Adaptive Robot Speed Control by Considering Map and Motion Uncertainty

Jun Miura [1], Yoshiro Negishi, and Yoshiaki Shirai

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

Abstract

This paper describes an adaptive robot speed control method for safe *and* efficient navigation in unknown environments. Safety and efficiency are usually in a trade-off relationship. Moving fast increases efficiency but decreases safety due to low reliability in localization and environment recognition; moving slowly decreases efficiency but increases safety. Speed control considering this trade-off is important in the following two cases. (1) When the robot enters a narrow free space, it needs to control the speed to avoid any collisions by considering the motion uncertainty. (2) When the robot enters a region whose vacancy (i.e., being free) has not been decided yet, it needs to control the speed so that it can observe the region sufficiently to be confident with the vacancy of the region. This paper proposes a simple but effective strategy for such a speed control that the robot selects the *safest fast speed*. To adopt this strategy, we define criteria for judging whether a speed is safe for the above two cases. The proposed method successfully made the robot move around in unknown static environments with adaptively controlling the speed.

Key words: Mobile robot, navigation in unknown environments, adaptive speed control, map generation.

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

Article published in Robotics and Autonomous Systems 54 (2006) 110–117

1 Introduction

Mobile robot navigation in unknown environments has been one of the active research areas in robotics. Many previous works have been concerned with localizing a robot and/or generating a reliable map from uncertain data obtained by internal (e.g., odometry) and external (e.g., vision and sonar) sensors [1–3]. Their main focuses have been on developing methodologies for realizing accurate and reliable localization and/or map generation, given a sequence of sensor data and a set of sensor uncertainty models.

Sensor planning is also important in designing sensor-based robots operating under various uncertainties. In the mobile robot domain, some dealt with observation planning issues, especially planning methods for increasing the quality of maps (e.g., [4]) or for exploring unobserved spaces (e.g., [5]) or both (e.g., [6,7]). The objectives of these planning methods are, however, mainly for obtaining more information. This paper, on the other hand, considers another aspect of planning, that is, a planning for increasing the efficiency of navigation.

Mobile robot navigation can roughly be divided into two levels: selecting a route and following the selected route. For the first level, several works deal with observation planning for selecting an efficient route using probabilistic models of sensor and motion uncertainties (e.g., [10,11]).

For the second level (i.e., following a route), there are many works on visual feedback motion control (e.g., [12]); they are mainly concerned with development or application of control theories.

A certain level of accuracy in robot localization is indispensable for a safe navigation. Several works deal with landmark selection problems in which an appropriate set of landmarks is selected for minimizing the predicted localization uncertainty (e.g., [13,14]). If we consider the cost of sensing, however, observing uncertainty-minimizing landmarks may not be optimal in terms of the cost of reaching a destination; so it is necessary to consider what accuracy is needed in each situation.

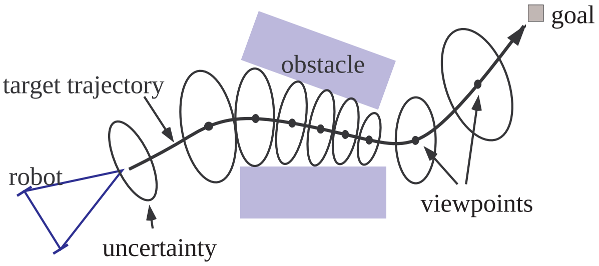


Fig. 1. An example of adaptive speed control. Viewpoint intervals are short in a narrow space.

A general goal of navigation methods is to realize a safe *and* efficient movement of the robot. *Safety* usually means that the robot does not collide with obstacles. On the other hand, *efficiency* here means that the robot can reach a destination in a small amount of time. These two requirements, safety and efficiency, are usually in a trade-off relationship. If the robot moves fast to increase efficiency, the number of observations usually decreases and uncertainties in localization and environment recognition thus increase; this most probably decreases safety. If the robot moves slowly to increase safety, efficiency will decrease.

To cope with the trade-off, we proposed the strategy that the robot moves at the *fastest safe speed* [15]. In this strategy, by assuming a constant time for one observation, we first relate the speed with the frequency of observations and thus reliability of localization or environment recognition. Then, we define criteria for judging whether a speed is safe. Finally the robot selects the fastest safe speed. In [15], we used the following criterion. Usually the deviation from a target trajectory increases as the robot moves faster (i.e., the interval becomes longer). If the robot can recover from the worst (i.e., most deviated) position, which is predicted for a speed, to the target trajectory without collision, the speed is judged as safe. The robot applied the criterion to all candidate speeds, and se-

