International Journal of Control, Automation, and Systems (2010) 8(2):328-338 http://www.springer.com/12555
DOI 10.1007/s12555-010-0219-3

Efficient and Reliable Monte Carlo Localization with Thinning Edges

Tae-Bum Kwon, Ju-Ho Yang, and Jae-Bok Song*

Abstract: The global convergence of MCL is time-consuming because of a large number of random samples. Moreover, its success is not guaranteed at all times. This paper presents a novel approach to reduce the number of samples of MCL and one heuristic approach to detect localization failure using thinning edges extracted in real time. Random samples are drawn only around the neighborhood of the thinning edges rather than over the entire free space and localization quality is estimated through the probabilistic analysis of samples added around the thinning edges. A series of experiments verified the performance of the proposed scheme.

Keywords: Kidnapped robot problems, Monte Carlo localization, particle filters, thinning edges.

1. INTRODUCTION

Successful localization is one of the key issues for reliable navigation of a mobile robot. Localization or pose estimation of a mobile robot can be classified into position tracking and global localization. Position tracking continually estimates the current robot pose from the known initial pose by tracking the change of the robot's pose during movement. On the other hand, global localization estimates the robot pose without use of prior knowledge of its initial pose. Hence, global localization is computationally more demanding than position tracking.

Several localization algorithms have been proposed. Typical examples include the Kalman filter [8,11], Markov localization [5,13], Monte Carlo localization (MCL) [4] and some hybrid approaches [6,12], and other ap-proaches [16,17]. The Kalman filter technique is commonly used when the initial pose is known. In this technique, the robot estimates its pose continually by compensating the odometric error using the range data. On the other hand, Markov localization is widely used for global localization. In this scheme, positional probabilities of all empty grids are computed. Thus, the Markov localization requires a large amount of computation time, and its accuracy of the localization is limited to the grid size if the grid type map is used. MCL (Monte Carlo localization) is another global localization technique. However, its implementation is faster than the Markov localization because the probability computation is carried out only for the random samples. MCL often provides more accurate results than the Markov localization because the samples can take any pose regardless of the grid size. The hybrid method is a combination of either Markov localization and a Kalman filter or a Kalman filter and MCL. Using the advantages of each method, these hybrid approaches attempt to improve the efficiency of pose estimation.

Among several localization techniques, MCL has been most widely used in recent years because of its various advantages. For examples, it can represent multi-modal probability distribution and thus can solve a global localization problem. However, its implementation requires more computation than the Kalman filter-based approach. Since the MCL computes probabilities of all random samples, its computational burden increases as the number of samples increases (especially for large-scale environments). This long localization update cycle may lead to a significant difference between the estimated and current robot poses [4]. Since this type of localization error becomes larger as the navigation speed increases, the speed of a robot should be limited to account for the long computational time of localization. Therefore, one important issue in the improvement of the MCL performance is the reduction of the number of random samples of the particle filter without degradation of the localization performance of MCL.

Adaptive particle filter using the KLD-sampling [8] focused on the reduction in the number of samples. As the samples converge to the robot's current pose, the number of samples can be reduced because the localization can be achieved with fewer samples than those required for the global localization. They adaptively change the number of samples according to the localization status, but fail to reduce the cycle time during the initial phase of global localization because the initial number of samples was the same as that of the standard MCL.

Manuscript received November 4, 2008; revised July 17, 2009; accepted September 1, 2009. Recommended by Editorial Board member Jang Myung Lee under the direction of Editor Jae Weon Choi. This research was supported in part by the Intelligent Robotics Development Program (21st Century Frontier R&D Program) and in part by the Agency for Defense Development and Unmanned Technology Research Center at KAIST.

Tae-Bum Kwon and Jae-Bok Song are with the School of Mechanical Engineering, Korea University, 5, Anam-dong, Sungbuk-gu, Seoul 136-713, Korea (e-mails: artistkwon@gmail.com, jbsong@korea.ac.kr).

Ju-Ho Yang was with the School of Mechanical Engineering, Korea University and is with the Mando Corporation, Korea (e-mail: rmfls200@korea.ac.kr).

* Corresponding author.

Another important issue of localization is the detection of a localization failure or kidnapping and the restoration of the localization quality. Even state-of-the-art localization algorithms such as MCL cannot guarantee successful performance for all situations. Thus, the robot should have the ability to autonomously detect and recover from failures. Once the estimated robot pose is determined unreliable, the correct robot pose should be re-estimated by global localization. Some popular methods observe the changes of probabilities of randomly distributed samples and determine the localization quality including a localization failure [4,12]. In these approaches, the important issue is the distribution of random samples in free space.

In this paper, we propose a novel approach to reduce the number of samples used in the particle filter and thus, to decrease the localization time during the initial operations of MCL. And a scheme using random samples for localization failure detection and recovery is also proposed. The thinning edges of the environment are exploited for these purposes. As a robot moves, a local grid map is built from the range data of a laser scanner by the occupancy grid mapping method [3], and the thinning edges are then extracted from the occupancy grids by a thinning algorithm in real time [7]. The robot then navigates along this local thinning edge, which is similar to the one obtained off-line from the given map data. Once the robot's motion is constrained on this local edge during global localization, samples are drawn only around the neighborhood of the thinning edges instead of over the entire free space to reduce the sample size. While the robot pose is estimated by the convergence of the samples, some random samples are added and distributed uniformly around the thinning edge. By analyzing the probabilities of the additional samples, the localization quality can be checked.

The remainder of this paper is organized as follows. Section 2 briefly introduces the Monte Carlo localization (MCL) and Section 3 presents a method for extracting the thinning edges in real time. The proposed MCL/TE (MCL with thinning edges) for reducing the operation time of MCL and for detecting a localization failure is discussed in detail in Sections 4 and 5. Conclusions are drawn in Section 6.

2. MONTE CARLO LOCALIZATION (MCL)

2.1. Bayes filtering

MCL represents the robot's positional certainty at an arbitrary location in a given grid map. A robot calculates the posterior probability $Bel(x_t)$ by using the Bayes filter [2] based on the odometry and range data as follows:

$$Bel(x_t) = p(x_t \mid z_{0:t}, u_{0:t}), \tag{1}$$

where x_t denotes the robot pose (x, y, θ) at time t, $z_{0:t} = \{z_0, z_1, \ldots, z_t\}$ denotes the measurements of the range sensor (e.g., laser scanner, sonar, IR sensor, etc.) up to t, and $u_{0:t} = \{u_0, u_1, \ldots, u_t\}$ is the odometric data from the wheel encoder. Reliability of measurements varies with the accuracy of the range sensor. In order to cope with various uncertainties, probability models are used to reflect the errors: sensor model (or perception model) and motion model (or action model). The Bayes filter is conducted in two steps (i.e., prediction and update), which can be represented by the following equation:

$$Bel(x_t) = \eta\, p(z_t \mid x_t) \int p(x_t \mid x_{t-1}, u_t)\, Bel(x_{t-1}) dx_{t-1}, \tag{2}$$

where η is the normalizing constant, $p(x_t \mid x_{t-1}, u_t)$ is the motion model, and $p(z_t \mid x_t)$ is the sensor model. We assume that both the motion and sensor models are described by a Gaussian distribution and that the noise can be modeled as Gaussian noise with zero mean.

