Proc. 2010 IEEE Int. Conf. on Multisensor Fusion and Integration in Intelligent Systems, pp. 144-149, Salt Lake City, USA, September 2010.

A Hierarchical SLAM for Uncertain Range Data

Kenta Kitajima, Hiroaki Masuzawa, Jun Miura, and Junji Satake
Department of Information and Computer Sciences
Toyohashi University of Technology

***Abstract*— This paper describes a new approach to SLAM problems using low quality range data. Vision sensors are useful for acquiring various kinds of environmental information but range data obtained by stereo vision is less reliable than other active sensors like laser range finders. False stereo matches often result in spurious obstacles, which may degrade the map when directly used in existing SLAM methods. We therefore propose a hierarchical approach in which local probabilistic occupancy maps are first generated to filter out such spurious obstacles and then used as inputs to an RBPF-based SLAM. Experimental results in simulation and in a real environment show that a consistent map can be generated by the proposed method with low quality stereo range data.**

***Index Terms*— SLAM, Rao-Blackwellized particle filter, mobile robots, stereo.**

I. Introduction

Maps are important media for mobile robots to communicate with people or other robots. Since it is tedious for people to make and maintain maps, automatic map making has been an important mobile robot task. When a robot makes a map while moving around the environment, we have to consider the accuracy of motion estimation. If we can obtain accurate position information by, for example, GPS, then it may not be difficult to make maps by putting observed data into a large fixed coordinate system. If such a reliable position information is not available, however, we need to simultaneously estimate the map and the motion. Therefore, simultaneous localization and mapping (SLAM) has been one of the most active research areas in mobile robotics [1].

A common standpoint in SLAM approaches is that SLAM is an estimation problem from a sequence of uncertain data and probabilistic inference is thus usually used.

Previous approaches can be categorized in several aspects. One aspect is the observed features used. One of the earlier SLAM works used features like corners and walls extracted in the range data to describe the map [2]. Recently visual SLAM has been popular and most of them also use specific visual features [3], [4], [5]. Others use an occupancy grid map [6].

Another aspect is the statistical tools used for integrating a sequence of uncertain data for SLAM. There are two major tools. One is (Extended) Kalman filter. SLAM methods using Kalman filter estimate a state including both the robot pose and the feature-based map [2], [7]. The other is particle filter [1] which represents the distribution of a state by a set of *particles*; each particle represents a possible state (i.e., a robot pose and a map). Although applying a naive particle filter to SLAM problem may be costly, an important variant, called Rao-Blackwellized particle filter (RBPF), has been applied successfully to feature-based [8] and grid-based [6] SLAM problems.

One objective of map making is to recognize free spaces where robots can move safely. For this objective, range information is indispensable. Many previous works use laser range finders (LRFs) to obtain range information [9]. LRFs can provide very accurate range data; most LRFs in use, however, provide only 2D range data on the scanning plane. If an environment includes not only simple walls but also obstacles with various shape, however, only 2D range data are not sufficient for correct map making.

Stereo is a passive range sensor that can obtain 3D range information and thus useful for map making in such an environment. One drawback in using stereo for obtaining dense range data is its relatively low reliability; frequent matching failures may result in many spurious obstacles. If an accurate ego-motion estimate is available or the total moving distance is short enough to avoid accumulated estimated errors, however, it is possible to apply Bayesian inference to filter out such spurious obstacles [10], [11], [12].

Local map joining [13] is a SLAM method which first generates a local map from a set of data from several consecutive frames, and then joins them to generate a global map. The method detects local features such as walls to make a local map. Such features are used for map joining using EKF. The method thus requires such explicit features.

Map match SLAM [14] is a similar approach but from a different standpoint. This uses a grid map as a medium for incorporating various sensor data into the grid-based RBPF framework. Although it is shown that sonar, stereo, and monocular camera can be used by the method, the effect of large uncertainty in range measurement to the resultant global map is not discussed.

A hierarchical version of DP-SLAM [15] uses a resultant map of the lower-level SLAM as an input to the higher-level SLAM, mainly for reducing the necessary number of particles for reliable mapping. This could fail to make a consistent map if the lower-level map is not well constructed due to large range data errors.

