

# Autonomous Visual Navigation of a Mobile Robot Using a Human-Guided Experience

Kiyosumi Kidono, Jun Miura<sup>1</sup>, and Yoshiaki Shirai

*Dept. of Computer-Controlled Mechanical Systems, Osaka University, Suita, Osaka  
565-0871, Japan*

---

## Abstract

Information on the surrounding environment is necessary for a robot to move autonomously. Many previous robots use a given map and landmarks. Making such a map is, however, a tedious work for the user. Therefore this paper proposes a navigation strategy which requires minimum user assistance. In this strategy, the user first guides a mobile robot to a destination by remote control. During this movement, the robot observes the surrounding environment to make a map. Once the map is generated, the robot computes and follows the shortest path to the destination autonomously. To realize this navigation strategy, we develop: (1) a method of map generation by integrating multiple observation results considering the uncertainties in observation and motion, (2) a fast robot localization method which does not use explicit feature correspondence, and (3) a method of selecting effective viewing directions using the history of observation during the guided movement. Experimental results using a real robot show the feasibility of the proposed strategy.

*Key words:* Vision-guided mobile robot, Teaching by guidance, Stereo vision, Observation planning

---

<sup>1</sup> Corresponding author. E-mail: jun@mech.eng.osaka-u.ac.jp, Tel: +81-6-6879-7332, Fax: +81-6-6879-4185.

# 1 Introduction

Recently, many studies have been conducted on autonomous navigation of mobile robots. Information on the surrounding environment is necessary for a robot to move autonomously. There are navigation methods [1,4,7] which use a given map and landmarks. It is, however, a tedious work for the user to make a map and to specify landmarks. Therefore we propose a navigation strategy which requires minimum user assistance. In our strategy, the robot operates in two phases: map generation and autonomous navigation. In the map generation phase, the user guides a mobile robot to a destination. During this movement, the robot observes the surrounding environment to make a map. Once the map is generated, the robot computes and follows the shortest path to the destination autonomously.

Kanbara et al. [2] proposed a similar two-phase method for an indoor mobile robot; they considered only one dimensional movement in a linear corridor. Maeyama et al. [5] also applied a similar approach to a two-dimensional outdoor robot navigation. In their method, the user selects as landmarks specific objects, such as trees and bushes, and provides the robot with recognition programs for such objects in advance; other objects cannot be used as landmarks. In addition, since only selected landmarks are recorded in the map, the robot should follow the guided path in the autonomous navigation phase. Matsumoto et al. [6] proposed to use the sequence of images which is obtained during a human-guided movement for navigating the robot. Their robot also has to follow the guided path in the autonomous navigation phase. In addition, such an image-based method is easily affected by the change of lighting condition.

There are many works on automatic map generation (e.g., [8,10,11]). By combining these methods with the results of the work on autonomous navigation with a given map, the above-mentioned two-phase approach may be realized. In our method, however, the robot records in the map not only feature positions but also the robot positions where such features were observed (i.e., viewpoints), and the information on viewpoints is used for selecting viewing directions in the autonomous navigation phase. Such information has not been considered in previous re-

searches on map generation, and in order to utilize the information, it is desired to develop the map generation and map-based navigation methods together.

The outline of the proposed method is as follows. During a human-guided, non-stop movement, the robot repeatedly observes the environment with stereo vision and generates a map. The map is incrementally generated by integrating each observation result into the latest map. Since the robot movement includes uncertainty, the robot position is estimated before integration by matching the latest map and the new observation. Once the map is generated, the robot can move autonomously from the start point to the destination. Since the guided path may not be optimal, the robot calculates a minimum-distance safe path and follows it. During autonomous movement, the robot also estimates its position as in the case of the map generation. Since better position estimation generally realizes more efficient robot navigation, the robot selects the viewing direction at each viewpoint so that the localization error is minimized.

The proposed method is developed for static environments; that is, the environment in the autonomous navigation phase is assumed to be the same as that in the map generation phase; we also suppose that if unknown obstacles appear during movement, the robot detects them using other sensors such as sonar.