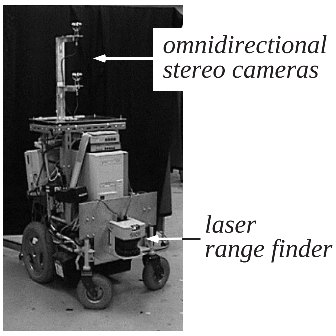
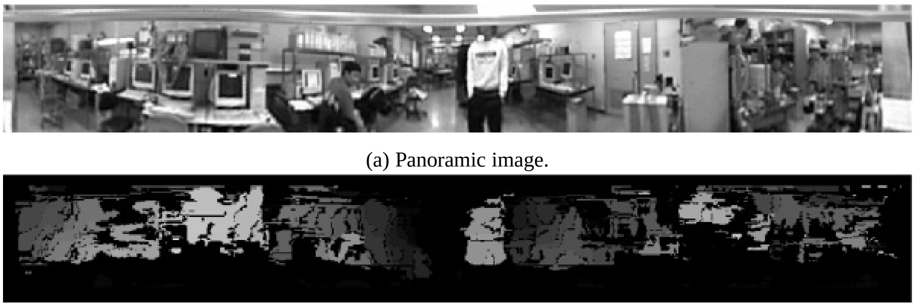


Fig. 2. Our mobile robot.



(a) Panoramic image.

(b) Panoramic disparity image obtained from (a).

Fig. 3. Omnidirectional stereo generates a panoramic disparity image.

lected the fastest safe one. Based on this simple but effective strategy, we can naturally realize an adaptive speed control as shown in Fig. 1; such a control scheme is analogous to what we are doing in driving cars.

The speed control method developed in [15] was for *completely-known* environments with *given* trajectories. This paper extends that method to unknown environments. In unknown environments, since the robot moves while incrementally generating a map, not only the distance to recognized obstacles but also that to *undecided regions*, which have not been sufficiently recognized as free, needs to be considered in robot speed control; it is not desirable for the robot to enter such an undecided region from the viewpoint of safety. An on-line path planning is also necessary in navigation in unknown environments.

Several speed control methods have been developed in path planning (e.g., [8]) or reactive control (e.g., [9]) contexts. These methods mainly focus on motion planning for given obstacle information, and do not explicitly consider the trade-off between safety and efficiency.

The rest of the paper is organized as follows. Section 2 briefly explains the map generation method that we have already developed. The process and the result of the map generation are used for the two kinds of speed limitations. Section 3 describes the speed limitation based on the distance to undecided regions. Section 4 describes the speed limitation based on the distance to obstacles represented in the robot local coordinates. Section 5 describes the speed control method considering both limitations, and section 6 shows experimental results using our mobile robot. Section 7 concludes the paper and discusses several future works.

2 Map Generation by Integrating Omnidirectional Stereo and Laser Range Finder

This section briefly describes our map generation method using an omnidirectional stereo and a laser range finder. Please refer to [16] for more details. Our speed control method is designed by considering what result is obtained and how it is obtained by this map generation method.

2.1 Two range sensors

Our stereo system uses a pair of vertically-aligned omnidirectional cameras (see Fig. 2). The system can generate a disparity image of 360×50 pixels in size and 40 pixels in disparity range in about every $0.18\,[s]$ (see Fig. 3). We also use a SICK laser range finder (LRF), which is set at the front of the robot so that it scans the horizontal plane at the height of $35\,[cm]$ from the floor (see Fig. 2). The resolution used is

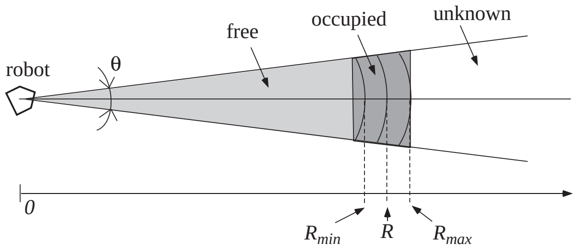


Fig. 4. Determination of cell attributes.

$1.0\,[deg]$ per point (i.e., 181 measurements for 180 degrees).

2.2 Probabilistic occupancy maps

We keep a probabilistic occupancy map [1] for each sensor. The map is the size of $10 \times 10\,[m]$ and divided into 200×200 cells. Each cell is the size of $5 \times 5\,[cm]$ and has the probability of obstacle existence at the corresponding position. The map is described in the robot local coordinates, centered at the current robot position and aligned with the robot orientation.

Temporal integration of sensor data is carried out for each map separately using *forward sensor model*[17,18]. We adopt the *independence assumption*, that is, update the probability of a cell independently of other cells. This assumption seems reasonable when a sensor has a fairly fine angular resolution.