2.2. Particle filters

The particle filter used in MCL represents the posterior distribution $p(x_t \mid z_{0:t}, u_{0:t})$ using a set of random samples. Each particle (or sample) corresponds to the robot pose (x, y, θ). Among the several variants of the particle filter, the SIR (Sampling Importance Resampling) algorithm is adopted in this research [6]. This algorithm is composed of the following three steps; sampling, importance weighting and resampling. In sampling, a new sample set X'_t is generated according to the motion model $p(x_t \mid x_{t-1}, u_t)$ from the past sample set X_{t-1} distributed by $Bel(x_{t-1})$. In importance weighting, the importance factor $\omega_t^{(i)}$ is evaluated using the sensor model.

$$\omega_t^{(i)} = \eta\, p(z_t \mid x_t'^{(i)}), \tag{3}$$

where η is a normalization constant and $x_t'^{(i)}$ is an element of set X'_t. $p(z_t \mid x_t'^{(i)})$ is calculated using the similarity measure function in [10], which calculates the range differences between the laser scan data z_t and the expected reference range data computed from a grid map at the expected position of the i-th sample, $x_t'^{(i)}$. In resampling, a new sample set X_t is randomly chosen from X'_t according to the distribution defined by the importance factor $\omega_t^{(i)}$.

$$\mathbf{X}_t = \{x_t^{(j)} \mid j = 1 \cdots N\} \sim \{x_t'^{(i)}, \omega_t^{(i)}\} \tag{4}$$

The prior probability of each sample of the new sample set X_t at time t is initialized to $1/N$. Through the recursive computation of three steps, the samples converge to the pose with the highest probability.

3. THINNING EDGE

3.1. Topological map building based on thinning algorithm

Topological information is an abstraction of the environment in terms of nodes representing distinct places and edges connecting these nodes together. Topological information can be generated either by the GVG (Generalized Voronoi Graph) method [1] or the thinning method described below. The GVG method is robust to various environments and can be extended to higher-dimensional space. However, the map creates boundary edges and weak meet points, which are

unnecessary in navigation. The thinning method is an alternative approach and does not create such unnecessary information. It is also robust to sensor noises and various environments because it is based on a probabilistic framework.

A thinning method is a type of image processing algorithm which is used to detect the skeletons of images [14]. Fig. 1 illustrates the concept of thinning. The objects on the left can be described satisfactorily by the structure composed of connected lines (i.e., 'T' shape drawn with thin lines on the right). Note that the connectivity of the structure is still preserved even with thin lines. In the case of mobile robots, these connected lines can be used as paths by which a robot can navigate without colliding with other objects.

The thinning process is applied to the free space shown in Fig. 2. Then, the free space is continually contracted from both the outside of the objects and the inside of the wall boundary. This thinning process is repeated until a skeleton corresponding to the thinnest line for the free space is extracted. After the edges are extracted through the thinning process, nodes can be extracted, as shown in Fig. 1. An end node corresponding to an end of each edge represents the dead end of the environment (e.g., dead end of the corridor). A branch node, at which more than three edges meet, represents a junction (e.g., intersection of corridors).

The thinning-based topological map (TTM) is constructed as follows. The robot collects the range data by scanning the environment using a laser rangefinder. Each cell in the occupancy grids is likely to be scanned several times. The occupancy probability for each cell is then updated based on the Bayesian update formula. This probabilistic approach to building the local occupancy

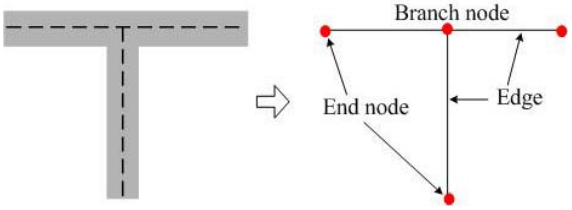


Fig. 1. Extraction of topological edges and nodes by thinning algorithm.

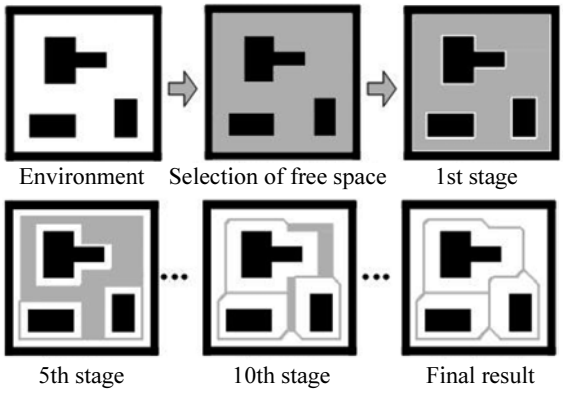


Fig. 2. Construction of a thinning-based topological map for given environment (Simulations).

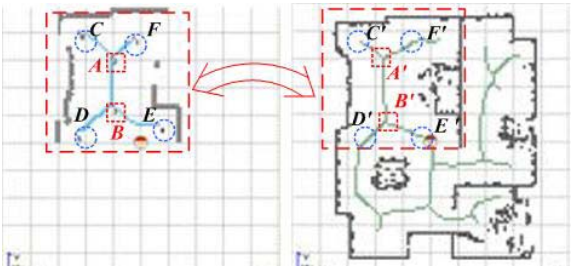
grids enhances the confidence of the underlying grid map used for building a local topological map. At each sampling instant, based on the range data, the local grid map and the subsequent local topological map are built. The local topological map even for the same space is continually changed as the underlying grid map is updated. The computation time to extract topological nodes and edges from the grid map of 10m x 10m in size was less than 30 ms with a 1.7 GHz Pentium 4 notebook computer. More detailed information on the thinning-based topological mapping can be found in [7,9].

The TTM is built in real time and its pure topological information is stored in the graph structure. In this case, the information on the nodes and edges is stored, while the metric information is discarded. This topological information can be exploited for other navigation tasks such as topological localization, topological exploration and so on. In this research, however, the metric information of the edges is added to the grid map and exploited to improve MCL.