## 2 Map Generation

### 2.1 Stereo Vision

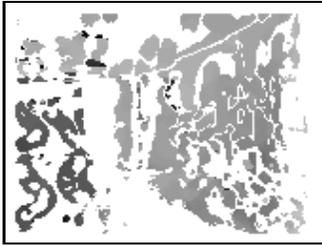
We use a stereo vision to obtain range data of the surroundings. The left and the right cameras are set on the mobile head and their optical axes are horizontal and in parallel with each other. For each feature point (edge with a high contrast) in the right image, the matched feature point is searched for along the horizontal epipolar line. The degree of matching is evaluated by the sum of absolute difference (SAD) of the intensity values in  $5 \times 5$  window,  $W$ , around a feature point:

$$\sum_{(i,j) \in W} |f_L(x+i+d, y+j) - f_R(x+i, y+j)|, (1)$$

where  $f_L(x, y)$  and  $f_R(x, y)$  indicate the intensity values of the point  $(x, y)$  in the left and the right images, respectively, and  $d$  represents the disparity. A pair of points is considered to match if the SAD value is small enough and minimum among all SAD values computed within the possible disparity range. Fig. 1(a) shows an example stereo images and Fig. 1(b) is the disparity image calculated from the stereo image. Darker points indicate larger disparities (nearer points).



(a) stereo image.



(b) disparity image.

Fig. 1. Stereo image and disparity image.

## 2.2 Object Boundary Extraction

We make a 2D map which records object boundaries. We extract points on the boundaries (called *object points*) as follows. The 3D position of each matched point in the image is calculated from its disparity. We first exclude the points which are on the ceiling or on the floor. Since the observation direction is specified by the column of the disparity image, we then compute the histogram of the disparity in each column and select the largest disparity whose frequency is higher than a certain threshold (see Fig. 2). The disparity with low frequency is not selected because it is likely to be caused by false stereo matches. We finally record the object point which has the selected disparity in each column; its positional uncertainty is also recorded.

The calculation of disparity images and the extraction of object points are performed on a realtime im-

age processor with multiple DSPs [9]. The processing speed is 7.5 frames per second.

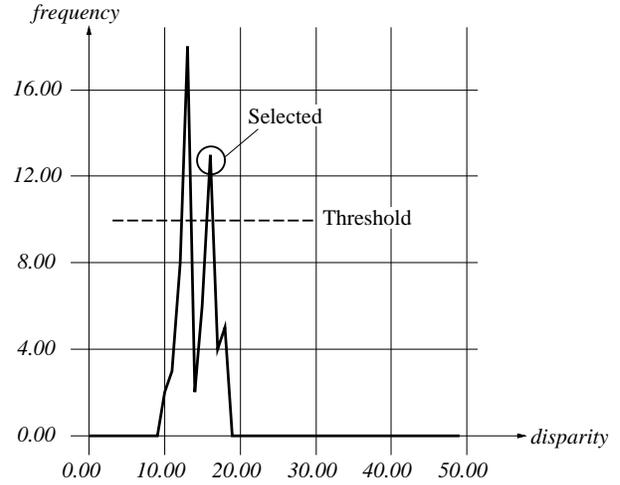


Fig. 2. Histogram of disparity and object boundary extraction.

## 2.3 Positional Uncertainty of Object Point

From the disparity of an object point, its 2D position  $\mathbf{x}_r$  in the scene with respect to the robot position is calculated as (see Fig. 3):

$$\mathbf{x}_r = F(\mathbf{I}) = \frac{1}{X_l - X_r} \begin{bmatrix} a(X_l + X_r) \\ 2af \end{bmatrix}, \quad (2)$$

where  $\mathbf{I} = [X_l, X_r]^T$  ( $X_l$  and  $X_r$  are the horizontal position in the left and the right images, respectively),  $f$  is the focal length, and  $2a$  is the baseline. The position of the object point in the world coordinates is then calculated using the robot position and orientation when the object point is observed.

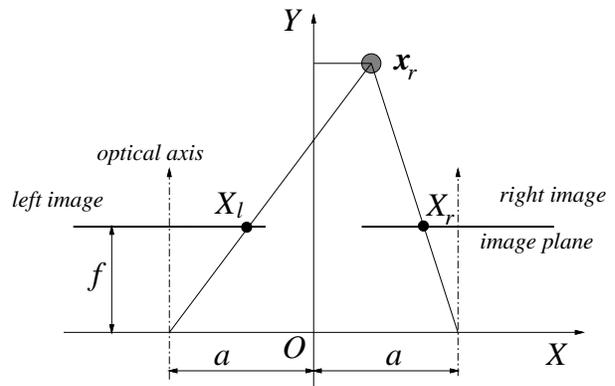


Fig. 3. Stereo geometry

The positional uncertainty is estimated as follows. We suppose that the position in the image includes the error caused by image quantization, and that the errors in both images are independent of each other; the covariance matrix  $\Lambda_{\mathbf{I}}$  of  $[X_l, X_r]^T$  is given by:

$$\Lambda_{\mathbf{I}} = \begin{bmatrix} \sigma_{X_l}^2 & 0 \\ 0 & \sigma_{X_r}^2 \end{bmatrix}. \quad (3)$$