From one observation, we determine the attribute of each cell as shown in Fig. 4. The figure shows the attribute determination for a region within one angular resolution. R is the observed distance (by omnidirectional stereo or LRF) to the nearest obstacle, and R_{min} and R_{max} indicate the uncertainty in range measurement [19].

Let O be the event that an obstacle is detected. O occurs at *occupied* cells; the inverse event $\overline{O}$ occurs at *free* cells. For these cells, the update of the probability is carried out as follows. Let E be the event that an obstacle exist, and let $P(E)$ be the probability that an obstacle exist (at a cell). The posterior probability to be obtained by integrating a new observation is given by the conditional probabilities: $P(E|O)$ and $P(E|\overline{O})$. These probabilities are calculated by the Bayes' theorem as follows:

$$\begin{aligned} P(E|O) &= \frac{P(O|E)P(E)}{P(O|E)P(E) + P(O|\overline{E})P(\overline{E})}, \\ P(E|\overline{O}) &= \frac{P(\overline{O}|E)P(E)}{P(\overline{O}|E)P(E) + P(\overline{O}|\overline{E})P(\overline{E})}, \end{aligned}$$

where $P(E)$ is the prior probability and $\overline{E}$ is the proposition that an obstacle does not exist. Among the terms in the above equations, $P(O|E)$ and $P(O|\overline{E})$ are observation models [16]; $P(\overline{O}|E) = 1 - P(O|E)$; $P(\overline{O}|\overline{E}) = 1 - P(O|\overline{E})$; $P(\overline{E}) = 1 - P(E)$. The temporal integration operation for each cell is performed independently of the others (*the independence assumption*).

To integrate observations at different robot positions, before performing the above probability updates, we transform the map in the previous robot local coordinates into the current local coordinates using the estimated ego-motion [16].

2.3 Integration of two maps

The two probabilistic maps are integrated as follows. Since the two sensors may detect different objects or different parts of an object at a 2D position, a direct integration of probability values by the Bayes' rule is not appropriate [16]. We, therefore, first classify each cell of a map into four classes and then integrate the classification results into the free space map.

The classification is carried out in two steps. In the first step, we use two thresholds. If the occupancy probability of a cell is larger than the higher threshold (currently, 0.7), the cell is classified as *obstacle*; if the probability is less than the lower threshold (currently, 0.2), the cell is classified as *free space*; otherwise, classified as *undecided*. We further classify *undecided* cells into two subclasses, *undecided with observation* (mostly for textureless objects in stereo, which should be observed using only the LRF) and *undecided without observation* (for the case of unobserved regions), using the number of observations of each cell position.

From the classification results, if both maps says a cell is *free space*, or if one map says *free space* and the other says *undecided with observation*, then the cell is determined to be free in the final map. Otherwise, the cell is determined to be occupied. The resultant free space map is used for the path planning of the mobile robot.

2.4 Map generation example

Fig. 5 shows an example movement of our robot. Fig. 6 shows the maps generated after the movement. In the probabilistic maps, brightness indicates the probability of each cell being occupied by an obstacle. In the free space map, white areas indicate free spaces, among which only the one around the robot is significant. The maps are drawn in the robot coordinates centered at the robot's final position. The table in front of the robot was correctly recognized by the stereo, while the LRF detected only its legs. On the other hand, the recognition by the stereo of the region near the door on the right failed at many positions because features are scarce on the door, while the LRF correctly recognized the region. In spite of recognition failures by one of the sensors at several positions, the integrated map reasonably represents the free space around the robot.

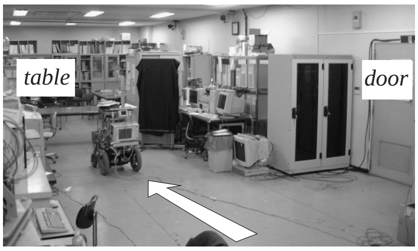


Fig. 5. An example scene.

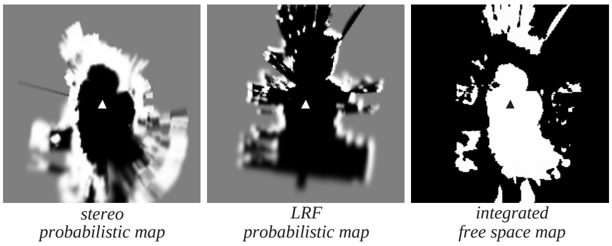


Fig. 6. Probabilistic maps and a free space.

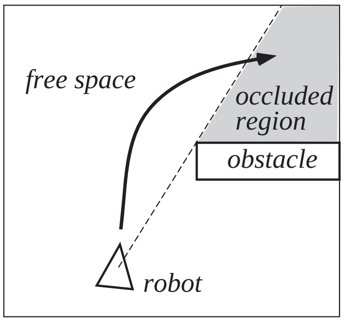


Fig. 7. A robot entering an occluded region.

