SLAM of a Mobile Robot using Thinning-based Topological Information

Yong-Ju Lee, Tae-Bum Kwon, and Jae-Bok Song*

Abstract: Simultaneous Localization and Mapping (SLAM) is the process of building a map of an unknown environment and simultaneously localizing a robot relative to this map. SLAM is very important for the indoor navigation of a mobile robot and much research has been conducted on this subject. Although feature-based SLAM using an Extended Kalman Filter (EKF) is widely used, it has shortcomings in that the computational complexity grows in proportion to the square of the number of features. This prohibits EKF-SLAM from operating in real time and makes it unfeasible in large environments where many features exist. This paper presents an algorithm which reduces the computational complexity of EKF-SLAM by using topological information (TI) extracted through a thinning process. The global map can be divided into local areas using the nodes of a thinning-based topological map. SLAM is then performed in local instead of global areas. Experimental results for various environments show that the performance and efficiency of the proposed EKF-SLAM/TI scheme are excellent.

Keywords: Mobile robots, navigation, SLAM, thinning.

1. INTRODUCTION

A mobile robot can build a map of an unknown environment and localize itself with respect to this map through on-board sensors using the SLAM (Simultaneous Localization and Mapping) technique. SLAM represents one of the most fundamental problems in the field of mobile robotics and much research has already been conducted, but a complete solution has not yet been presented. This is partly due to the fact that the estimation of robot pose is closely correlated to map building. That is, the accurate estimation of the robot pose depends heavily on an accurate map and vice versa.

Among the several SLAM algorithms, the EKF (Extended Kalman Filter)-based SLAM [1,2] is the scheme most widely used. The EKF is the optimal sensor fusion method which has been used for a long time. The odometeric error caused by an encoder can be compensated by an EKF, which fuses different types of sensor data with weights proportional to the uncertainty of each sensor. In many cases the EKF-based SLAM requires artificially installed features, which causes difficulty in actual implementation. Moreover, the computational complexity involved in an EKF increases as the number of features increases.

Recently, FastSLAM, which combines a particle filter with an EKF, was proposed [3]. The particle filter is used to estimate the robot path and the EKF is employed to update the probability of the samples, which represent the candidates for the robot pose. It can very closely model the actual motion of a robot. FastSLAM uses as many low-dimensional EKFs as the number of samples and it adopts a kd-tree to reduce the computational complexity. However, it requires a large amount of unnecessary memory to represent the covariance and expectations of the robot pose and features. In addition, since multiple maps are converged after loop-closing, it is difficult to implement FastSLAM on a real-time basis.

The compressed filter algorithm was proposed to restrict the computational complexity and improve performance [4]. In the compressed filter, a global map is decomposed into several sub-maps. The robot uses the features in the sub-map that are located in the vicinity of the robot to reduce the computational complexity to $O(N_L^2)$, where N_L is the number of features in the sub-maps.

Recursive Unscented Kalman Filtering was proposed as an efficient method to deal with a large number of landmarks [5]. In this scheme, the robot pose and landmark positions are represented by their marginal Gaussian probability and updated individually, which reduces the filtering dimensionality and the computation requirements.

As the number of features increases, the computa-

Manuscript received December 11, 2006; accepted June 4, 2007. Recommended by Editorial Board member Jang Myung Lee under the direction of Editor Jae Weon Choi. This paper 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.

Yong-Ju Lee, Tae-Bum Kwon, and Jae-Bok Song are with the Department of Mechanical Engineering, Korea University, 5, Anam-dong, Sungbuk-gu, Seoul 136-713, Korea (e-mails: {yongju_lee, haptics, jbsong}@korea.ac.kr).
* Corresponding author.

tional burden of matching and updating features also increases, which prevents the SLAM from being conducted in real time. If the robot knows which features are nearby, it is unnecessary to update the information of all the features that are far away from the robot and thus are not used to update the robot pose. Therefore, the EKF can be carried out efficiently if it can identify the features near the robot. This paper proposes an algorithm that can find the local area in which matching and updating the features are conducted for the EKF-based SLAM. This local area can be searched for by the topological information which can represent the environment in the neighborhood of the robot. The topological information, which is composed of topological edges and nodes, is extracted through a thinning algorithm.

The remainder of this paper is organized as follows. Section 2 presents a brief introduction of the EKF-based SLAM. Section 3 discusses how to obtain the topological information using a thinning algorithm. Section 4 describes the way of finding the local features near the robot. Finally, Sections 5 and 6 present the experimental results and conclusions.