By using the Taylor series expansion with neglecting higher order terms, the stereo uncertainty  $\Lambda_r$  of  $\mathbf{x}_r$  is calculated by [3]:

$$\Lambda_r = \frac{\partial F}{\partial \mathbf{I}} \Lambda_{\mathbf{I}} \frac{\partial F^T}{\partial \mathbf{I}}. \quad (4)$$

The positional uncertainty  $\Lambda$  of the object point in the world coordinates is calculated by first applying the rotation between the robot and the world coordinates (let  $\Lambda'_r$  be the uncertainty after rotation) and by then adding the uncertainty of the robot position  $\Lambda_l$ .  $\Lambda$  is given by:

$$\Lambda = \Lambda'_r + \Lambda_l. \quad (5)$$

We will describe later how to calculate the robot position and its uncertainty in Sec. 2.5.

## 2.4 Integration of Multiple Observations

### 2.4.1 Finding Correspondence between Observation and Map

We integrate the observation result with the map to reduce the positional uncertainty of object points. For this purpose, we need to find the correspondence between the object points in the observed data and the ones in the map.

For each observed point, we search for the corresponding point in the map which satisfies the following condition:

$$(\mathbf{x}_o - \mathbf{x}_m)^T (\Lambda_o + \Lambda_m)^{-1} (\mathbf{x}_o - \mathbf{x}_m) \leq 9, \quad (6)$$

where subscripts  $o$  and  $m$  represent the observed point and the point in the map, respectively. The left side of this inequality is the Mahalanobis distance between

two points; this condition states that the two points can be matched if their relative distance is within the so-called  $3\sigma$  range. If there are several candidates for corresponding points, the one with the minimum Mahalanobis distance is considered the corresponding point.

Even if the robot observes the same object surface several times, the corresponding point may not always be observed for each object point on the surface. This happens when different object points are observed on the same object surface as shown in Fig. 4. In that case, if the corresponding point is not found but two object points are found on the both sides in the neighborhood of the observed point, such as  $\mathbf{x}_1$  and  $\mathbf{x}_2$  in Fig. 4, we estimate the position of the corresponding point as follows.

First, assuming that the object surface is sufficiently smooth, we consider that the surface is approximated by the line connecting  $\mathbf{x}_1$  and  $\mathbf{x}_2$ . We then consider that the corresponding point exists at the position where the Mahalanobis distance to the new observed point is minimum (see Fig. 4). The mean position  $\mathbf{x}$  and its covariance matrix  $\Lambda$  of the corresponding point is calculated by the linear interpolation:

$$\mathbf{x} = (1 - \omega)\mathbf{x}_1 + \omega\mathbf{x}_2, \quad (7)$$

$$\Lambda = (1 - \omega)^2 \Lambda_1 + \omega^2 \Lambda_2, \quad (8)$$

where  $\omega$  indicates the ratio of interpolation.

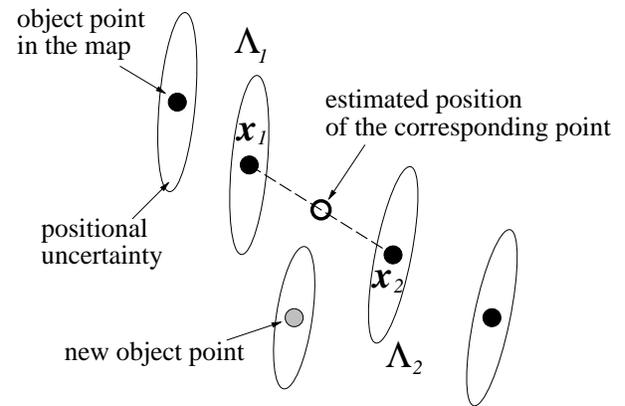


Fig. 4. Estimation of corresponding point

If the object point is found only on either side of the observed point in the neighborhood, or if any point is not found which satisfies eq. (6), we consider that the observed point belongs to an unobserved part of a known object or to an unknown object, and simply

add the point to the map.

In order to exclude the object points caused by false correspondence in stereo, the object points which have not been integrated for a certain consecutive observations are deleted from the map.

#### 2.4.2 Integration of Positional Information of Observed Points

The observed object points for which the corresponding points are found are integrated into the map by the maximum likelihood estimate [11]. The new mean position and the covariance matrix are given by:

$$\mathbf{x} = \frac{\Lambda_o^{-1}\mathbf{x}_o + \Lambda_m^{-1}\mathbf{x}_m}{\Lambda_o^{-1} + \Lambda_m^{-1}} \quad (9)$$

$$\Lambda^{-1} = \Lambda_o^{-1} + \Lambda_m^{-1} \quad (10)$$

where subscripts  $o$  and  $m$  represent the new observation and the map, respectively.

