Proc. 1992 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 1847-1854, Raleigh, NC, 1992.

Hierarchical Vision-Motion Planning with Uncertainty: Local Path Planning and Global Route Selection

Jun Miura Yoshiaki Shirai
Dept. of Mech. Eng. for Computer-Controlled Machinery
Osaka University
Suita, Osaka 565, Japan
jun@ccm.osaka-u.ac.jp

Abstract — **A new framework of a hierarchical vision-motion planning for a mobile robot under uncertainty is proposed. An optimal plan is generated by two-level planning: global route selection and local path planning. In global route selection, a sequence of observation points to acquire sufficient information to reach a destination is determined. Both the cost and the uncertainty of vision are considered in this planning. In local path planning, given two successive observation points, trajectories, moving speeds, and reference points for robot localization are determined so that a robot can reach the second point safely with a minimum cost. Both the error in localization and that in motion control are considered in this planning. A local path planner is repeatedly invoked in global route selection to determine an actual path between observation points. Our hierarchical planner can generate an optimal plan for a mobile robot planning problem.**

I. Introduction

A mobile robot with vision moving to a destination in a cluttered environment needs two kinds of planning. One is to plan a route (a sequence of free areas that the robot passes through) to the destination and the other is to plan actual paths that the robot follows. Since the cost of taking a wrong route and turning back is usually high, a robot should have enough information about the environment to select a correct route. Visual recognition for obtaining such information requires much time in general. Thus, it is important to select efficient views. On the other hand, when a robot follows a path, avoiding collision with obstacles is the most important problem. It is, therefore, necessary to take the uncertainty of motion control into consideration in generating an actual path.

There have been many works on path planning of mobile robots (ex. [2][7][10]). The goal of most researches is development of algorithms for generating a minimum cost (or feasible) path in a cluttered environment. To cope with a complex environment, several assumptions are made: a robot has complete information about the environment in advance; a robot can move without uncertainty. Clearly, such assumptions are inappropriate in a real world. Some works [8] treat path planning in an unknown environment, while the uncertainty and the cost of sensing is not yet considered. Shekhar et al. [12] proposed a motion planning strategy based on the preimage approach [3] considering uncertainties in control and sensing, while their model of uncertainty is too simple and is not constructed for actual robots or sensors.

Visual guidance of a mobile robot is also one of the recent topics. Pollard et al. [11] describes a vehicle control method using predictive feed-forward stereo. Blake et al. [1] proposed an approach to path-planning around smooth obstacles. Kubota et al. [6] described a method of generating heuristically an obstacle-avoiding path using image data.

There are also many works on the error analysis of mobile robot localization [5] [14]. The results of the analyses, however, have been rarely used for generating an actual path. Error in trajectory-following control [4] [13] should also be considered in path planning.

In a previous paper [9], we proposed a framework of a vision-motion planning for a mobile robot under uncertainty. In that framework, an optimal sequence of vision and motion operations is generated considering the cost and the uncertainty of visual recognition. It was assumed that an actual path is a straight line connecting neighboring observation points, and that a robot can move on it without uncertainty. These assumptions are unsuitable for a real robot. For example, a behavior such that a robot changes its speed according to the width of a route cannot be expressed. In this paper, therefore, we improve our framework in the following two points: first, an actual path is generated considering both the uncertainty in localization and that in motion control; second, a total plan is generated by two-level planning, where the previous planner works as a higher level planner.

In order to concentrate the argument on structure of a planning system, we simplify a planning problem by assuming that a robot moves in a static environment, and that a graph of possible routes is given in advance. In such situation, the object of a higher level planner is to determine a sequence of observation points to acquire enough information for selecting a correct route at each branch.

Section II explains how a vision-motion planning problem is hierarchically decomposed into two subproblems. Section III briefly reviews a global vision-motion planning framework we have already proposed. Section IV describes models of uncertainty in localization and control and explains an algorithm of generating a local path using such models. In section V, we show a simulation result. In the final section, we summarize this paper and describe future works.

II. Hierarchical Decomposition of Vision-Motion Planning

Planning of a mobile robot moving to a destination among obstacles can be decomposed into two problems: the selection of a route (a sequence of free areas) to pass; and the determination of actual paths. We call the former *global route selection* and the latter *local path planning* (Fig.1).