2. EKF-BASED SLAM

The EKF which is based on Bayes filtering [6], is used to handle the nonlinearities involved in the motion of a mobile robot. The state vector is defined as follows:

$$\mathbf{X} = [\mathbf{X}_r^T, \mathbf{X}_{L_1}^T, \cdots, \mathbf{X}_{L_N}^T]^T, \tag{1}$$

where $\mathbf{X}_r$ is the robot pose given by (${}^W x_r$, ${}^W y_r$, ${}^W\theta_r$) and $\mathbf{X}_{Li}$ is the position of the i-th feature denoted by (${}^W r_i$, ${}^W\alpha_i$), as shown in Fig. 1. The superscript W represents the quantity written in the world frame. A robot estimates its pose by continuous prediction and update based on the EKF algorithm.

2.1. Update

If the robot observes a feature, it compares this feature to those in the state vector $\mathbf{X}$. If the feature coincides with one of the stored features, then an update stage is conducted. At the update stage, the posteriors of the states are estimated by incorporating the predicted and actual observations.

The predicted observations $\hat{\mathbf{Z}}_t$ is defined by

$$\hat{\mathbf{Z}}_t = [\hat{\mathbf{z}}_{t,i} \mid 1 \le i \le n_p], \tag{2}$$

where $\hat{\mathbf{z}}_{t,i}$ is computed based on the prior state vector at time t, $\hat{\mathbf{X}}_t^-$, and n_p is the number of predicted observations (e.g., line feature i in Fig. 2). For example, the predicted observation of line feature i is computed as follows:

$$\hat{\mathbf{z}}_{t,i} = \mathbf{h}_i(\hat{\mathbf{x}}_t^-) = \begin{bmatrix} {}^R\hat{\alpha}_i \\ {}^R\hat{r}_i \end{bmatrix} = \begin{bmatrix} {}^W\alpha_i - {}^W\hat{\theta}_t^- \\ {}^W r_i - ({}^W\hat{x}_t^- \cdot \cos {}^W\alpha_i + {}^W\hat{y}_t^- \cdot \sin {}^W\alpha_i) \end{bmatrix}. \tag{3}$$

The actual observation $\mathbf{Z}_t$ obtained by the range sensor is defined as

$$\mathbf{Z}_t = [\mathbf{z}_{t,j} \mid 1 \le i \le n_a], \tag{4}$$

where n_a is the number of actual observations (e.g., line feature j in Fig. 2) and the observation for line feature j is given by

$$\mathbf{z}_{t,j} = [{}^R\alpha_j, {}^R r_j]^T, \tag{5}$$

where ${}^R\alpha_j$ is the angle of the line normal to feature line j relative to the robot frame, and ${}^R r_j$ is the distance to the feature line from the robot center.

By using the difference between the predicted and the actual observations, the state $\hat{\mathbf{X}}_t$ and its covariance $\mathbf{P}_t$ are updated as follows:

$$\mathbf{K}_t = \mathbf{P}_t^- \mathbf{H}_t^T (\mathbf{H}_t \mathbf{P}_t^- \mathbf{H}_t^T + \mathbf{R}_t)^{-1}, \tag{6}$$

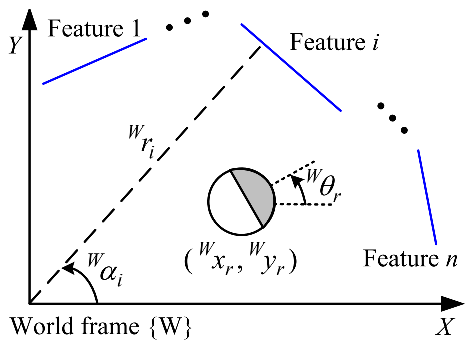


Fig. 1. Robot pose and feature position relative to the world frame.

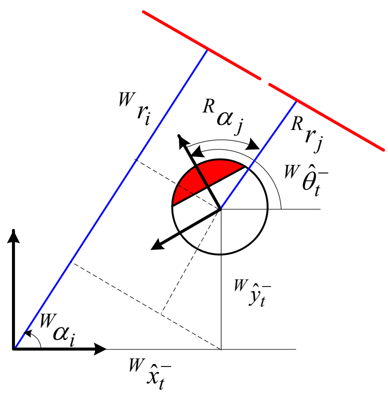


Fig. 2. Representation of features in the world and robot frames.

$$\hat{\mathbf{X}}_t = \hat{\mathbf{X}}_t^- + \mathbf{K}_t(\mathbf{Z}_t - \hat{\mathbf{Z}}_t), \quad (7)$$

$$\mathbf{P}_t = (\mathbf{I} - \mathbf{K}_t\mathbf{H}_t)\mathbf{P}_t^-, \quad (8)$$

$$\mathbf{H}_t = \frac{\partial \mathbf{h}(\hat{\mathbf{X}}_t^-)}{\partial \mathbf{X}_t}, \quad (9)$$

where **K** represents the Kalman gain, and **H** is the Jacobian matrix of the sensor model with respect to the state vector.