#### 2.5 Estimation of Robot Position

To integrate observation results at different viewpoints, we need to estimate the motion between the viewpoints. Since the positional uncertainty accumulates if the robot position is estimated only by dead reckoning, we employ the following vision-based localization method.

We use a three-wheeled electric scooter as a mobile robot as shown in Fig. 6. We have previously developed an uncertainty model of robot motion caused by the errors of the odometer and the steering angle [7]. Using this model, we predict the uncertainty region for the current robot position. The motion error monotonically increases as the robot moves, and the actual error depends on the moving distance. In our experiments, the error accumulated by the movement between viewpoints (i.e., the movement covered during the processing of one image) does not exceed  $20 [cm] \times 20 [cm]$ . On the other hand, the maximum error of our localization method described below is empirically estimated to be around  $8 [cm]$ . From these data, we decided to divide the uncertainty region into  $3 \times 3$  subregions and consider the center

of each subregion as a candidate for the robot position.

As described in Sec 2.2, at each observation, the set of object points which are nearest to the robot in every direction are obtained. From the distances of such object points, we can obtain a profile of distances, called a *range profile*, which describes the distance to the object in each direction. Fig. 5 shows an example of range profile computed from the disparity data shown in Fig. 1(b). On the other hand, if the robot orientation is given in each candidate position, another range profile is also computed from the map. By comparing these two range profiles, we determine the robot position and orientation which maximizes the *similarity* of the two range profiles.

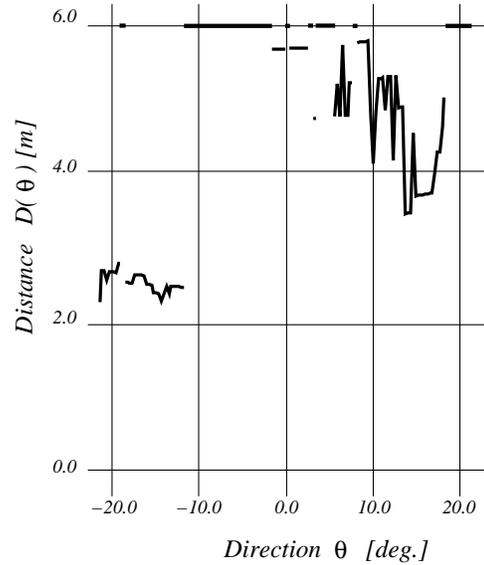


Fig. 5. The range profile computed from the range image shown in Fig. 1(b). Since the robot detects object points whose distances from the robot are within  $6[m]$ , if any points are not found within the range in a direction, the distance in the direction is set to  $6[m]$ .

The similarity of range profiles is evaluated by:

$$S(i, \phi) = \sum_{\theta=\theta_{min}}^{\theta_{max}} |D_o(\theta) - D_m^i(\theta - \phi)|, \quad (11)$$

where  $i$  is a candidate position and  $\phi$  is the orientation of the robot;  $D(\theta)$  is the distance to an object in direction  $\theta$ ; subscripts  $o$  and  $m$  represent the observation and the map, respectively;  $[\theta_{min}, \theta_{max}]$  is the range of possible viewing direction. This expression calculates the sum of absolute difference between profiles; thus, smaller  $S(i, \phi)$  indicates higher similarity.

Using this equation, for each candidate position, the best viewing direction  $\phi^*(i)$  is determined by:

$$\phi^*(i) = \arg \min_{\phi} S(i, \phi). \quad (12)$$

Then, the best position  $i^*$  is determined by:

$$i^* = \arg \min_i S(i, \phi^*(i)). \quad (13)$$

After  $i^*$  is determined, the positional uncertainty  $\Lambda_l$  of the robot is updated so that the corresponding uncertainty region circumscribe the selected candidate region. Concerning the orientation uncertainty, we are currently set the uncertainty of orientation to zero after each localization.

The merit of this localization method based on the comparison of range profiles is that the computational cost is low since explicit point correspondence is not necessary.

