Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 2737-2742, Sendai, Japan, Sep./Oct. 2004.

Mobile Robot Navigation in Unknown Environments Using Omnidirectional Stereo and Laser Range Finder

Yoshiro Negishi, Jun Miura, and Yoshiaki Shirai
Department of Computer-Controlled Mechanical Systems, Osaka University
{negishi,jun,shirai}@cv.mech.eng.osaka-u.ac.jp

***Abstract*— This paper describes the navigation of a mobile robot in unknown static environments using an omnidirectional stereo and a laser range finder. The robot detects obstacles by the sensors, estimates the ego-motion, integrates the sensor data to generate a probabilistic occupancy map, and plans a safe motion. This paper focuses on the ego-motion estimation and the data integration for map generation. We extend our previous methods to increase the robustness of navigation. Experimental results show the feasibility of our navigation method.**

I. Introduction

Vision-based navigation of mobile robots is one of the active research areas in robotics. Many useful techniques have been developed such as landmark-based navigation [5], scan matching-based navigation [9], statistical localization [4], map generation [11], and simultaneous localization and mapping [1], [8].

We have been working on an autonomous robot system with omnidirectional stereo [6], [7]. We have developed a map generation method using the omnidirectional stereo and a laser range finder [10]. This map generation method has an advantage of being able to recognize complex environments by compensating the drawback of one sensor by the data from the other.

The method basically works as follows; if an object is detected, the corresponding grids of the map increase their probabilities of obstacle existence and the grids corresponding to the space in front of the object decrease the probabilities. In this way, we expect that erroneously observed obstacles are deleted by the subsequent observations of the true obstacles behind. The method considered only 2D positions of obstacles because a full 3D reconstruction will be very costly. This, however, sometimes caused a problem that a true obstacle was erroneously *deleted* by observing other obstacles behind it. This paper, therefore, improves the method so that true obstacles are not deleted by considering the heights and the visibilities of objects.

For temporal integration of sensor data obtained while a robot moves, ego-motion estimation between observation points is necessary. Our previous estimation method used the correspondence between point features in two consecutive range data, and was not fully reliable due to occasional false correspondences. This paper also improves the ego-motion estimation method such that it robustly searches for the correspondences and it uses line features when point features are scarce.

These two improvements increase the robustness of navigation of our robot. The experimental results show the effectiveness of the improved system in the navigation in unknown environments.

II. Previous Map Generation Method

This section briefly describes our previous map generation method using an omnidirectional stereo and a laser range finder. Please refer to [10] for the details.

A. Two Range Sensors

Our stereo system uses a pair of vertically-aligned omnidirectional cameras (see Fig. 1). The system can generate a disparity image of 360x50 in size and 40 in disparity range in every about $0.1\,[s]$ (see Fig. 2). 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. 1). The resolution used is $1.0\,[deg]$ per point (i.e., 181 measurements for 180 degrees).

B. Map Generation Algorithm

We keep a separate map for each sensor. We use as a map a probabilistic occupancy grid representation [2]. 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 [10]. We, therefore, perform temporal integration of sensor data for each map separately, and integrate them using a set of logical rules.

1) Temporal integration of sensor data: The probability of each grid of a map is updated as follows. From one observation, we determine the attribute of each grid: *occupied*, *free*, and *unknown* (see Fig. 3). 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[1]. The region between R_{min} and R_{max} is labeled as *occupied*. The region before the occupied region is labeled as *free*. The region behind the occupied region is labeled as *unknown*.

Let O be the event that an obstacle is detected. O occurs at *occupied* grids; the inverse event $\overline{O}$ occurs at *free* grids. For these grids, 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 grid). The new probability map to be obtained by integrating a new observation is given by the conditional probabilities:

[1]Refer to [6] for the uncertainty estimate of omnidirectional stereo. The uncertainty in LRF measurement is constant regardless of the measured value.