In order to determine an optimal route, a robot needs enough information about the environment such as types, sizes, and positions of obstacles. Especially the passability of a space between obstacles is important information. The cost of obtaining such information is generally high because not merely a measurement but recognition of a complex scene is often necessary. We assume, therefore, that a robot stops at each time to recognize the environment. Under this assumption, the object of global route planning is to determine an optimal sequence of observation points and directions. A route is determined by connecting successive observation points, from a start point to a goal point. We have formulated this problem by using statistical decision theory in [9].

If a robot is given two successive observation points, it must determine an actual path to follow. In local path planning, trajectories, moving speeds and reference points for robot localization are determined. The position of a robot relative to an obstacle is used for describing trajectories because it is more useful for obstacle avoidance than the absolute positions. Note that the cost of visual processing during trajectory-following control is rather low. This is because recognition of the scene has been already finished in global route selection and merely measurement is carried out. Therefore visual processing is performed simultaneously with the robot motion control.

We employ a two-level cost minimization in order to construct an optimal (minimum time) plan. In global route selection, the total cost of vision and motion operations is minimized. A local path planner is invoked whenever a minimum cost of each motion is needed in global route selection. Since possible routes are compared in terms of the cost which has been minimized within each route, a generated plan is globally optimal.

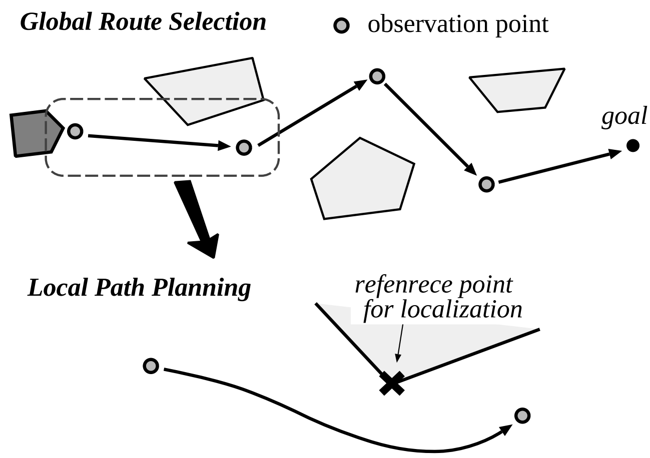

Fig. 1: Two-level planning.

III. Global Route Selection

In global route selection, a robot determines a sequence of observation points to acquire enough information to select a correct route. The cost and the uncertainty of visual recognition is considered. This section briefly describes the algorithm of global route selection [9].

Here, we define that the optimal route is a route which minimizes the expectation of the total cost. We assume that the state of the environment is represented by a multivariate distribution, each variable of which is a property of the environment. We call such multivariate distribution *distribution information* and describe it by D. We derive a recurrence formula which relates the current observation point $\boldsymbol{x}_i$ and distribution information $\boldsymbol{D}_i$ with the optimal next observation point $\boldsymbol{x}_{i+1}$ and observation $\boldsymbol{o}_{i+1}$.

Since an observed data is a vector of properties, the uncertainty of an observed data becomes a multivariate distribution. Let $P_{obsd}(\boldsymbol{s}; \boldsymbol{D}_i, \boldsymbol{x}_{i+1}, \boldsymbol{o}_{i+1})$ denote the probability of getting an observed vector $\boldsymbol{s}$. Also, let $fuse(\boldsymbol{D}, \boldsymbol{s}, \boldsymbol{x}, \boldsymbol{o})$ be a function which computes a new distribution information from $\boldsymbol{D}$, $\boldsymbol{s}$, $\boldsymbol{x}$, and $\boldsymbol{o}$ using statistical data integration. A robot can predict that a distribution information $fuse(\boldsymbol{D}_i, \boldsymbol{s}, \boldsymbol{x}_{i+1}, \boldsymbol{o}_{i+1})$ will be acquired with the probability $P_{obsd}(\boldsymbol{s}; \boldsymbol{D}_i, \boldsymbol{x}_{i+1}, \boldsymbol{o}_{i+1})$ after an observation $\boldsymbol{o}_{i+1}$ at $\boldsymbol{x}_{i+1}$.