In this paper, we deal with a SLAM problem with uncertain range data, which is a typical case for stereo-based range measurements. Large uncertainty may cause a failure of correct registration of data as well as ego-motion estimation. We therefore take a similar approach to these map-based SLAM methods, but put a more focus on the robustness to large errors in range data.

Fig. 1 shows an overview of the proposed method, which is composed of two stages. In the first stage, local maps, which are robot-centered probabilistic occupancy grid maps, are

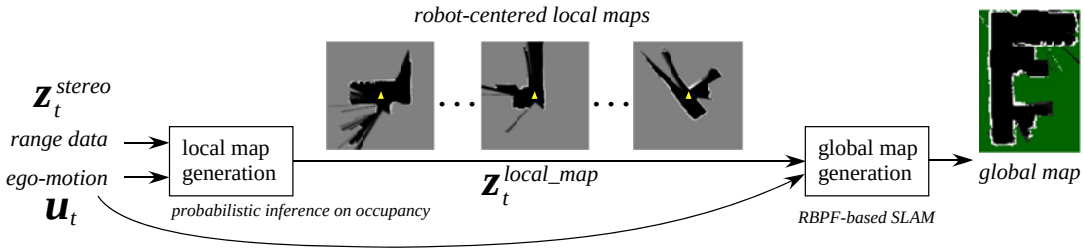


Fig. 1. Overview of the hierarchical SLAM.

repeatedly generated from sequences of range data z_t^{stereo} and controls u_t. In the second stage, an RBPF-based SLAM is performed using the local maps $z_t^{local\_map}$ and the controls as inputs in order to generate a global map in some fixed coordinates. The method can generate a consistent map from a low quality stereo range data sequence.

The rest of the paper is organized as follows. Sec. II explains how to make local maps and Sec. III details the RBPF-based global map generation. Sec. IV compares the proposed and a previous method in a simulated environment for various error rates of stereo range data to show the robustness of the proposed one. Sec. V show experimental results in a real environment. Sec. VI concludes the paper and discusses future work.

II. Local map generation

A. Range data and ego-motion

The local map generation part calculates robot-centered local maps from range data and ego-motion information. Each set of range data is represented by a series of range measurements for directions with a fixed interval. 2D range finders usually provide this type of representation. In the case of stereo, we extract the nearest distance in each column of the dense depth map to obtain the same representation.

To integrate observations during the robot movement into a robot-centered local map, we apply coordinate transformation to an observation sequence in order to map them in the latest robot coordinates. We assume that the ego-motion estimation is accurate enough for a short movement. If we have an accurate odometry or other positioning systems, we can use them; this is sometimes the case for indoor robots.

B. Integration of range data

We use the occupancy grid representation [12] for local maps; each map holds the probability that an obstacle exists there. We set the size of each cell to $50\,[mm] \times 50\,[mm]$. Bayesian inference is used for updating the maps [10]. The procedure is briefly explained below.

For one observation, we classify the attribute of each cell into one of the three categories: *occupied*, *free*, and *unknown*. Occupied cells are the one with observed obstacles. Free cells are between the occupied ones and the robot. Unknown ones are behind the occupied cells where no information is available in the observation of this time. Based on this interpretation, we update each cell of the grid map as follows.

Let O be the event that an obstacle is detected; O occurs at occupied cells; the inverse event $\overline{O}$ occurs at free cells. Probabilities are updated for these two types of cells. Let E be the event that an obstacle exists at a cell, and let $P(E)$ be the probability of the obstacle existence there. The probability update is carried out by calculating either of the following posterior probabilities:

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

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

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 update equations, $P(O|E)$ and $P(O|\overline{E})$ are observation models depending on the sensors used. Other terms can be calculated from the models: $P(\overline{O}|E) = 1 - P(O|E)$, $P(\overline{O}|\overline{E}) = 1 - P(O|\overline{E})$, and $P(\overline{E}) = 1 - P(E)$. Each grid is updated independently of the others (*the independence assumption*).

