Proc. 1999 IEEE Int. Conf. on Systems, Man, and Cybernetics
pp. IV-692-698, Tokyo, Japan, Oct. 1999.

Mobile Robot Motion Planning Considering the Motion Uncertainty of Moving Obstacles

Jun Miura, Hiroshi Uozumi, and Yoshiaki Shirai
Department of Computer-Controlled Mechanical Systems,
Osaka University, Suita, Osaka 565-0871, Japan
jun@mech.eng.osaka-u.ac.jp
http://www-cv.mech.eng.osaka-u.ac.jp/~jun

Abstract

This paper proposes a motion planning method for a mobile robot in the situation where there are both static and moving obstacles. If the robot cannot communicate with moving obstacles, it has to predict their future movement in order to plan the safe and efficient motion. Since such a prediction includes uncertainty, the proposed method explicitly considers the uncertainty in motion planning. We use a probabilistic model of the uncertainty and select the motion which minimizes the expected time of reaching the destination. We also utilize the knowledge of possible paths of moving obstacles, which is applicable to usual structured environments. Simulation results validate the effectiveness of the method.

1 INTRODUCTION

Mobile robot motion planning in dynamic environments has recently been studied extensively [1]. In the case where a robot cannot communicate with moving obstacles, it is necessary for the robot to predict the future motion of them. Most of past research can be classified, in terms of the knowledge of the movement of obstacles, into two categories. In one category, the movement of obstacles is completely *unknown*, thus, the reactive motion planning is only reasonable way for a robot to cope with moving obstacles [2]; not the optimality of robot motion but the safety is an important issue there. In the other category, the movement of obstacles is completely *known*, thus, the optimal motion can be generated by employing a *planning in space-time* [3]. Tsubouchi and Arimoto [4] applied this idea to the robot with a limited range of sensing; from the sensed position and velocity of obstacles, the robot predicts their future movement by assuming that they will continue to move at a constant velocity, and plans the next best action. Such prediction and planning are repeatedly performed.

In between these categories, several works consider the uncertainty in obstacle motion. Inoue et al. [5] proposed a method to predict the motion of an obstacle and its uncertainty from the history of its movement. They considered only the range of uncertainty; that is, the robot generates a plan which is safe regardless of actual obstacle motion. This method may result in an inefficient robot motion if the positional distribution of the obstacle is not uniform within the range.

This paper considers the following two characteristics of motion prediction:

- Prediction quality usually increases as time advances.
- The probability that each motion actually occurs within the possible range is not uniform in general.

Let us consider a simple example. Suppose you are going to cross a street and a car is approaching you. You have to decide when to begin crossing the street, i.e., before or after the car passes. When the car is far away, predicting when the car will pass in front of you suffers from a large uncertainty because it is the prediction of a far future, and because the observation uncertainty is large for a far object. However, as time advances, the situation will be more certain and, at some time point, you will be able to make a decision with confidence. Also, the time of the car passing will usually be non-uniformly distributed around the most likely predicted value.

In addition to the above characteristics, we consider to use the knowledge of the environment; that is, in usual structured environment, we can predict the obstacle motion to some extent. They never move randomly; each has its own start and goal points and the path connecting them should be generated in some rational manner (e.g., by a minimum-length criterion). For example, in a typical office environment, flow of people is restricted by the displacements of walls, doors, furniture and so on. We use the *tangent graph* [6] to represent such a restriction.

The proposed planner first calculates possible paths in the tangent graph generated only from given information of static obstacles. We deal with the case where the trajectories of moving obstacles and the desired trajectory of a robot are restricted to the ones on the possible paths. In this situation, we can enumerate points on the paths where the robot and each moving obstacle may meet. Then the planner calculates the probabilistic distribution of the moving obstacle coming to such a point; this distribution is used to estimate the expectation of the time of the robot reaching the destination. The best motion is then selected which minimizes the expected time.

2 MODELING MOTION UNCERTAINTY ON A PATH

Modeling Velocity Uncertainty

This paper deals with the case where the path of a moving obstacle is given as a sequence of segments on a tangent graph. Thus we needs to model only the velocity variation of the obstacle and to predict its 1-D position along the path.