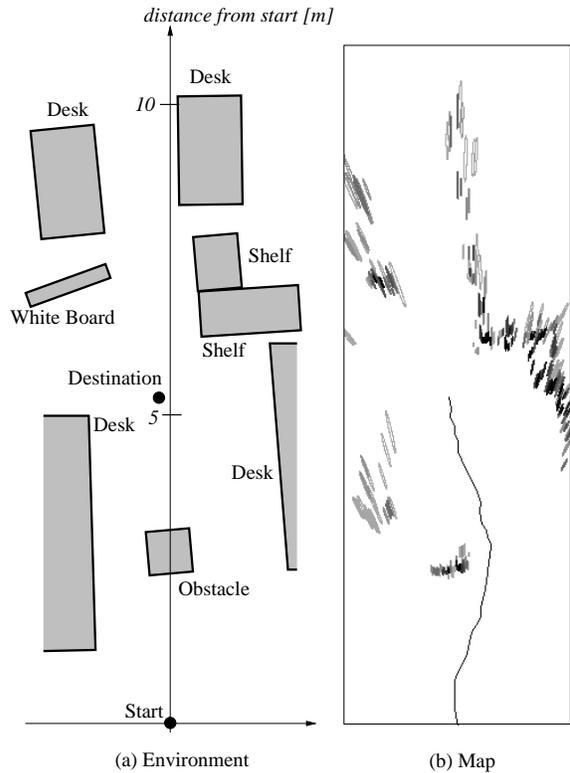


Fig. 7. A map generation result after a guided movement.

## 2.6 Experimental Result of Human-Guided Movement

The user guides the robot using the keyboard as shown in Fig. 6. During this movement, the robot repeatedly observes the surrounding environment without stopping. The robot swings the camera head horizontally to observe a wide area. Fig. 7 shows an example of the environment and the generated map. In the map, each object point is represented by the *uncertainty ellipse*, which is the so-called  $3\sigma$  ellipse calculated from the covariance matrix. The black line is the robot movement estimated by the visual localization method.



Fig. 6. A snapshot of human-guided movement

## 3 Autonomous Navigation with Observation Planning

### 3.1 Generation of Shortest Path

Fig. 8 illustrates the generation of the shortest path. For path generation, we first make a grid-based map. We regard the grids which include parts of uncertainty ellipses as occupied. The set of occupied grids constitutes object regions. To consider the robot width and the motion uncertainty, we then enlarge the object regions by  $w/2 + a$ , where  $w$  is the robot width and  $a$  is the motion error of the robot estimated empirically (currently,  $20[cm]$ ). Finally, we compute the shortest path outside the enlarged regions. The path consists of straight line segments and circular segments; the radius of the circular segments is set to the minimum turning radius of the robot.

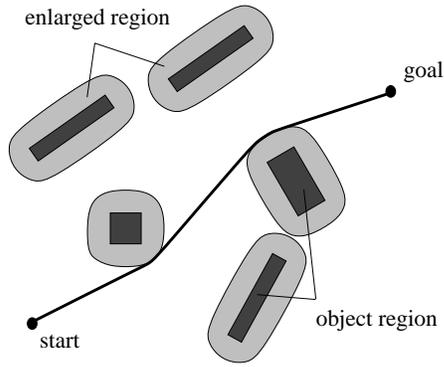


Fig. 8. Shortest path

### 3.2 Observation Planning

Our robot moves at a constant speed; it also makes observations at an almost constant time interval. Viewpoints (i.e., robot positions where the image is input) are thus determined from the speed, the interval, and the generated path. We here describe a method to determine an appropriate viewing direction for each viewpoint. The method could be used in combination with our method of selecting optimal viewpoints online [7].

#### 3.2.1 Determining a set of observable object points

To select the best viewing direction, we first determine a set of observable object points (i.e., feature points on object boundaries) for each viewpoint. Since the robot was once guided by the user from the start point to the destination, it knows at which viewpoint each object point was observed. This knowledge is used to determine the observability of object points.

In general, the path in the autonomous movement phase is different from the guided one. We therefore make the following two assumptions. We first assume that an object point is observable on the line connecting the object point and the viewpoint where the object point was observed before. We further assume that if an object point is observed at two consecutive viewpoints, the object point is observable at any point between the two viewpoints. Under these assumptions, for each object point, an *observable area* is determined where the object point is observable as shown in Fig. 9. We calculate the corresponding ob-

servable area for every object points in the map and determine the set of observable points for each viewpoint.

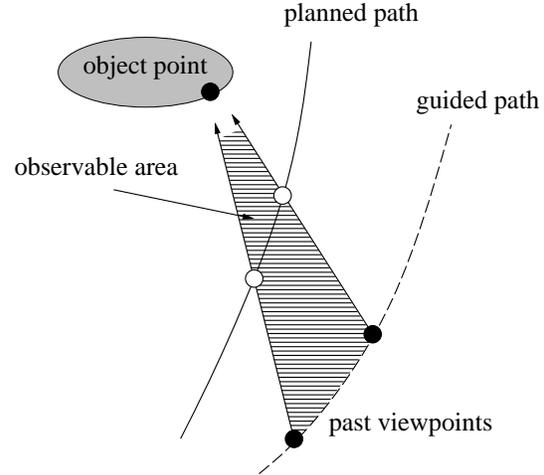


Fig. 9. Observable area of an object point. The object point is expected to be observable from the viewpoints between two white circles on the planned path.

