

## Reduction in Sample Size Using Topological Information for Monte Carlo Localization

Ju-Ho Yang\*, Jae-Bok Song, and Woojin Chung

\* Department of Mechanical Engineering, Korea University, Seoul, Korea  
(Tel : +82-2-929-8501; E-mail: [rmfls200@korea.ac.kr](mailto:rmfls200@korea.ac.kr))

**Abstract:** Monte Carlo localization is known to be one of the most reliable methods for pose estimation of a mobile robot. Much research has been done to improve performance of MCL so far. Although MCL is capable of estimating the robot pose even for a completely unknown initial pose in the known environment, it takes considerable time to give an initial estimate because the number of random samples is usually very large especially for a large-scale environment. For practical implementation of the MCL, therefore, a reduction in sample size is desirable. This paper presents a novel approach to reducing the number of samples used in the particle filter for efficient implementation of MCL. To this end, the topological information generated off-line using a thinning method, which is commonly used in image processing, is employed. The topological map is first created from the given grid map for the environment. The robot scans the local environment using a laser rangefinder and generates a local topological map. The robot then navigates only on this local topological edge, which is likely to be the same as the one obtained off-line from the given grid map. Random samples are drawn near the off-line topological edge instead of being taken with uniform distribution, since the robot traverses along the edge. In this way, the sample size required for MCL can be drastically reduced, thus leading to reduced initial operation time. Experimental results using the proposed method show that the number of samples can be reduced considerably, and the time required for robot pose estimation can also be substantially decreased.

**Keywords:** Mobile robots, Monte Carlo localization, Particle filters, Topological map

### 1. INTRODUCTION

To navigate reliably in the environment, a mobile robot must know where it is. Localization is a method for estimating the pose of a robot. It can be divided into the global localization and local localization. Position tracking (or local localization) calculates the robot pose with the knowledge of the initial robot pose and tracks its change during movement. On the other hand, the global localization estimates the robot pose without the prior knowledge of its initial pose. Hence, the global localization is more difficult than the position tracking.

Several localization algorithms have been proposed so far; Kalman filter [1], Markov localization [2], Monte Carlo localization (MCL) [3,4] etc. and a hybrid method [5]. The Kalman filter technique is usually used when the initial pose is known. The robot estimates its pose continuously by compensating the odometric error using the range sensor data. Hence, if the initial pose is accurate and the sensor error is small, the Kalman filter approach can offer an accurate and efficient localization result. On the other hand, Markov localization is used for the global localization. In such a case using a grid map, this algorithm must calculate probability that the robot is located in each grid for all empty grids of the occupancy grid map. Thus it requires a large amount of computation time, and the accuracy of the localization is limited to the grid size because the probability is calculated by the grid. MCL (Monte Carlo localization) is also one of the global localization techniques but the implementation is faster than Markov localization because the probability computation is done only for the random samples. It is more accurate because the samples can take any pose regardless of the grid of the grid map instead of a grid on the global map. The hybrid method takes a combination of either Markov localization and a Kalman filter or a Kalman filter and MCL. Using advantages of each method, this hybrid method improves the efficiency in pose estimation.

Until now, MCL has been considered best in dealing with the pose estimation, but its implementation takes more computation time than the Kalman filter approach. Thus much research has focused on reducing the computation time by appropriately selecting the samples called particles in the particle filter. The adaptive particle filter using KLD-sampling

[6] addressed this problem. This approach chose a small number of samples if they are populated densely on a small part of the state space, or it chooses a large number of samples if the state uncertainty is high. But the initial number of samples is still the same as the method that is not applied.

If the initial number of samples in MCL can decrease, the computing time required for the MCL algorithm can be reduced accordingly. If samples are drawn from the region with high probability of the robot being located, the number of sample can be reduced significantly. This research uses topological information to find the region with higher probability. If the mobile robot travels on the voronoi diagram which is the equidistant line between obstacles, its surrounding is a good candidate for sampling. The topological edge corresponding to the voronoi diagram is generated in real-time from range sensor data with thinning-based topological mapping technique that our laboratory developed.

### 2. Monte Carlo Localization (MCL)

#### 2.1 Bayes Filtering

MCL represents the robot's positional certainty at an arbitrary location in the given grid map. A robot calculates the posterior probability, called belief, using the Bayes filter [7] based on the odometric data and the range data as follows.