2.2. Prediction

At the prediction stage, the prior of the state vector, $\hat{\mathbf{X}}_{t+1}^-$, and the prior of its covariance matrix, $\mathbf{P}_{t+1}^-$, at time t+1 are computed from the state vector $\hat{\mathbf{X}}_t$, its covariance matrix $\mathbf{P}_t$ and the encoder reading $\mathbf{u}_t$ as follows:

$$\hat{\mathbf{X}}_{t+1}^- = \mathbf{f}(\hat{\mathbf{X}}_t, \mathbf{u}_t, t) + \mathbf{w}_t, \quad (10)$$

$$\mathbf{P}_{t+1}^- = \mathbf{F}_\mathbf{x}\mathbf{P}_t\mathbf{F}_\mathbf{x}^T + \mathbf{F}_\mathbf{u}\mathbf{Q}_t\mathbf{F}_\mathbf{u}^T, \quad (11)$$

$$\mathbf{F}_\mathbf{x} = \frac{\partial \mathbf{f}}{\partial \hat{\mathbf{X}}_t}, \quad \mathbf{F}_\mathbf{u} = \frac{\partial \mathbf{f}}{\partial \mathbf{u}_t}, \quad (12)$$

where $\mathbf{w}_t$ represents the process noise with zero mean and the covariance $\mathbf{Q}_t$. The matrices $\mathbf{F}_\mathbf{x}$ and $\mathbf{F}_\mathbf{u}$ are the Jacobian matrices of the nonlinear motion model $\mathbf{f}(\cdot)$ with respect to the state and the input, respectively. Note that the superscript '-' in (10) and (11) indicates the *prior* variables before the observation at time t+1.

3. THINNING-BASED TOPOLOGICAL INFORMATION

Topological information is an abstraction of the environment in terms of the nodes representing discrete places and the edges connecting them together. Topological information can be generated by various methods including a thinning method which is used in this research [7]. The thinning method is one of the popular image processing algorithms used to detect the skeletons of images. Fig. 3 illustrates the concept of thinning. The objects on the left can be adequately described by the structure composed of connected lines (i.e., the 'T' shape drawn with thin lines on the right). Note that the connectivity of the

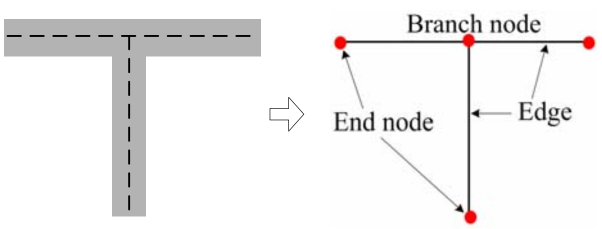


Fig. 3. Extraction of topological edges and nodes by thinning algorithm.

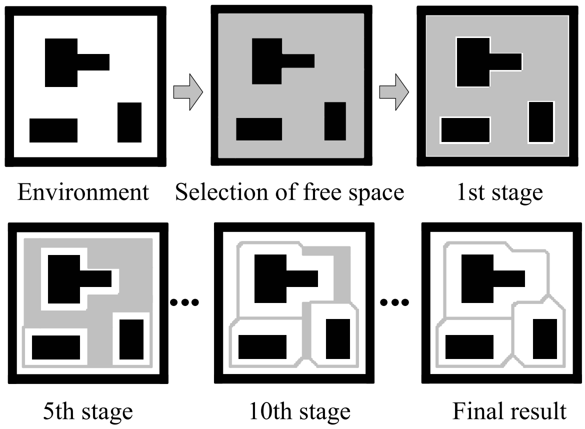


Fig. 4. Construction of a thinning-based topological map (TTM) for a given environment (Simulations).

structure is still preserved even with thin lines. In the case of mobile robots, the connected lines can be used as paths by which a robot can navigate without colliding into other objects.