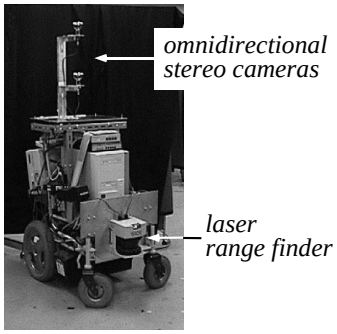
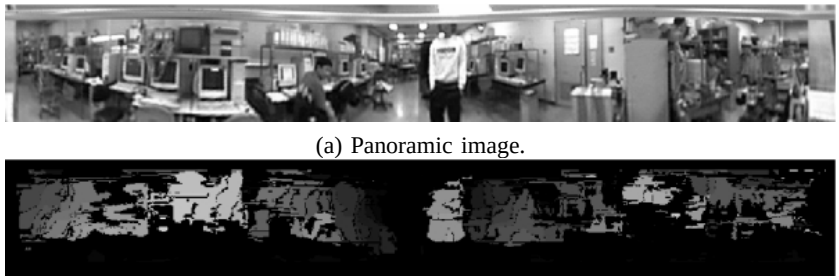


Fig. 1. Our mobile robot.



(a) Panoramic image.

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

Fig. 2. Omnidirectional stereo generates a panoramic disparity image.

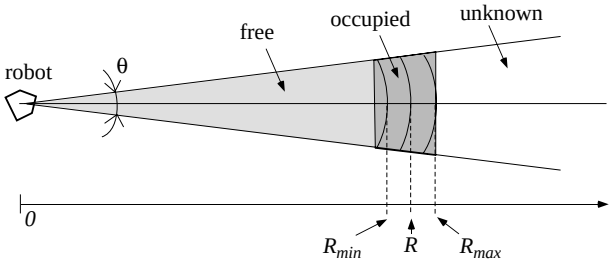


Fig. 3. Determination of grid attributes.

$P(E|O)$ and $P(E|\overline{O})$. These probabilities are calculated by the Bayes' theorem as follows:

$$P(E|O) = \frac{P(O|E)P(E)}{P(O|E)P(E) + P(O|\overline{E})P(\overline{E})}, \quad (1)$$

$$P(E|\overline{O}) = \frac{P(\overline{O}|E)P(E)}{P(\overline{O}|E)P(E) + P(\overline{O}|\overline{E})P(\overline{E})}, \quad (2)$$

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 [10]; $P(\overline{O}|E) = 1 - P(O|E)$; $P(\overline{O}|\overline{E}) = 1 - P(O|\overline{E})$; $P(\overline{E}) = 1 - P(E)$. The integration operation for each grid is performed independently of the others (*the independence assumption*).

2) Integration of two maps: The two probabilistic maps are integrated by first classifying each grid of each map into four classes and then integrating the two classification results into the final 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 grid is larger than the higher threshold (currently, 0.7), the grid is classified as *obstacle*; if the probability is less than the lower threshold (currently, 0.2), the grid is classified as *free space*; otherwise, classified as *undecided*. We further classify *undecided* grids into two subclasses[2].

The second step is for discriminating the following two cases. (1) The occupancy of a grid is undecided because enough information has not been obtained although the robot has observed there many times. A typical situation is the one where the omnidirectional stereo cannot obtain range data for textureless objects. For a grid in the direction of such an object, its class remains *undecided* even if the stereo has tried to observe the object many times. In this case, we believe in the LRF's interpretation. (2) The occupancy is undecided because enough observation has not been made yet. Since we use a probabilistic uncertainty model and integrate observations statistically, each sensor needs a certain number of observations to determine the occupancy (i.e., free space or obstacle) of a grid with confidence. Therefore, if the number of observations of a grid is small, we have to wait until the situation becomes clearer by further observing it. We discriminate the two cases using a threshold for the number of observations. If the number is larger than the threshold (case (1)), we classify a grid as *undecided with observation*; otherwise *undecided without observation*. The number of observations is incremented when a grid has event O or $\overline{O}$ (see Sec. II-B.1), or when a grid is in the direction for which any range measurement is not obtained. The threshold for stereo is five and that for LRF is one.