3.2. Local and global thinning edges and their similarity

A thinning edge is a skeleton of the environment extracted by applying the thinning algorithm to a grid map. In this research, two types of grid maps (i.e., local and global grid maps) are used; therefore, two types of thinning edges, the local and global thinning edges, can be considered. The global grid map, which represents the whole environment, is usually given in advance for localization. When the thinning algorithm is applied to this map, the whole skeleton of the environment can be extracted to obtain global thinning edges. The global thinning edges cannot be extracted in real time for a large grid map. This type of extraction is, however, not important because the global grid map is not changed; thus, the global thinning edges are extracted just once. The metric information of the global thinning edges is stored in the global grid map in this research. On the other hand, the local thinning edges are extracted from the local grid map that is built during navigation using both the odometric and range data. These edges can be continuously changed when a new environment is sensed and mapped. In this research, both local grid and topological maps are 10m x 10m in size, and the recently sensed region around a robot can be mapped in this local map. This size can be changed depending on the computational ability of the system.

The key idea of the proposed scheme is the distribution of random samples of the particle filter around the global thinning edges while robot motion is constrained such that it moves along the local thinning edges. Therefore, it is important to investigate whether the local thinning edges coincide with the global thinning edges. Fig. 3(a) shows one set of local thinning edges obtained in real time. Fig. 3(b) shows the global thinning edges that were computed from a given global grid map by applying the thinning algorithm to the free space. These edges show strong similarity. More specifically, the resulting locations of nodes A, B, C, D, E well match the corresponding locations of nodes or edges A′, B′, C′, D′,



(a) (b)

Fig. 3. Comparison of local with global thinning edges (Experiments); (a) local thinning edges generated using sensor data, and (b) global thinning edges generated using the given grid map.

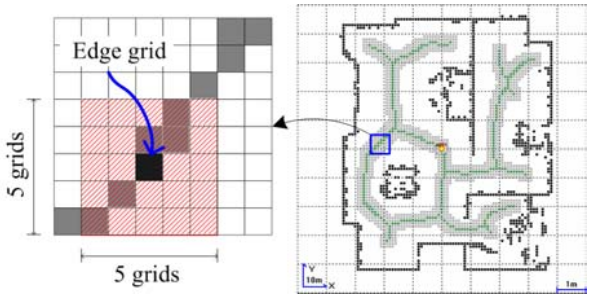
E′, respectively. Although there is a slight positional difference between F and F′, the difference is smaller than 0.8m, which is acceptable in the proposed method. The positional difference can be overcome by spreading the samples around the global thinning edges (not on the edges alone).

The thinning edge is especially useful in an environment of significant geometrical connectivity between sub-spaces. For example, one of the best target environments is an indoor building, which includes office rooms and corridors. In such an environment, the shapes of the thinning edges are quite robust and invariant [15]. However, there are some environments where the differences between global and local thinning edges should be taken into account, such as in the case when the robot faces a small region around the corner or travels in a large-scale, open environment. Sometimes, the local thinning edges are affected by the visibility of the sensor. However, such cases are not common, and the robot can carry out successful localization as it continues to move along the local thinning edges.

4. MCL/TE FOR INCREASING CONVERGENCE SPEED

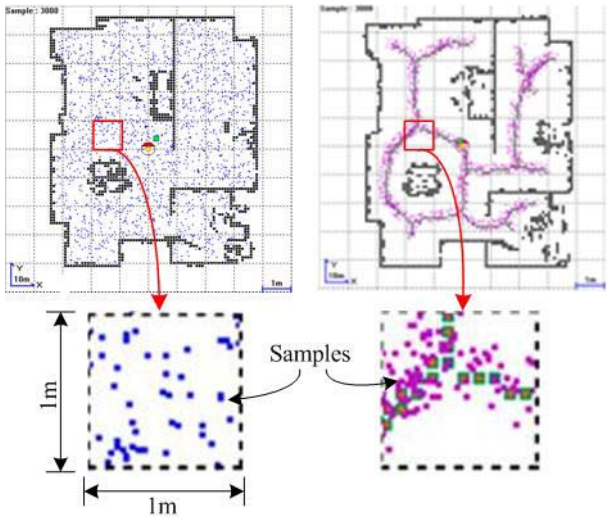
4.1. Sampling on region near edges

MCL estimates the robot pose by comparing the range data measured from a laser scanner with the reference range data computed from a given environment map (in this case, an occupancy grid map). In the proposed MCL/TE (MCL with thinning edges), a global thinning edge is extracted, often off-line, by applying the thinning algorithm to a given global grid map, and then random samples for the particle filter are then drawn only around the global thinning edges rather than over the entire free space. This approach can be made feasible when the robot is located on the thinning edge before its pose is estimated globally since the samples representing the possible robot poses are drawn only around these edges. To ensure appropriate positioning of the robot, a local grid map and its thinning edges are generated from the range data measured by the laser scanner. As explained in section 3B, the local thinning edges are slightly different from the global ones. In order to deal with these differences, we extended the sampling area to the



(a) (b)

Fig. 4. Sampling region near thinning edges.



(a) Standard MCL. (b) MCL/TE.

Fig. 5. Difference in sampling area.

neighborhood of the thinning edges. In this experiment, the size of the extended region was set to 25cm, which corresponded to 2.5 grids when the size of the grid was set to 10cm × 10cm. As shown in Fig. 4(a), the sampling area is a 5 grid by 5 grid rectangular region including the edge itself. Fig. 4(b) illustrates the region where the samples are drawn according to the proposed scheme. If the robot's location is sufficiently close to the edge, MCL can be successfully carried out. The region of sample distribution can be remarkably reduced from the entire free space.

Fig. 5 illustrates the difference in sampling area between the standard MCL (Fig. 5(a)) and the proposed MCL/TE for an environment of 10m × 10m. Figs. 5(a) and 5(b) show that MCL/TE has much smaller region of sample distribution than the standard MCL. Therefore, if the number of samples is fixed, the sample density of the MCL/TE can be much higher than that of the standard MCL, implying that the accuracy of MCL can be improved significantly. If the sampling densities are identical, the number of samples can be significantly reduced when MCL/TE is employed, so the computational cost of MCL can be considerably reduced.

4.2. Procedure for MCL/TE for increasing convergence speed

In this paper, the thinning edge is used for two purposes.

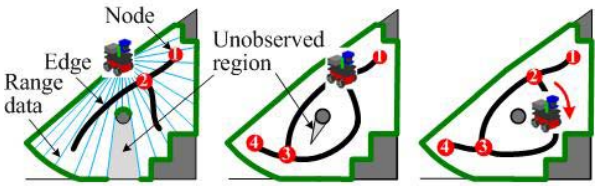


Fig. 6. Robot motion using thinning edges.

