Autonomous Visual Navigation of a Mobile Robot Using a Human-Guided Experience[†]

Kiyosumi Kidono, Jun Miura, and Yoshiaki Shirai

Dept. of Computer-Controlled Mechanical Systems, Osaka University
2-1 Yamadaoka, Suita, Osaka 565-0871, Japan
{k-kidono, jun, shirai}@cv.mech.eng.osaka-u.ac.jp

Abstract. It is necessary for a robot to have environmental information in order to move autonomously. Although we can usually give a map to the robot, making such a map is quite a tedious work for the user. So we propose a navigation strategy which requires the minimum user assistance. In the method, we first guide a mobile robot to a destination by a 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. 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 planning 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.

1 Introduction

Recently, many studies have been conducted on autonomous mobile robots. It is necessary for a mobile robot to know environmental information in order to move autonomously. There are methods [1][2] for planning the navigation using a given map and landmarks. In these methods, the robot position is estimated by observing known landmarks in order to reach the destination safely. But it costs too much for users to give a map and specify landmarks.

We propose a navigation strategy which requires the minimum user assistance. Our approach is composed of two-phase: map making and autonomous navigation. In the map making phase, we guide 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.

Kanbara et al. [3] proposed a similar two-phase approach. They considered only one dimensional movement in a linear corridor. Since we make a two dimensional map, our method can be applied to more general navigation problem. Matsumoto et al. [4] propose to use a sequence of images which is obtained during a human-guided movement to guide the autonomous movement. Since the robot is navigated to generate the same image sequence, the robot has to follow the original path even if it is inefficient.

In this paper, we propose the following two-phase strategy. In the map making phase, a mobile robot is guided by a remote control. During this movement, the robot

[†]In Proc. of the 6th Symp. on Intelligent Autonomous Systems, pp. 620-627, 2000.

observes the surroundings with stereo vision and makes a map by integrating multiple observations. In the autonomous navigation phase, the robot computes the shortest path to reach the destination quickly because the human-guided path is not always the best. Taking advantage of the human-guided experiment, the robot makes an observation plan in order to estimate the robot position more accurately, thereby reducing the number of observations. Then the robot moves along the computed path with estimating its position by matching observed information with the map.

The proposed method assumes a static environment; that is, the environment in the autonomous navigation phase is assumed to be the same as that in the map-making phase. If an unknown obstacle appears in the autonomous navigation phase, it is assumed to be detected using ultrasonic sensors.

2 Map Generation

In this phase, we make a map considering observation uncertainty and motion uncertainty. The following subsections describe the method for making the map from stereo images.

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 a camera head and their optical axes horizontal and in parallel with each other. So 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 5×5 window W around a feature point as follows:

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

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.

(a) Original stereo image (b) Disparity image

Fig. 1: The process of calculating a disparity image

Fig. 1(a) shows an example stereo images and Fig. 1(b) is the disparity image calculated from the stereo image. The darker points indicate larger disparities (nearer points).

2.2 Object Boundary Extraction

Once a set of matched points are obtained, their three dimensional positions are computed by triangulation. We make a 2D map which represents object boundaries. So we extract points on the object boundaries (called *object points*) from the 3D information of matched points.

Since the observation direction is specified by the column of the disparity image, we 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 correspondence.

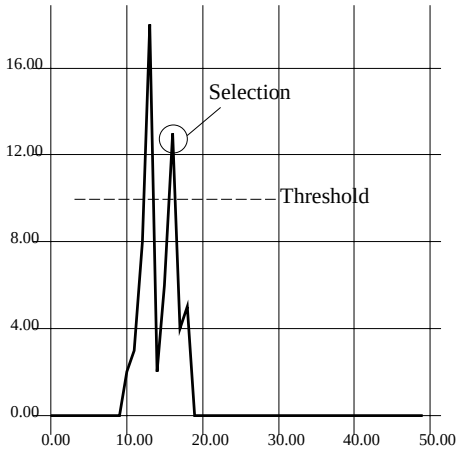


Fig. 2: Histogram of disparity

To estimate the robot position, we use the *distance profile* which represents the object boundaries observed from a specific position. The distance profile is computed by the set of disparities selected in each column of the disparity image. In subsection *2.6*, we explain the method of estimating the robot position.