3 Speed Limitation by the Distance to Undecided Regions

This section describes a method for limiting the robot speed so that the robot does not enter an *undecided region* whose vacancy has not been sufficiently decided.

3.1 Basic strategy

Fig. 7 illustrates an example situation where a robot is entering an occluded region; the vacancy of the occluded region or the region near that occluded is still undecided. As the robot moves, a free region gradually expands with the accumulation of newly observed data. Due to observation uncertainties, in order to be confident with the vacancy of a cell (or a region), the robot needs to observe it several times.

One motion strategy of the robot is to reach an undecided region at the highest speed and observe there; but this may result in an undesirable sudden acceleration/deceleration. We, therefore, control the robot speed so that the robot can make a number of observations large enough to be confident with the vacancy of the region until it reaches there. We call such a speed a *safe speed*.

We define a safety criterion for judging whether a speed is safe as follows. Let N be the necessary number of observations of an undecided region; in other words, we suppose that if the robot observes the region no less than N times, then the robot becomes confident with the vacancy of the region. Also let d be the distance to the region and T be the time for one observation (considered to be constant). If the robot speed is v, it can observe the undecided region d/vT times before reaching the region. The safety criterion is, therefore, given by:

$$\frac{d}{vT} \geq N.$$

This inequality represents the speed limitation by the distance to undecided regions.

To adopt this safety criterion, we need to determine N and d. N is determined by considering the observation and the map uncertainty model; d is calculated from the result of a path planning. The next two subsections will explain how to calculate N and d.

3.2 Determining the Necessary Number of Observations for Obtaining Confidence

To determine the necessary number of observations (N), we examined how the probability of obstacle existence changes as more observations are obtained, using the observation uncertainty models and the map generation method. A typical case examined is the one where the robot is initially $500\,[cm]$ (which is equal to the maximum observable distance of the stereo) distant from a front object and the robot moves at $50\,[cm/frame]$ while observing the object using stereo. Fig. 8 shows the relationship between the number of observations and the probability $P(E)$ that an object actually exists at cells in front of the object in the case where the initial probability is 0.5 (i.e., completely undecided). In this case, five observations are needed for the robot to be confident with the vacancy of the cells (currently, the threshold for judge the vacancy is 0.2).

We also examined experimental data which were obtained by moving the robot along a route similar to the one in Fig. 14. For each cell which was classified as *free space*, we examined how many observations were carried out for changing the cell from completely undecided to free, and summarized the examination result into the histogram shown in Fig. 9. The mean of the necessary number of observations is about five.

From the above results, we consider that the necessary number of observations, N, determined from the typical case analysis is comparable to that obtained from the experiments; so we currently use five as N. If we use a different sensor and a map generation method, the appropriate value of N may be different from the current one. In such a case, a reasonable value could be determined only by a typical case analysis.

The robot recognizes a currently-undecided region as free by accumulating observations of the object be-

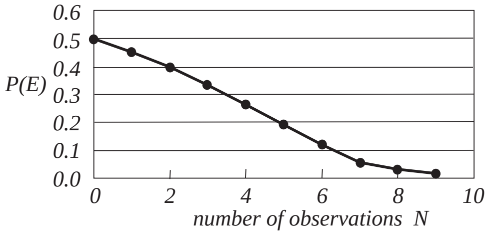


Fig. 8. Decrease of the probability of obstacle existence according to the number of observations.

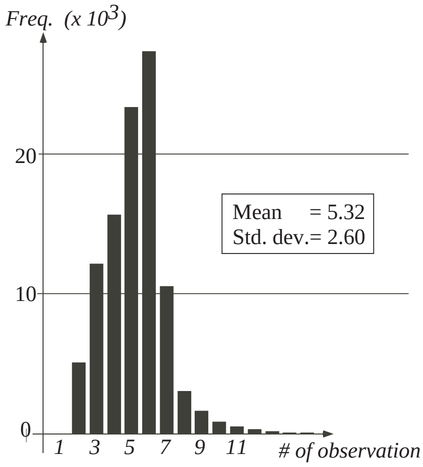


Fig. 9. Histogram of the number of observations for recognizing free spaces.

hind the region. If the distance to the object is not correctly measured due to, for example, failure of stereo matching, the actual number of observations of the region may become larger than expected, and the distance to the undecided region may thus become short. The histogram in Fig. 9 includes such cases. Even if the distance is shorter than expected, however, the robot can still move safely because the robot speed is controlled on-line according to the *actual* distance to the region.