The thinning edges can be used as both the region of sample distribution and the navigation path on which a robot can navigate without colliding with nearby objects until the initial localization is completed. The thinning edges are suitable as navigation paths because of the wide field of view of the range sensor such as the laser rangefinder. That is, if the robot travels along the mid-lines between obstacles, which correspond to the thinning edges, it has a better chance of collecting more environmental information than traveling on other paths. With more environmental information, MCL can be performed more successfully.

Fig. 6 illustrates the robot movement on the edges. First, the robot builds a local grid map from the range data, and then extracts the thinning edges and nodes as shown in Fig. 6(a). Some regions behind obstacles may not be observed at this time. In Fig. 6(b), the robot moves to the nearest node, and MCL is conducted. While MCL is in progress to globally estimate the robot pose, the robot moves along the newly extracted edges to collect more environmental information for MCL. As the robot moves, the unobserved regions can be sensed as shown in Fig. 6(b) and (c), and the thinning edges extracted from the local grid map become more similar to those from the global grid map as explained in Section 3.2.

If the robot knows that it is moving along the local thinning edges, the robot computes its pose based on the samples drawn in the neighborhood of the global thinning edges. Fig. 7 shows the localization procedure for the proposed MCL/TE. The local thinning edges around the robot's current pose are built as shown in Fig. 7(a). In Fig. 7(b), the initial random samples are drawn around the global thinning edges for MCL/TE. Samples converge to the several regions in Fig. 7(c) after the robot's movement and MCL execution. Fig. 7(d) represents the result of pose estimation after the completion of global localization using MCL/TE. The real robot pose is in the circle in Fig. 7(d). Once the robot pose is globally estimated by MCL/TE, the robot does not have to move along the thinning edges any further, and the robot pose is estimated by the standard MCL.

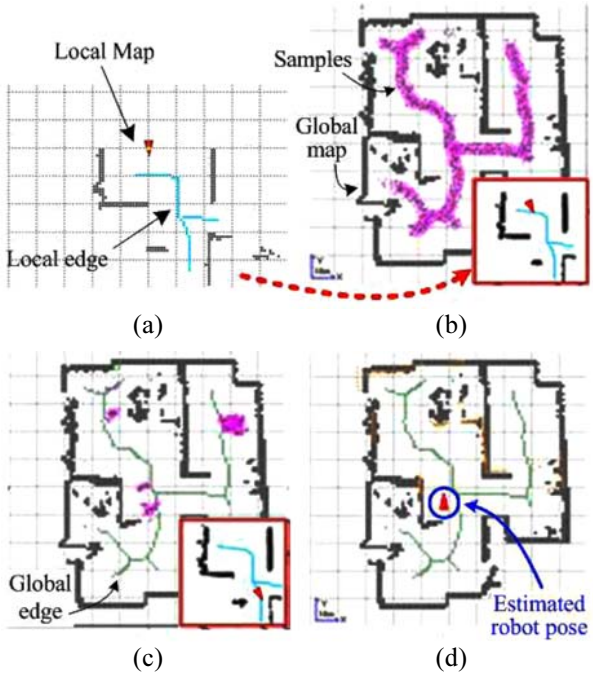


Fig. 7. MCL process using thinning edges.

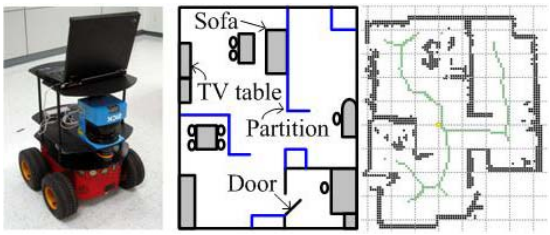


Fig. 8. (a) Experimental setup, and (b) experimental environment and its grid map and global thinning edges.

4.3. Experiments

A Pioneer 3 robot equipped with the SICK LMS 200 laser rangefinder was used in this study (see Fig. 8(a)). A Pentium 4 notebook was used to control a robot and execute the proposed algorithm. Its clock cycle was 1.7 GHz and it had 1 GB main memory. The average velocity of the robot was 0.2m/s. Fig. 8(b) shows the room with different pieces of furniture including tables, a sofa, and chairs. The local map of this room is 7m x 9m in size and its grid size is set to 10cm x 10cm.

4.3.1 Number of samples

In this section, we employ two performance measures to evaluate the localization performance of MCL/TE. One is the localization failure rate and the other is the computation time. The standard MCL and the proposed MCL/TE were compared experimentally to investigate the usefulness of the proposed scheme.

If the number of samples is too small, the localization failure rate of MCL increases because the samples are not sufficient in number to represent the robot pose. On the other hand, the computational burden becomes a problem when too many samples are used. A localization failure rate is defined as the ratio of the number of failed localization experiments to the total number of experiments. Fig. 9(a) shows the experimental results of the localization failure rates. In all experiments, the robot motion was always constrained on the thinning edges for both MCL/TE and standard MCL.

The localization failure rate of MCL/TE was signifi-

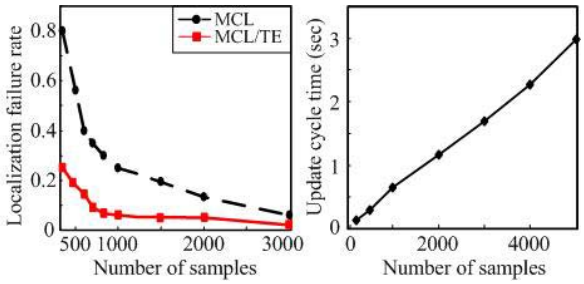


Fig. 9. Localization failure rate and cycle time according to sample size; (a) localization failure rates of MCL and MCL/TE, and (b) cycle time of MCL algorithm.

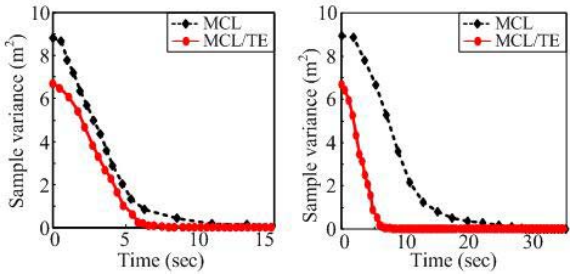


Fig. 10. Sample variances with time; (a) variance of sample poses for identical sample size (900 samples), and (b) variance of sample poses for identical localization failure rate with 3000 samples for MCL and 900 samples for MCL/TE.

cantly lower than that of the standard MCL in Fig. 9(a) for the same number of samples. For example, when 1,000 samples were used for the map of Fig. 8, MCL/TE had a failure rate of 6%, which was much lower than that of 25% for the standard MCL. The proposed scheme performed better because the sample density of MCL/TE was much higher than that of the standard MCL for the same number of samples. Moreover, the localization failure rate of MCL using 3,000 samples was similar to that of MCL/TE using 900 samples. This result indicates that the required number of samples can be reduced to 30% of that required for the standard MCL for a typical indoor environment like Fig. 8.