Because sensor data are integrated statistically (by Bayes rule), the integration result of them can be considered to includes all information acquired in the past recognition process. Therefore the optimal plan based on some specific position and distribution information is independent of how such information has been acquired. Thus, the minimum cost at $\boldsymbol{x}_i$ with distribution information $\boldsymbol{D}_i$ is given by the minimum of the sum of the following (Fig.2):

1. the cost of motion to the next observation point $\boldsymbol{x}_{i+1}$.
2. the cost of the next observation $\boldsymbol{o}_{i+1}$.
3. the minimum cost from $\boldsymbol{x}_{i+1}$ to the goal point. This is a weighted sum of minimum expectations of the cost, each of which depends on each possible sensor information $\boldsymbol{s}$ obtained by $\boldsymbol{o}_{i+1}$ and weighted with the probability of $\boldsymbol{s}$.

Therefore the problem of global route selection is formulated as follows:

$$\begin{aligned} C_o(\boldsymbol{x}_i, \boldsymbol{D}_i) &= \min_{\substack{\boldsymbol{x}_{i+1} \in \mathcal{X} \\ \boldsymbol{o}_{i+1} \in \mathcal{O}}} \begin{pmatrix} C_m(\boldsymbol{x}_i, \boldsymbol{x}_{i+1}) + C_v(\boldsymbol{o}_{i+1}) + \\ min\_cost(\boldsymbol{x}_{i+1}, goal) \end{pmatrix}, \\ min\_cost(\boldsymbol{x}_{i+1}, goal) &= \\ \int P_{obsd}(\boldsymbol{s}; \boldsymbol{D}_i, \boldsymbol{x}_{i+1}, \boldsymbol{o}_{i+1}) \cdot & \\ C_o(\boldsymbol{x}_{i+1}, fuse(\boldsymbol{D}_i, \boldsymbol{s}, \boldsymbol{x}_{i+1}, \boldsymbol{o}_{i+1}))d\boldsymbol{s} & \end{aligned} \quad (1)$$

$C_o(\boldsymbol{x}, \boldsymbol{D})$: The optimal cost with $\boldsymbol{D}$ at $\boldsymbol{x}$.
$C_m(\boldsymbol{x}, \boldsymbol{y})$: The cost of motion from $\boldsymbol{x}$ to $\boldsymbol{y}$, computed by a local path planner.
$C_v(\boldsymbol{o})$: The cost of observation $\boldsymbol{o}$.
$\mathcal{X}$: A possible range of $\boldsymbol{x}_{i+1}$.
$\mathcal{O}$: A possible range of $\boldsymbol{o}_{i+1}$.

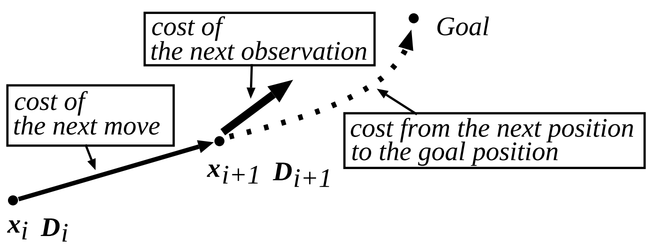

Fig. 2: Calculating the cost to the goal point

Since a planning problem is represented by a recurrence formula, a robot can get an optimal solution using dynamic programming (DP). When a robot can decide the final motion to the goal point with the current information, since no further observation is necessary, the cost to the goal point can be computed without using (1). Recursive computation terminates at that time. In order to compute the minimum cost of each local path, a local path planner is repeatedly invoked in global route selection.

IV. Local Path Planning

In local path planning, given two successive observation points, a robot determines the minimum cost, collision-free path connecting them. Generally speaking, a robot can move fast in an environment with few obstacles, while in a cluttered environment, a robot must slow down. In this section, we model such behavior considering the two kinds of errors: the error in localization and that in control.

A. *Error in Localization and Selection of a Reference Point*

When a robot follows a local path among obstacles, geometrical relationship between the robot and obstacles provides important information for collision avoidance. Thus, we assume that a path is represented with respect to the positions of obstacles, and that a robot continuously selects a reference point (RP) for localization in the scene. We also assume that a robot can control the vision system so that the viewing direction is always toward a selected reference point as shown in Fig.3.

If a robot follows a two dimensional path on a flat floor, the localization error in the direction perpendicular to the trajectory is most important for avoiding collision. Thus, we consider only this error as the localization error and represent it by e_l.