3.3 Determining the Distance to an Undecided Region by Path Planning

The distance d to an undecided region is given not by the Euclidean distance but by the distance along a robot path from the current position to the region; so the path needs to be calculated in the current free space. Any path planners can be used because the speed control strategy proposed in this paper is independent of the path planner used. We here describe our heuristic path planner which considers the driving mechanism of our mobile robot.

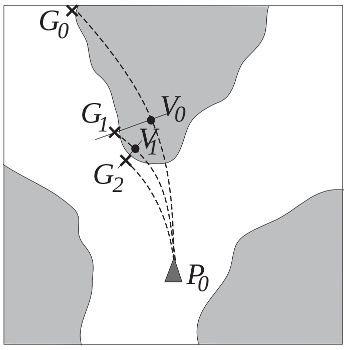


Fig. 10. Find a feasible via point.

We currently give the robot a destination in the world coordinates; the robot transforms it to the local coordinates using the estimated robot position. If the destination is in the local map, the robot uses it for path planning. Otherwise, the robot selects a temporary destination for path planning in the current free space which is on the boundary between the free space and an undecided region and is nearest to the given destination. Considering the motion constraint of our robot driven by two powered wheels, we approximately represent a path of the robot by a sequence of circular paths. We use the length of a generated path as the distance d to an undecided region.

Fig. 10 illustrates the process of path planning. The planner first calculates the circular path which connects the current robot position (P_0 in the figure) and a destination (G_0) and satisfies the orientation constraint at the current position (arc $P_0V_0G_0$). If this path is safe, it is selected. Otherwise, the planner first searches for the point on the circular path which is farthest from the free space (V_0 is selected) and draws a line perpendicular to the tangent line of the circular path there, and selects a temporary des-

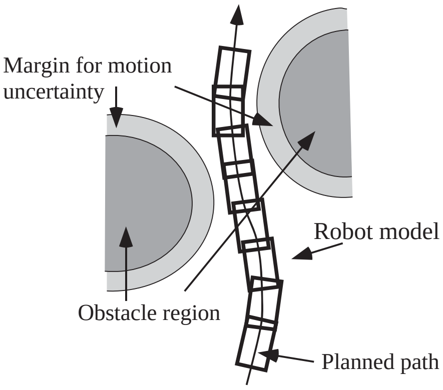


Fig. 11. Collision check of a path.

tination (G_1) on the line in the free space. For this temporary destination, the planner repeats the same operation until a safe circular path is found (try arc $P_0V_1G_1$, select G_2, and find P_0G_2). Then this process is iterated with the selected via point (G_2) being the initial position. Currently we limit the maximum number of iterations of this process to two; in a complex environment, the endpoint of the planned path may not be the given destination. The robot repeatedly plans a new path every time the free space map is updated, and the path is used for determining the speed and the turning radius of the robot for the current feedback cycle.

In the path planning, original obstacle regions are expanded by two types of *margins*; one is for considering the motion uncertainty and the other is for the robot size. Since the motion uncertainty depends on the robot speed, we estimate the margin for each candidate speed in advance (see Section 4). Concerning the margin for the robot size, the planner uses the robot width as the margin in planning a path, and then verifies the safety of the path by checking collision on points with a certain interval on the path using the robot shape, as shown in Fig. 11. If a collision is detected, the planner goes back to the selection of via points described above.

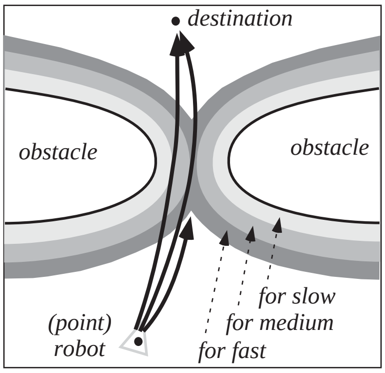


Fig. 12. Path planning for a narrow space with multiple speeds.

4 Speed Limitation by the Distance to Obstacles

This section describes a method for limiting the robot speed based on the distance to nearby obstacles. Since our previous method [15] was for completely-known environments and given trajectories, the collision check was relatively easy. In the current navigation problem, however, since the robot determines not only the speed but also the path to follow, we needs to develop a method for determining both simultaneously.

4.1 Path Planning for Several Robot Speeds

We determine the safety of a speed by judging whether the robot can pass through a narrow space using the speed because we are mainly concerned with speed control in such spaces.