Fig. 9(b) shows the update cycle time of MCL, which is required to update the robot pose based on the sensor data. In this experiment, 900 samples were used in MCL/TE, and 3000 samples in MCL, as shown in Fig. 9(a). The average cycle time of MCL was 1.8sec, and that of MCL/TE was 0.6sec, as shown in Fig. 9(b). Note that the update cycle time depends only on the number of samples for both MCL and MCL/TE since these two algorithms are identical except for the distribution of samples.

This reduced cycle time is very advantageous in practical applications for the following reasons. The measurement time required to collect all 181 range data using a laser rangefinder and make them available to MCL operation is usually 0.2 to 0.3 seconds. When 3,000 samples are used, 6 to 9 sensor scans are missed during one localization update cycle, and the traveling distance during this update is 0.36m at a robot speed of 0.2m/s. Hence, the current robot position estimate may differ from the correct one by as much as 0.36m, thus leading to an incorrect position estimate. If the number of samples is reduced to 900, only 2~3 sensor scans are missed and the traveling distance is reduced to 0.12m during one update, which results in more accurate pose estimation. Similar comparison has been studied in [4], where the localization accuracy was analyzed according to the number of samples. The result also showed that the localization accuracy could be deteriorated for large sample sets because of the computational burden.

4.3.2 Transition of variance of sample pose

Since random samples tend to converge to the real robot pose if MCL is conducted successfully, it is meaningful to investigate the sample variances. Fig. 10(a) shows the variances over time. For both cases, 900 samples were used. The variances started to decrease from the start of MCL, and it took about 6 seconds to converge for an average of 20 experiments. The dots marked the instants at which each MCL cycle was completed. The initial sample variance in MCL/TE was lower than that of MCL. Furthermore, MCL/TE showed faster convergence than MCL as expected. On the other hand, the localization failure rate of the standard MCL with 3000 samples was similar to MCL/TE with 900 samples, as shown in Fig. 9(a). The experimental results for these conditions are shown in Fig. 10(b). Although the failure rates of two methods were similar, the convergence rate of MCL/TE was much faster than that of the standard MCL.

5. MCL/TE FOR DETECTING AND SOLVING LOCALIZATION FAILURE

5.1. Criterion for evaluation of localization quality

Localization often fails to provide the correct estimation of the robot pose for various reasons such as incorrect sensor data, cluttered environments, kidnapping and so on. Because localization failure is inevitable to some extent, the robot should be able to autonomously detect and recover from failures. In this paper, a heuristic method for detecting localization failure is proposed. This method utilizes the thinning edges to detect localization failure.

One common method for detecting localization failure is to add a small number of random samples *globally* during the MCL operation. This scheme assumes there is a small probability that the robot will lose track of its pose. The key issue is the distribution scheme of these additional samples over the entire environment. If a vision sensor and artificial landmarks are used, random samples can be distributed on relatively limited regions around the landmark, which the robot sees at that time [12]. If a range sensor and natural landmarks are used, it is not easy to determine the distribution of random samples over the entire free space. However, if thinning

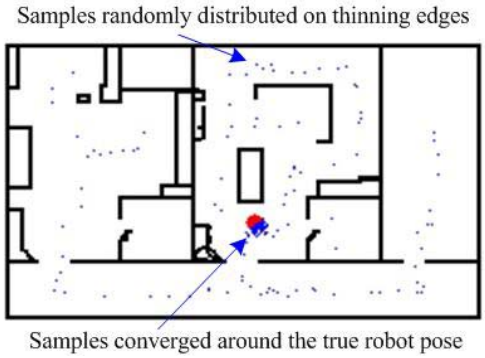


Fig. 11. Additional random samples are distributed on the thinning edges while the original samples are converged around the robot pose. There are 100 converged samples and 100 random samples on the thinning edges.

edges are exploited, samples can be drawn less randomly than the normal random distribution.

In the proposed scheme, a small number of random samples are added on the thinning edges as the original random samples converge to a certain small region containing the true robot pose for tracking, as shown in Fig. 11. Note that in this paper, *random samples on the thinning edges* and *converged samples* indicate two different sample groups; the random samples on the thinning edges are the additional ones used to cope with localization failure. As long as the robot navigates on a path which is away from nearby obstacles, some samples belonging to the random samples on the thinning edges are likely to represent the real robot pose because the thinning edges are similar to the mid-lines of the free space. Then, localization quality can be evaluated by comparing the probabilistic characteristics of the converged samples with those of the random samples on the thinning edges. The actual robot pose can be represented by the converged samples only for a successful localization. Some variables are described in Table 1 to introduce a criterion for evaluating localiza-tion quality. A_{conv} and A_{rand} mean the average probabilities of the converged samples and the random samples on the thinning edges, respectively. N_{conv} and N_{rand} are the numbers of samples for two groups. The prior probability of each sample is set to $1/(N_{\text{conv}}+N_{\text{rand}})$ in the resampling step of MCL, as described in section 2.

Fig. 12 shows the changes in the probabilistic characteristics of the converged samples and random samples on the thinning edges according to the change of the localization state. The localization quality in intervals A, B and C corresponds to *normal*, *warning* and *failure*, respectively. Since every converged sample is near the

Table 1. Some variables defined for analysis.

	Converged samples	Random samples on the thinning edges
Average probability of samples	A_{conv}	A_{rand}
Number of samples	N_{conv}	N_{rand}
Prior probability	$P_{prior} = 1/(N_{conv}+N_{rand})$	

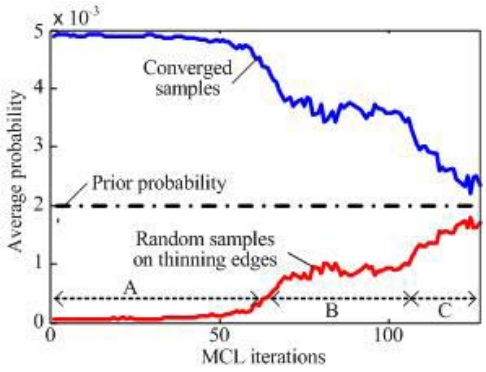


Fig. 12. Average probability according to the state of localization.

actual robot pose in interval A, the average probability of the converged samples is higher than that of the random samples on the thinning edges. Because the robot pose is rather inaccurately estimated in interval B, the difference between the probabilities of the converged samples and the random samples on the thinning edges decreases. In interval C, the average probabilities of the converged samples and random samples on the thinning edges approach the prior probability because the probability of the converged samples are as incorrect as those of the random samples on the thinning edges.