Using a model of uncertainty of stereo vision system described in [9], in which only quantization error of digital images is considered, we can represent the relative positional error of the robot to an RP by a normal distribution. If we set the x axis and the z axis so that the z axis is aligned with the viewing direction (Fig.4), the covariance matrix C_{pos} of the normal distribution becomes a function of the distance d to the RP and is represented by

$$C_{pos} = \begin{pmatrix} \sigma_x(d)^2 & 0 \\ 0 & \sigma_z(d)^2 \end{pmatrix}, \quad (2)$$

Letting ϕ be the angle between the z-axis and the trajectory, we can define e_l as

$$e_l(d, \phi) = k \max(\sigma_z(d) \sin\phi, \ \sigma_x(d) \cos\phi), \quad (3)$$

where k is a constant.

Given a set of candidates of reference points in the scene, the robot continuously chooses one with the minimum e_l. For example in Fig.5, from start to $\boldsymbol{b}$ the first RP is used, while the second RP (same as goal) is used from $\boldsymbol{b}$ to goal. Fig.6 is a graph of the localization errors for both RPs for the circular part of the path. At point $\boldsymbol{b}$, both RPs have the same localization error. The robot changes the RP at that point. Note that the time for rotating cameras is currently neglected.

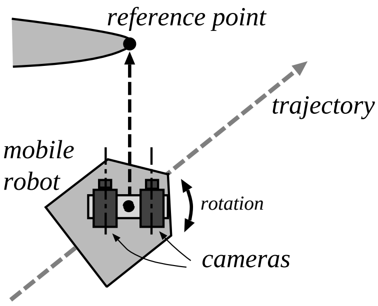

Fig. 3: Camera direction.

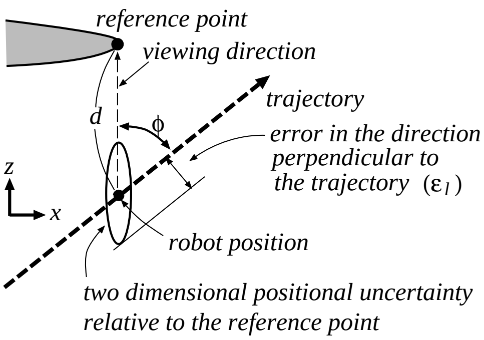

Fig. 4: Localization error.

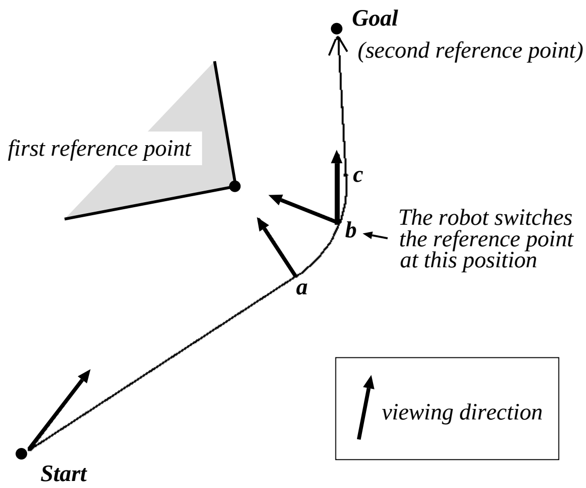

Fig. 5: Change of reference points.

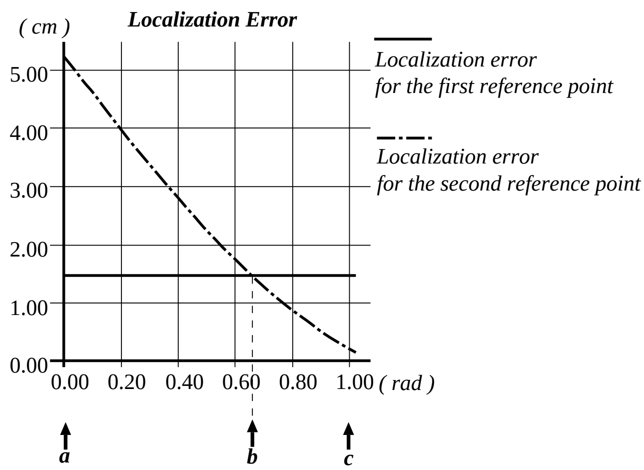

Fig. 6: A graph of localization error.