A faster speed results in a larger motion uncertainty (i.e., a larger margin for path planning) and thus may make it impossible for the robot to pass through a narrow space. To find the fastest safe speed, we apply the path planner to the current free space for all of possible robot speeds (i.e., multiple margins for motion uncertainty) and generate the list of planned paths in the ascending order of their lengths. Then we see if there is a gap in the list of the lengths; the existence of such a gap implies that there is a narrow space in front of the robot, and that the robot cannot pass through it at its fastest speed. In the case of Fig. 12, for example, the robot can generate paths to the destination through a narrow space for *slow* and *medium* speeds thanks to relatively small margins for motion uncertainty, while it fails to generate such a path for *fast* speed; as a result, the path length is longer for the first two speeds and is shorter for the last one, and there is a gap in the list of path length.

If there is a gap larger than a certain threshold, we divide the path length list into two groups at the gap, and consider that the speeds in the longer group are safe. Otherwise, all speeds are considered safe. Then the path for the maximum safe speed is selected. In the case of Fig. 12, for example, the path for speed *medium* is selected. At present, we use a set of four speeds for our robot.

4.2 Selection of Safe Speeds for the Path

The generated path is for a safe speed; the robot, however, may be able to move faster at the first part of the path if the obstacles are distant enough. So we check the collision possibility on the path for the speeds faster than the corresponding safe speed, and if a faster speed is safe during at least two cycles of visual feedback movement, the speed is considered to be safe. In Fig. 12, for example, the robot may be able to move at speed *fast* for a moment on the path generated for speed *medium*.

5 Speed Selection Based on Two Speed Limitations

The robot determines the speed by considering the above two speed limitations. Fig. 13 shows the flow of speed control. The robot first obtains sensor information, updates the map, and sets the temporary destination in the local coordinates. Next, the robot calculates the paths for the candidate speeds, and determines the path and a set of safe speeds, using the method described in Section 4. Finally, the robot selects the fastest speed among them which satisfies the limitation by the distance to undecided regions, using the method described in Section 3. These processes are repeated every time new sensor data are obtained.

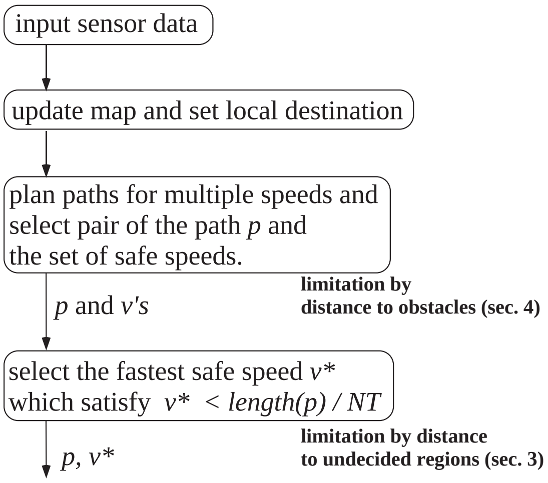

input sensor data
update map and set local destination
plan paths for multiple speeds and select pair of the path *p* and the set of safe speeds.
p and *v's*
limitation by distance to obstacles (sec. 4)
select the fastest safe speed *v** which satisfy *v** < *length(p) / NT*
p, *v**
limitation by distance to undecided regions (sec. 3)

Fig. 13. Flow of speed selection procedure.

6 Experimental Results

Fig. 14 illustrates a navigation experiment. The environment we used for the experiment include various situations such as a simple corridor with flat walls and an area with many complex-shaped objects in a room; it is sufficiently complicated for demonstrating the performance of our adaptive speed control method.

We gave the robot only a sequence of four destinations in the world coordinates, and the robot autonomously moved on a path as shown in the figure by switching the destinations one after another, with generating maps and adaptively controlling the speed. The observation cycle, including the stereo

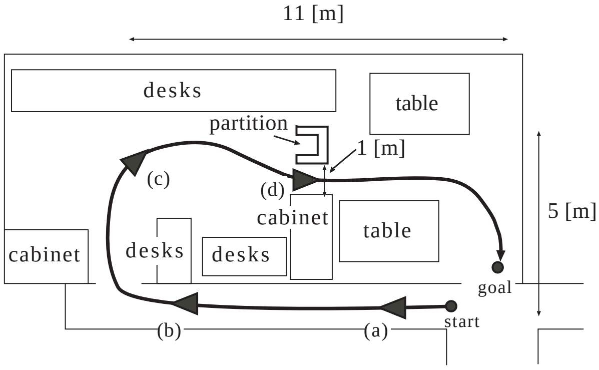


Fig. 14. Experimental setup.

