horizontal differentiated image between corresponding feature lines of both images)
+ $\sum$ (absolute difference value of position of the feature lines between corresponding feature lines of both images).

Using these rules and the evaluation function, the correspondence of feature lines between the two images can usually be found easily when there is not a large difference between the two images.

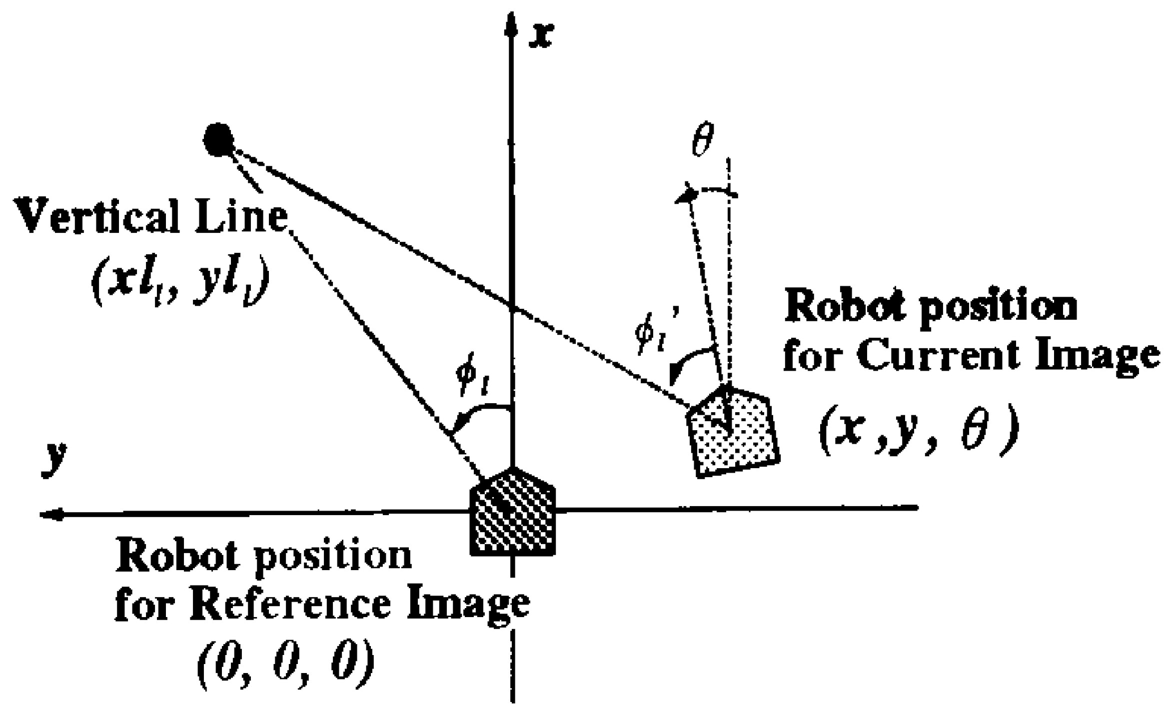


Figure 3: Location of a vertical line and robots for a reference and a current image. The origin is set at the position of the robot for the reference image. Let the angle of the vertical line for the reference image be ϕ_l, and the angle for the current image be ϕ_l'

3.2 Relation between the Location of the Robot and the Position of the Feature Lines

Figure 3 shows the location of a vertical line in the robot coordinate system and the robot's position for a reference and a current images. The origin is set at the position of the robot for the reference image. ϕ_l denotes the direction of the vertical line for the reference image, and ϕ_l' denotes the direction for the current image. $\Delta\phi_l$ is defined as

$$\Delta\phi_l = \phi_l - \phi_l' \quad . \tag{1}$$

The differences of position and orientation (x, y, θ) may be estimated using the relation between $\Delta\phi$ and ϕ which is obtained from many vertical lines. So, we examined the relation of ϕ and $\Delta\phi$ to the differences of position and orientation of robot. Figure 4 shows the case that only the orientation θ is different, figure 5 shows the case that only the position x is different, figure 6 shows the case that only the position y is different, and figure 7 shows the case that both x and y are different.

From these graphs, we observed the following relationships. Figure 4 shows that the difference of orientations θ just leads to a translation in the $\Delta\phi$–axis, so θ can be estimated from the distance of its translation. Figure 5 shows that the graph can be approximated by an inclined straight line roughly when there is a difference in position x, so x can be estimated from its slope. Figure 6 shows that the difference of positions y makes a graph formed like the letter "V", so y can be estimated from the extent of its opening.

The most likely case is that the differences of both x and y occur as in figure 7, but figure 7 can be considered as the sum of figure 5 and 6, so it is possible to estimate x and y to some extent. In the case that differences among all positions and orientations exist, it can be considered the same as the case that the difference of both x and y exist because the influence of θ can be easily removed by the translation in the $\Delta\phi$–axis.

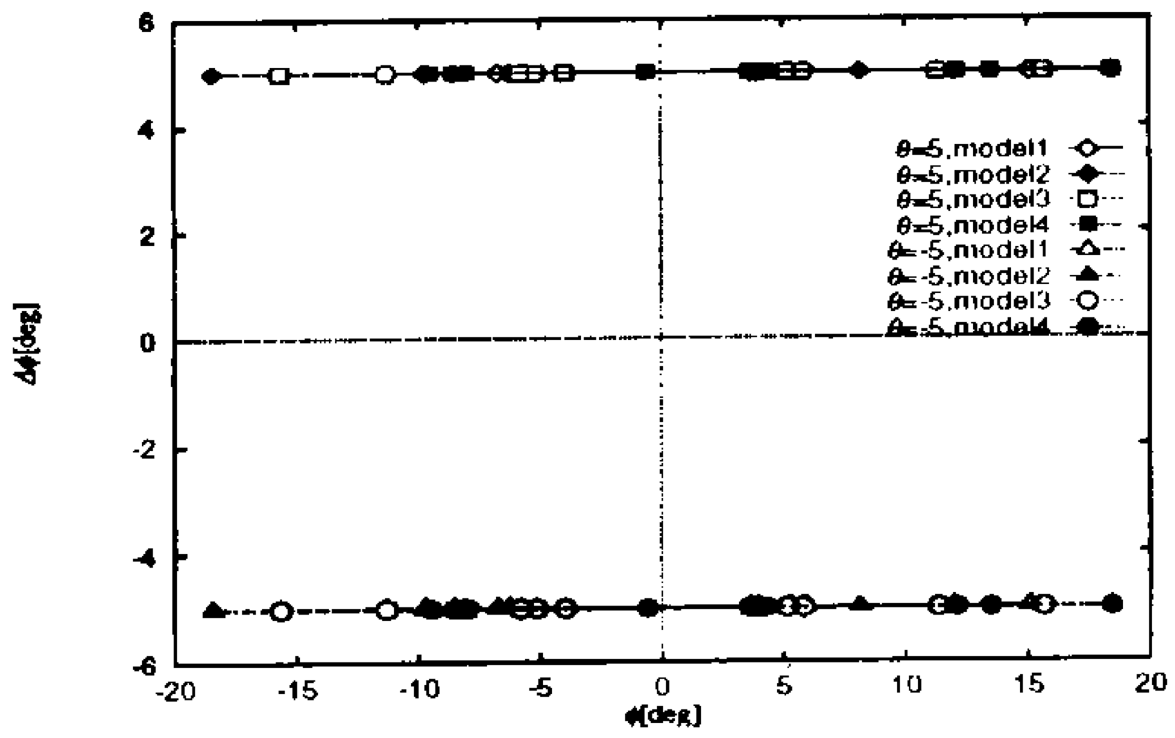


Figure 4: Graph of $\Delta\phi$ vs. ϕ. Cases where Only the orientation is different.