C. Sensor models

$P(O|E)$ is the probability that an obstacle is observed in a cell when it actually exists there. In the case of stereo, there is always a possibility that the correct stereo match is not obtained depending on, for example, textures on obstacle surfaces and lighting conditions. $P(O|\overline{E})$ is the probability that a spurious obstacle is observed at a position where no obstacle actually exists; this corresponds to a false stereo match. This probability also depends on similar factors. It is, however, difficult to know the actual condition of a specific position on an object surface.

Fig. 2 shows a few pairs of input and depth images. Colors in depth images shows the depth of each pixels. Black pixels in depth images indicate that no matches are obtained for calculating the depth. Depth data for image set 1 is mostly correct while those with image set 2 and 3 have false matches. In addition, the number of pixels with calculated depths varies from image to image. It is, therefore, difficult to construct a parametrized model of stereo uncertainty. We thus examine many images captured in the experimental site and estimate the models. As a result, we obtain the following values for our stereo system: $P(O|E) = 0.54$ and $P(O|\overline{E}) = 0.1$. A more detailed stereo error model, which is used for simulation, will be described in Sec. IV-C.

D. Local map examples

Fig. 3 illustrates how errors in range data are filtered out by temporal data integration in local map generation. The top row is a sequence of input range data; this sequence

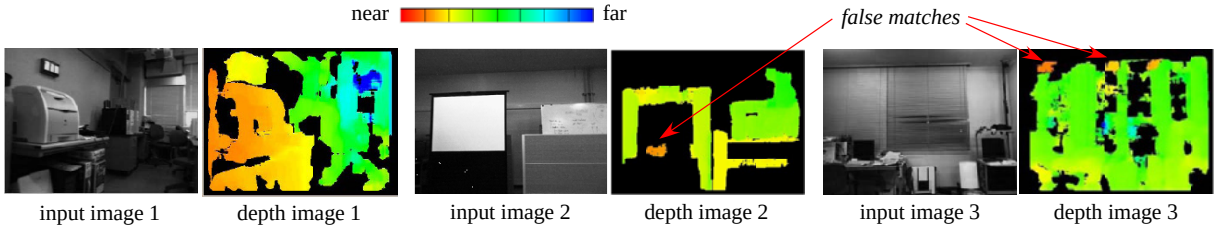


Fig. 2. Examples of input and depth images.

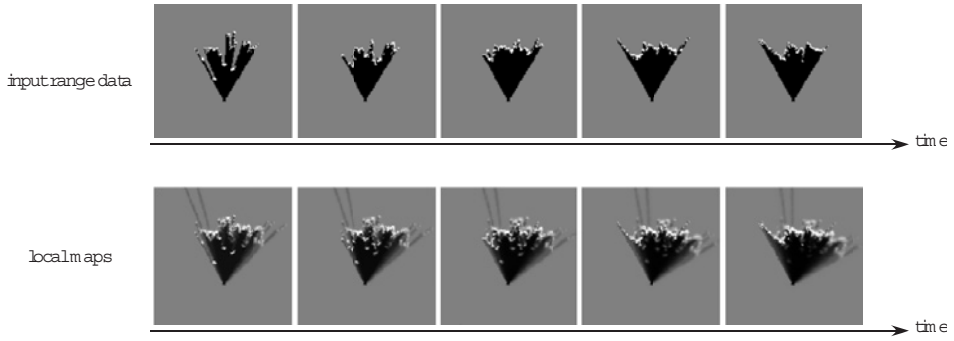


Fig. 3. A sequence of input range data and generated local maps.

corresponds to the scene shown by input image 3 in Fig. 2. Large spurious objects protruding towards the robot appear in the leftmost input, although no such objects exist in the actual scene. The bottom row of Fig. 3 shows the sequence of generated local maps. The spurious objects are erased as correct range data are input in the subsequent observations.

E. Frequency of local map generation

An important point of in the hierarchical SLAM approach is how frequently local maps are generated and used for the global map generation. To obtain a reliable local map, it is desirable to integrate many observations. However, the local map generation relies on the quality of ego-motion estimation, which is carried out by dead reckoning using odometry in our case and thus suffers from an accumulated error in a long movement. It is also necessary to consider that two subsequent local maps have to have a certain amount of common areas so that odometry errors can be reduced by local map matching in a SLAM framework.