2.3 Observation Uncertainty

We model the uncertainty of object points by two dimensional normal distribution [1]. The mean position $\boldsymbol{x}$ of each object point is calculated from the disparity by triangulation. The disparities of object points include an error caused by image quantization. Using the Taylor series expansion, we calculate the 2D positional uncertainty Λ of object points. In the map, a set of $\boldsymbol{x}$ and Λ is recorded.

In this paper, we consider the so-called 3σ ellipse obtained from Λ as the range of uncertainty, called an uncertainty ellipse.

2.4 Finding Correspondence between Observation and Map

We integrate observed object points in order to reduce the uncertainty of their position. So we must find the correspondence between object points in observed data and ones in the map.

First, for each observed point, we search for the corresponding point in the map. We consider the point which satisfies the following equation as the corresponding point.

$$(\boldsymbol{x}_n - \boldsymbol{x}_m)^T(\Lambda_n + \Lambda_m)^{-1}(\boldsymbol{x}_n - \boldsymbol{x}_m) \leq 9 \quad (2)$$

where indices n and m represent the new observed point and the point in the map, respectively.

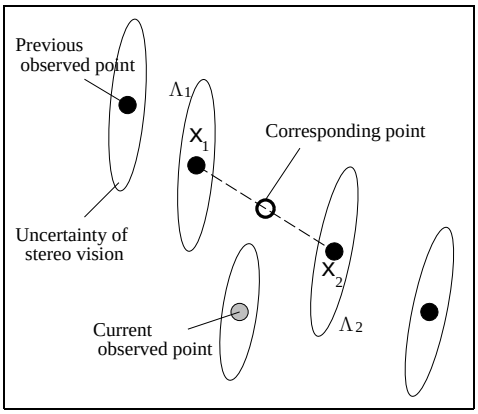


Fig. 3: Estimation of corresponding point

If the corresponding point is not found, we consider the following two situations. One is the situation where the observed object points are a part of an unknown object boundary. In this situation, the object point is simply added to the map. The other is

the situation where the observed object points belong to the known object boundaries as shown in Fig. 3. In this situation, the corresponding point is computed by the following procedure.

As shown in Fig. 3, we first search for two object points near the current observed point. Assuming that the object boundary between the two object points is linear, we consider the point on the linear boundary which is the nearest, in terms of the Mahalanobis distance, from the observed point as the corresponding point. Since the corresponding point has not been observed actually, the mean position $\boldsymbol{x}$ and the covariance matrix Λ are estimated by:

$$\boldsymbol{x} = (1-\omega)\boldsymbol{x}_1 + \omega\boldsymbol{x}_2, \tag{3}$$

$$\Lambda = (1-\omega)^2\Lambda_1 + \omega^2\Lambda_2, \tag{4}$$

where ω indicates the ratio of interpolation.

2.5 Integration of Observed Points

The observed object points for which the corresponding points are found are integrated into the map by Kalman filter [5].

$$\boldsymbol{x} = \frac{\Lambda_c^{-1}\boldsymbol{x}_c + \Lambda_p^{-1}\boldsymbol{x}_p}{\Lambda_c^{-1} + \Lambda_p^{-1}} \tag{5}$$

$$\Lambda^{-1} = \Lambda_c^{-1} + \Lambda_p^{-1} \tag{6}$$

where indices c and p indicate the current and the previous observation, respectively.

During the human-guided movement, the robot repeatedly observes the surrounding environment without stopping. The robot swings the camera head to observe a wide area. To integrate observation results at different viewpoints, we estimate the motion between the viewpoints. The method for estimating the robot position will be described in next subsection. In order to exclude the object points caused by false correspondence of stereo image, the object points which have never been integrated are deleted from the map.

2.6 Estimation of Robot Position

In the calculation of the robot position by dead reckoning, the positional uncertainty is accumulated due to the uncertainty of the robot motion. So we estimate the robot position by additionally using visual information as follows.