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

III. Selective Probability Update with Visibility Check Using Object Height

The map generation method described in the previous section has a problem that a true obstacle is sometimes erroneously deleted in the map for stereo. Fig. 4 shows a typical situation; a robot is approaching obstacles with repeatedly observing them. When the robot is distant from the obstacles (see the left figure), the lower obstacle is observable and the corresponding occupancy probabilities increase. When the robot comes near (see the right figure), however, the lower obstacle goes out of the robot's field of view. If the taller obstacle is observed behind the lower one, the grids for the lower one are considered to be *free* and their probabilities will decrease; the robot may think those grids are traversable. This problem comes from the strategy that only 2D range information is used in map generation. The same problem may arise when the robot fails to detect the front obstacle due to, for example, stereo

[2]In the original map generation method[10], these two types of *undecided* were not distinguished.

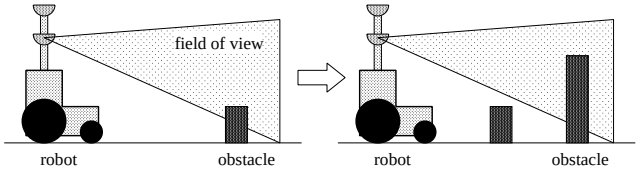


Fig. 4. Visibility changes as the robot moves.

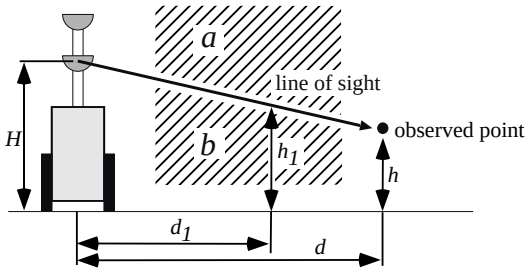


Fig. 5. Visibility check.

matching failure even if the obstacle is not out of sight.

To cope with this problem, we keep the height of obstacles, and check the visibility of obstacles behind the previously-recognized obstacles. That is, if the line of sight to the current observed point does not intersect the previously-recognized obstacles, then the space in front of the current observed point is labeled as *unknown* (i.e., no probability updates are performed for the space); otherwise, the space is labeled as *free* and the above-mentioned probability updates are carried out. In the case of Fig. 5, if obstacles exist only in the region below the line of sight (region *b* in the figure), the probabilities in front of the observed point do not change; otherwise (i.e., obstacles exist in the upper region (region *a*)) the probabilities are updated.

Every grid which has been recognized as an obstacle keeps the mean and the variance of the past observed heights. Suppose that the robot of height H observes an obstacle with height h at distance d as shown in Fig. 5. A previously-observed obstacle at distance d_1 with mean height μ_{h_1} and variance $\sigma^2_{h_1}$ is considered not to conflict with the current observation if the following inequality holds:

$$\mu_{h_1} + 2.0\,\sigma_{h_1} \leq \frac{(h-H)d_1}{d} + H. \tag{3}$$

If all previously-observed obstacles in the space in front of the newly-observed obstacle satisfy this inequality, the probability updates are not performed in that space.

IV. Ego-Motion Estimation

Ego-motion estimation is indispensable for integrating range data obtained at different positions. Odometry is often used for ego-motion estimation but such a dead reckoning suffers from accumulated positional errors. We here, therefore, investigate the use of data from an external sensor, an LRF, for ego-motion estimation.

In our previous work [10], we used distinctive points (e.g., legs of tables and chairs and corners of furniture) in LRF data and estimated the ego-motion from the correspondence of such points. Since the correspondences are based only on the relative distances, however, there were sometimes incorrect matches which degraded the ego-motion estimation results. To exclude such incorrect matches, this paper adopts a robust matching algorithm for finding correspondence. In addition, we use line features as well as point ones because some area in typical indoor environments may has few distinctive points, for example, in the case of corridors.