B. Error in Control

If positional information is obtained by vision, and if the time for acquiring such information once is constant and not negligible, the faster a robot moves, the larger the control error becomes because that the distance that a robot moves during visual data processing is proportional to the speed. As same as the localization error, we consider only the control error in the perpendicular direction to the trajectory.

We assume a car-like mobile robot (Fig.7). The front wheel is steered and actuated. The position of the robot is represented by the midpoint of two rear wheels. As a cause of the control error of the robot, we currently consider only the error of the steering angle.

First, we examine the case the robot follows a straight line (Fig.7). Letting $\Delta\psi$ be the maximum error of the steering angle, the rotation radius R in the worst case is given by

$$R = \frac{L}{\tan\Delta\psi} \tag{4}$$

where L is the wheelbase of the robot. If t_v is the cycle time of visual feedback, the maximum control error $e_c(v)$ when the robot moves at speed v is given by

$$e_c^{straight}(v) = R\big(1-\cos(v\cdot t_v/R)\big). \tag{5}$$

Next, we examine the case that the robot follows a curved trajectory (Fig.8). If the desired steering angle is ψ and the maximum error of it is $\Delta\psi$, the desired rotation radius (R^*) and the actual rotation radius (R) are given by

$$R^* = \frac{L}{\tan\psi}, \tag{6}$$

$$R = \frac{L}{\tan(\psi+\Delta\psi)}. \tag{7}$$

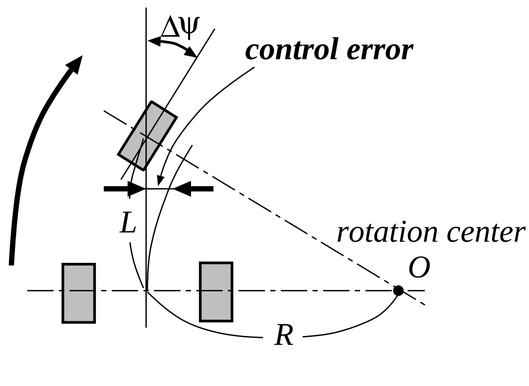

Fig. 7: Control error on a straight trajectory.

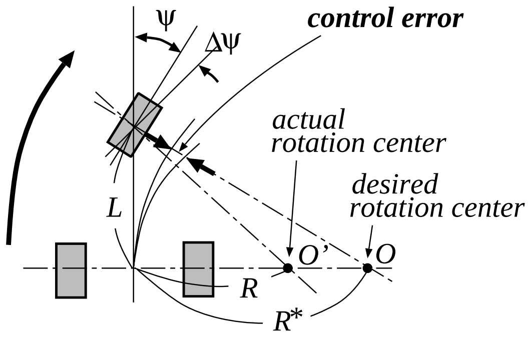

Fig. 8: Control error on a curved trajectory.

Using these radii, the maximum control error for a curved trajectory is represented by

$$e_c^{curved}(v) = R^* - \sqrt{R^{*2} - 2R(R^*-R)\cdot\big(1-\cos(v\cdot t_v/R)\big)}\,. \tag{8}$$

C. Generating an Optimal Local Path

This subsection describes an algorithm of generating an optimal local path. We currently assume that a trajectory consists of straight lines and circular ones, and that the speed of a robot is limited to discrete predetermined values v_i ($i = 1 \sim n$, $v_i < v_{i+1}$). We also assume for simplicity that the robot does not collide with an obstacle if the distance to it is larger than a given value R_0, although the shape of the robot is not a circle.

Suppose a trajectory is already given. By choosing reference points along the trajectory, the robot can calculate the localization error using (3). By choosing the speeds along the trajectory, the robot can calculate the control error using (5) and (8). The sum of these errors is considered to be the total error. Since the total error shows how near the robot can be to obstacles, we can decide whether the current trajectory is safe (collision free) or not. However it is generally difficult to generate an optimal (minimum time) path because the localization error depends on the relative position of the robot to obstacles, that is, a trajectory itself. In order to make the path generation problem easier, we make more assumptions as follows: there are only polyhedral obstacles in the environment; only vertices of obstacles are candidates of RPs (reference points); the robot tries to avoid no more than two obstacles at a time; a local path is composed of five parts (*approach, turn1, intermediate, turn2* and *departure*) as shown in Fig.9; *turn1* and *turn2* have the same radius; the robot moves on *turn1, turn2* and *intermediate* at the same speed. Of course, some parts of the path can be omitted according to the placement of obstacles.