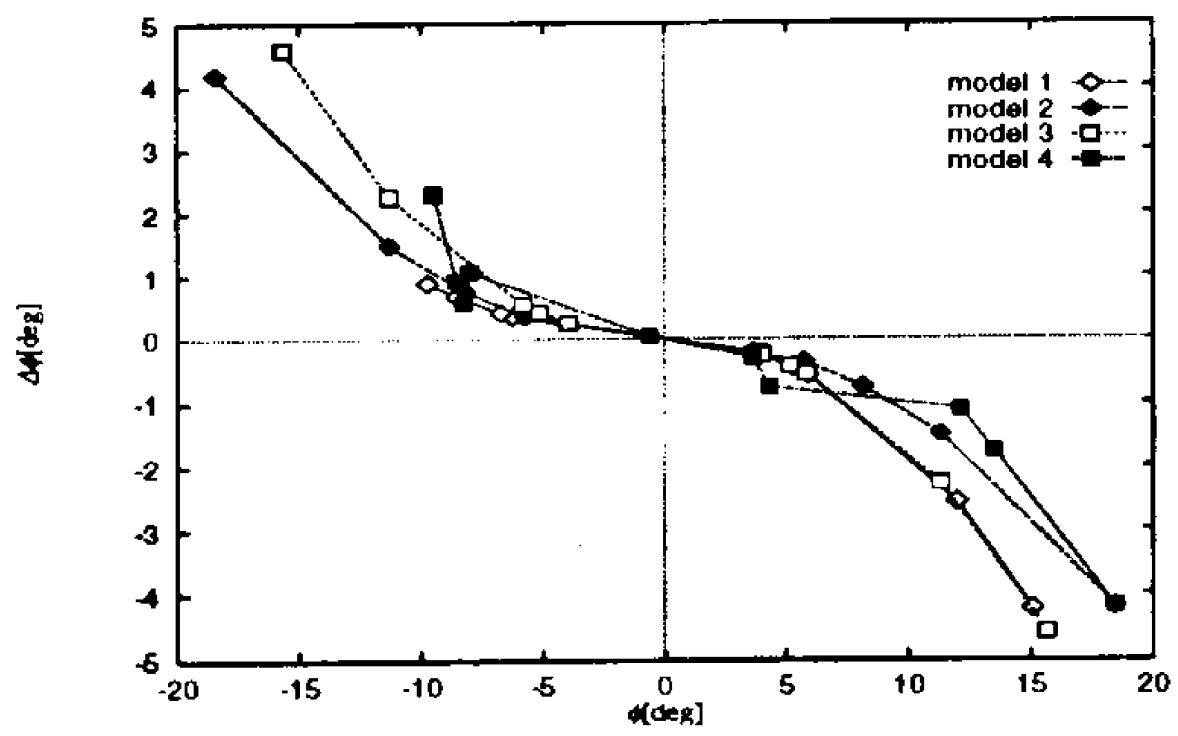


Figure 5: Graph of $\Delta\phi$ vs. ϕ. Only the position x is different.

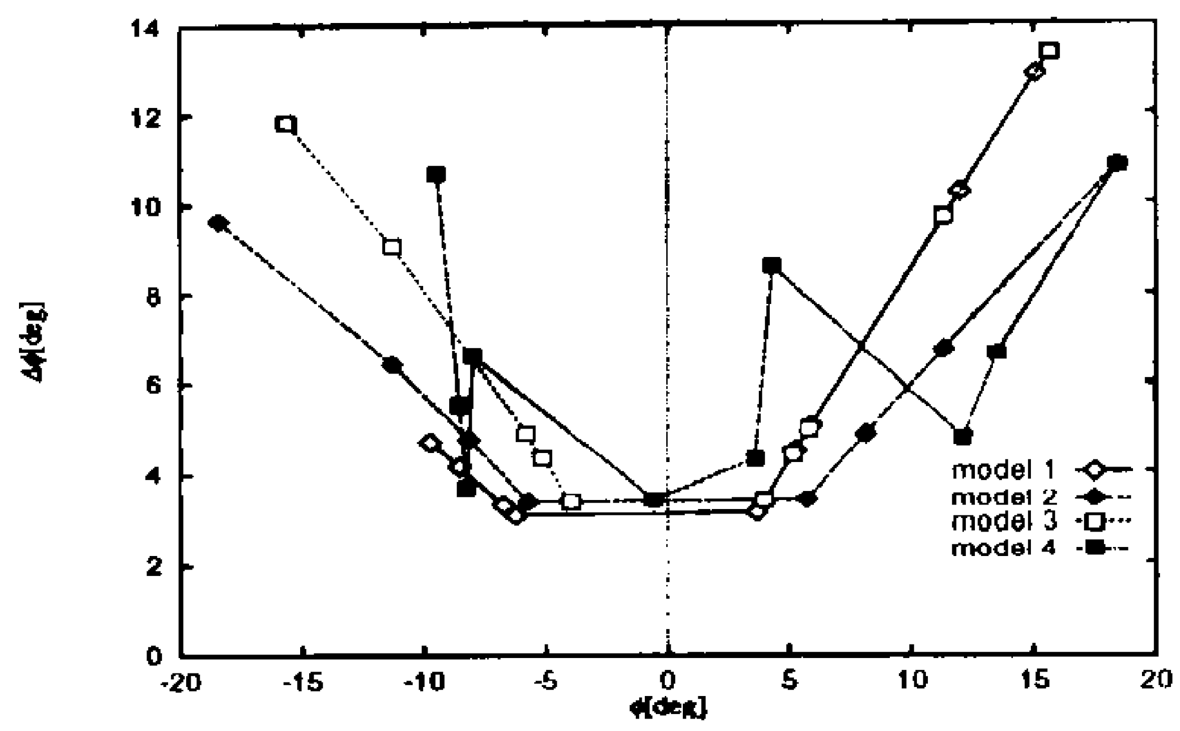


Figure 6: Graph of $\Delta\phi$ vs. ϕ. Only the position y is different.

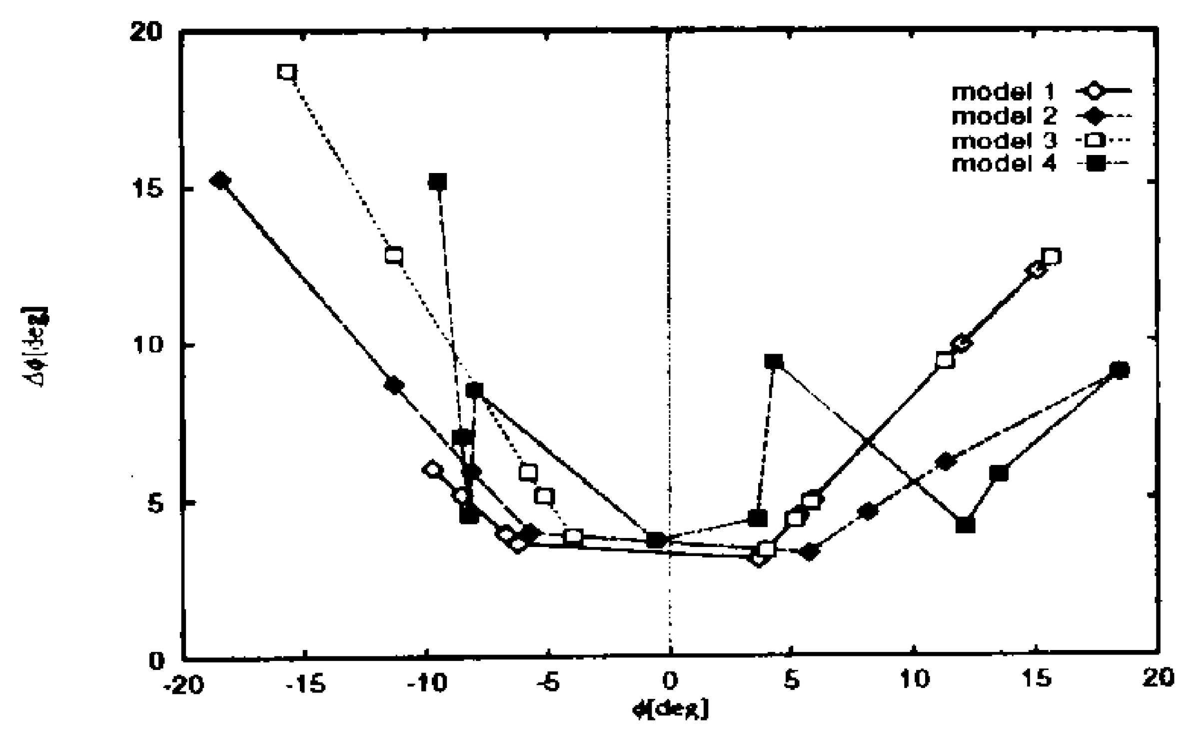


Figure 7: Graph of $\Delta\phi$ vs. ϕ. Both the positions x and y are different.