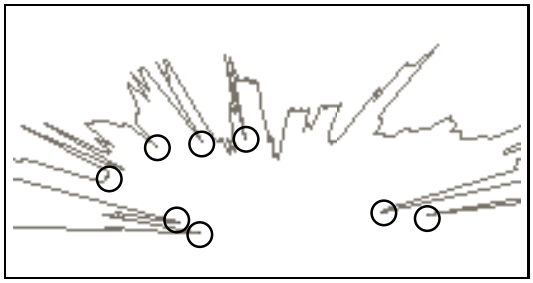


Fig. 6. Extracted feature points in LRF data.

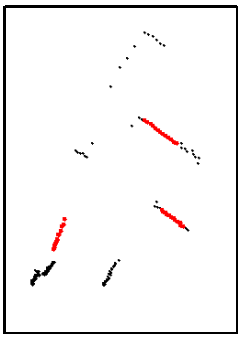


Fig. 7. Extracted lines in LRF data.

The method described here uses only LRF data and does not integrate them with odometry data, but such an integration can easily be incorporated into our method by, for example, using the Kalman filter.

A. Ego-motion Estimation by Point-to-Point Correspondence

Se et al. [13] proposed a RANSAC [3]-based method for matching features. They extracted visual features called SIFT (scale invariant feature transform) and find correspondence between features in the map and the those in the current image for global localization. We apply a similar approach to finding correspondence of point features in LRF data. Figs. 6 and 7 show examples of extracted point and line features.

For an ego-motion parameter vector $(\Delta x, \Delta y, \Delta\theta)$, a feature at (X^t, Y^t) at the current frame is transformed into $(\tilde{X}^{t-1}, \tilde{Y}^{t-1})$ in the previous frame using the following equation (see Fig. 8):

$$\begin{pmatrix} \tilde{X}^{t-1} \\ \tilde{Y}^{t-1} \end{pmatrix} = \begin{pmatrix} \cos\Delta\theta & \sin\Delta\theta \\ -\sin\Delta\theta & \cos\Delta\theta \end{pmatrix} \begin{pmatrix} X^t \\ Y^t \end{pmatrix} + \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}.$$

For each point (X_i^t, Y_i^t) in the current frame, we search for the corresponding point (X_i^{t-1}, Y_i^{t-1}) in the previous frame. Let $(\tilde{X}_i^{t-1}, \tilde{Y}_i^{t-1})$ be the transformed point obtained by using the above equation. The necessary condition for

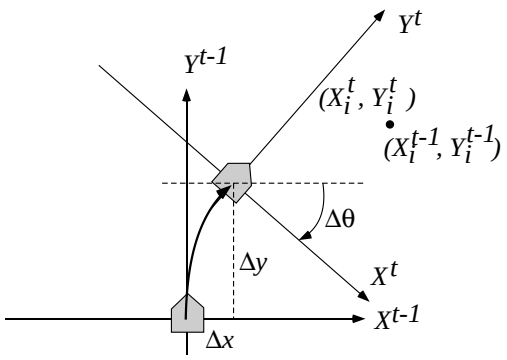


Fig. 8. Ego-motion estimation using point features.

these points corresponding to each other is then given by:

$$dp_i^2 = \left(\tilde{X}_i^{t-1} - X_i^{t-1}\right)^2 + \left(\tilde{Y}_i^{t-1} - Y_i^{t-1}\right)^2 < t_{dist}, \quad (4)$$

where dp_i^2 is the squared distance between the points and t_{dist} is a threshold. Using only this condition for making correspondences may result in false correspondences. So we use the following RANSAC-like algorithm.