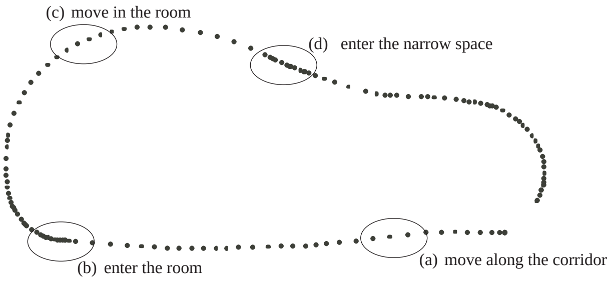


Fig. 15. Trace of observation points.

and LRF observation, the map updating, and the speed control, is about $0.3\,[sec]$. The robot has four candidate speeds, $1.1\,[m/s]$, $0.7\,[m/s]$, $0.35\,[m/s]$, $0.2\,[m/s]$; the maximum speed is almost the same as the walking speed of an ordinary person. The margins for the speeds are empirically set to $30\,[cm]$, $20\,[cm]$, $10\,[cm]$, $5\,[cm]$, respectively.

Fig. 15 shows the trace of observation points (where the robot obtained stereo and LRF data) using our LRF-based ego-motion estimation method [16]. Points (a)-(d) in the figure correspond to those in Fig. 14, respectively. Since the observation cycle is almost constant, the interval between observation points indicates the robot speed. Although the estimated position includes accumulated errors, the result of Fig. 15 is enough for examining the changes of the robot speed, since the accumulated error is sufficiently small within a short duration.

Fig. 16 shows generated free space maps and planned paths with snapshots at the above four points (a)-(d). The robot position is at the center of the maps. At point (a), the planned path was long enough to enable the robot to move at its fastest speed. At point (b), since the observed area was limited by a wall (that is, the distance to an undecided region is short), the robot planned only a short path and slowed down. At point (c), the robot was able to observe a wide area, so it moved fast again. At point (d), although the robot had a view beyond a narrow space between a partition and a cabinet, it slowed down because only the slowest speed was feasible for passing through the narrow space. Note that in cases (b) and (d), the robot slows down *before* reaching the undecided or the narrow regions (also see Fig. 15); controlling the speed beforehand in such a way is important for smooth movements of the robot.

The total moving distance was about $30\,[m]$ and the total time was about $45\,[sec]$. When the robot moved at the lowest speed throughout the route, it took about $150\,[sec]$. The proposed speed control method improved the efficiency of navigation about three times with keeping the same safety.

7 Conclusion and Discussions

This paper has proposed an adaptive robot speed control method under the map and the motion uncertainty. To solve the trade-off between safety and efficiency in navigation, we first defined the criteria for judging whether a robot speed is safe. In the case of moving towards undecided regions, we determined the number of observations needed for recognizing the vacancy of undecided regions with confidence, from the sensor uncertainty model and the experimental data; the number is then used with the distance to an undecided region for defining the criterion. In the case of passing through a narrow space, another criterion is defined which uses the results of path planning with several robot speeds. Once the criteria are defined, the robot selects the fastest safe

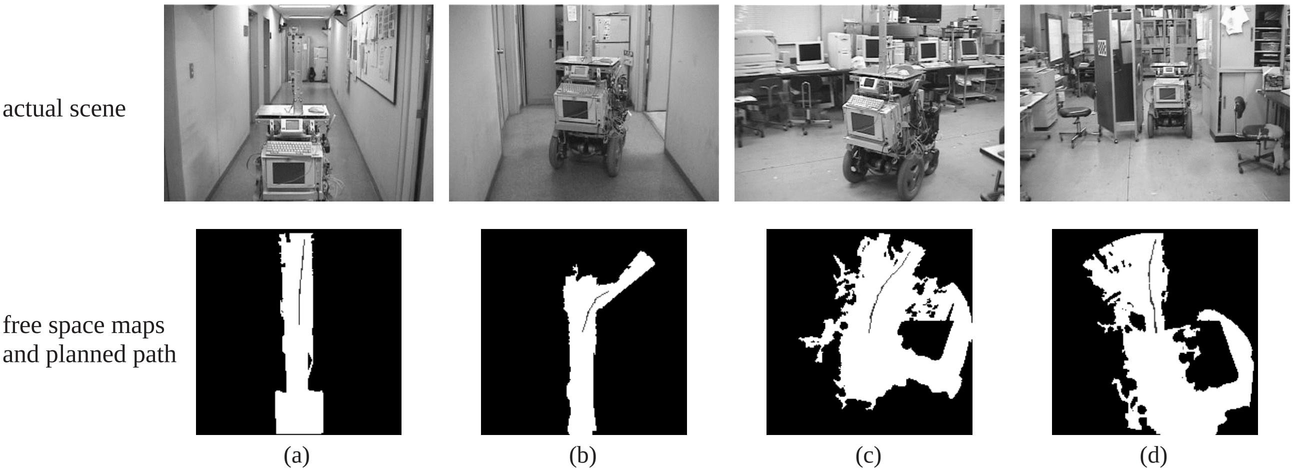


Fig. 16. Experimental results on free space maps and planned path. Black regions indicate obstacles and are expanded by the margins for the motion uncertainty and the robot size in path planning.

speed. This simple strategy has been shown to be effective in controlling a real mobile robot under uncertainties of observation and motion.