3.3 Algorithm of Position Estimation

Based on the relations described in the preceding paragraph, we developed the following algorithm to estimate the differences of position and orientation.

1. Calculate ϕ and $\Delta\phi$ for each corresponding pair of vertical lines.

2. The graph of $\Delta\phi$ vs. ϕ is fit to the straight lines using a least squares method in the regions of $\phi > 0$, $\phi < 0$ and the whole ϕ, respectively. Let these slopes be a_+, a_- and a_{all}, and their values when $\phi = 0$ be b_+, b_- and b_{all}, respectively.

3. If the signs of a_+ and a_- are

   (a) the same, it can be considered that there is no difference of position y like figure 5, so the position difference is estimated as follows (See figure 8).

$$\begin{aligned} x &= a_{all} \times CONV\_X. && (2) \\ y &= 0. && (3) \\ \theta &= b_{all}. && (4) \end{aligned}$$

   (b) different, the graph is shaped like the letter "V", and it can be considered that there is a difference of position y as in figure 6, so the position difference is estimated as follows (See figure 9).

$$\begin{aligned} x &= \frac{a_+ + a_-}{2} \times conv\_x(|y|). && (5) \\ y &= \frac{|a_+| + |a_-|}{2} \times CONV\_Y \\ &\times \ (\text{sign of } a_+). && (6) \\ \theta &= \frac{b_+ + b_-}{2}. && (7) \end{aligned}$$

where $CONV\_X$, $CONV\_Y$ and $conv\_x(y)$ are the parameters to calculate x or y from the slope. $CONV\_X$ and $CONV\_Y$ are constants, and $conv\_x(y)$ is a function of y.

We calculated the differences of position and orientation with this algorithm for many patterns of line arrangements and actual images. As a result, the difference of orientation was able to be calculated with high accuracy, but the accuracy of calculating the difference of position was not high. However, if the estimation of the position difference is almost correct, the robot can navigate on the expected path. So this degree of accuracy is satisfactory.

4 Simulation

To verify the accuracy and the availability of this position estimation method, we made some computer simulations on a workstation.

To simplify the problem, let us consider that there are reference points at regular intervals in front of the robot. The robot estimates its position at a reference point on the robot's coordinate system, adjusts it, and moves to the next reference point. Because

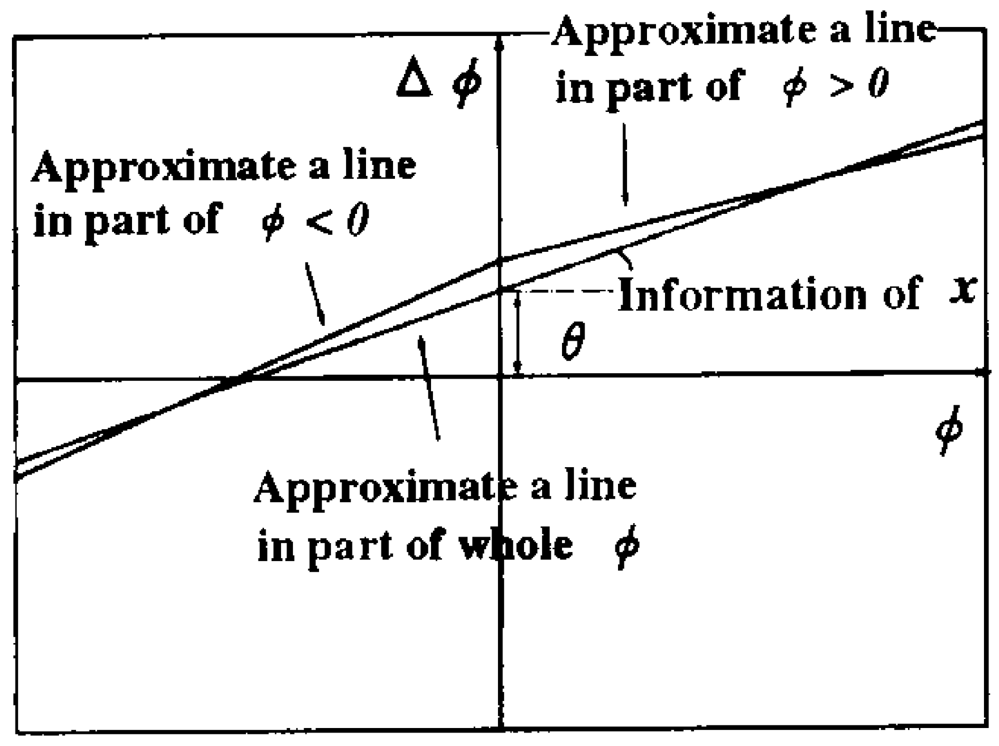


Figure 8: The case that the slopes of the graph of $\Delta\phi$ vs. ϕ have same signs. x is estimated from the slope, and θ is estimated from $\Delta\phi$ ($\phi = 0$).