The procedure of generating a local path is as follows. First, the robot selects a turning speed v_t, and consequently the control error becomes $e_c(v_t)$. Since a vertex of the front obstacle is the reference point at the former part of a turn as shown in Fig.5, the localization error for a turning radius r becomes $e_l(r, \pi/2)$. To make this turn safe, the following unequality must be satisfied (Fig.10).

$$r > R_0 + e_c(v_t) + e_l(r, \pi/2). \tag{9}$$

The minimum radius which satisfies this inequality is selected. The selected radius is common to both *turn1* and *turn2*. Once a turning radius has been determined, the geometry of five parts is completely determined so that *approach*, *intermediate* and *departure* become tangent lines of *turn1* and *turn2*, where *approach* and *departure* pass through *Start* and *Goal*, respectively (see Fig.9).

Then, we determine speeds on *approach*. When a robot approaches an obstacle from a distant point, it can move at the highest speed at first. As the robot comes near to the obstacle, it must slow down. To make the total time minimum, therefore, the robot needs to know the nearest point to the obstacle to which the robot can move at some speed v_i. Let us consider Fig.11. A point $\boldsymbol{p}_1$ is on *approach*, and d is the distance from $\boldsymbol{p}_1$ to the endpoint of *approach*. We calculate the localization error e_l and the control error e_c at $\boldsymbol{p}_1$ and set a point $\boldsymbol{p}_2$ so that the distance from $\boldsymbol{p}_2$ to *approach* is $e_l + e_c$. We decide the speed v_i at point $\boldsymbol{p}_1$ is safe if the distance D from $\boldsymbol{p}_2$ to the reference point is larger than R_0. By calculating the minimum value of d for each speed, from the highest speed to the turning speed v_t, the positions for slowing down are determined. For *departure*, the similar calculation is carried out, while the speed increases as the robot goes away from the obstacles.

By using the above procedure and by selecting reference points as described in A, the robot can generate the complete path (trajectories, speeds, and reference points) for some selected turning speed v_t and can calculate the time to follow the path. By repeating this procedure for possible v_t's, the minimum time path is finally determined. The flow of local path planning is summarized in Fig.12. Fig.13 shows a result of local path planning. The speed and the required time for each path are indicated. Viewing directions to three RPs are also indicated. In actual, parts with very small required time will be eliminated.

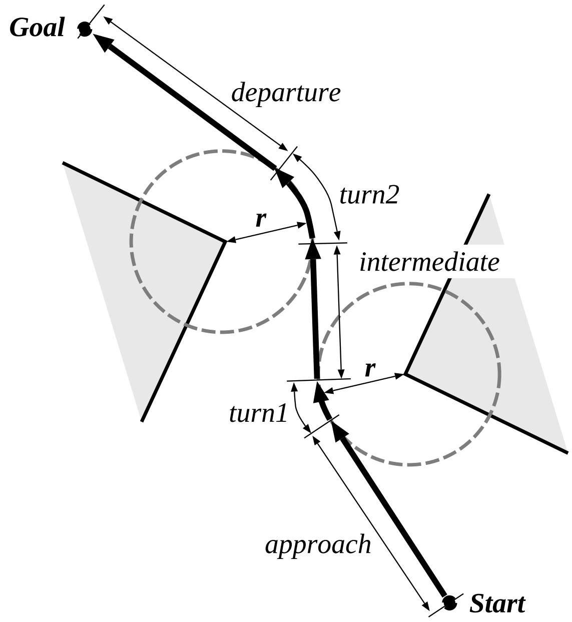

Fig. 9: Structure of a local path.

Fig. 10: A safe turn.

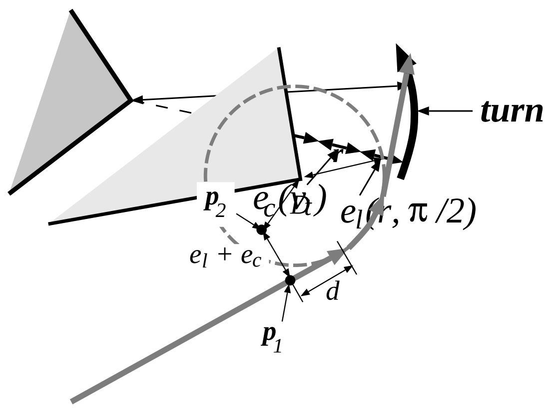

Fig. 11: Determination of positions for decreasing the approach speed.