We assume the following on the movement of obstacles: each moving obstacle has the possible range of its velocity, represented as $[v_{min}, v_{max}]$; it changes the velocity at every *time step* ΔT; the velocity of a time step is constant and randomly *and* independently selected within the above range[1].

Under these assumptions, we can predict the future position of a moving obstacle as follows. Let x_0 and σ_0^2 be the current position and the variance of an obstacle and v_k be the velocity at the kth time step. Then the position x_i after i steps is given by:

$$x_i = x_0 + \sum_{k=1}^{i} v_k \Delta T. \quad (1)$$

Since every v_k follows the same but independent uniform distribution within the above velocity range, the distribution of x_i can be approximated by a normal distribution (by *central limit theorem* [7]). The variance σ_{step}^2 of the movement added by one step is calculated as:

$$\begin{aligned} \sigma_{step}^2 &= \frac{1}{v_{max} - v_{min}} \int_{v_{min}}^{v_{max}} (v - \bar{v})^2 dv \\ &= \frac{1}{12}(v_{max} - v_{min})^2, \quad (2) \end{aligned}$$

where $\bar{v} = (v_{max} + v_{min})/2$ is the mean of the obstacle velocity. The probability density function $p(x; i)$ of the obstacle being at x after moving for i steps is then given by

$$\begin{aligned} p(x; i) &= \frac{1}{\sqrt{2\pi\sigma_i^2}} \, exp\left(-\frac{(x - \bar{x}_i)^2}{2\sigma_i^2}\right), \quad (3) \\ \bar{x}_i &= x_0 + i\bar{v}\Delta T, \\ \sigma_i^2 &= \sigma_0^2 + i\sigma_{step}^2. \end{aligned}$$

[1]More constraining knowledge could be used depending on the actual environment and the problem settings.

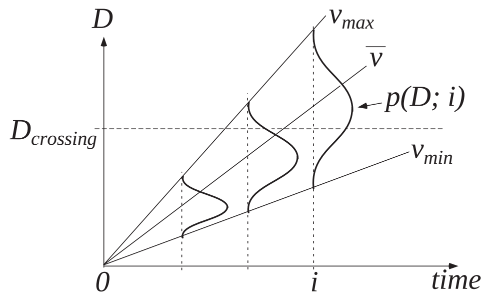


Fig. 1: Prediction of arrival time.

Predicting the Arrival Time of Moving Obstacle at a Crossing

Movement of an obstacle affects that of the robot near the crossings of their paths. Thus it is necessary to calculate the distribution of the arrival time of the obstacle at a crossing. Using the velocity uncertainty model described above, the distribution is calculated as follows (see Fig. 1).

In the figure, the vertical axis indicates the moving distance of the obstacle from the current position; the horizontal axis indicates the time (or time step). $D_{crossing}$ is the distance to a specific crossing on the path. Since the positional distribution of the obstacle at some time point is calculated by eq. (3), the probability $P(i)$ of the obstacle reaching the crossing at the ith time step can be *approximated* by:

$$P(i) = \alpha p(D_{crossing}; i), \quad (4)$$

where α is a normalization constant and is calculated as $1/\sum_i p(D_{crossing}; i)$. The resultant probability distribution is not symmetric; its mass center is a little biased to the less-time side.

Modeling Sensing Uncertainty

Another source of uncertainty in prediction of the movement of an obstacle is the sensing uncertainty. We suppose a vision-based mobile robot, which uses stereo vision to detect an obstacle and to measure its position and velocity. We use the probabilistic uncertainty model of stereo vision which we have previously developed [8].