#### 3.2.2 Determining viewing direction

The set of observable object points is calculated for each viewpoint, as described above. Since a part of the points are actually observed for a specific viewing direction due to the limited field of view, we examine which part of the points is most effective for robot localization.

As described in Sec. 3.1, the shortest path consists of straight line segments and circular segments; a circular segment is determined by its nearby object points. Since the relative distance from the robot to such points is important when entering a circular segments, the robot directs the camera head to the object point which is nearest to the circular path (called *nearest point*) among such nearby points, whenever it is possible. Fig. 10 shows the regions where nearest points are observable; if the viewpoint is in one of the regions, the robot always observes the corresponding nearest point.

If such a nearest point is not observable from the current viewpoint, the robot selects the viewing direction so that the positional error of the robot is minimized after observation.

Let us consider the case where the robot observes object point at  $x_m$  and its uncertainty in the map is

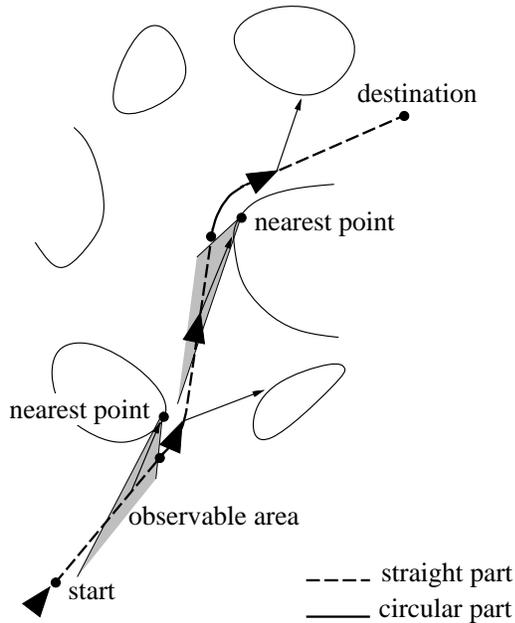


Fig. 10. Selection of viewing direction

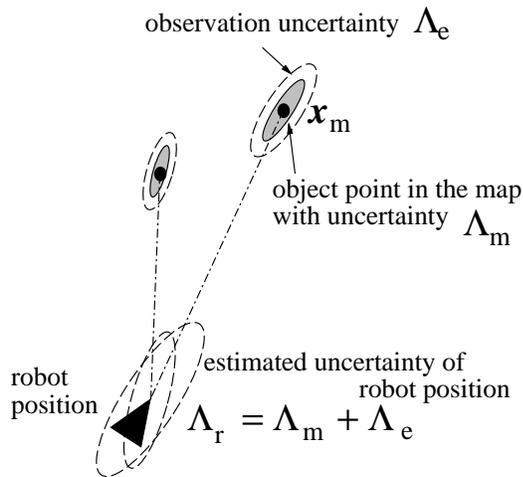


Fig. 11. Estimating the uncertainty in localization

$\Lambda_m$  (see Fig. 11). The object point will be observed within the uncertainty ellipse  $\Lambda_m + \Lambda_e(\psi)$  centered at  $\mathbf{x}_m$ , where  $\psi$  is the viewing direction and  $\Lambda_e(\psi)$  is the observation error determined by the relative position between the point and the robot (note that these uncertainties are all supposed to be represented in the world coordinates for simplicity). Therefore, by observing the point, the robot position is constrained by the uncertainty ellipse represented by:

$$\Lambda_l(\psi) = \Lambda_m + \Lambda_e(\psi) \quad (14)$$

For each observable object point, the above constraint on the robot position can be calculated. Therefore, as in the case of eq. (10), the uncertainty  $U(\psi)$  to be

obtained by viewing direction  $\psi$  can be estimated by integrating information of each object point based on the Maximum likelihood estimate:

$$U(\psi)^{-1} = \sum_{i \in VR(\psi)} \Lambda_l^i(\psi)^{-1}, \quad (15)$$

where  $\Lambda_l^i(\psi)$  is the constraint by object point  $i$ ;  $VR(\psi)$  is the set of observable points when observing direction  $\psi$ .

We search for the best direction  $\psi^*$  which minimizes the uncertainty of the robot position:

$$\psi^* = \arg \min_{\psi} |U(\psi)| \quad (16)$$

### 3.3 Experimental Result of Autonomous Navigation

We performed experiments on autonomous navigation using a real robot. During the autonomous movement, the robot estimates the position using the visual localization method described in Sec. 2.5 and adjust its moving direction so that it follows the generated shortest path.