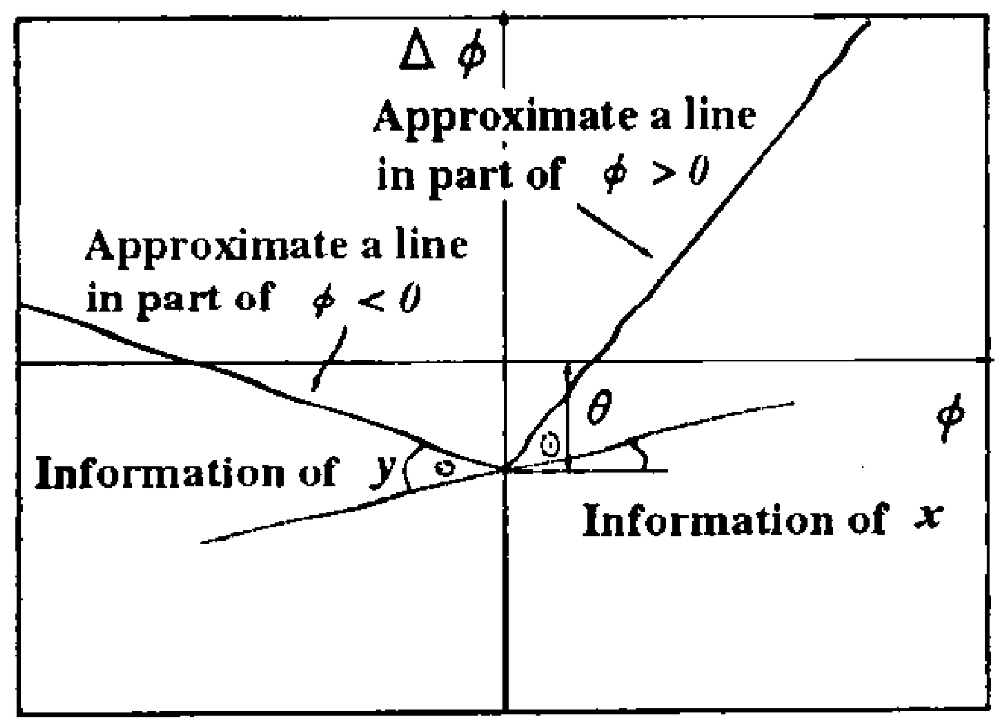


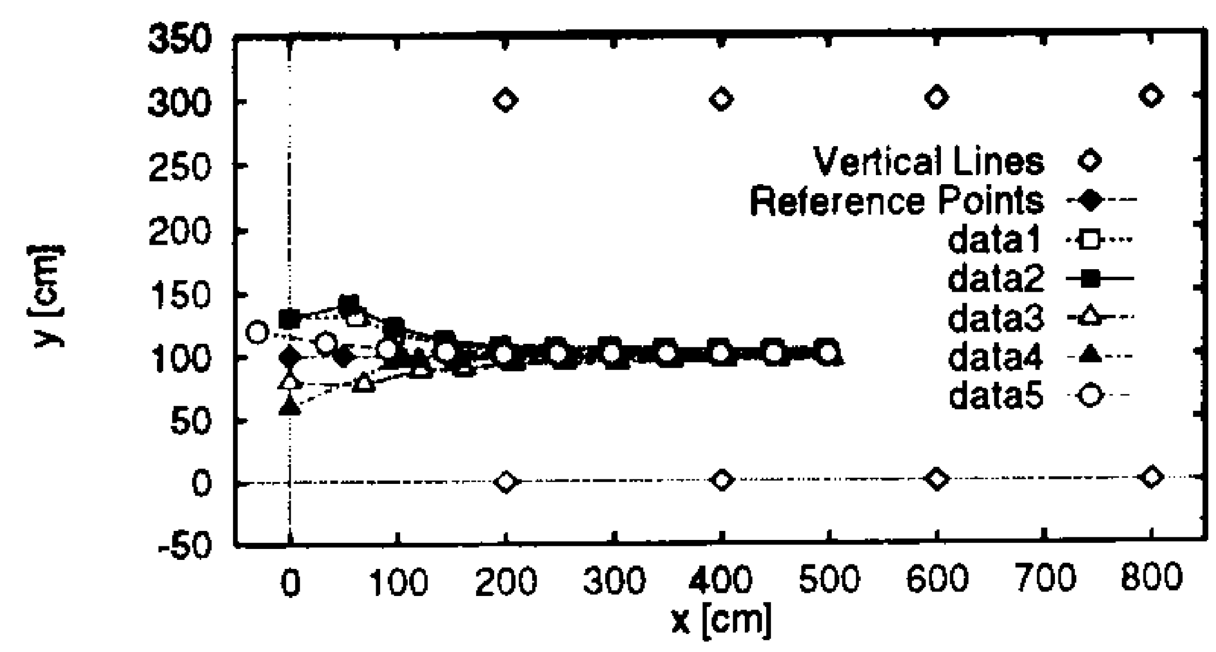
Figure 9: The case that the slopes of the graph of $\Delta\phi$ vs. ϕ have different signs. y is estimated from the extent of opening, x is estimated from the inclination of the whole graph, and θ is estimated from $\Delta\phi$ ($\phi = 0$).

the position estimation at this time is not exact, the estimated robot's position is not always equal to the actual robot's position. But by repeating this process, the trajectory of the robot converges to the line of the reference points.

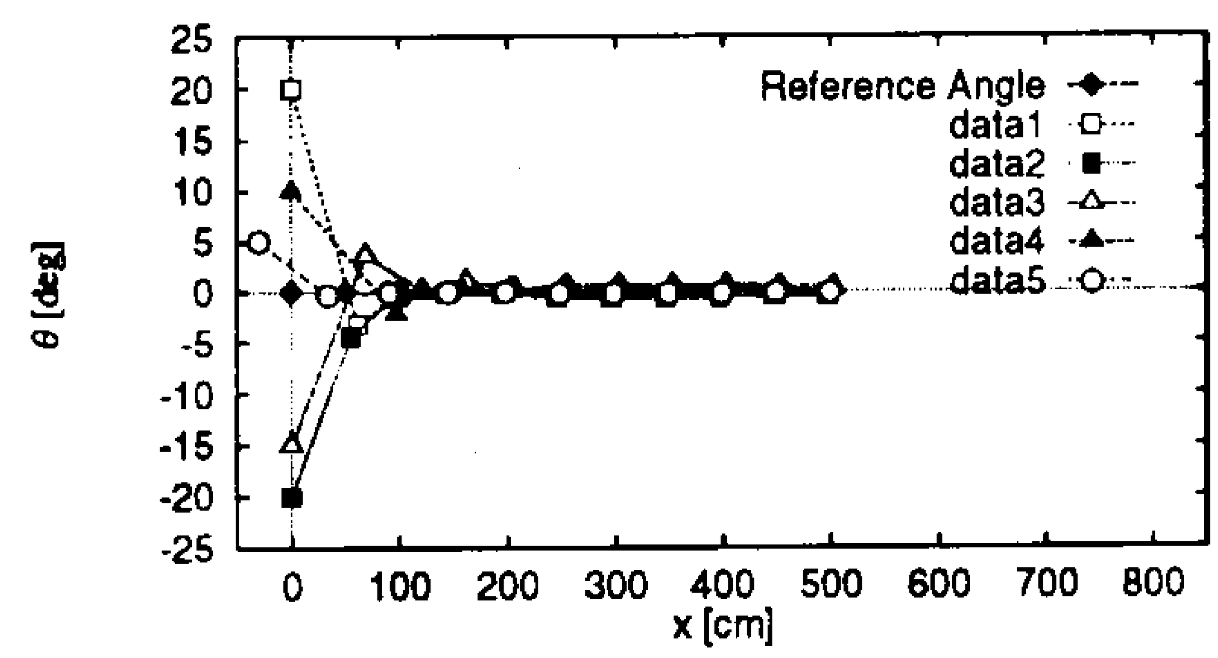
In these simulations, the interval of the reference points is set as 50cm, the field of view of camera is set as ±25 degrees, and the vertical lines are arranged like a corridor. Also, it is considered that exact correspondence between the feature lines can be found.

First, we simulated a case where reference points lie on a straight line, and a robot starts at various points. Simulation results are shown in figure 10. The mark ◇ expresses the position of vertical lines, and reference points lie on the line y = 100cm. As shown in these graphs, the position and orientation of the robot approached that of the reference point.

Also, we made the simulation in the case including turn. In this case, reference points lie on an arc. In this simulation, the difference of the robot's orientation between adjacent reference points on the arc is set as 18 degrees. Results are shown in figure 11. First, the reference points lie on a straight line, next on an

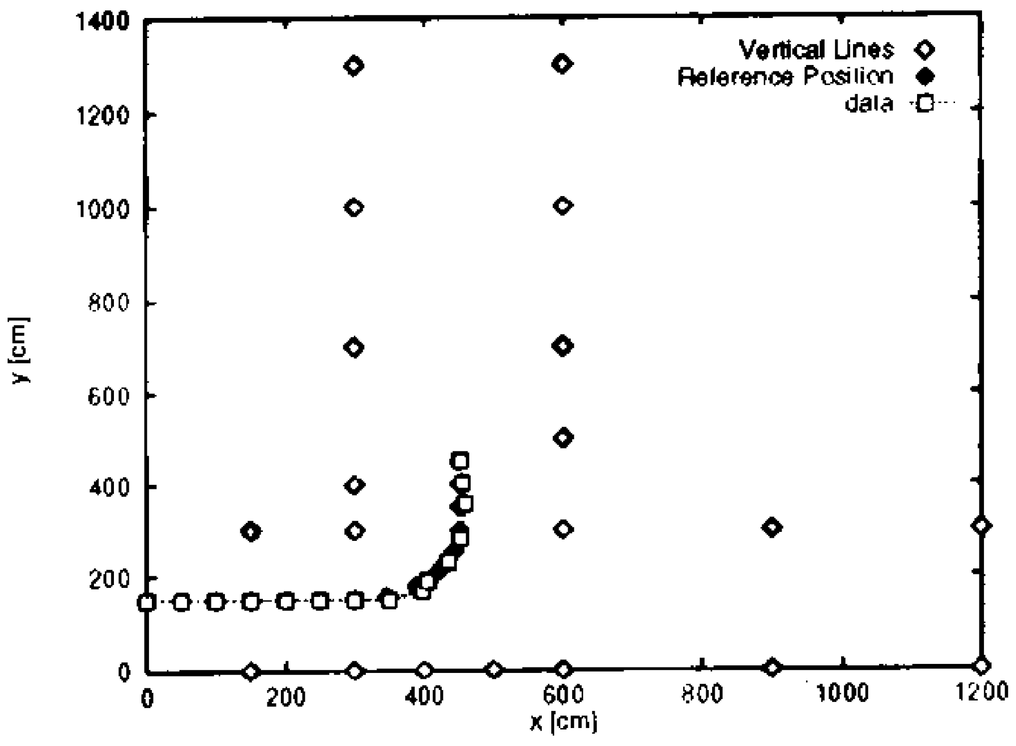


(a) Graph of x vs. y.