The model represents the positional uncertainty of an obstacle due to vision uncertainty by a normal distribution. We here describe how to predict the positional uncertainty after the next time step. Let $N(\mu_0, \sigma_0^2)$ be the *predicted* distribution of obstacle position x after the next step; this distribution is calculated from the current distribution and the predicted motion uncertainty added by the next step. Let x_{obs} be the observation result obtained after the motion for the next step. Assuming that the variance σ_{obs}^2 of x_{obs} is constant regardless of the true value of x, x_{obs} follows $N(\mu_0, \sigma_0^2 + \sigma_{obs}^2)$ [8]. The information on x after the observation is obtained by integrating the predicted distribution $N(\mu_0, \sigma_0^2)$ and the observation result. Let $N(\mu_1, \sigma_1^2)$ be the distribution after the integration. σ_1^2 is given by

$$\sigma_1^2 = \frac{\sigma_0^2 \sigma_{obs}^2}{\sigma_0^2 + \sigma_{obs}^2}. \tag{5}$$

This value is always the same for any observation result. On the other hand, μ_1 depends on the observation result and we cannot know it beforehand. We can, however, calculate the *distribution* of μ_1; the mean μ_{μ_1} and the variance $\sigma_{\mu_1}^2$ are given by [8]:

$$\mu_{\mu_1} = \mu_0, \tag{6}$$

$$\sigma_{\mu_1}^2 = \frac{\sigma_0^4}{\sigma_0^2 + \sigma_{obs}^2}. \tag{7}$$

We use this distribution to enumerate a set of possible states after the next step.

Gradual Reduction of Prediction Uncertainty

As explained with an example in Sec. 1, the current situation may become more certain as time advances. Therefore, if the robot has several candidate paths to the destination, it may be better to defer the decision of path selection, instead of immediately committing to one path. Fig. 2 shows the current probability distribution of an obstacle arriving at a crossing and a set of *predicted* probability distributions after one time step passes and the new observation result is integrated[2]. The set of distributions covers all possible situation which is represented by the current distribution. Only one of which, however, will actually occur. The variance of each distribution in the set is smaller than that of the current one, that is, the situation will become more certain. The reasons why this happens are: (1) the distance to the crossing becomes smaller; (2) observed data are statistically integrated to reduce the uncertainty.

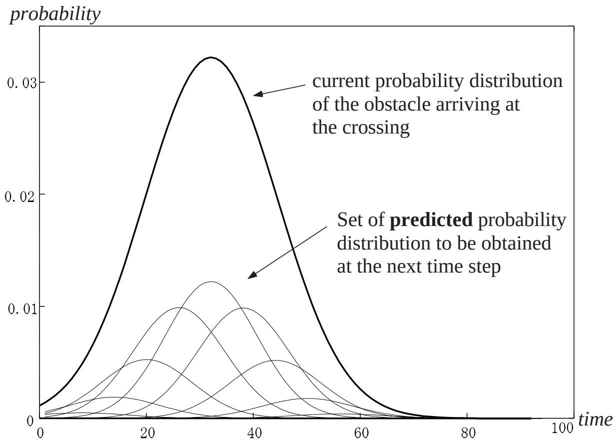


Fig. 2: Reduction of prediction uncertainty. Note that each predicted probability distribution is weighted with its occurrence probability.

3 MOTION PLANNING FOR FIXED PATH MOVING OBSTACLES

Planning the Next Motion

The robot basically follows a path on the tangent graph to minimize the moving distance to the destination as long as there is no influence from moving obstacles. If the robot has to consider avoidance of collision with them, the robot selects a certain number of nodes as the candidates of an intermediate goal and enumerates a set of candidate motions which does not conflict with the candidate nodes (see Fig. 3). After each time step, the robot observes obstacles, estimates their positional uncertainty, and performs one-step look-ahead search for the next motion. Once a node is known to be far superior to the others, the commitment is made to the node.

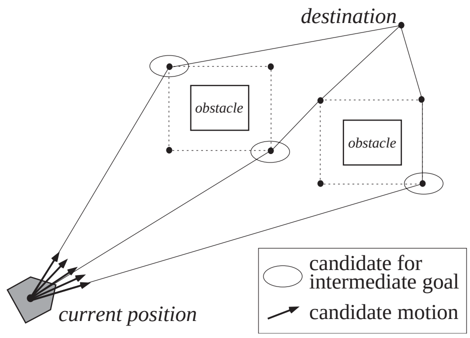


Fig. 3: Generating candidate motions.