Considering these points, we empirically determined the number of observations to be used for local map generation to be ten. In the current setting, we move the robot at about $0.34\,[m/s]$ and the frequency of observation is about $2\,[Hz]$; the robot moves about $1.7\,[m]$ for obtaining ten observations.

III. Global Map Generation

A. Grid-based FastSLAM

The global map generation step uses a sequence of local maps as observations and ego-motion estimation to make a map in a fixed world coordinate system. We adopt a grid-based FastSLAM algorithm [6] in this step.

The full SLAM problem is to estimate the following posterior [1]:

$$p(\boldsymbol{x}_{1:t}, \boldsymbol{m} \mid \boldsymbol{z}_{1:t}, \boldsymbol{u}_{1:t}), \tag{3}$$

where $\boldsymbol{x}_{1:t}$ is a sequence of robot poses, $\boldsymbol{m}$ is a map, $\boldsymbol{z}_{1:t}$ is a sequence of observations, and $\boldsymbol{u}_{1:t}$ is that of control commands. We then factorize this expression as follows:

$$\begin{aligned} &p(\boldsymbol{x}_{1:t}, \boldsymbol{m} \mid \boldsymbol{z}_{1:t}, \boldsymbol{u}_{1:t}) \\ &\quad = p(\boldsymbol{x}_{1:t} \mid \boldsymbol{z}_{1:t}, \boldsymbol{u}_{1:t}) \cdot p(\boldsymbol{m} \mid \boldsymbol{x}_{1:t}, \boldsymbol{z}_{1:t}). \end{aligned} \tag{4}$$

This factorization decomposes the full SLAM problem into two sequential estimation problems: (1) estimating the robot *path* using observation and control sequences and (2) making a map with *known* robot path and observations. This greatly reduces the computational cost of the SLAM problem. Particle filters based on this factorization are called *Rao-Blackwellized particle filters* (RBPFs).

B. Local map-based SLAM algorithm

Each particle has robot path (sequence of robot poses) $\boldsymbol{x}_{1:t}$ and global map $\boldsymbol{m}$ with accumulated likelihood w_t. A global map is also a probabilistic occupancy grid map. An observation $\boldsymbol{z}_t$ is a local map and a control command $\boldsymbol{u}_t$ is an estimate of ego-motion.

The local map-based SLAM algorithm is given as follows:

1. For each particle $\boldsymbol{y}_{t-1}^{[k]}$ $(k \in [1 \ldots N])$ in particle set Y_{t-1}, do the following:
   a. Retrieve pose $\boldsymbol{x}_{t-1}^{[k]}$ from particle $\boldsymbol{y}_{t-1}^{[k]}$, sample a new pose $\boldsymbol{x}_t^{[k]} \sim p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}^{[k]}, \boldsymbol{u}_t)$, and add it to path:
   $$\boldsymbol{x}_{1:t}^{[k]} \leftarrow \boldsymbol{x}_{1:t-1}^{[k]} \cup \boldsymbol{x}_t^{[k]}. \tag{5}$$
   b. Calculate likelihood $l^{[k]}$ of the particle from the degree of matching between observation $\boldsymbol{z}_t$ (the current local map) and $\boldsymbol{m}^{[k]}$ (global map) with

new pose $\boldsymbol{x}_t^{[k]}$, and update the accumulated likelihood:

$$w_t^{[k]} = w_{t-1}^{[k]} * l^{[k]}. \quad (6)$$

c. Update the global map $\boldsymbol{m}^{[k]}$.

2. Do selective resampling [6]:
   a. Calculate the weight of each particle as the normalized likelihood.
   b. Calculate the effective number of particles N_{eff}:

$$N_{eff} = 1/\sum_{k=1}^{N}\left(\hat{w}_t^{[k]}\right)^2,\ \hat{w}_t^{[k]} = w_t^{[k]}/\sum_{i=1}^{N} w_t^{[i]} \quad (7)$$

   c. If $N_{eff} <$ then sample N particles with replacement using $w_t^{[k]}$ as weights and reset all weights to 1.0 after finishing the sampling; otherwise skip the resampling step.