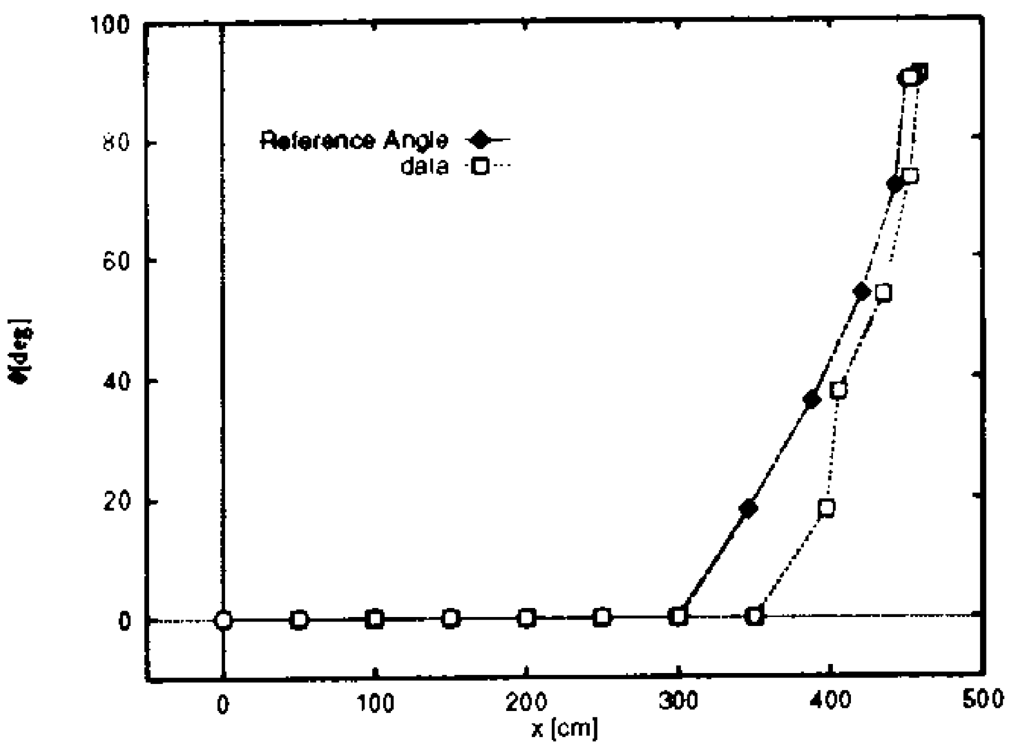


(b) Graph of x vs. θ.

Figure 10: Simulation result in the case that reference point lie on a straight line ($y = 100, \theta = 0$). Graph (a) shows the location, and (b) shows the orientaion at that time. The mark ◇ expresses vertical lines. The robot starts at various points.



(a) Graph of x vs. y.



(b) Graph of x vs. θ.

Figure 11: Simulation result in the case including a turn. Graph (a) shows the location, and (b) shows the orientation at that time.The mark ◇ expresses vertical lines, ◆ expresses reference points, and □ expresses the trajectory of the robot.

arc to turn left, and on another straight line again. The robot started at the same position as the reference. As shown in these graphs, the robot was able to follow the reference points laid on the arc.

It is found from these results that the accuracy of the position estimation is satisfactory, also it is possible to navigate the robot with this method.

5 Experiments with an Actual Autonomous Mobile Robot

We made some experiments with an actual autonomous mobile robot using this navigation method. As an autonomous mobile robot, we used the "Yamabico" (Figure 12) [6] which we have researched and developed. The Yamabico has a vision and a locomotion control system. The vision system can grab an image of 256 × 240 pixels. We use one CCD camera whose field of view is ±25 degrees.

Because of hardware restriction, we didn't use video images as reference. In this simulation, the pairs of horizontal position and the differential values of the vertical lines are only stored in the memory as the data of a reference image. The interval of the reference points is set as the same 50cm as in the simulation.

We made two kind of experiments: the case that the robot stops while estimating position and the case that the robot estimates the position with moving.

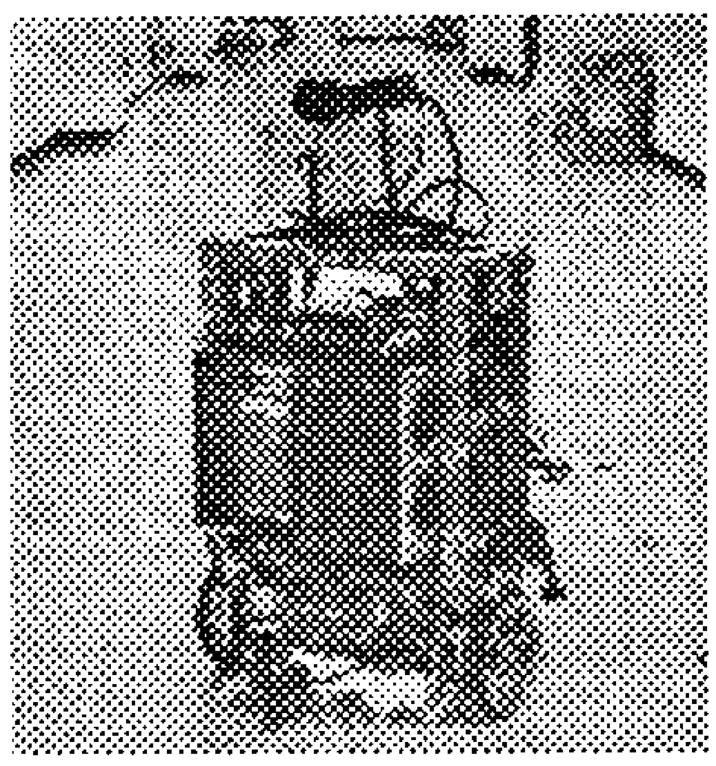
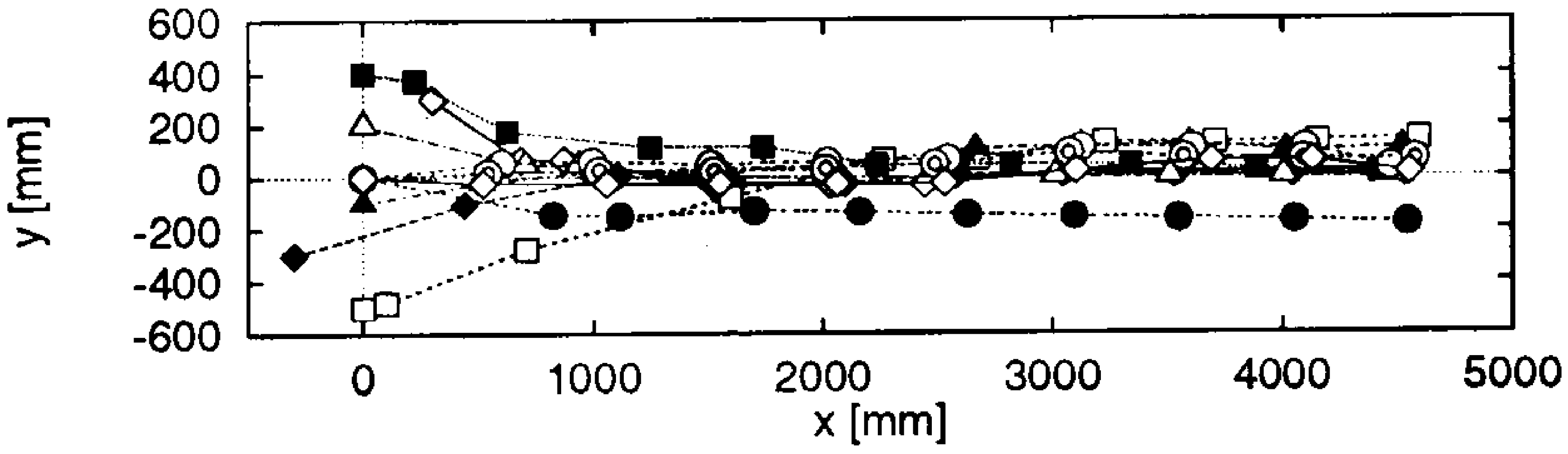


Figure 12: Autonomous mobile robot "Yamabico".