An experimental result is shown in Fig. 12. The start point and the destination are the same as those shown in Fig. 7. In Fig. 12, a light line indicates the shortest path and dots around the path represent the estimated viewpoint. The arrow on each dot represents the viewing direction which the robot observed in this experiment. At viewpoints 1–3, the robot observed the nearest point of the object on the left side of the path. At viewpoints 4 and after, since the robot was not able to observe nearest points, it observes the best viewing direction which are selected to minimize the uncertainty in robot localization. Fig. 13 shows snapshots of the robot autonomously moving.

In order to show the effectiveness of our viewing direction selection based on the minimum localization error criterion, we compare our method with the method which always directs the camera head forward. The results are summarized in Tables 1 and 2.

Table 1 shows the predicted positional uncertainty  $|U(\phi)|$  of the robot. Smaller values indicate higher accuracy. The values in the parentheses in the left

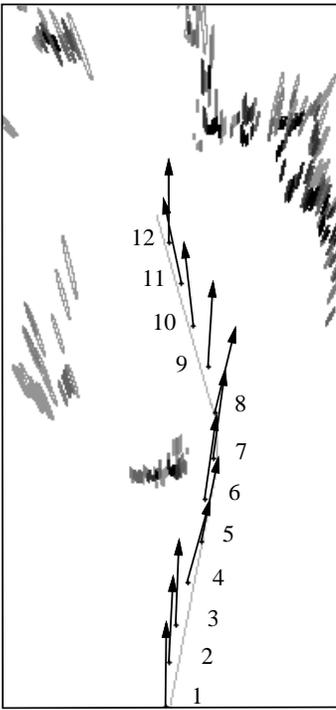


Fig. 12. The result of a navigation based on an observation plan

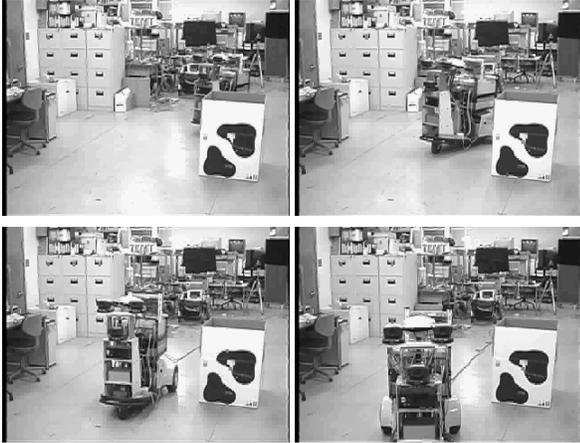


Fig. 13. snapshots of autonomous navigation

column indicate selected view directions. Our method (indicated as *planned*) always provides better localization.

Table 2 shows the result of robot localization using the method in Sec. 2.5. The table describes the similarity value  $S(i^*, \phi^*(i^*))$  of the selected candidate (indicated as  $S^*$ ), the mean similarity value  $\bar{S}$  for all candidates, and the value  $\alpha$  calculated by the following expression:

$$\alpha = (\bar{S} - S(i^*, \phi^*(i^*))) / \sigma, \quad (17)$$

where  $\sigma$  is the standard deviation of the similarity

value.  $\alpha$  indicates the distinctiveness of the best position estimate among all estimates. In our localization method (see Sec. 2.5), the larger the difference between the range profiles at the selected position candidate and at the others is, the smaller the error in localization is expected to be. So another way of selecting the viewing direction is to search for the direction which maximizes the difference. It is, however, costly to calculate the difference for every possible cases; so we use the method described in Sec. 3.2.2 as an alternative. Table 2 shows that the viewing directions selected by our method always increase  $\alpha$  and, therefore, that the viewing direction selection using the equation (16) is effective for more accurate estimation of the robot position. Note that the selected viewing direction does not necessarily maximize  $\alpha$ ; however the method seems practical because it can improve the localization accuracy with a relatively small amount of computation.

## 4 Conclusion

We have proposed a novel visual navigation method based on human-guided experiences. In the human-guided phase, the robot observes the environment with a stereo vision and generates a map. In the autonomous movement phase, the robot utilizes the map and the past experience on observation in order to select viewing directions which are effective for safe and efficient navigation. To realize this navigation strategy, we developed a map generation method considering uncertainties, a fast visual localization method, and a method of selecting appropriate viewing directions. In our strategy, all the user has to do is to just guide the robot from the start point to the destination.

A future work is to combine the proposed method with the method for selecting viewpoints [7] to simultaneously determine both the viewpoint and the viewing direction for more efficient navigation. Another future work is to cope with the case where the environment changes between the map generation and the autonomous navigation phase.