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

where  $x_t$  denotes the robot pose  $(x, y, \theta)$  at time  $t$ ,  $z_{0:t} = \{z_0, z_1, \dots, z_t\}$  denotes the measurements of the range sensor (e.g., laser scanner, sonar, IR sensor, etc.) starting at time 0 up to time  $t$ , and  $u_{0:t} = \{u_0, u_1, \dots, u_t\}$  is the odometry data from the encoder. The reliability of measurements varies with the accuracy of the range sensor. The large odometric error usually occurs for various reasons such as the surface conditions (e.g., slippage, roughness) or the precision of the robot mechanism (difference in the size of both wheels, mechanical alignment of wheel axis, etc.). In order to cope with such imperfect perception, the 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, prediction and update, and given by

$$Bel(x_t) = \int p(x_t | x_{t-1}, u_{t-1}) Bel(x_{t-1}) dx_{t-1} \quad (2)$$

where  $p(x_t | x_{t-1}, u_{t-1})$  is the motion model,  $p(z_t | x_t)$  is the perception model, and  $\eta$  is the normalizing constant.

**2.2 Particle Filters**

The particle filter used in MCL represents a posterior distribution  $p(x_t | z_{0:t}, u_{0:t})$  by a set of random samples drawn from this distribution. The samples of a posterior distribution are called particles. Each particle corresponds to the robot pose  $(x, y, \theta)$ . Among several variants of the particle filter, the SIR (Sampling Importance Resampling) algorithm [8] is adopted in this research. It is performed in the next three steps.

**Sampling:** The new sample set  $X'_t$  is generated according to the motion model  $p(x_t | x_{t-1}, u_{t-1})$  from the past sample set  $X_{t-1}$  distributed by  $Bel(x_{t-1})$ .

**Importance weighting:** The importance factor  $\omega_t^{(i)}$  is evaluated using the sensor model.

$$\omega_t^{(i)} = \frac{p(z_t | x_t^{(i)})}{\eta} \quad (3)$$

where  $\eta$  is a normalization constant.

**Resampling:** The new sample set  $X_t$  is randomly drawn from  $X'_t$  according to the distribution defined by importance factor  $\omega_t^{(i)}$ .

$$X_t = \{x_t^{(j)} | j = 1 \dots N\} \sim \{x_t^{(i)}, \omega_t^{(i)}\} \quad (4)$$

At this stage, the importance factor  $\omega_t^{(i)}$  of the sample set  $X_t$  at time  $t$  is initialized to  $1/N$ . Through the recursive three steps, the samples converge to those with high probability.

**3. TOPOLOGICAL MAP BUILDING**

**3.1 Thinning Algorithm**

In the feature-based topological map, the environment is modeled by a set of geometric primitives such as edges and nodes. It has several advantages such as compactness, fast computation, natural expression to human, and so on. A topological map, however, is not appropriate for localization which requires comparison of the current map with the reference map because it has only limited feature information compared to a grid map.

To build a topological map using range sensors, a Voronoi diagram is commonly used. But it is complex and difficult to apply to arbitrarily shaped objects. In building a topological map, therefore, a thinning method was proposed in [9], which needs simpler computation than the Voronoi diagram, but can show similar performance.

A thinning method is one of the popular image processing algorithms, which have been used to detect the skeleton of images. 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 connectivity of the structure is still preserved even for representation with thin lines. In the case of mobile robots, the connected lines are used as paths on which a robot navigates without colliding with other objects.

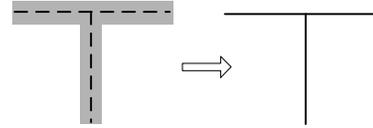


Fig. 1 Concept of thinning.

The detailed algorithm of the thinning process can be referred to in [10]. As shown in Fig. 2, the free space to which the thinning process is applied is selected. Then the free space continued to be contracted from both the outside of the objects and the inside of the wall boundary. Thinning process is repeated until the skeleton corresponding to the thinnest line for the free space is extracted.

The thinning-based topological map is constructed as follows. The robot collects the range data by scanning the environment using a laser rangefinder. Since the scanning rate is about 5 Hz and the robot navigates slowly, each cell 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 a local grid map enhances the confidence of the underlying grid map for the local topological map. At each sampling instant, based on the range data, the local grid map and the subsequent local topological map is built. The local topological map even for the same space is constantly changing as the underlying grid map is updated. Since the thinning-based topological map building can be executed fast, the topological map can be constructed on the real-time basis.