The detailed planning algorithm is as follows (see Fig. 4). For each motion $i\,(i = 1, \ldots, N)$, the robot first predicts the set of possible states $\{S_{ij} | j = 1, \ldots, M\}$ and their probability P_{ij}, which are to be obtained after the motion, as described in the previous section. Then for each state S_{ij}, the robot calculates the expected time T_k^{ij} of reaching the destination when selecting candidate node $k\,(k = 1, \ldots, L)$[3] and selects the best (minimum-time) candidate node k_{ij}^* as:

$$k_{ij}^* = \arg\min_{k=1}^{L} T_k^{ij}. \tag{8}$$

[2]Note that the velocity range of the obstacle is discretized with some granularity for a computational purpose.

[3]The next subsection will explain how to calculate the expected time.

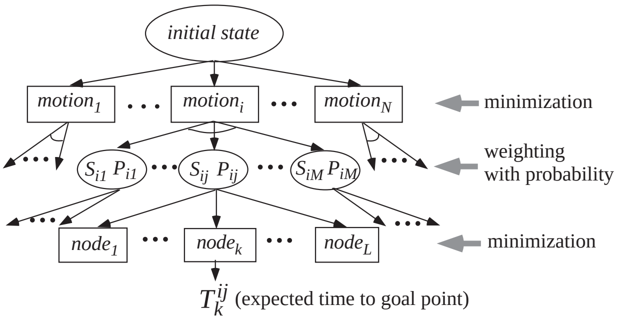


Fig. 4: Selecting the next motion.

Then the expected time T_i of reaching the destination when taking candidate motion i is given by

$$T_i = \sum_{j=1}^{M} P_{ij} T^{ij}_{k^*_{ij}}. \tag{9}$$

Finally the best motion i^* is selected as:

$$i^* = \arg \min_{i=1}^{N} T_i. \tag{10}$$

Calculating Expected Time to Destination

Collision Avoidance by Stopping The robot controller combines the global planning using tangent graphs with the local planning using a potential function. The attractive force comes from the next node on the graph, while the repulsive forces come from dynamic and static obstacles. However this strategy may sometimes result in a very inefficient behavior of the robot; for example, an obstacle could *push* the robot to deviate largely from the desired path. Therefore, we set a safety distance L_{safe} and controls the robot so as not to enter within the distance L_{safe} from any obstacles. L_{safe} is determined such that if the distance of the robot and an obstacle is larger than L_{safe}, the robot motion is not affected by the obstacle within the above-mentioned control scheme.

Basically the robot moves at a constant speed on the shortest path. If the path of the robot and that of an obstacle intersect, and if the robot knows the obstacle will come to the distance less than L_{safe}, the robot stops before the intersection point (crossing) and waits for the obstacle to pass by[4].

Calculating Waiting Time The period during which the robot has to wait is calculated as follows. Let us consider Fig. 5. The two paths intersect at P_c with angle θ. The robot waits at point P_0 whose distance to the path of the obstacle is L_{safe}. Let t_0 be the time at which the robot reaches P_0. To calculate the waiting period, we first calculate two distances, D^{in}_{safe} and D^{out}_{safe}. D^{in}_{safe} indicates the distance of the obstacle to the crossing P_c such that the robot can pass the crossing before the obstacle if the obstacle is further than D^{in}_{safe} at t_0 (see Fig. 5(a)). D^{out}_{safe} is the distance from the crossing such that the robot can pass the crossing after the obstacle if the obstacle is further than D^{out}_{safe} at t_0 (see Fig. 5(b)). Assuming that the obstacle and the robot moves at constant speed v_o and v_r respectively, these two distances are given by:

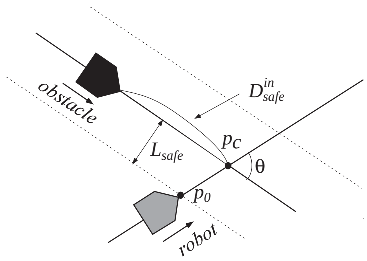
$$D^{in}_{safe} = \frac{L_{safe}}{\sin\theta}\left\{\sqrt{\frac{v_r^2+v_o^2-v_r v_o\cos\theta}{v_r^2}}+\frac{v_o}{v_r}\right\}, \tag{11}$$

$$D^{out}_{safe} = \frac{L_{safe}}{\sin\theta}\left\{\sqrt{\frac{v_r^2+v_o^2-v_r v_o\cos\theta}{v_r^2}}-\frac{v_o}{v_r}\right\}. \tag{12}$$

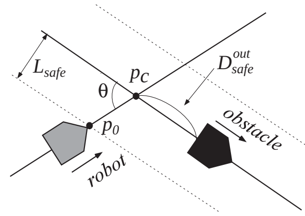
If the obstacle is within the range $[P_c - D^{in}_{safe}, P_c + D^{out}_{safe}]$ at t_0, the robot has to wait for the obstacle exiting from the range. From this condition, we can obtain the following. (1) If the time of the obstacle arriving at the crossing is within the range $[t_0 - D^{out}_{safe}/v_o, t_0 + D^{in}_{safe}/v_o]$, the robot has to wait[5]. (2) In addition, for a time of arrival t within the range, the robot has to wait for the duration of $t - (t_0 - D^{out}_{safe}/v_o)$. (see Fig. 6); this is explained as follows. If the obstacle arrives at P_c at t, we know that it was at the distance of $v_o(t-t_0)$ to P_c at time t_0. Thus the robot has to wait while the obstacle moves by the distance $D^{out}_{safe} + v_o(t-t_0)$. Dividing this distance by v_o leads to the above waiting time.

[4]Note that the above combined motion control strategy is still effectively used in case of an unanticipated movement of an obstacle.

[5]We assume again that the obstacle's speed is constant v_o near the crossing.



(a) Condition for the robot to pass before obstacle.



(b) Condition for the robot to start moving.

Fig. 5: Collision avoidance by stopping.

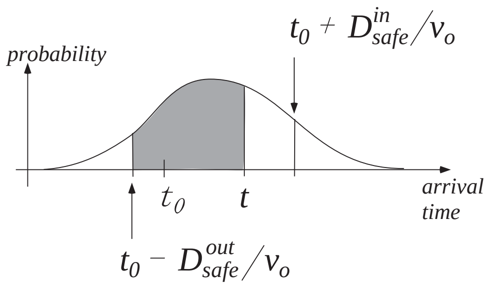


Fig. 6: Expected time to wait.

Expected Time to Destination From the above result and the probability distribution $P(i)$ of the obstacle arriving at the crossing (see eq. (4)), we can calculate the expected time of the robot reaching the destination on a certain path. The expectation of the extra time needed for waiting, T_{wait}, is calculated by:

$$\begin{aligned} T_{wait} &= \sum_{t \in [t_{min}, t_{max}]} P(t)(t - (t_0 - D_{safe}^{out}/v_o)), \quad (13) \\ t_{min} &= t_0 - D_{safe}^{out}/v_o, \\ t_{max} &= t_0 + D_{safe}^{in}/v_o. \end{aligned}$$

The expected time to the destination is then calculated as the sum of T_{wait} and the time needed in the case where the robot encounters no obstacles.

4 SIMULATION

Realtime Simulation Environment

Fig. 7 shows our setup for the *realtime* simulation. *Simulator* is a graphics simulator which controls the movement of all objects in the environment. *Planner* repeats the cycle of (1) receiving the sensed information of moving obstacles from *Simulator*, (2) calculating an appropriate motion of the robot, and (3) sending the motion command to *Simulator*. The interface protocol between the two systems is designed to emulate the one between *Planner* and our real robot so that the planning algorithms developed in this simulation environment can be used for the real robot with a minimum modification.