C. Map matching and map update

1) Likelihood calculation by map matching: The likelihood of a particle is calculated in step 1b. The likelihood depends on how well the local and the global map match with each other. Since the obstacle cells are more important for matching [14], and since the global map has more reliable information than the local map, we see what percentage of obstacle cells of the global map matches with those of the local map.

We first extract *obstacle cells* from the global map $\boldsymbol{m}^{[k]}$ and the local map $\boldsymbol{z}_t$. Let S be the set of obstacle cells in the global map within the overlapping region of both maps; the region is calculated using new pose $\boldsymbol{x}_t^{[k]}$. We examine each cell in the local map corresponding to S and count the number n of the obstacle cells. The likelihood $l^{[k]}$ is then given by:

$$l^{[k]} = \exp\left\{-k \cdot (n/|S|)\right\}, \quad (8)$$

where k is a parameter (currently set to 50).

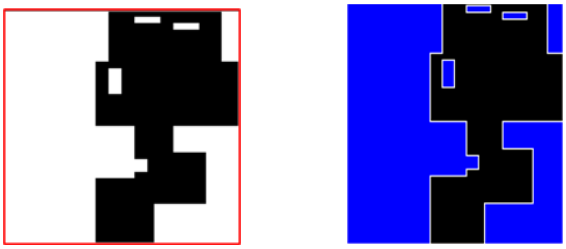
2) Global map update: A global map is updated similarly to the case of local map update. In the case of local map, we classified the attribute of each cell into *occupied*, *free*, and *unknown* based on the observed obstacles and their position in the robot local coordinate system. We similarly classify *the cells of the local map* into the three classes using two thresholds (currently, 0.7 and 0.2). After the classification, we use eqs. (1) and (2) for updating the global map.

IV. Simulation Results

This section describes comparison of the proposed and a previous method (i.e., an ordinary grid-based FastSLAM [6]). To compare them in various error rates of stereo range data, we develop a simulator with a stereo error model.

A. Simulation setting

We use a simulated environment of $18\,[m] \times 18\,[m]$, as shown in Fig. 4(a). The black region corresponds to free cells and white regions to occupied ones. A simulated robot moves around in this environment and obtains stereo and odometry data sequences. Fig. 4(b) is the ideal map to be obtained by perfect sensors. White, black, and blue regions correspond to occupied, free, and unknown cells, respectively.



(a) Simulation environment. (b) Ideal map.

Fig. 4. A simulation environment and the ideal map.

B. Map quality evaluation

The quality of a map is evaluated by the dissimilarity between the map and the ideal one. The dissimilarity E is defined by:

$$E = \sum_{(i,j)\in\mathcal{S}} diff(i,j), \quad (9)$$

$$diff(i,j) = \begin{cases} 0 & g(i,j) = -1 \\ 0.1 & g(i,j) \neq -1 \text{ and} \\ & t(i,j) = -1 \\ |g(i,j) - t(i,j)| & \text{otherwise} \end{cases}, \quad (10)$$

where $g(i,j)$ and $t(i,j)$ indicates the probability of cell (i,j) for a generated and the ideal map, respectively. Value -1 is set to cells with no information. For a certain range of robot pose (i.e., two translational and one rotational values), we search for the minimum dissimilarity and the minimum value is used as the dissimilarity between the pair of maps.

C. Stereo error model for generating simulation data

Many stereo uncertainty models have been developed so far. Most of them are, however, for modeling range data perturbation due to, for example, the quantization error of pixels (e.g., [16], [17]). Although some works propose other types of errors such as false correspondence of feature points [18], they are not for simulating the stereo data generation process.

While the quantization errors do not have large effects on the mapping quality, errors due to false correspondence do. We thus examined how these errors appear in stereo range data and constructed a stereo error model which covers both types of errors. The field of view is $66\,[deg.]$ and the depth range is $0 \sim 8\,[m]$ for the stereo camera used. The two types of errors are modeled as follows.