The reasonable number of samples for tracking the robot pose depends on the type of environment, and the average probabilities of the two groups of samples also depend on the numbers of the samples. For example, the values of the average probabilities of the two groups in Fig. 12 become smaller for a larger number of converged samples. The localization state, however, needs to be determined regardless of the number of samples. To do this, random samples on the thinning edges are added in number equal to that of the converged samples, or $N_{\text{conv}}=N_{\text{rand}}$. It is reasonable that these additional samples do not increase the computational burden much because MCL tracks the robot pose with a small number of converged samples. With this condition, a new criterion C_{state} is introduced as follows;

$$C_{\text{state}} = \frac{A_{conv}}{A_{rand}}, \quad \text{when} \quad N_{conv} = N_{rand}. \tag{5}$$

Two sample sets are considered in C_{state}, and each sample set can be considered as one candidate of the robot pose because N_{conv} is equal to N_{rand}. The robot pose can be estimated by these two candidates, and A_{conv} and A_{rand} are the probabilities of the two candidates, respectively. A_{conv} and A_{rand} become high and low, respectively, if the converged samples successfully represent the true robot pose while A_{conv} becomes low with the poor localization quality. Therefore, the ratio of the two probabilities, as given by (5), can be exploited to evaluate the localization quality, and the localization state can be determined according to C_{state} if two thresholds between normal and warning states and between warning and failure states are set, respectively. To set the thresholds, the meaning of C_{state}, the ratio of

the probabilities of two candidates, should be considered. For example, C_{state} equal to 1 means that the reliability of the estimated pose by the converged sample set is equal to that by the random samples on the thinning edges. In this experiment, two threshold values were selected for the following reasons.

- When the reliability of the pose estimated by the converged sample set is more than 85%, the localization state is classified as a normal state. The sum of (A_{conv}* N_{conv}) and (A_{rand}* N_{rand}) is the total probability equal to 1.0 and N_{conv} is equal to N_{rand}. Therefore, A_{conv} and A_{rand} are equal to $0.85/N_{\text{conv}}$ and $0.15/N_{\text{conv}}$, respectively, when the localization state is normal. In this case, C_{state} is 5.7 by equation (5), and the pose estimated by the converged samples is 5.7 times more reliable than that by the random samples on the thinning edges.
- Similarly, when the reliability of the pose estimated by the converged sample set is less than 70%, the localization state is classified as a failure or kidnapped state. Therefore, A_{conv} and A_{rand} are equal to $0.70/N_{\text{conv}}$ and $0.30/N_{\text{conv}}$, respectively, when the localization state is a failure. In this case, C_{state} is 2.3 by equation (5) and the pose estimated by the converged samples is 2.3 times more reliable than that by the random samples on the thinning edges.
- If C_{state} is between two thresholds, 2.3 and 5.7, the localization state is classified as a warning state. Therefore, the two threshold values, 2.3 and 5.7, are set to classify localization into three states.

There are two modes of MCL, namely, the position tracking mode and the global localization mode. The mode is determined according to the standard deviation of the samples. The criterion, C_{state}, is used only during the position tracking mode, in which the robot pose is estimated by some samples converging near the true robot pose. In this mode, the standard deviation of the samples is small. However, when a robot continues to move after the samples converge to the incorrect pose or kidnapping occurs, samples diverge and thus, the standard deviation of the samples becomes large. Regardless of C_{state}, localization switches into the global localization mode, provided that the standard deviation of the converged samples becomes larger than the predefined threshold. In this case, the MCL/TE algorithm, described in Section 4, globally seeks the robot pose and stops adding random samples until the standard deviations of all samples become smaller than the threshold deviation. If this global localization is successful, the robot can estimate its pose correctly again.

Table 2. States of localization and possible actions.

State of localization	Values of variables	Possible actions to overcome problems
Normal	$C_{\text{state}} > 5.7$, $A_{conv} >> P_{prior} >> A_{rand}$	
Warning	$2.3 \leq C_{\text{state}} \leq 5.7$, $A_{conv} > P_{prior} > A_{rand}$	Motion model with large uncertainty
Failure (Kidnapping)	$C_{\text{state}} < 2.3$, $A_{conv} \approx P_{prior} \approx A_{rand}$	Motion model with large uncertainty + addition of samples
Global localization	Standard deviation of converged samples is larger than predefined threshold.	Global localization by MCL/TE

If the localization state is determined as warning by C_{state}, it was suspected that the wheel slippage occurred and the estimated pose had much uncertainty. There are several solutions to this problem, and in this research, a different motion model for warning and failure states from the original motion model was used. This motion model considers more the uncertainties of odometric data than the original motion model for the normal state, and therefore, the converged samples spread more widely. If a discrepancy between the true and the estimated robot pose is not large, the localization state will be recovered to normal. On the contrary, if the localization quality is not recovered and a robot continuously moves, the localization state can be changed into failure. Other situations such as kidnapping can also lead the localization state to failure. In this case, the change of motion model is not sufficient, and more samples are required to recover the localization quality. The additional samples, therefore, were distributed around the estimated pose in this experiment. If the localization quality is not recovered by these actions, the samples continue to diverge and the localization mode will be changed from the tracking mode to the global localization mode. Table 2 shows the states of localization and possible actions according to C_{state}.

Fig. 13 shows one example. The range data and the given map are shown in Fig. 13(a). The robot is successful in estimating its true pose in case *A* and thus the range data coincide with the map. In this case, the value of C_{state} is greater than 5.7, indicating that the robot is in the normal localization state. In case *B*, C_{state} drops below 2.3 due to wheel slippage. The robot is then aware of the localization failure, changes the motion model, and widely re-distributes the converged samples in the subsequent resampling step. In this experiment, both 250 converged samples and 250 random samples are used, and the total number of samples is fixed to 500. The number of samples is sufficiently large and no additional samples are needed to recover the robot pose in the localization failure state. After some MCL iterations, the robot can estimates its pose near the true pose, and then C_{state} becomes greater than 2.3 in case *C*. However, the robot will still distribute the converged samples rather widely since the localization is in a warning state. After all, the robot accurately finds its true pose in case *D* and the C_{state} value becomes greater than 5.7 again.