First a predicted uncertainty of a robot position can be computed by considering the error of the odometer and the steering angle [1]. We divide the predicted uncertainty region into 3×3 regions and consider the center of each divided region as a candidate for the robot position. If the robot orientation is given in each candidate position, a distance profile is computed from the map. The distance profile is also computed from the observation. The candidate position is evaluated by the similarity between the two distance profiles.

The similarity $S(i, \phi)$ is calculated by

$$S(i,\phi) = \sum_{\theta=\theta_{min}}^{\theta_{max}} \left| D_o(\theta) - D_m^i(\theta - \phi) \right|, \tag{7}$$

where $D_o(\theta)$ and $D_m(\theta)$ indicate the profile based on the observation and the map, respectively; and i and ϕ indicate the candidate position and orientation of the robot,

respectively. First, we determine the best robot orientation in each candidate position by

$$\phi^*(i) = \arg\min_{\phi} S(i, \phi). \tag{8}$$

Then we select the best candidate position in all candidates by

$$i^* = \arg\min_{i} S(i, \phi^*(i)). \tag{9}$$

2.7 Experimental Result of Human-Guided Movement

We conducted an experiment in our laboratory as shown in Fig. 4(a). Fig. 4(b) show the map made by the robot. The map represents the uncertainty ellipses of the integrated object points. The black line is the estimated robot movement.

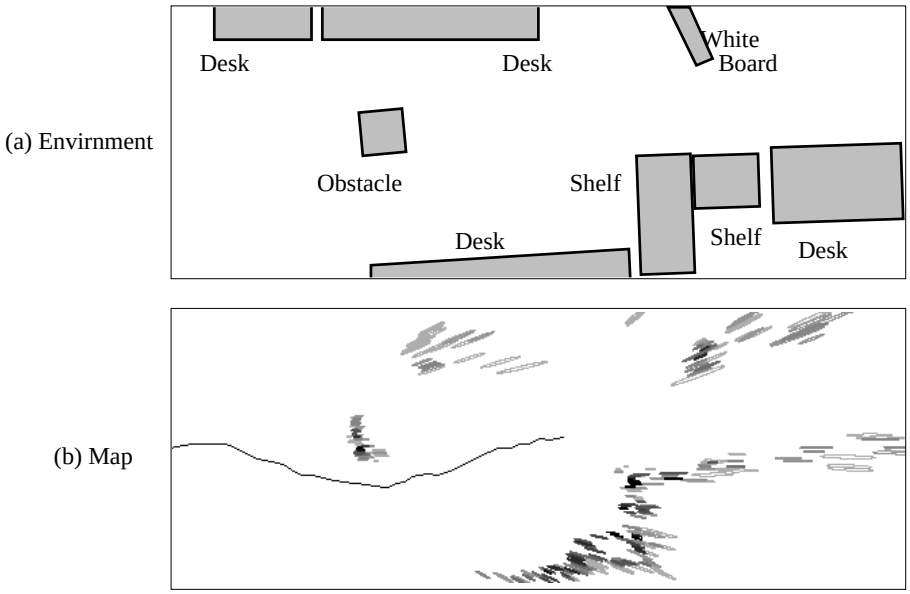


Fig. 4: The experimental result of map making

3 Autonomous Navigation

In this phase, our goal is to reach the destination quickly and safely. After making the map, we detect the shortest path to reach the destination quickly. To move safely, we need to estimate the robot position in the map.

3.1 Detection of Shortest Path

We represent the map by the grids. We regard the grids which include parts of uncertainty ellipses as the object boundaries. In path detection, the object boundary are enlarged by a certain width in order to consider the motion uncertainty and the robot size. We compute the shortest path outside the enlarged regions. As shown in Fig. 5, the path is made from the straight line segments and the circular segments which are defined by the minimum turning radius of the robot.