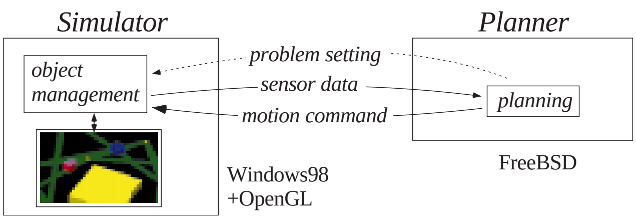


Fig. 7: Realtime simulation environment.

Simulation Result

Fig. 8 shows a simulation result. There are a static obstacle and a moving obstacle in the environment and the robot considers two routes, among which the left one is shorter. The figure shows the movement of the robot and the obstacle until the robot reached the goal point. Since the left route is shorter, the robot started toward the left route; as the situation became more certain, the evaluation of the right route went up, while that of the left one fell down. So the robot gradually shifted its direction towards the right route and, at time $t = 17$, it committed to the right route and followed it to the goal point. The parameters used in this simulation are: obstacle velocity range is $[4.5 \pm 1.0][cm/s]$; the robot velocity is $7.5[cm/s]$; the variance of uncertainty in measuring distance is $6.25e - 7 \cdot d^4$ (d is the distance to the obstacle); the length of the left and the right route are $317.0[cm]$ and $332.0[cm]$, respectively. The number of candidate motions is 5.

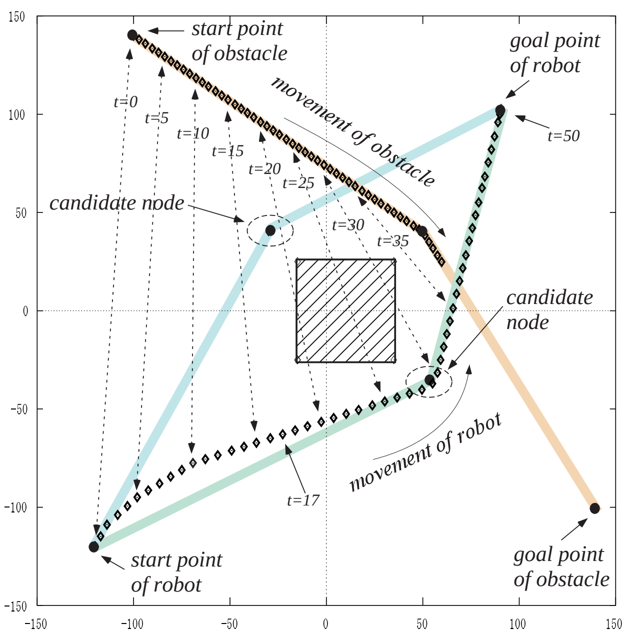


Fig. 8: A simulation result.

Fig. 9 shows a set of predicted states (the distribution of the predicted time of the obstacle arriving at the crossing) at time $t = 4$ for the left (see Fig. 9(a)) and the right (see Fig. 9(b)) route, respectively. Although the expected waiting time of the left route is longer than that of the right route, since the left route is shorter, the two routes are competing in terms of time; thus, the robot selects at this time point to move toward exactly the middle of the directions to both routes.

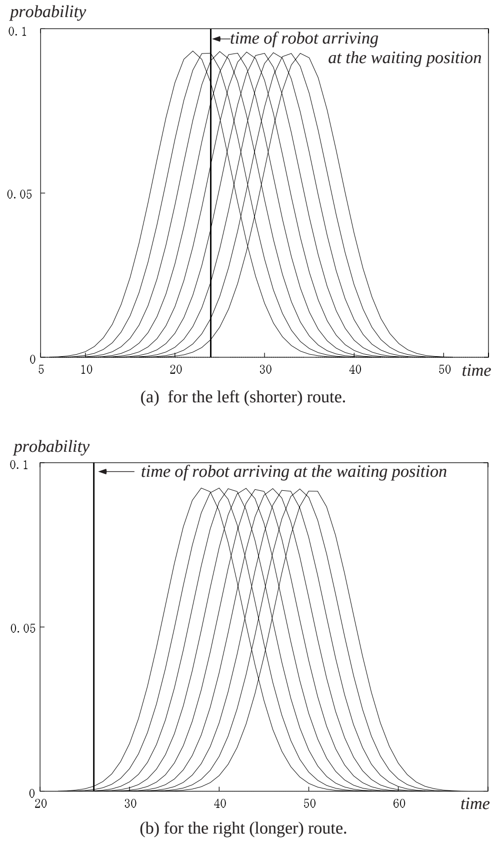


Fig. 9: Two sets of arrival time probabilities and the time robot's arrival at the waiting position.