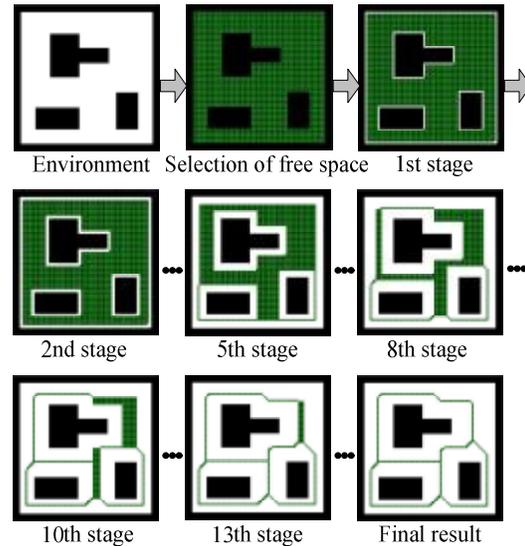


Fig. 2 Edge generation of a thinning-based topological map for given environment (Simulations).

**4. LOCALIZATION USING TOPOLOGICAL INFORMATION**

**4.1 Sampling on Region near Edge**

A Grid map is widely used for environmental representation in many applications. A topological map can be build from the grid map by using the thinning algorithm. Our idea is that the MCL samples can be distributed to the specific region, which is limited to the close neighborhood of the topological edge.

The proposed idea can be made feasible when the robot is located on the topological edge. In order to guarantee appropriate positioning of the robot, a local topological map can be build from the scanned sensor data from the laser range finder. In ideal case, the local topological map well matches with the global map. However, two maps show slight difference in practice. This difference is caused by a limited sensing range of a single scanner. In order to obtain complete geometry of the environment, a robot can be rotated at the same position.

In practice, samples can be distributed around the region which is close to the topological edge. One example is shown in Figure 3(a). Samples can be placed on the 5x5 rectangular region including the edge. Figure 3(b) illustrated the region where the samples are distributed according to the proposed idea. If the robot's location is sufficiently close to the edge, the MCL can be successfully carried out. The robot is controlled to move along the topological edge.

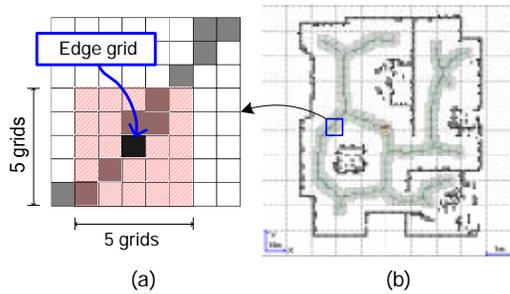


Fig. 3 Region near edge (a) and topological map applied it (b)

Figure 4(a) shows the initial sample distribution of the conventional MCL. Figure 4(b) shows the samples of the proposed idea.

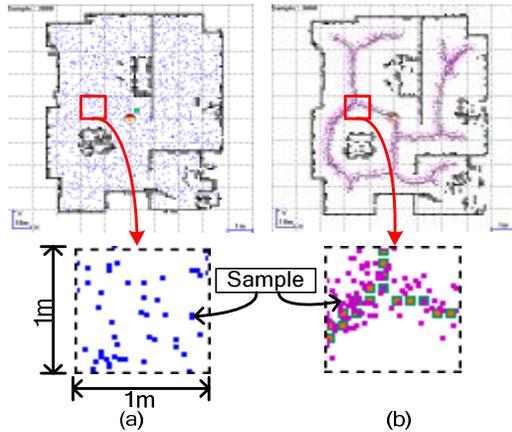


Fig. 4 General random sampling on global region (a) and sampling on region near the edge (b)

From Figure 4(a) and Figure 4(b), it is clear that the proposed method decreases the region of sample distribution. Therefore, a density of samples is higher in the proposed method, when the number of samples is fixed. This fact implies that the accuracy of the MCL can be improved. If the sampling densities are same, the number of samples can be reduced. This fact indicates that the computational cost of the MCL can be reduced.

#### 4.2 Procedure of MCL Using Topological Information

In this paper, topological information is used for two purposes. One is to use the edge as a robot's reference path. The other is to determine the region of sample distribution.