The error in depth measurement is modeled by a normal distribution, the mean and the variance of which are zero and two percentage of the measured value, respectively.

The large error due to false correspondence usually does not appear in a systematic way. As the result of examination of actual stereo data, we found such errors occur not at each pixel independently but occur as a *block*, which is a continuous region in stereo range data. Some examples are shown in the "input range data" row in Fig. 3. The mean and the standard deviation of such depth errors are $4.886\,[m]$ and $1.773\,[m]$, respectively, in the data taken in our experimental site. These values are used for generating false depths in the model. The size of the block was between $5 \sim 10\,[deg.]$ in angle, and it becomes smaller as the depth increases. We thus model the error in the size of block as follows. For false

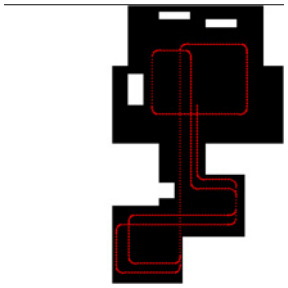


Fig. 5. The robot trajectory for data collection.

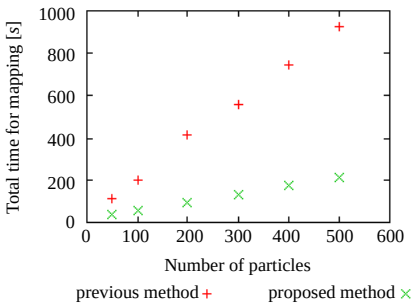


Fig. 6. Comparison in computation time.

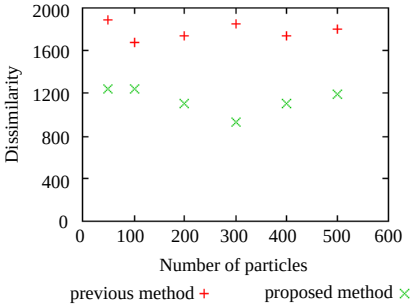


Fig. 7. Comparison in map quality.

depth $d\,[m]$, the size follows a normal distribution with the standard deviation being $3\,[deg.]$ and the mean being

$$\frac{5(8-d)}{8} + 5\,[deg.]. \tag{11}$$

The position of the block is determined randomly in the field of view. Since the number of occurrence of false matches per frame was about one in average, we put one false block in each stereo range data in simulation.

D. Simulation data generation

The robot moves on a predetermined route and calculates the odometry values and the stereo data every $200\,[mm]$ travel. The errors in odometry are added to the calculated values; the standard deviations are $20\,[mm]$ in the translational components and $0.7\,[deg.]$ in the rotational one, which are determined from the experimental data. The errors in stereo data are added by following the stereo error model described in the previous subsection.

E. Comparison results

This subsection describes comparison results using the data of 785 measurement positions obtained by moving the robot on the trajectory shown in Fig. 5.

1) Changes of computation time and quality due to change of the number of particles: Fig. 6 compares the computation times for map generation with different numbers of particles; the values are the averaged ones for five runs. The proposed method is much faster because the SLAM step is performed only when each local map is generated. Fig. 7 compares the quality of the maps. The vertical axis indicates the dissimilarity of the generated map and the true one. The proposed method outperforms the previous one both in the computation time and the map quality.

2) Changes of map quality due to change of the degree of stereo errors: Most methods can generate a nice map with high-quality data, but will soon be degraded with low-quality data if they are not robust. We therefore compares the two methods with various error rates in stereo data.

In order to change the data quality, we use the following expression for the mean of the size of the block with a false depth being

$$\frac{3\alpha(8-d)}{8} + 3\alpha\,[deg.], \tag{12}$$

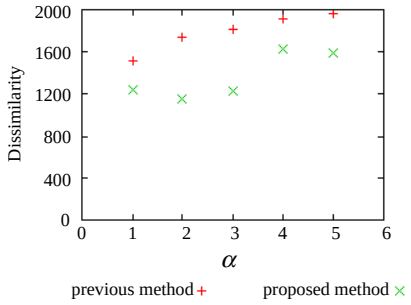


Fig. 8. Comparison with various data qualities.