1) Choose two pairs of points and check if the distance between the points in one frame is similar to that in the other frame. If the difference of the distances is less than a threshold, proceed to the next step. Otherwise, choose another two pairs of points.
2) Using the selected pairs of points, calculate the ego-motion using eqs. (6-8).
3) Using the calculated ego-motion to transform other points in the current frame to the previous one, and see if each transformed point has a corresponding point which satisfies the condition on the distance for correspondence (eq. (4)).
4) Count the number of points for which the corresponding points are found.
5) If all possible pairs are tested, then determine the ego-motion as the one gains the maximum number of corresponding points. Otherwise, go back to the first step.

Since the number of matched pairs is relatively small in our case (around 10 to 15), we test all possible pairs, unlike the original RANSAC.

Once a set of N_p matched points is obtained, we calculate the best ego-motion which minimizes the following sum of the squared distances:

$$S_p = \sum_{i=1}^{N_p} dp_i^2. \quad (5)$$

This minimization problem has the following analytical solution [10]:

$$\Delta\theta = \tan^{-1} \frac{\begin{array}{r} N_p\left[X_i^{t-1}Y_i^t\right] - N_p\left[X_i^t Y_i^{t-1}\right] \\ -\left[X_i^{t-1}\right]\left[Y_i^t\right] + \left[Y_i^{t-1}\right]\left[X_i^t\right] \end{array}}{\begin{array}{r} N_p\left[X_i^{t-1}X_i^t\right] + N_p\left[Y_i^{t-1}Y_i^t\right] \\ -\left[X_i^{t-1}\right]\left[X_i^t\right] - \left[Y_i^{t-1}\right]\left[Y_i^t\right] \end{array}}, \quad (6)$$

$$\Delta x = \frac{\left[X_i^{t-1}\right] - \left[X_i^t\right]\cos\Delta\theta - \left[Y_i^t\right]\sin\Delta\theta}{N_p}, \quad (7)$$

$$\Delta y = \frac{\left[Y_i^{t-1}\right] + \left[X_i^t\right]\sin\Delta\theta - \left[Y_i^t\right]\cos\Delta\theta}{N_p}, \quad (8)$$

where $[\cdot]$ indicates the summation from 1 to N_p.

B. Use of Line-to-Line Correspondence

There are often places in indoor environments where point features are scarce such as corridors. So we also use line features as well as point features. Lines are extracted in LRF data as follows. We first extract candidates of points on lines by using k-curvature. That is, for each measured point, take a pair of points which are k points apart from the measured point, and draw two lines connecting the measured point and one of the point pair. If the angle between the lines is less than a threshold, then the measured point is a candidate point on a line. Such extracted candidates are clustered into line candidates and each of the line candidates is then verified by fitting a line to its points and by estimating the residual.

A line detected in one frame and another line detected in the subsequent frame are considered to be matched if their center points are near enough and the relative direction is small enough. Since the number of detected lines are not large in our environment, such a simple matching method works well.

Let the ith line at the previous frame be $\rho_i = X\cos\phi_i + Y\sin\phi_i$ and the points of its corresponding line at the current frame be $\{X_j^t(i), Y_j^t(i)\}\,(j(i) = 1, \ldots, N_i)$. For an ego-motion $(\Delta x, \Delta y, \Delta\theta)$, after transforming the points on the line to the current frame to the previous one using eq. (4), we evaluate the squared distance between this ith pair of lines by:

$$dl_i^2 = \sum_{j(i)=1}^{N_i} \left(\tilde{X}_{j(i)}^{t-1}\cos\phi_i + \tilde{Y}_{j(i)}^{t-1}\sin\phi_i - \rho_i\right)^2. \quad (9)$$

Then for N_l pairs of lines, the sum of the squared distances is given by:

$$S_l = \sum_{i=1}^{N_l} dl_i^2. \quad (10)$$

C. Use of Both Types of Correspondence

We would like to minimize the following sum of squared distances:

$$\begin{aligned} S_{pl} &= S_p + S_l \\ &= \sum_{i=1}^{N_p} dp_i^2 + \sum_{i=1}^{N_l} dl_i^2. \end{aligned} \quad (11)$$

Since we cannot obtain an analytical solution of this minimization problem, we take the following two-step procedure.