The thinning process is applied to the free space shown in Fig. 4. Then, this 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. 3. An end node corresponding to the end of each edge represents the dead end of the environment (e.g., dead end of the corridor). A branch node where more than three edges meet represents the 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. Since the scanning rate is about 5Hz, 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 local occupancy grids enhances the confidence of the underlying grid map for a local topological map. At each sampling instant, based on

Occupancy grid map	Edge extraction by thinning process	Topological map with nodes and edges
(a)	(b)	(c)

Fig. 5. Example of thinning-based topological map.

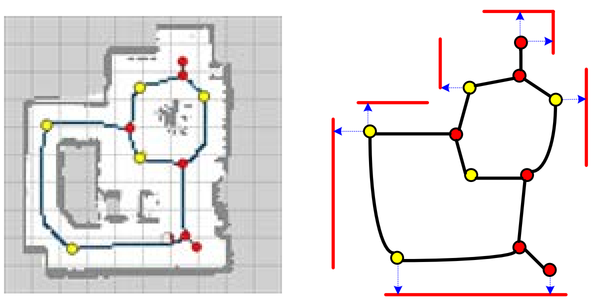
the range data, the local grid map and the subsequent local topological map is built, as shown in Fig. 5. The local topological map even for the same space is constantly changing as the underlying grid map is updated. The TTM is built in real time.

4. SLAM USING TOPOLOGICAL INFORMATION

In order to compensate for the robot pose, the robot should know the features that are being currently observed by its sensor. The process of relating the observed features to the features stored in the map is called *data association* [8]. Correct data association is crucial to the navigation of a mobile robot as it allows the robot to determine its pose with respect to the feature map. This paper proposes a SLAM with topological information (SLAM/TI) that can perform efficient data association using topological information.

Topological information extracted by a thinning algorithm covers the environment reasonably well. Information on the features that are near a certain node is stored in that node. If the topological information is used, the efficient data association can be achieved because all features do not have to be compared with the observation for matching. Fig. 6(a) shows a hybrid grid/topological map of an indoor environment. Fig. 6(b) represents the topological nodes and their corresponding features. The arrows indicate the features related to each node. In this paper, the least-square method is employed to extract line features from the raw sensor data [9].

First, the robot finds the nearest node. If more than two neighboring nodes are detected, the node that can be reached faster with the current orientation of the robot is selected. Then, the robot finds all nodes along the edges connected to this selected node. At this time the nodes behind the robot are not considered because they are beyond the limit of visibility. In this way the robot detects all nodes that are placed near the robot.



(a) (b)

Fig. 6. Topological representation of the environment; (a) hybrid grid/topological map, and (b) relationships between topological nodes and features.

The robot registers the features stored at the detected nodes on a temporary database. Since only the neighboring features are considered, the number of features stored in this database is not large. Hence this proposed approach can reduce the operation time of data association based on feature matching.

Fig. 7 illustrates an example of the process. At the initial position, the robot detects node 6 as the nearest node. Then the robot detects the nodes 2, 3, and 5, which are all connected to node 6 through the edges, and adds these nodes to the search list. The robot uses the features contained at these four nodes (i.e., 6, 2, 3, and 5) for data association.

Fig. 8 depicts the local area generated by the node search. The features in the dotted region, instead of all the features, are used to compensate for the robot pose. After the features for comparison are determined, the observations are associated with the previously known features by using a Chi-squared test given by

$$\chi^2 \geq \mathbf{v}_{ij}^T \mathbf{K}_i^{-1} \mathbf{v}_{ij}, \qquad (13)$$

where $\mathbf{v}_{ij} = \mathbf{z}_{t,j} - \hat{\mathbf{z}}_{t,i}$ is the innovation that is the difference between the actual observation of feature j, $\mathbf{z}_{t,j}$, and the predicted observation of feature i, $\hat{\mathbf{z}}_{t,i}$. The innovation covariance is $\mathbf{K}_t = \mathbf{H}_t \mathbf{P}_t^- \mathbf{H}_t^T + \mathbf{R}_t$. The value of χ^2 can be obtained from a Chi-square table. Because the observation has 2 degrees of freedom (i.e., r and α), χ^2 is 5.99 for a confidence level of 95%. If (13) is satisfied with $\chi^2 = 5.99$, the predicted feature is considered to coincide with the

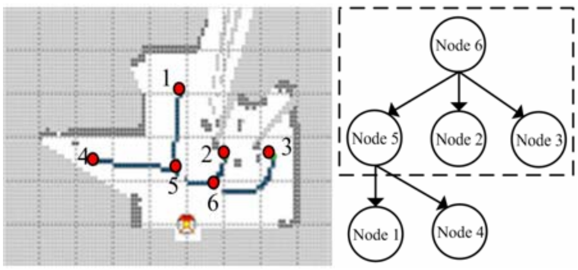


Fig. 7. Example of selection of nodes.