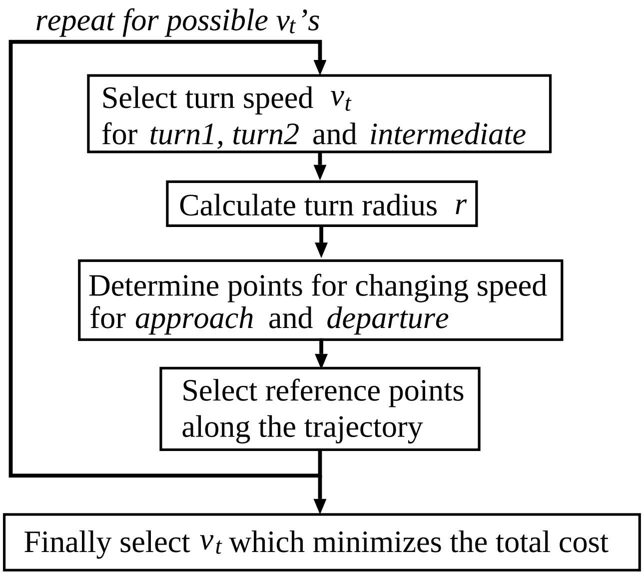

Fig. 12: Flow of local path planning.

V. Simulation Results

This section describes a simulation result of hierarchical vision-motion planning. Fig.14 shows the problem and the optimal solution. There are two routes to reach the goal point (its position is given) from the initial position. One route passes through between obstacles and the other one passes the outside of the right obstacle. The former route is shorter than the latter. To take the former route, the robot needs to estimate the width of the space between obstacles. As the robot cannot decide whether the shorter route is passable or not by the initial observation at the start point, it needs to generate a plan by the recursive search of observation points. In this problem, we made the following assumptions: the cost of taking a detour is known in advance; possible observation points are limited to predetermined discrete positions; discrete distributions are actually used to represent positional uncertainties.

Since the positions of the obstacle vertices are discretely distributed, for each combination of the possible positions of the left and the right vertices, an optimal local path to the goal was determined. Using the expectation of the cost of local paths, an optimal sequence of observation points was determined in global route selection. In this problem, enough information for construct the final plan was acquired by the second observation.