We first determine $\Delta\theta$. From point-to-point correspondences, we obtain $\Delta\theta$ as eq. (6). We also obtain it from line-to-line correspondence as the mean of directional differences of line pairs. We calculate the weighted mean of these two values as $\Delta\theta$, in which the number of LRF data points is used as the weight. Once the rotational component

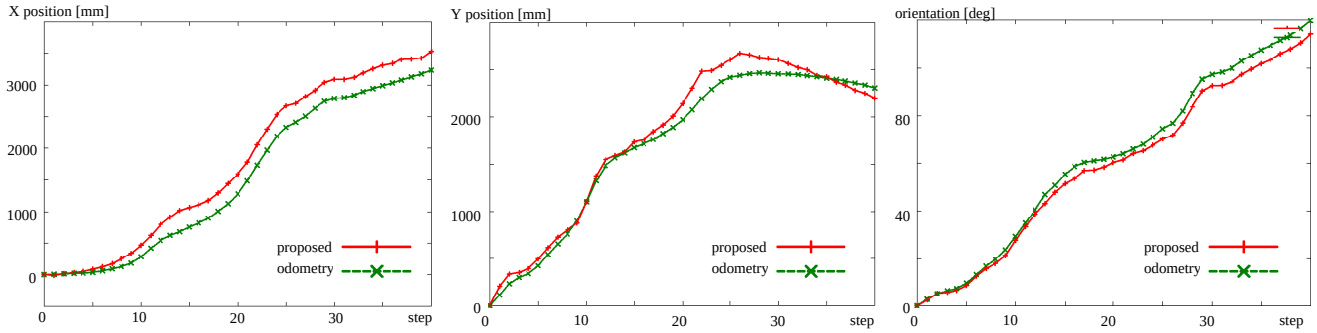


Fig. 9. Ego-motion estimation with comparison with odometry.

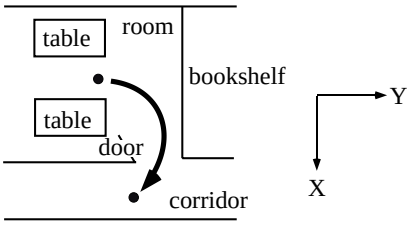


Fig. 10. An example robot motion.

is determined, we search the ΔX-ΔY space for the best translational component which minimizes eq. (11).

D. Ego-motion Estimation Experiment

Fig. 9 shows the result of ego-motion estimation for the robot motion shown in Fig. 10. There are many points features inside the room, while in the corridor, only line features are available. We compared the estimation result obtained by the proposed method with that calculated from odometry data. The figure shows that the proposed method is comparable to the dead reckoning by odometry, although position estimation by odometry suffers from accumulation errors. Since the ego-motion estimation is used for integrating observation data during a short period for local map generation, the proposed ego-motion estimation method shows a reasonable performance; this is also indicated by a navigation result shown in the next section.

V. Navigation Experiment

We conducted navigation experiments in our laboratory. The robot repeatedly updates the map and plans a collision-free motion towards a specified destination. The robot maintains local probabilistic and free space maps in the robot-centered coordinate; each map is composed of 200×200 grids, each of which has the size of $5\,[cm] \times 5\,[cm]$. The robot plans a safe motion inside the free space by using a heuristic path planner [12], which generates a path composed of up to two circular segments, and by an adaptive speed control method [12].

Fig. 11(a)-(d) shows a result of map generation and navigation. In each row, figures indicate a snapshot of experiment, the probabilistic map for omnidirectional stereo, that for the laser range finder (LRF), and the integrated free space map, from left to right. Triangles in the maps indicate the robot position. In probabilistic maps, brighter pixels indicate higher probability of obstacle existence. White areas in the free space maps indicate free spaces.