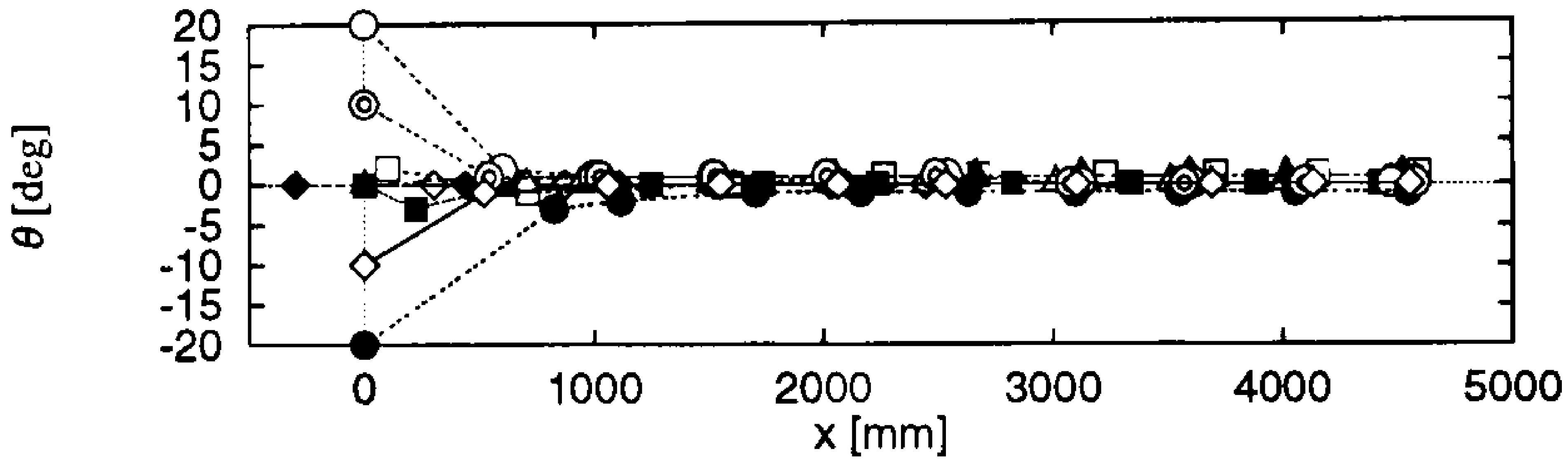
5.1 Position Estimation with Stopping

The robot estimated its position at a reference point on the coordinate system, and moved to the next reference point. The robot repeated this process.

First, we made experiments in the case that 10 reference points lay on a straight line. The photograph of the experimental environment is shown in figure 13. The results for the robot starting at various positions are shown in figure 14. The reference points are not shown in the graphs, but they lie on the axis of x. As



(a) Graph of x vs. y.



(b) Graph of x vs. θ.

Figure 14: Experimental result. Reference points lie on the straight line (x–axis). Graph (a) shows the location, and (b) shows the orientation at that time. The robot started at various positions.

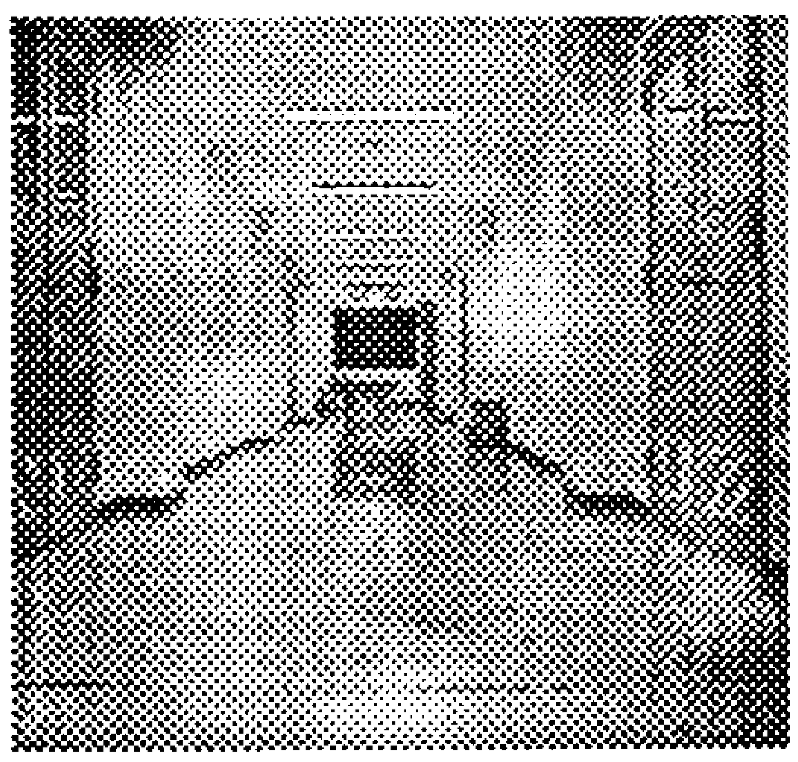


Figure 13: Experimental environment.

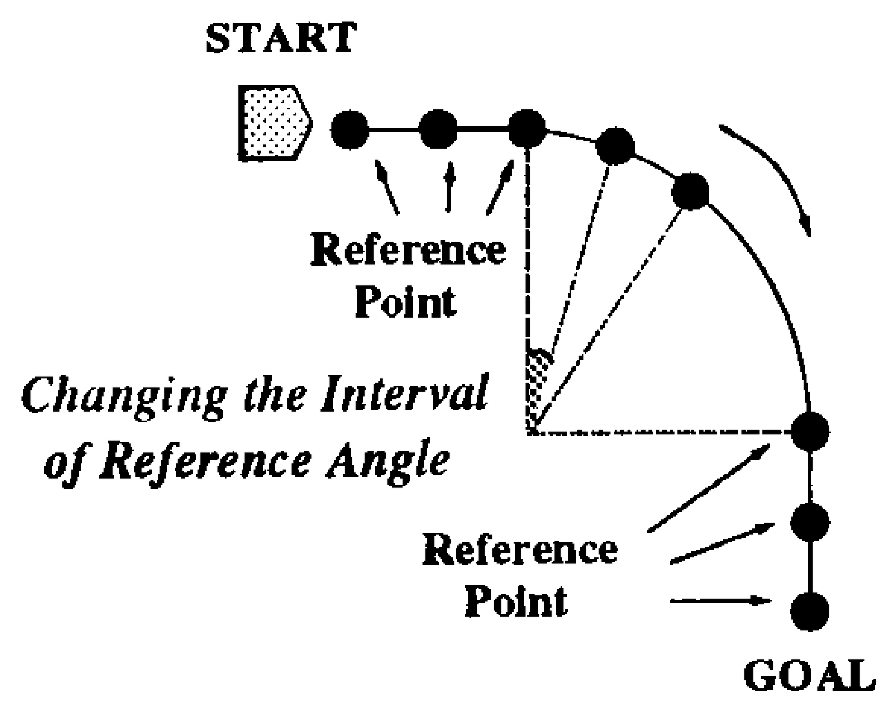


Figure 15: Outline of the path in an experiment.

shown in these graphs, the trajectories of the robot approached the sequence of the reference points.