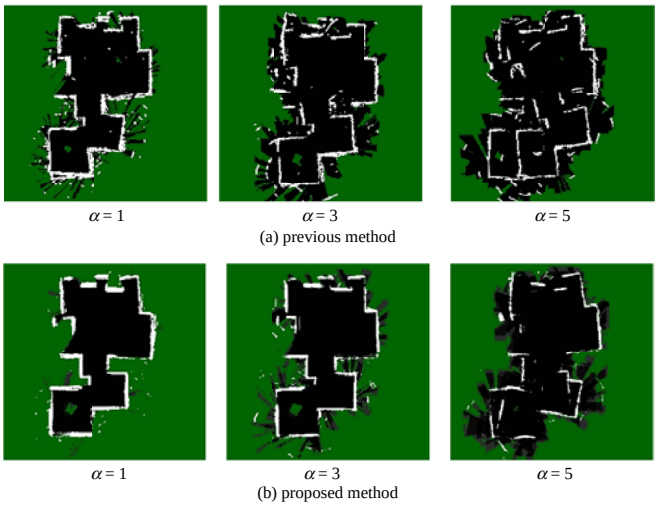


Fig. 9. Comparison of generated maps.

where α is a parameter to control the data quality and setting $3\alpha = 5$ makes this expression be the same as the previous one (see eq. (11)).

Fig. 8 shows the change of map quality for various α's. The map quality decreases as the data quality decreases in both methods, but the proposed one is more robust than the previous one. Fig. 9 shows the generated maps for both methods with various α's. This also illustrates the robustness of the proposed method.

V. Experimental results for real data

We performed SLAM experiments in an actual office environment. Fig. 10 shows the experimental room with

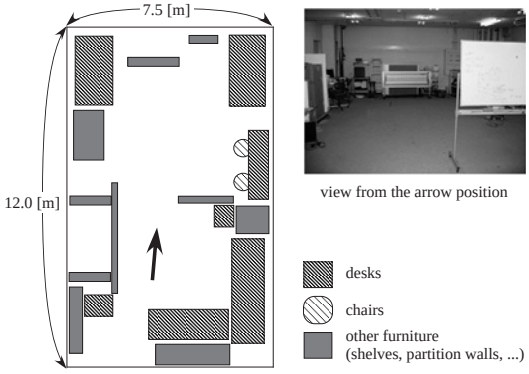


Fig. 10. Experimental environment.

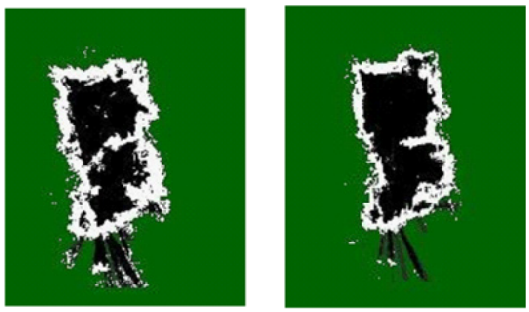


Fig. 11. Comparison in global map generation using real data.

the size of $12.0 \times 7.5\,[m]$. Many desks, chairs, and other furniture are naturally placed throughout the room. We use a robot with a stereo camera (MiniBEE by ViewPlus) with an encoder-based odometry system. We manually moved the robot on an arbitrary path to take sensor data.

Fig. 11 shows the generated maps for the previous and the proposed method. Both maps represent roughly the shape of the room but the one by the proposed method is more consistent with the actual room.

VI. Conclusions and Future Work

This paper has described a hierarchical SLAM method for uncertain range data. The method continuously generates local maps by integrating consecutive observations and then uses them as inputs to the higher-level SLAM process to generate a global map. By filtering out many spurious obstacles existing in the input range data in the local map generation process, the SLAM process can run without failure.

We first tested the proposed method using a simulation environment. We analyzed how errors in stereo data are generated to make a stereo error model and use it in simulation. Using the various simulation data, we compare our method with a grid-based SLAM method to show its robustness against low-quality range data. We also compares the methods for mapping in a real environment.