Each time the robot pose is estimated in the tracking mode, some random samples on the thinning edges are added to the thinning edges over the entire environment and C_{state} is computed to determine the state of localization. In this experiment, the motion model, which has larger uncertainty than that of the normal state, is

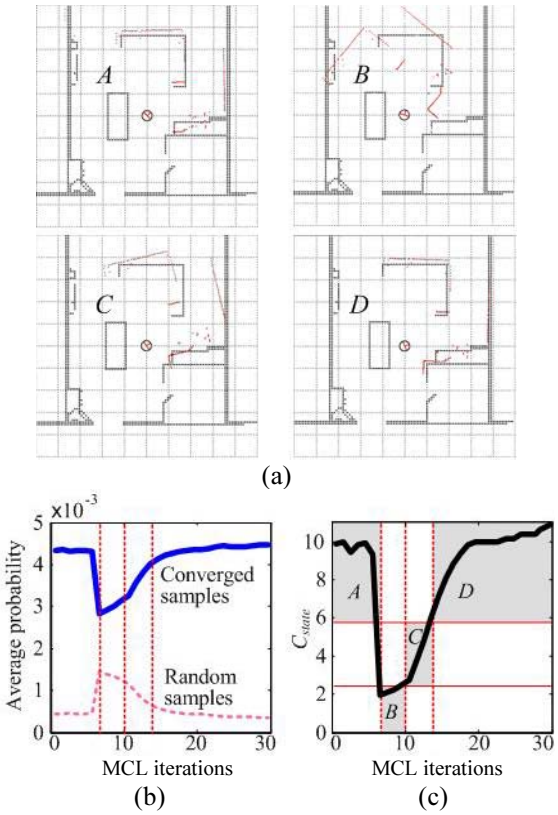


Fig. 13. Wheel slip detection and recovery using MCL/TE; (a) range data and map, (b) average probabilities changing with MCL iterations, and (c) localization states, *A*: normal, *B*: localization failure, *C*: localization warning, and *D*: normal.

used in the sampling step of MCL to recover to the correct robot pose if the state is determined as a localization warning or failure. Then C_{state} and the standard deviation are evaluated at each MCL iteration as localization continues.

5.2. Experiments

Though the samples converge around small regions, it is possible that the estimated robot pose is not the true one. This is due to either a localization failure or kidnapping. To detect this situation, MCL/TE generates some random samples on the thinning edges and distributes them around the thinning edges in every MCL cycle, as shown in Fig. 11. Then, the robot calculates C_{state} and investigates the localization state, as described above. At this moment, the numbers of converged samples and random samples on the thinning edges do not affect the value of C_{state}. In this experiment, a small number of random samples (equal to the number of converged samples for local tracking) are always added around the thinning edges over the entire environment whose size is about 20m by 10m.

Fig. 13 shows that the robot lost track of its pose but the samples were still near the true pose. In this case, the robot can recover its pose by distributing samples more widely. However, if the difference between the true and the estimated pose is large, the robot will not likely be able to recover its pose by this approach. This situation is shown in Figs. 14 and 15. In Fig. 14(a), the robot mistakes the incorrect pose for the true one because the two rooms look very similar to each other and region *A*, especially, has the same shape as B, as shown in Fig. 14(b). Therefore, the C_{state} value is large, as shown in Fig. 15(d). Fig. 14(b) shows that as the robot moves, the accumulated odometric error increases and C_{state} continuously decreases since the range data do not coincide with the map. Regions *A*, *B* and *C* in Fig. 15(d) also represent this situation. If the robot moves further, the converged samples gradually diverge because the sensor data are not similar to the map of the environment and the standard deviation of samples becomes larger, as

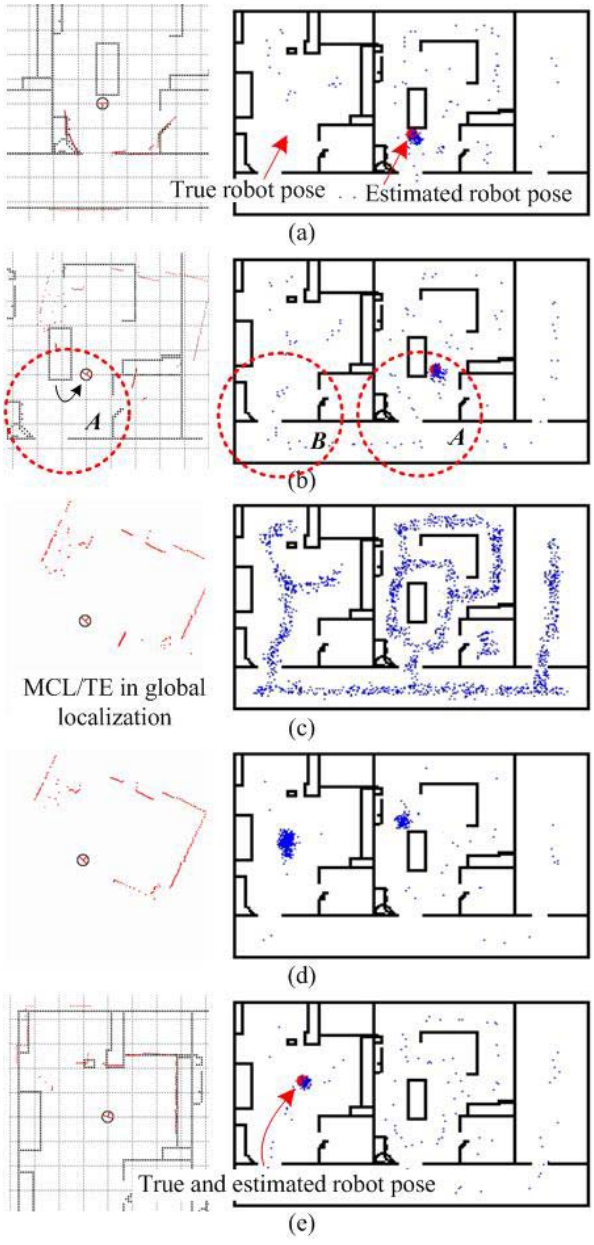


Fig. 14. Detection of localization failure and recovery using random samples on thinning edges. Global maps and samples (right) and range data and map (left).

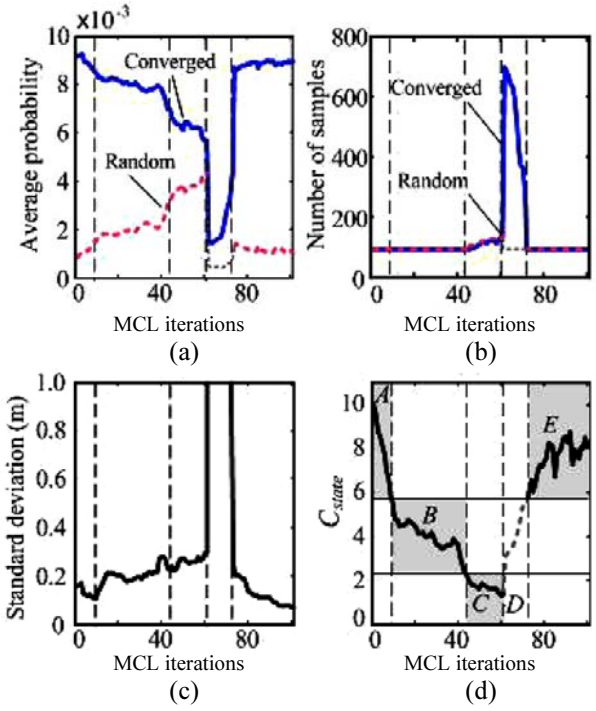


Fig. 15. Detection of localization failure and recovery using MCL/TE; (a) average probabilities changing with MCL iterations, (b) number of samples, (c) standard deviation of converged samples with MCL iterations, and (d) localization states, A, E: normal, B: warning, C: failure, D: global localization.