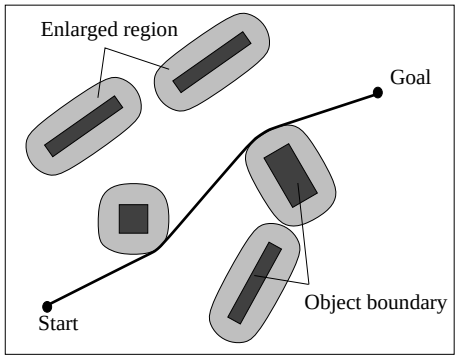
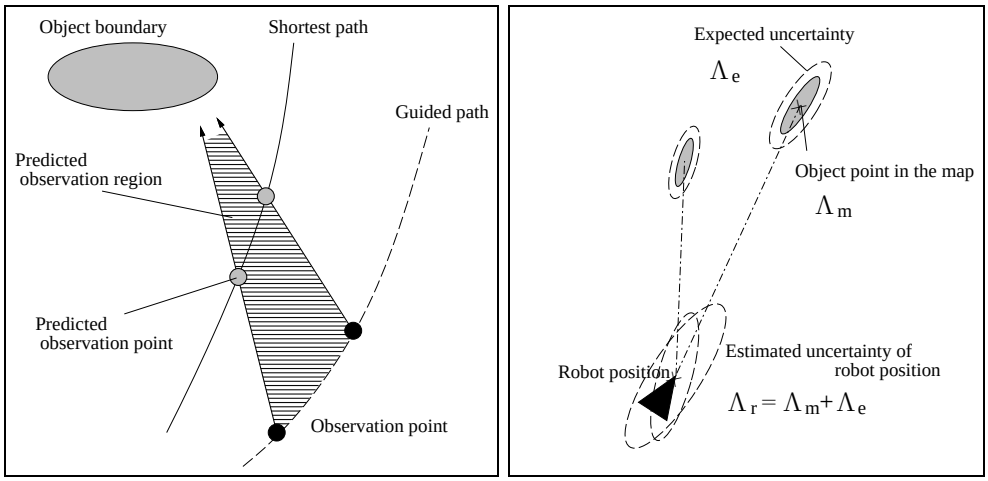


Fig. 5: Shortest path

3.2 Observation Planning

By planning viewpoints and viewing directions, the robot could estimate its position more accurately, thereby reducing the number of observations and being able to reach to the destination. We [1] have already developed a method for planning viewpoints. In this paper, therefore, we propose a method of planning the viewing direction for the case where the robot observes the environment at constant intervals. Once we guided the robot to a destination, the robot knows at which viewpoint each object point was observed. This knowledge is useful in planning an effective observation strategy.

First, we want to know which object points are observable from a viewpoint. We estimate the region where each object point in the map is able to be observed. If the robot is located on the line connecting an object point and the viewpoint where the robot observed the object point on the guided path, the robot is assumed to be able to observe the object point. In addition, an object point is observed at two consecutive viewpoints on the guided path, the robot is assumed to be able to observe the object point between these viewpoints. Based on these assumptions, for each object point, we can calculate the region within which the object point is observable as shown in Fig. 6(a). Using such regions, we can obtain the set of object points observable for a viewpoint.



(a) Prediction of observation region

(b) The localization estimated by the uncertainty of object points

Fig. 6: Our ideas on the autonomous navigation

Next, we determine the viewing direction at each viewpoint. The shortest path consists of straight line segments and circular segments. Since the shape of a circular segment is determined by its nearby object points, it is important for the robot to observe such points before entering the circular segment. Therefore the robot directs the camera head so that such object points are located at the center of the observed images, whenever it is possible.

If such object points are not observable from the current viewpoint, we select the viewing direction as follows. For each viewing direction, we can predict a set of object points to be observed. Then we can predict the positional uncertainty of the robot to be obtained by observing the set by considering the uncertainty of the position of each object point and that of the observation. Finally, we select the best viewing direction which gives the minimum positional uncertainty of the robot.

The predicted uncertainty of the robot position is calculated as follows. For each viewing direction ϕ, a set of observable object points $\mathcal{O}(\phi)$ is determined. Suppose that the robot observes an object point j in $\mathcal{O}(\phi)$ whose uncertainty is Λ_{mj} in the map. The positional uncertainty $\Lambda_{lj}(\phi)$ of the robot is computed by the following equation (see Fig. 6(b)).