Currently, we treat the reliability of undecided regions uniformly; that is, the necessary number of observations is set to the same for all undecided regions, regardless of the number of observations so far. The reliability of each cell should differ from each other, depending on the observation history of the cell. A future work is to consider this factor in determining the necessary number of subsequent observations for each cell.

The current method assumes static environments, where a future situation such as the confidence of the vacancy of a cell (i.e., the probability of obstacle existence at a cell) and the distance to obstacles is sufficiently predictable for speed control. Another future work is to cope with dynamic environments where moving obstacles such as walking persons make such predictions and speed control more difficult.

References

[1] A. Elfes, Sonar-based real-world mapping and navigation, Int. J. of Robotics and Automat. 3 (3) (1987) 249–265.

[2] J. Leonard, H. Durrant-Whyte, Simultaneous map building and localization for an autonomous mobile robot, in: Proceedings of 1991 IEEE/RSJ Int. Workshop on Intelligent Robots and Systems, 1991, pp. 1442–1447.

[3] S. Thrun, W. Burgard, D. Fox, A probabilistic approach to concurrent mapping and localization for mobile robots, Machine Learning 31 (1998) 29–53.

[4] W. Burgard, D. Fox, S. Thrun, Active mobile robot localization, in: Proceedings of the 15th Int. Joint Conf. on Artificial Intelligence, 1997, pp. 1346–1352.

[5] H. Choset, J. Burdick, S. Walker, K. Eiamsa-Ard, Sensor based exploration: Incremental construction of the hierarchical generalized voronoi graph, Int. J. of Robotics Research 19 (2) (2000) 126–148.

[6] H. Feder, J. Leonard, C. Smith, Adaptive

mobile robot navigation and mapping, Int. J. of Robotics Research 18 (7) (1999) 650–668.

[7] A. Makarenko, S. Williams, F. Bourgault, H. Durrant-Whyte, An experiment in integrated exploration, in: Proceedings of 2002 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2002, pp. 534–539.

[8] Y. Hwang, N. Ahuja, Gross motion planning – a survey, ACM Computing Surveys 24-3 (1993) 219–292.

[9] R. Arkin, Integrating behavioral, perceptual, and world knowledge in reactive navigation, Robotics and Autonomous Systems 6 (1990) 105–122.

[10] H. Hu, M. Brady, A bayesian approach to real-time obstacle avoidance for a mobile robot, Autonomous Robots 1 (1) (1994) 69–92.

[11] J. Miura, Y. Shirai, Vision and motion planning for a mobile robot under uncertainty, Int. J. of Robotics Research 16 (6) (1997) 806–825.

[12] R. S. Oropeza, M. Devy, V. Cadenat, Controlling the execution of a visual servoing task, J. of Intelligent and Robotic Systems 25 (4) (1999) 357–369.

[13] N. Roy, W. Burgard, D. Fox, S. Thrun, Coastal navigation - mobile robot navigation with uncertainty in dynamic environments, in: Proceedings of 1999 IEEE Int. Conf. on Robotics and Automation, 1999, pp. 35–40.

[14] K. Kidono, J. Miura, Y. Shirai, Autonomous visual navigation of a mobile robot using a human-guided experience, Robotics and Autonomous Systems 40 (2-3) (2002) 121–130.

[15] I. Moon, J. Miura, Y. Shirai, On-line viewpoint and motion planning for efficient visual navigation under uncertainty, Robotics and Autonomous Systems 28 (2-3) (1999) 237–248.

[16] J. Miura, Y. Negishi, Y. Shirai, Mobile robot map generation by integrating omnidirectional stereo and laser range finder, in: Proceedings of 2002 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2002, pp. 250–255.

[17] Y. Sawano, J. Miura, Y. Shirai, Man chasing robot by an environment recognition using stereo vision, in: Proceedings of the 2000 Int. Conf. on Machine Automation, 2000, pp. 389–394.

[18] S. Thrun, Learning occupancy grids with forward models, in: Proceedings of 2001 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2001, pp. 1676–1681.

[19] H. Koyasu, J. Miura, Y. Shirai, Realtime omnidirectional stereo for obstacle detection and tracking in dynamic environments, in: Proceedings of the 2001 IEEE/RSJ Int. Conf. on Intelligent Robots and Sysetms, 2001, pp. 31–36.