In rows (a) and (b), a partition with a blackout curtain, which is indicated by gray circles in the free space maps, is not fully recognized by the omnidirectional stereo because image features are scarce. Since the LRF recognizes it clearly, however, the partition is recognized as an obstacle in the free space map. On the other hand, in rows (c) and (d), concerning the table, which is again indicated by gray circles in the free space maps, the stereo recognizes its table top clearly, while the LRF detects only its legs. As a result, the table region is correctly recognized as an obstacle in the final free space map. The robot was able to safely move around in unknown environments, for which only a single sensor may not be sufficient.

We then compare the map generation results for the improved method (considering the height of objects) and the previous one (not considering the height). Row (e) in Fig. 11 indicates the stereo probabilistic map and the integrated free space map at point (d) generated by the previous method. Comparing these two results, the table on the right of the robot is clearly visible in the free space map generated by the improved method, while the table almost disappears in the map by the previous method. For the other places, both methods do not have distinct differences. It is, therefore, concluded that the improved method which considers high information in map updating can generate a more reliable map than the previous one.

VI. Conclusions and Discussion

This paper has described the navigation of a mobile robot in unknown environments using an omnidirectional stereo and a laser range finder. We have improved our previous method in the following two points. One is to consider the height of obstacles in map generation. By checking the visibility of a previously-recognized obstacle in comparison with the current obstacle, we determine if the information that the space before the current obstacle is free has a conflict with the current map information and selectively update the probability in the space. The other improvement is about the ego-motion estimation method. By employing a RANSAC-like point feature matching method and by additionally using line features, we can robustly estimate