$$\Lambda_{lj}(\phi) = \Lambda_{mj} + \Lambda_{ej}(\phi), \tag{10}$$

where $\Lambda_{ej}(\phi)$ is the observation uncertainty of object point j calculated by the vision uncertainty model [1]. The positional error of the robot is computed from all observable object points by the following equation based on the maximum likelihood estimate [6] as follows:

$$U^{-1}(\phi) = \sum_{j \in VR(\phi)} \Lambda_{lj}^{-1}(\phi). \tag{11}$$

We finally select the best viewing direction ϕ^* which minimizes the positional uncertainty of the robot as follows:

$$\phi^* = \arg\min_{\phi} |U(\phi)|. \tag{12}$$

3.3 Experimental Result of Autonomous Navigation

We performed experiments on autonomous navigation using a real robot. An experimental result is shown in Fig. 7. In the experiment, the map shown in Fig. 4(b) was used. In Fig. 7, 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. Fig. 8 shows snapshots of the robot autonomously moving.

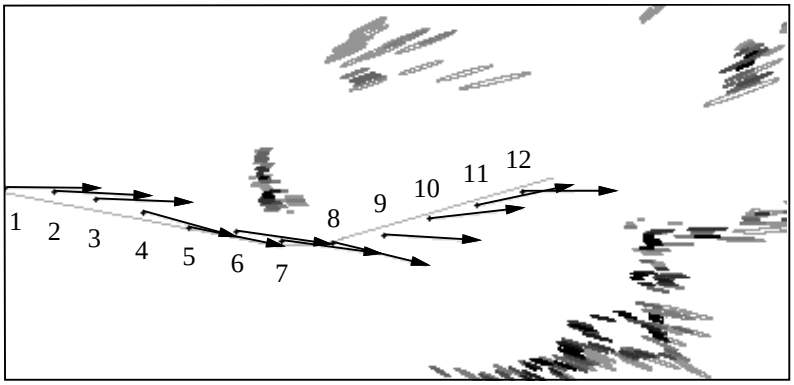


Fig. 7: An experimental result of autonomous movement

Fig. 8: Scene of autonomous navigation

In order to show the effectiveness of our view planning method, we compare our method with the method which always directs the camera head to the right front. The results are shown in Table 1 and in Table 2. The positional uncertainty $|U(\phi)|$ of the

robot is shown in Table 1. The smaller values indicate higher accuracy. The values in Table 2 are calculated by the following expression:

$$\{S(i^*, \phi^*(i^*)) - \hat{S}(i, \phi^*(i))\}/\sigma, \tag{13}$$

where S is the value calculated by equation (7) and $\hat{S}$ and σ is the average and the standard deviation of the similarity. This value indicates the distinctiveness of the best position estimate. The larger values indicate more accurate estimation. These results shows that the viewing direction calculated by the equation (12) is effective for accurately estimating the robot position.

Table 1: The comparison of the estimated positional uncertainty

No.	Front	Our Method (ϕ_*[deg])
5	4.77	4.23 (-1.04)
8	1.14	0.665 (-8.49)
11	2.18	1.61 (-9.76)

Table 2: The difference of the estimation of the robot position

No.	Front	Our Method
5	0.542	0.895
8	0.331	0.343
11	0.295	0.371

4 Conclusion

We have proposed a navigation method of a mobile robot based on the map which is autonomously generated by the robot. During human-guided movement, the robot makes the map which represents object boundaries by integrating the observed object points. After making the map, the robot detects the shortest path and plans viewing directions at each viewpoint. The robot can reach to the destination safely and efficiently using the plan. The feature of our approach is that user have only to guide the robot to a destination for once. A future work is to cope with the case where the environment changes between the map-making and the autonomous navigation phase.

References

[1] I. Moon, J. Miura, Y. Shirai, "On-line viewpoint and motion planning for efficient visual navigation under uncertainty", *Robotics and Autonomous Systems* 28 (1999) 237-248.

[2] 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.

[3] 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.

[4] 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.

[5] N. Ayache and O. D. Faugeras, "Maintaining Representations of the Environment of a Mobile Robot", *IEEE Trans. on Robotics and Automation*, vol. 5, no. 6, pp. 804–819, 1989.

[6] Zhengyou Zhang and Olivier Faugeras: "A 3D World Model Builder with a Mobile Robot", *The International Journal of Robotics Research*, vol. 11, no. 4 August 1992.