shown in Fig. 15(c). After all, the standard deviation of samples becomes larger than the predefined threshold deviation, and the localization state is switched to global localization. Fig. 14(c) and (d) and region *D* in Fig. 15(d) show this state. During global localization, the random samples on the thinning edges to evaluate the localization state are not required, so the probability of region *D* in Fig. 15(a) and C_{state} of region *D* in Fig. 15(d) are meaningless. Random samples are distributed around the thinning edges by the MCL/TE to increase the convergence speed; this procedure was described in section 4. The probabilities of the random samples are updated continuously, as shown in Fig. 14(d). After a few iterations (about 12 in this experiment), the samples converge around the true robot pose, as shown in Fig. 14(e) and in region *E* in Fig. 15(d), and the C_{state} value increases.

6. CONCLUSIONS

In this paper, we proposed a novel approach to reduce the number of samples used for the initial phase of the MCL process. A simple but useful method for the detection of localization failure and recovery from such failure was also proposed. With this method, the robot can globally seek its pose faster than it would with the standard MCL algorithm, and then it can autonomously detect and recover from a localization failure. This method, MCL/TE (MCL with thinning edges), was investigated by a series of experiments. From this research, the following conclusions have been drawn.

1) By constraining the robot motion on the thinning edges, which can be generated in real time from the range data, the region of sample distribution can be significantly reduced during global localization.
2) The proposed MCL/TE method improved the localization performance in terms of the convergence speed and failure rate. The increased update rate of MCL contributed to the reliability of practical localization performance.
3) The localization state can be determined by analyzing the ratio between the probabilities of the samples converged near the robot and the samples randomly distributed around the thinning edges.
4) Depending on the localization quality, the proper localization scheme can be selected, such as MCL/TE for increasing convergence speed. Then, the localization state can be restored to the 'normal' state.

As mentioned above, there are some environments where the differences between the global and the local thinning edges are large. In this case, the proposed method might not be able to find the robot pose correctly because of these differences, even though the robot continues to move and the difference decreases. Of course, the proposed MCL/TE method can detect the localization failure and recover the robot pose, but this response may take longer than the standard MCL. This is a weak point of this proposed approach, and the research on checking the differences between the global and local thinning edges is under way.

REFERENCES

[1] H. Choset and J. Burdick, "Sensor based planning. Part I: the generalized Voronoi graph," *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1649-1655, 1995.

[2] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statics and Computing*, vol. 10, pp. 197-208, 2000.

[3] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *IEEE Journal of Computer*, vol. 22, pp. 46-57, 1989.

[4] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, "Monte Carlo localization: efficient position estimation for mobile robots," *Proc. of the 6th National Conference on Artificial Intelligence*, pp. 343-349, 1999.

[5] D. Fox, W. Burgard, and S. Thrun, "Active markov localization for mobile robots," *Robotics and Autonomous Systems*, vol. 25, pp. 195-207, 1998.

[6] J.-S. Gutmann, W. Burgard, D. Fox, and K. Konolige, "An experimental comparison of localization methods," *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 736-743, 1998.

[7] T. B. Kwon and J.-B. Song, "Real-time building of a thinning-based topological map," *Intelligent*

Service Robotics, vol. 1, pp. 211-220, 2008.
[8] A. Widyotriatmo, B. Hong, and K.-S. Hong, "Predictive navigation of an autonomous vehicle with nonholonomic and minimum turning radius constraints," *Journal of Mechanical Science and Technology*, vol. 23, no. 2, pp. 381-388, 2009.
[9] T. B. Kwon and J.-B. Song, "Thinning-based topological exploration using position possibility of topological nodes," *Advanced Robotics*, vol. 22, pp. 339-359, 2008.
[10] D. H. Lee, Woojin Chung, and M. S. Kim, "A reliable position estimation method of the service robot by map matching," *Proc. of IEEE/RSJ International Conference on Robotics and Automation*, pp. 2830-2835, 2003.
[11] J. Leonard and H. Durrrant-Whyte, "Mobile robot localization by tracking geometric beacon," *IEEE Trans. on Robotics and Automation*, vol. 7, pp. 376-382, 1991.
[12] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, The MIT Press, Cambridge, 2005.
[13] Y. J. Lee, B. D. Yim, and J.-B. Song, "Mobile robot localization based on effective combination of vision and range sensors," *International Journal of Control, Automation, and Systems*, vol. 7, no. 1, pp. 97-104, 2009.
[14] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital pattern," *Trans. on Communication of the ACM*, vol. 27, pp. 236-239, 1984.
[15] Y. J. Lee, T. B. Kwon, and J.-B. Song, "SLAM of a mobile robot using thinning-based topological information," *International Journal of Control, Automation, and Systems*, vol. 5, no. 5, pp. 557-583, 2007.
[16] J. Bae, S. Lee, and J.-B. Song, "Use of coded infrared light for mobile robot localization," *Journal of Mechanical Science and Technology*, vol. 22, no. 7, pp. 1279-1286, 2008.
[17] C. Kwok, D. Fox, and M. Meila, "Real-time particle filters," *Advances in Neural Information Processing Systems 15*, pp. 1057-1064, 2004.

Tae-Bum Kwon received his B.S. and Ph.D. degrees in Mechanical Engineering from Korea University in 2003 and 2009. He is currently a Research Assistant Professor in Mechanical Engineering at Korea University. His research interests include mobile robotics.

Ju-Ho Yang received his B.S. and M.S. degrees in Mechanical Engineering from Korea University in 2004 and 2006, respectively. He is currently a Research Engineer in brake system team at Mando Coorporation.

Jae-Bok Song received his B.S. and M.S. degrees in Mechanical Engineering from Seoul National University in 1983 and 1985, respectively, and his Ph.D. degree in Mechanical Engineering from MIT in 1992. He joined the faculty of the Department of Mechanical Engineering, Korea University in 1993. Currently, he is a Director of Intelligent Robotics Research Center at Korea University. He is also an Editor for International Journal of Control, Automation and Systems. His current research interests include mobile robotics, and design and control of intelligent robotics systems.