We currently use only range data from stereo. Although range data are especially useful in space recognition, other data such as image features are also useful in mapping. Since many visual SLAM techniques have been developed, integrating visual features and range data obtained from a stereo camera for more robustness is a future work. Simultaneous use of stereo and laser range finder is another future work. Since stereo and a laser range finder have complementary characteristics [10] in terms of reliability/accuracy and field of view, integrating them would enhance the robustness *and* the accuracy of the map.

Acknowledgment

This work is supported by NEDO (New Energy and Industrial Technology Development Organization, Japan) Intelligent RT Software Project.

References

[1] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, 2005.

[2] 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.

[3] S. Se, D. Lowe, and J. Little. Mobile Robot Localization and Mapping with Uncertainty using Scale-Invariant Visual Landmarks. *Int. J. of Robotics Research*, Vol. 21, No. 8, pp. 735–758, 2002.

[4] T. Lemaire, S. Lacroix, and J. Solà. A Practical 3D Bearing-Only SLAM Algorithm. In *Proceedings of 2005 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 2449–2454, 2005.

[5] M. Tomono. Monocular SLAM Using a Rao-Blackwellised Particle Filter with Exhaustive Pose Space Search. In *Proceedings of 2007 IEEE Int. Conf. on Robotics and Automation*, pp. 2421–2426, 2007.

[6] G. Grisetti, C. Stachniss, and W. Burgard. Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling. In *Proceedings of 2005 IEEE Int. Conf. on Robotics and Automation*, pp. 2432–2437, 2005.

[7] J.E. Guivant and E.M. Nebot. Optimization of the Simultaneous Localization and Map-Building Algorithm for Real-Time Implementation. *IEEE Trans. on Robotics and Automation*, Vol. 17, No. 3, pp. 242–257, 2001.

[8] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. In *Proceedings of AAAI National Conf. on Artificial Intelligence*, pp. 593–598, 2002.

[9] D. Hähnel, W. Burgard, D. Fox, and S. Thrun. An Efficient FastSLAM Algorithm for Generating Maps of Large-Scale Cyclic Environments from Raw Laser Range Measurements. In *Proceedings of 2003 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 206–211, 2003.

[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] S. Thrun. Learning Occupancy Grids with Forward Models. In *Proceedings of 2001 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 1676–1681, 2001.

[12] A. Elfes. Sonar-Based Real-World Mapping and Navigation. *Int. J. of Robotics and Automat.*, Vol. 3, No. 3, pp. 249–265, 1987.

[13] J.D. Tardos, J. Neira, P.M. Newmann, and J.J. Leonard. Robust Mapping and Localization in Indoor Environments using Sonar Data. *Int. J. of Robotics Research*, Vol. 21, No. 2, pp. 311–330, 2002.

[14] C. Schröter and H.-M. Gross. A Sensor-Independent Approach to RBPF SLAM – Map Match SLAM Applied to Visual Mapping. In *Proceedings of 2008 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 2078–2083, 2008.

[15] A.I. Eliazar and R. Parr. Hierarchical Linear / Constant Time SLAM Using Particle Filters for Dense Maps. In *Advances in Neural Information Processing Systems*, No. 18, pp. 339–346, 2005.

[16] L. Matthies and S.A. Shafer. Error Modeling in Stereo Navigation. *IEEE J. of Robotics and Automat.*, Vol. RA-3, No. 3, pp. 239–248, 1987.

[17] D.J. Kriegman, E. Triendl, and T.O. Binford. Stereo Vision and Navigation in Buildings for Mobile Robots. *IEEE Trans. on Robotics and Automat.*, Vol. RA-5, No. 6, pp. 792–803, 1989.

[18] J. Miura and Y. Shirai. An Uncertainty Model of Stereo Vision and Its Application to Vision-Motion Planning of Robot. In *Proceedings of the 13th Int. Joint Conf. on Artificial Intelligence*, pp. 1618–1623, Chambéry, France, August 1993.