Secondly, we made experiments in the case that reference points lay on an arc. As shown in Figure 15, the robot went straight, turned 90 degrees on the arc, and went straight again. We changed the difference of the robot's orientation between adjacent reference points on the arc. In autonomous navigation, the robot started at the same point as in teaching.

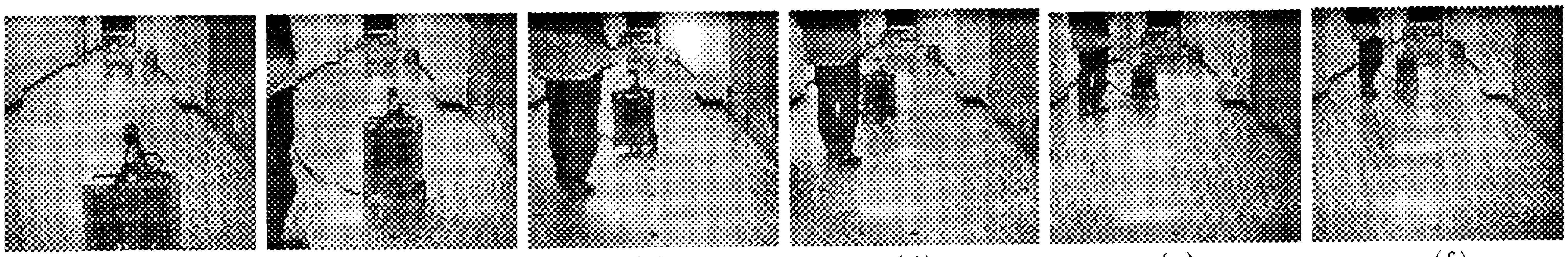
Figure 16 shows the results when the difference of the robot's orientation between adjacent reference points is 8 degrees. In this experiment, we made 4 trials and the robot was able to turn the curve in 3 cases.

In the case that the robot couldn't turn the curve, the position estimation failed because the correspondence between the feature lines could not be found. In this experiment, the robot couldn't turn the curve when the interval of the reference angle was set to more than 8 degree. In most of the failed cases, the robot failed to find the correspondence around 30 degrees in the curve. It can be considered that it was difficult to find correspondence because the images changed greatly around 30 degrees in this curve.

5.2 Position Estimation with Moving

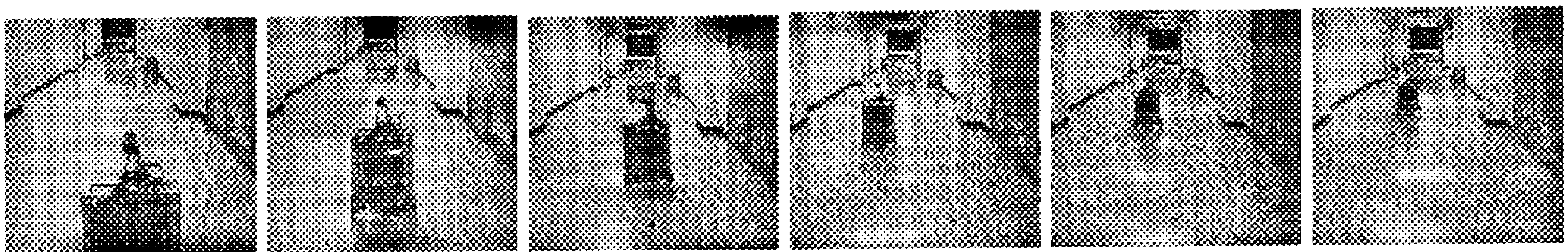
In contrast to the preceding paragraph, in this experiment, the robot took an image, estimated and adjusted its position while moving. If the robot recognized that it was already ahead of the next reference point, the robot stopped at that point and did the next estimation.

In teaching to record reference images, the robot was driven arbitrarily by a manual controller. Photographs from teaching are shown in figure 17, and ones during autonomous navigation are shown in fig-



(a) (b) (c) (d) (e) (f)

Figure 17: Teaching stage. A human controlled the robot with a manual controller. The robot moved ahead, turned left, and moved ahead again.



(a) (b) (c) (d) (e) (f)

Figure 18: Autonomous navigation stage. The robot moved ahead, turned left, and moved ahead again, similar to teaching.

ure 18. At this time, the speed of robot was 15 cm/s.

As seen in these figures, we were able to achieve an autonomous navigation even in the case with moving.

6 Discussions

Based on the preceding section, we examine some problems of this navigation method.

6.1 Applicable Environments

To apply this navigation method, a certain number of vertical lines have to be measured. Especially, several vertical lines must be seen from the robot if the robot turns at the corner.

6.2 Correspondence Finding

This position estimation method assumes that correspondence finding is exact. Since it was assumed that the correspondence was found correctly in the simulation, the robot could turn a curve when the difference of the robot's orientation between adjacent reference points was 18 degrees. However, in practice mis-correspondence will sometimes occur. Because it is easy to make a mistake in finding the correspondence when the reference image is very different from the current camera image, the robot could barely turn when the difference of robot's orientation is 8 degrees in the real environment.

To decrease the mis-correspondence, the interval of reference points must be shortened. But this is related to the processing time described in the following paragraph.

6.3 Processing Time and Interval of Reference Points

In these experiments, the image processing always takes 0.25 second, but the time for the correspondence finding becomes longer when the difference between images is large; less than 1 second in a short case, a few seconds in a long case. In the case of taking a long time for the correspondence, the robot may already go past the next reference point when finishing the position estimation if the robot doesn't stop during calculation. Thus, the interval between reference points should be set large enough relative to both the processing time and the speed of the robot in order to finish the processing before reaching the next reference point.

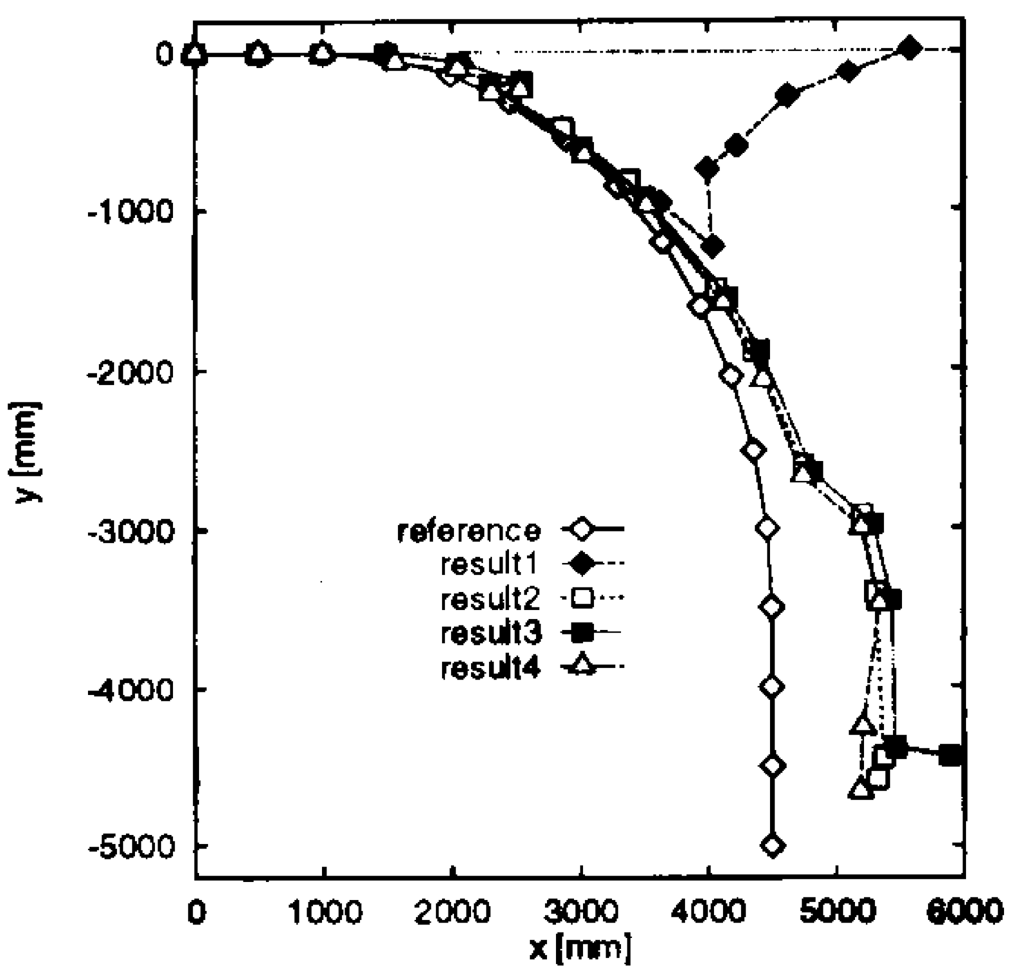
However, because the difference of the robot's orientation between reference points cannot be large as mentioned in the previous section, the radius of the curve cannot help being large when the interval of reference points becomes large. In an indoor environment like a corridor, the spatial upper limit of the radius is determined, so it is difficult to make the interval of reference points large.