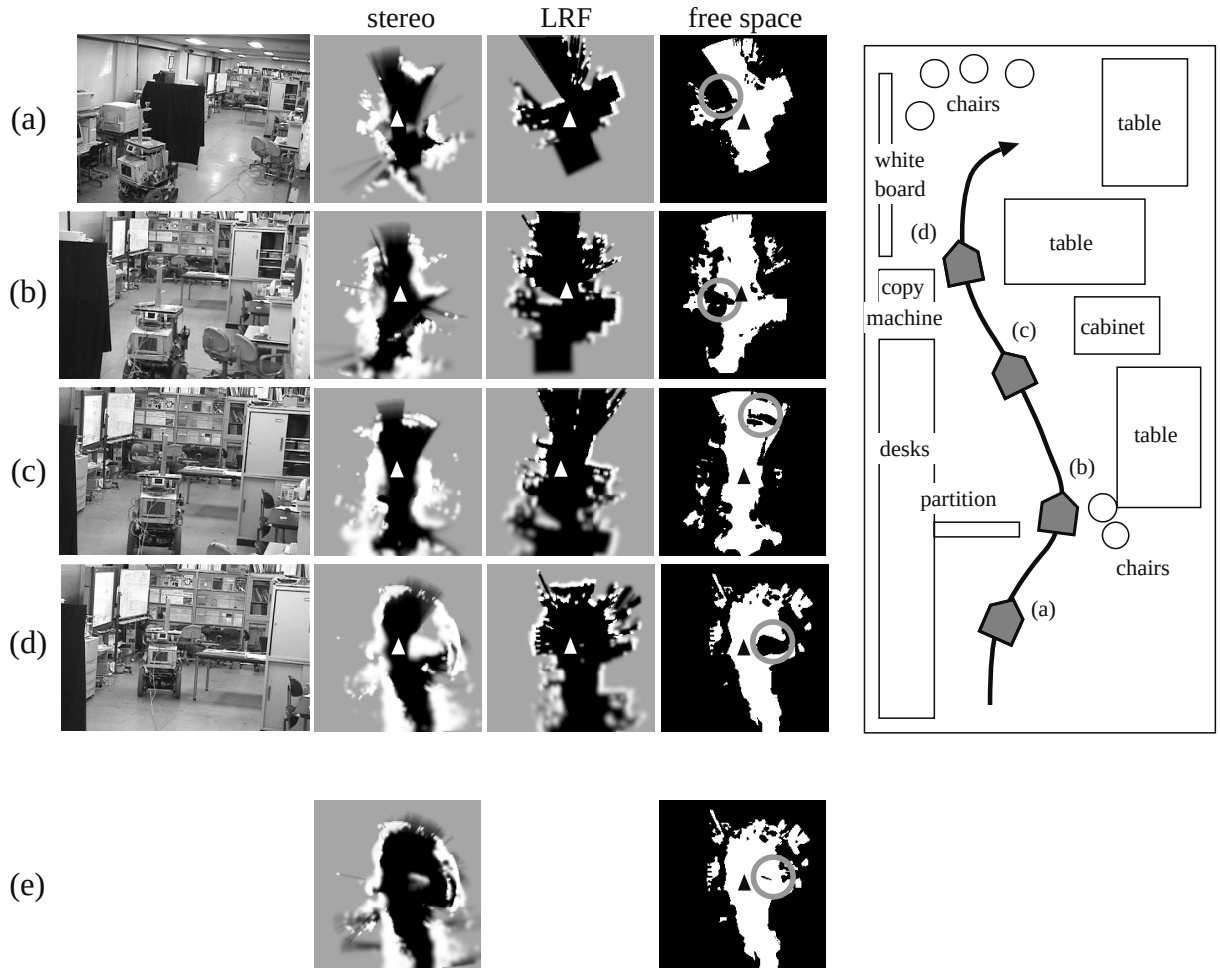


Fig. 11. A map generation and navigation example and a comparison with the previous method.

the ego-motion both in complex in-room environments and in simple, feature-scarce corridor environments.

We currently deal with static environments. A future work is to extend the method to cope with dynamic obstacles by, for example, adopting obstacle detection method such as [7].

Acknowledgment

The authors would like to thank Toshifumi Atsuta for implementing the ego-motion estimation method.

REFERENCES

[1] G. Dissanayake, H. Durrant-Whyte, and T. Bailey. A Computationally Efficient Solution to the Simultaneous Localization and Map Building (SLAM) Problem. In *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 1009–1014, 2000.

[2] A. Elfes. Sonar-based real-world mapping and navigation. *Int. J. of Robotics and Automat.*, Vol. 3, No. 3, pp. 249–265, 1987.

[3] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, Vol. 24, No. 6, pp. 381–395, 1981.

[4] D. Fox, W. Burgard, and S. Thrun. Active markov localization for mobile robots. *Robotics and Autonomous Systems*, Vol. 25, No. 3-4, pp. 195–207, 1998.

[5] K. Kidono, J. Miura, and Y. Shirai. Autonomous visual navigation of a mobile robot using a human-guided experience. *Robotics and Autonomous Systems*, Vol. 40, No. 2-3, pp. 121–130, 2002.

[6] H. Koyasu, J. Miura, and 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*, pp. 31–36, 2001.

[7] H. Koyasu, J. Miura, and Y. Shirai. Mobile robot navigation in dynamic environments using omnidirectional stereo. In *Proceedings of 2003 IEEE Int. Conf. on Robotics and Automation*, pp. 893–898, 2003.

[8] J.J. Leonard and H.F. 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*, pp. 1442–1447, 1991.

[9] F. Lu and E. Milios. Robot Pose Estimation in Unknown Environments by matching 2D Range Scans. *Journal of Intelligent and Robotic Systems*, Vol. 18, pp. 249–275, 1997.

[10] J. Miura, Y. Negishi, and 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*, pp. 250–255, 2002.

[11] D. Murray and J. Little. Using real-time stereo vision for mobile robot navigation. *Autonomous Robots*, Vol. 8, No. 2, pp. 161–171, 2000.

[12] Y. Negishi, J. Miura, and Y. Shirai. Adaptive robot speed control by considering map and localization uncertainty. In *Proceedings of the 8th Int. Conf. on Intelligent Autonomous Systems*, pp. 873–880, 2004.

[13] S. Se, D. Lowe, and J. Little. Global localization using distinctive visual features. In *Proceedings of 2002 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 226–231, 2003.