Table 1

The comparison of predicted uncertainty of the robot position with and without observation planning.

| viewpoint | planned           |  | forward           |
|-----------|-------------------|--|-------------------|
|           | viewing direction |  | viewing direction |
| 4         | 5.21 (-5.47)      |  | 9.24              |
| 5         | 4.23 (-1.04)      |  | 4.77              |
| 6         | 1.64 (2.04)       |  | 1.98              |
| 7         | 1.75 (2.44)       |  | 2.20              |
| 8         | 0.665 (-8.49)     |  | 1.14              |
| 9         | 1.13 (-7.08)      |  | 2.60              |
| 10        | 2.87 (-1.89)      |  | 3.41              |
| 11        | 1.40 (0.66)       |  | 1.91              |
| 12        | 1.61 (-9.76)      |  | 2.18              |

Table 2

The comparison of estimation result on localization with and without observation planning.

| viewpoint | planned           |           |          | forward           |           |          |
|-----------|-------------------|-----------|----------|-------------------|-----------|----------|
|           | viewing direction |           |          | viewing direction |           |          |
|           | $S^*$             | $\bar{S}$ | $\alpha$ | $S^*$             | $\bar{S}$ | $\alpha$ |
| 4         | 231.11            | 248.52    | 0.324    | 238.55            | 244.54    | 0.113    |
| 5         | 264.07            | 271.65    | 0.612    | 256.18            | 261.01    | 0.314    |
| 6         | 242.53            | 248.21    | 0.572    | 206.85            | 211.80    | 0.406    |
| 7         | 248.17            | 256.95    | 0.447    | 133.31            | 137.39    | 0.315    |
| 8         | 184.77            | 192.63    | 0.271    | 227.98            | 233.27    | 0.264    |
| 9         | 199.66            | 211.94    | 0.365    | 194.14            | 206.43    | 0.253    |
| 10        | 164.11            | 179.13    | 0.744    | 156.35            | 163.53    | 0.314    |
| 11        | 186.26            | 194.61    | 0.276    | 201.45            | 214.51    | 0.199    |
| 12        | 207.10            | 223.54    | 0.321    | 190.03            | 200.90    | 0.251    |

## References

- [1] C.J. Taylor and D.J. Kriegman: "Vision-Based Motion Planning and Exploration Algorithms for Mobile Robots," *IEEE Trans. on Robotics and Automation*, vol. 14, no. 3, 1998.
- [2] T. Kanbara, J. Miura, and Y. Shirai, "Selection of Efficient Landmarks for an Autonomous Vehicle", *Proc. of 1993 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 1332-1338, 1993.
- [3] D.J. Kriegman, E. Triendl, and T.O. Binford: "Stereo Vision and Navigation in Building for Mobile Robots," *IEEE Trans. on Robotics and Automation*, vol. 5, no. 6, pp. 792-803, 1989.
- [4] S. Maeyama, A. Ohya, and S. Yuta, "Non-stop Outdoor Navigation of a Mobile Robot," *Proc. of the 1995 IEEE/RSJ int. Conf. on Intelligent Robotics and Systems*, pp. 130-135, 1995.

- [5] S. Maeyama, A. Ohya, and S. Yuta: "Outdoor Landmark Map Generation through Human Assisted Route Teaching for Mobile Robot Navigation," Proc. of the 1996 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 957-962, 1996.
- [6] Y. Matsumoto, M. Inaba, H. Inoue, "Visual Navigation using View-Sequenced Route Representation," *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 83-88, 1996.
- [7] I. Moon, J. Miura, Y. Shirai, "On-line viewpoint and motion planning for efficient visual navigation under uncertainty," *Robotics and Autonomous Systems* Vol. 28, pp. 237-248, 1999.
- [8] G. Oriolo, G. Ulivi, M. Vendittelli: "Real-Time Map Building and Navigation for Autonomous Robots in Unknown Environments," *IEEE Trans. on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 28, no. 3, pp. 316-333, 1998.
- [9] Y. Yamane, Y. Shirai, J. Miura, and Y. Kuno: "Person Tracking Using Optical Flow and Uniform Brightness Regions," *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 3267-3273, 1998.
- [10] T. Yoneda, Y. Shirai, and J. Miura: "Navigation of a Vision-Guided Mobile Robot in an Unknown Environment," Proc. Japan-U.S.A. Symp. on Flexible Automation, Vol. 2, pp. 569-572, 1998.
- [11] Z. Zhang and O.D. Faugeras: "A 3D World Model Builder with a Mobile Robot," *The International Journal of Robotics Research*, Vol. 11, No. 4, pp. 269-285, 1992.