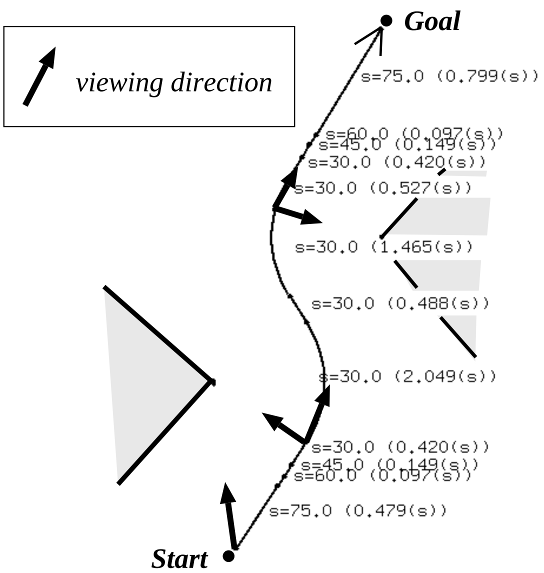

Fig. 13: An example of planned local path

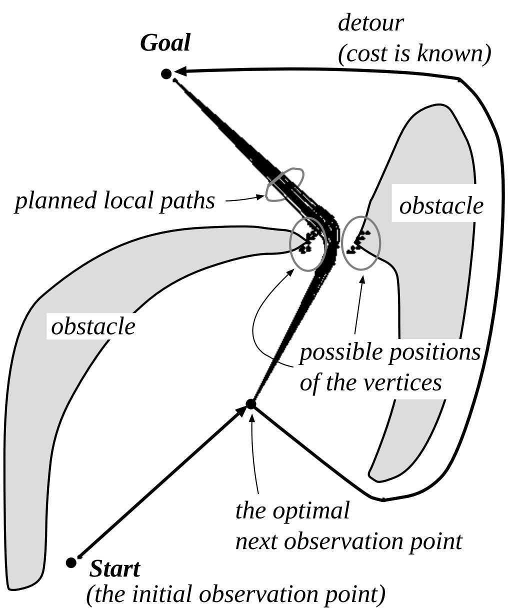

Fig. 14: Simulation result

VI. Conclusions and Discussion

We have proposed a new framework of a hierarchical vision-motion planning under uncertainty. The planning problem has been hierarchically decomposed into two subproblems: global route selection and local path planning. In global route selection, an optimal sequence of observation positions is determined by considering both the cost and the uncertainty of vision. In local path planning, given two successive observation positions, collision-free trajectories, moving speeds, and reference points for robot localization are determined by considering both the error in localization and that in control.

Planning of a local path by taking the localization error and the control error into consideration is an important feature of this paper. However we need the test and the refinement of the error models through experiments by a real robot. Especially, when visual data are integrated with the data acquired by dead reckoning, the localization error and the strategy of selecting reference points must be modified. It is also necessary to improve our local path planning algorithm to cope with more general situation.

Acknowledgement

This work is supported in part by Grant-In-Aid Scientific Research from the Ministry of Education, Science, and Culture, Japanese Government.

References

[1] A. Blake, M. Brady, R. Cipolla, Z. Xie, and A. Zisserman. Visual navigation around curved obstacles. In *Proceedings of 1991 IEEE Int. Conf. Robotics and Automat.*, pages 2490–2495, 1991.

[2] R.A. Brooks and T. Lozano-Pérez. A subdivision algorithm in configuration space for findpath with rotation. *IEEE Trans. on Systems, Man, and Cybanet.*, SMC-15(2):224–233, 1985.

[3] B.R. Donald. Planning multi-step error detection and recovery strategies. *Int. J. Robotics Res.*, 9(1):3–60, 1990.

[4] T. Hongo, H. Arakawa, G. Sugimoto, K. Tange, and Y. Yamamoto. An automatic guidance system of a self-controlled vehicle. *IEEE Trans. Industrial Electronics*, IE-34(1):5–10, 1987.

[5] D.J. Kriegman, E. Triendl, and T.O. Binford. Stereo vision and navigation in buildings for mobile robots. *IEEE Trans. on Robotics and Automat.*, RA-5(6):792–803, 1989.

[6] T. Kubota, H. Hashimoto, and F. Harashima. Path searching of mobile robot based on cooperation of sensors. In *Proceedings of 1988 IEEE Int. Workshop on Intelligent Robots and Systems*, pages 569–574, 1988.

[7] J.C. Latombe. A fast path planner for a car-like indoor mobile robot. In *Proceedings of AAAI-91*, pages 659–665, 1991.

[8] V.L. Lumelsky and A.A. Stepanov. Path-planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape. *Algorithmica*, 2:403–430, 1987.

[9] J. Miura and Y. Shirai. Planning of vision and motion for a mobile robot using a probabilistic model of uncertainty. In *Proceedings of 1991 IEEE/RSJ Int. Workshop on Intelligent Robots and Systems*, pages 403–408, Osaka, Japan, November 1991.

[10] H. Noborio, T. Naniwa, and S. Arimoto. A feasible motion-planning algorithm using the quadtree representation. In *Proceedings of 1988 IEEE Int. Workshop on Intelligent Robots and Systems*, pages 769–774, 1988.

[11] S.B. Pollard, J. Porrill, and E.W. Mayhew. Experiments in vehicle control using predictive feed-forward stereo. In H. Miura and S. Arimoto, editors, *Robotics Research 5*. The MIT Press, 1990.

[12] S. Shekhar and J.C. Latombe. On goal recognizability in motion planning with uncertainty. In *Proceedings of 1991 IEEE Int. Conf. on Robotics and Automat.*, pages 1728–1733, 1991.

[13] T. Washizawa, Y. Kaku, J. Hirai, and K. Ukon. Motion control analysis of automatic guided vehicle. In *Proceedings of 8th Annu. Conf. Robotics Society of Japan*, 1990. (in Japanese).

[14] Y. Watanabe and S. Yuta. Estimation of position and its uncertainty in the dead reckoning system for the wheeled mobile robot. In *Proceedings of the 20th Int. Symp. Industrial Robots*, 1989.