In summary, if the processing time can be shortened, the interval of distance and difference of robot's orientation between reference points can be shortened, so the reliability of this method will be improved.

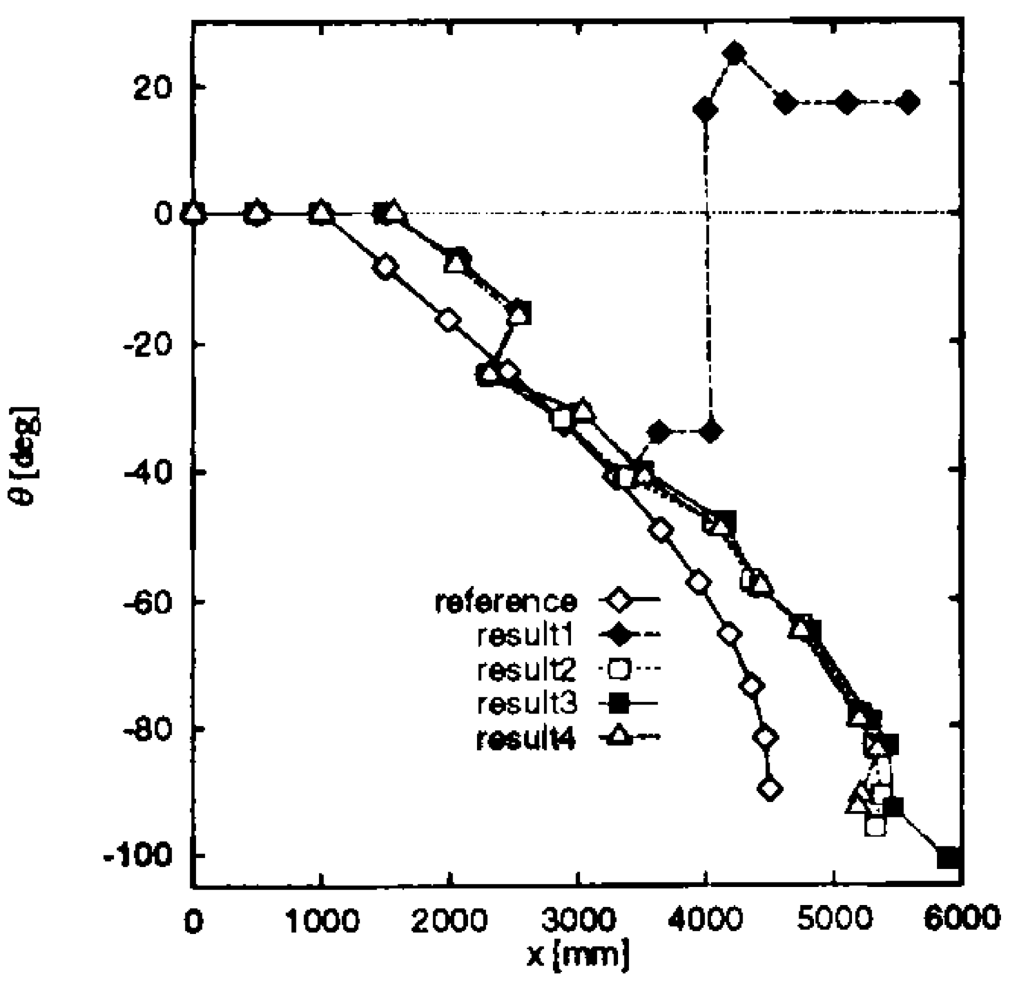
7 Conclusions

We studied an approach to autonomous navigation using pre-recorded images.

In this method, first, a robot records camera images while moving along a designated path. During autonomous navigation, the robot calculates the differences of position and orientation between the recorded image and a current camera image, and adjusts its own position. We used vertical lines as features. It was easy to find correspondence between the feature lines if there was little difference between images. The differences of position and orientation of the robot were



(a) Graph of x vs. y.



(b) Graph of x vs. θ.

Figure 16: Experimental result in the case that reference points lie on an arc. The difference of the robot's orientation between adjacent reference points is 8 degrees. Graph (a) shows the location, and (b) shows the orientation at that time. The mark ◇ expresses the reference points, and other marks express the trajectories of the robot.

estimated using the difference of position of feature lines between the two images.

We made some computer simulations for this estimation method. As a result, we could verify that the accuracy of the position estimation is satisfactory for navigation.

We also carried out several experiments with an actual mobile robot. In these experiments, with stopping at each reference point, the robot can follow not only a straight path but also a curve. And also the robot can estimate the position without stopping at each reference point.

We also examined the applicable environments, the mis-correspondence between feature lines, the processing time and the interval between reference points.

The correspondence finding problem between prerecorded and current images is one of the most important subjects required to achieve navigation by this method. Precision trajectory following will make it easier to find an exact correspondence of feature lines of both images because the difference of these images will be small. So, we have to consider fast image processing hardware or an algorithm to shorten the cycle time of feedback control.

We didn't use video images as reference because of hardware restriction in our experiments. So, in future work, we will use faster hardware which makes the intervals between reference points short, mount a VTR on the robot and will use video images as the reference images. At this time, the speed control is required to match the speed of the video images.

Acknowledgements

We express our gratitude to the members of the Intelligent Robot Laboratory of the University of Tsukuba for their support of this research, and specially Mr. T. Kurosu of the Maeda Construction Corp. for his contribution to the first stage of this research. We also thank very much to Prof. A. C. Sanderson of the Rensselaer Polytechnic Institute for his useful suggestion in the preparation of this paper.

References

[1] J. J. Leonard and H. F. Durrant-whyte. *Directed Sonar Sensing for Mobile Robot Navigation.* Kluwer Academic Publishers, 1992.

[2] K. Hyppä. Optical navigation system using passive, indentical beacons. *Proc. of the Intelligent Autonomous Systems*, pp. 737–742, 1986.

[3] S. Maeyama, A. Ohya, and S. Yuta. Positioning by Tree detection sensor and Dead reckoning for Outdoor Navigation of a Mobile Robot. *Proc. of the IEEE Int. Conf. on Multisensor Fusion and Integration for Intelligent systems*, pp. 653–660, 1994.

[4] Y. Matsumoto, M. Inaba, and H. Inoue. Navigation based on memorization of scene image sequences — position identification and obstacle detection in corridor environment —. *Proc. of the Annual Conf. of Robotics Society of Japan*, pp. 313–314, 1995 (in Japanese).

[5] T. Kurosu, A. Ohya and S. Yuta. Positioning and Guidance of Mobile Robot by Real-time Vision. *Proc. of the JSME Annual Conf. on Robotics and Mechatronics*, pp. 1178–1183, 1994 (in Japanese).

[6] S. Yuta, S. Suzuki, and S. Iida. Implementation of a small size experimental self-contained autonomous robot—sensors, vehicle control, and description of sensor based on behavior—. R. Chatila et al. Eds, *Experimental Robotics II*, pp. 344–359. Springar-Verlag, 1993.