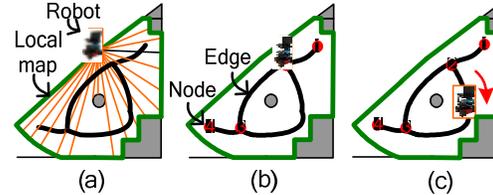


Fig. 5 Robot motion using topological information

In order to use topological information as a navigation path, the robot builds the local map and generates the edge. Because the topological node and edge information are relatively accurate and similar to the global topological map, the robot moves along the closest edge. Figure 5 illustrates the move on the edge. First the robot builds the local map and then the edges are generated as shown in Figure 5(a). In Figure 5(b), the robot moves to the nearest node after extracting a node. After reaching the node, MCL is applied. In this paper, topological information is used for two purposes. One is to use the edge as a robot's reference path. The other is to determine the region of sample distribution.

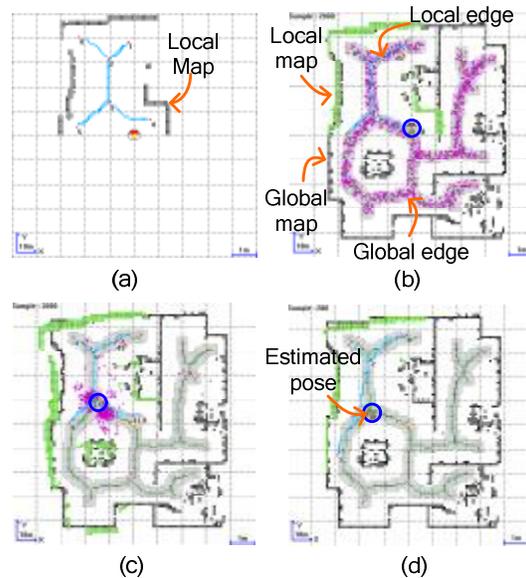


Fig. 6 MCL process using topological information (experiments)

If the robot knows that it is moving along the local topological edge, it is clear that the robot's position can be computed by samples distributed on the neighborhood on the neighborhood of the global topological edge. Therefore, global MCL can be carried out. Figure 6 shows the proposed localization procedure. The local topological edge can be obtained as shown in Figure 6(a). Figure 6(b) shows the initial sample distribution of the proposed scheme. Figure 6(c) shows

the converged samples after the robot's motion and the MCL. Figure 6(d) represents the result of pose estimation after completely MCL.

**5. EXPERIMENTAL RESULTS**

The experiments were performed by Pioneer 3 robot which is equipped with the SICK LMS 2000 Laser finder. The average velocity of the pioneer 3 was 0.2m/s. We constructed experimental environment 9.5m × 7.5m like a living room with table, couch, chairs and so on (Fig 7). A map size is 10 by 10 meters (in Figure 7(b)) and a grid resolution is 10cm.

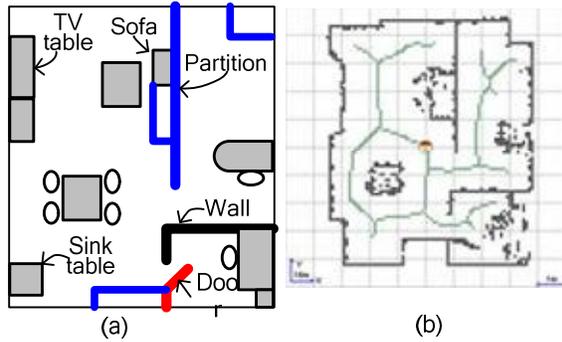


Fig. 7 Experiments environment (a) and global map (b)

In this paper, we employ two performance measures for the localization. One is a failure rate and the other is computation time. Comparison experiments were carried out between the conventional and the proposed method in order to investigate the usefulness of the proposed scheme. First, the required number of samples was considered for the case of similar failure rate. Second, resultant computational costs were compared with respect to the computing time of each algorithm.

**5.1 Number of Samples**

If the number of samples is too small, failure rate of the MCL increases. On the other hand, computational burden becomes a problem when too much samples are used. In this experiment, the MCL failure rate is investigated according to the change of the number of samples. Comparison is made between conventional MCL and the proposed scheme where the size of the environment is 10m × 10m.

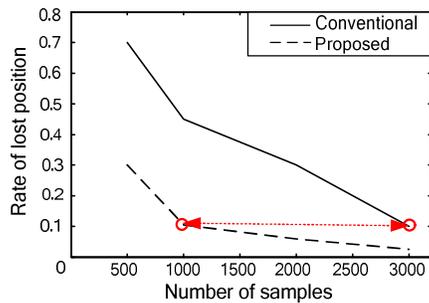


Fig. 8 Rate of lost position of standard MCL (solid line) and proposed (dash line)

From Figure 8, it can be seen that the failure rate of using 3000 samples in the conventional MCL was similar to the case of 1000 samples in the proposed approach. Therefore, it is clear the number of required samples decreased by 70%. The area of distributed samples in the proposed scheme was 1/3 of the entire environment. As the number of samples increase, the failure rate was reduced.