5 CONCLUSIONS AND DISCUSSION

This paper have proposed a novel method of generating mobile robot motion in a dynamic environment considering the motion uncertainty of moving obstacles. Using the probabilistic model of the uncertainty of motion itself and that of observation uncertainty, we can model the gradual reduction of the uncertainty in motion prediction, which we usually experience in many situations. Based on this probabilistic model, the method repeatedly selects the best motion in a decision-theoretic manner, that is, by one-step look-ahead search in a probabilistic search tree.

An immediate extension is to consider the path ambiguity. At a junction of paths, an obstacle may take any branch and this is also difficult for the robot to know in advance. By applying a probabilistic method, as used in this paper, to prediction of an obstacle's path, we will be able to estimate the distribution of the time of the obstacle arriving at the branch and the probability of taking each path at a time. Similarly to the case of motion prediction, the situation (which path to take) will be more certain as time advances.

Another important extension is to consider the planning cost. Currently, since planning is simple and very fast, the planning time is negligible. However, when the planning cost becomes higher due to, for example, increase of the number of possible situations to examine, we would have to consider the tradeoff between planning quality and planning cost [9]. This tradeoff will be very important especially in a dynamic environment because the time allowed for planning may be very short; an interesting example would be the case of an obstacle approaching the robot.

References

[1] T. Tsubouchi and M. Rude. Motion Planning for Mobile Robots in a Time-Varying Environment. *J. of Robotics and Mechatronics*, Vol. 8, No. 1, pp. 15–24, 1996.

[2] S. Ishikawa. A Method of Indoor Mobile Robot Navigation by Using Fuzzy Control. In *Proc. 1991 IEEE/RSJ Int. Workshop on Intelligent Robots and Systems*, pp. 1013–1018, 1991.

[3] K. Fujimura. Time-Minimum Routes in Time-Dependent Networks. *IEEE Trans. on Robotics and Automation*, Vol. 11, No. 3, pp. 343–351, 1995.

[4] T. Tsubouchi and S. Arimoto. Behavior of a Mobile Robot Navigated by an Iterated Forecast and Planning Scheme in the Presence of Multiple Moving Obstacles. In *Proc. 1994 IEEE Int. Conf. on Robotics and Automation*, pp. 2470–2475, 1994.

[5] A. Inoue, K. Inoue, and Y. Okawa. On-line Motion Planning of an Autonomous Mobile Robot to Avoid Multiple Moving Obstacles Based on the Prediction of Their Future Trajectories. *J. of Robotics Soc. of Japan*, Vol. 15, No. 2, pp. 249–260, 1997. (in Japanese).

[6] Y.-H. Liu and S. Arimoto. Path Planning Using a Tangent Graph for Mobile Robots Among Polygonal and Curved Objects. *Int. J. of Robotics Research*, Vol. 11, No. 4, pp. 376–382, 1992.

[7] M. Fisz. *Probability Theory and Mathematical Statistics*. Wiley, 1963.

[8] J. Miura and Y. Shirai. Vision and Motion Planning for a Mobile Robot under Uncertainty. *Int. J. of Robotics Research*, Vol. 16, No. 6, pp. 806–825, 1997.

[9] J. Miura and Y. Shirai. Vision-Motion Planning for a Mobile Robot considering Vision Uncertainty and Planning Cost. In *Proceedings of the 15th Int. Joint Conf. on Artificial Intelligence*, pp. 1194–1200, Nagoya, Japan, August 1997.