**5.2 Pose Estimation Time**

Experiments were carried out in order to investigate computation time for the different number of samples. The criterion of the MCL completion was determined by a threshold of sample covariance. Since the covariance shows the convergence properties of samples, covariance does not change a lot after convergence. From Figure 9, it can be seen that the computation time can be reduced by 70% when the number of samples changed from 3000 to 1000. Two experiments showed similar failure rate. This fact implies that the reduced number of samples result in fast convergence of MCL algorithm. When there are 2000 samples, failure rate decreased, and the 30% of the computational time can be saved. From the Figure 9, it is obvious that the convergence of the proposed algorithm is fast even though the number of samples is same in both in both cases. In the conventional scheme, the computing time is increased due to the additional processing of the samples which move to the occupied time is same in both cases. In the conventional scheme, the computing time is increased due to the additional processing of the samples which move to the occupied grids or out of range. The proposed scheme does not result in such inefficiency of computation. This is the reason why the propose scheme results in a short computing time at each cycle. This fact can be clearly seen from Figure 9.

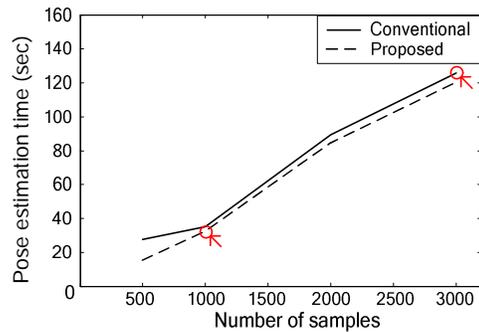


Fig. 9 Pose estimation time of standard MCL (solid line) and proposed (dash line)

**. CONCLUSION**

In this paper, we proposed a scheme to reduce the number of samples using the topological information when executing MCL. Topological edges are reference path of the robot, and the robot's locations limited to the neighborhood of the topological edges. This method reduces the number of samples and a pose estimation time when it is compared with standard MCL because efficient sampling is available by exploiting topological information. As a result, high speed computation of MCL can be achieved by the proposed scheme, while the failure rate is the same with the conventional approach. Experimental results clearly scheme that the proposed scheme

contributes proposed scheme contributes to improve the MCL performance.

#### Acknowledgments

This research was performed for the Intelligent Robotics Development Program, one of the 21st Century Frontier R&D Programs funded by the Ministry of Commerce, Industry and Energy of Korea.

#### REFERENCES

- [1] F. Lu and E. Milius, "Globally consistent range scan alignment for environment mapping," *Autonomous Robots*, Vol. 43, pp. 333-349, 1997.
- [2] D. Fox, W. Burgard, and S. Thrun. "Active markov localization for mobile robots," *Robotics and Autonomous Systems*, Vol. 25(3-4), pp 195-207, 1998.
- [3] S. Thrun, D. Fox, W. Burgard and F. Dellaert, "Robust Monte Carlo localization for mobile robots," *Artificial Intelligence*, Vol. 128(1-2), 2001.
- [4] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, "Monte Carlo Localization: Efficient position estimation for mobile robots," In *Proc. of the National Conference on Artificial Intelligence*, 1999.
- [5] J.-S. Gutmann, W. Burgard, D. Fox and K. Konolige, "An experimental comparison of localization methods," In *Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems IROS*, 1998.
- [6] C. Kwok, D. Fox and M. Meila, "Real-time particle filters," *Proceedings of the IEEE*, Vol. 92(3), March, 2004.
- [7] S. Thrun, "Probabilistic algorithms in robotics," *AI Magazine*, 2000.
- [8] A. Doucet, N.J. Gordon and J.F.G. de Freitas, editors. *Sequential Monte Carlo Methods in Practice*, forthcoming, 2000.
- [9] G. A. Baxes, *Digital Image Processing*, John Wiley & Sons, 1994.
- [10] B. Ko, J.B. Song, "Real-time Building of Thinning-Based Topological Map with Metric Features," *Proc. of IROS*, pp. 1524-1529, 2004.