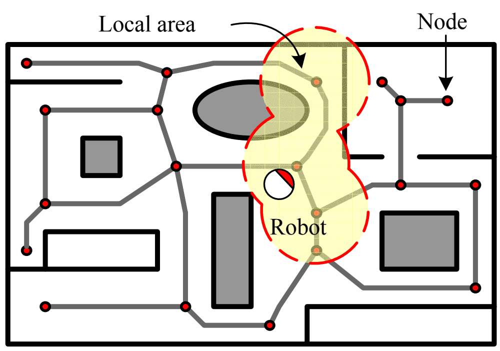


Fig. 8. Local area selected by topological nodes.

actual features. After finding the features in its vicinity, the robot estimates its pose based on the information on the detected features.

The features which are not associated with the observation are not likely to offer much information in estimation of the robot pose. In other words, the features chosen by the topological nodes are sufficient to maintain the accuracy for the update of the robot pose. The features that are not chosen by the topological nodes can be removed without sacrificing accuracy, which greatly decreases the computational load.

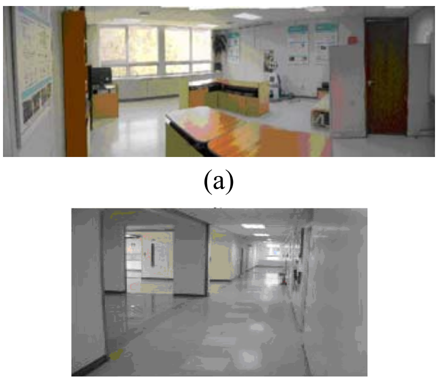
5. EXPERIMENTAL RESULTS

The Pioneer-3AT robot equipped with a SICK LMS200 laser rangefinder was used for the experiments. The robot moves at an average speed of 0.2m/s. Experiments were conducted in the environments of a living room and an office. The area of the environment was 9.5m x 7.5m with several pieces of furniture, as shown in Fig. 9(a). Also, experiments were also conducted for the floor of a building which was 40m x 10m, as shown in Fig. 9(b). All environments were modeled by a grid map having a cell size of 10cm x 10cm.

5.1. Maps constructed by SLAM/TI

A grid map constructed by the proposed SLAM/TI for an office and a building floor are shown in Figs. 10(b) and 11(b), respectively. These grid maps are used to certify that the proposed SLAM/TI scheme does not fail in indoor environments. These figures show that the environment can be modeled using the SLAM/TI algorithm without distorting the environments.

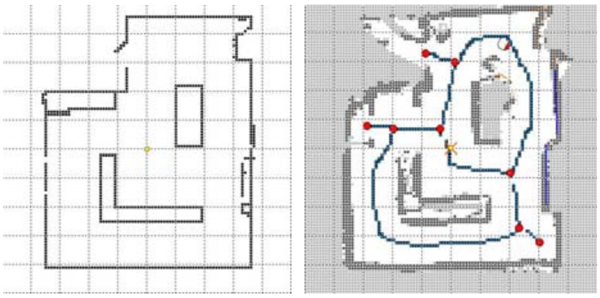
Fig. 12(a) shows the path of the robot and the line features for the environment of a building floor. In Fig. 12(a), the solid line denotes the robot path obtained only by the odometeric data and the dashed line represents the estimated robot path based on the



(a)

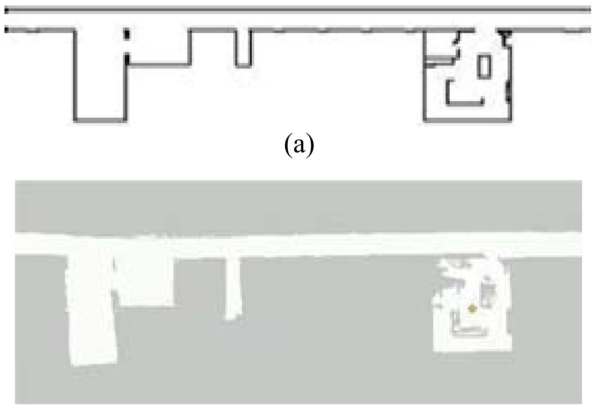
(b)

Fig. 9. Experimental environments; (a) an office, and (b) a building floor.



(a) (b)

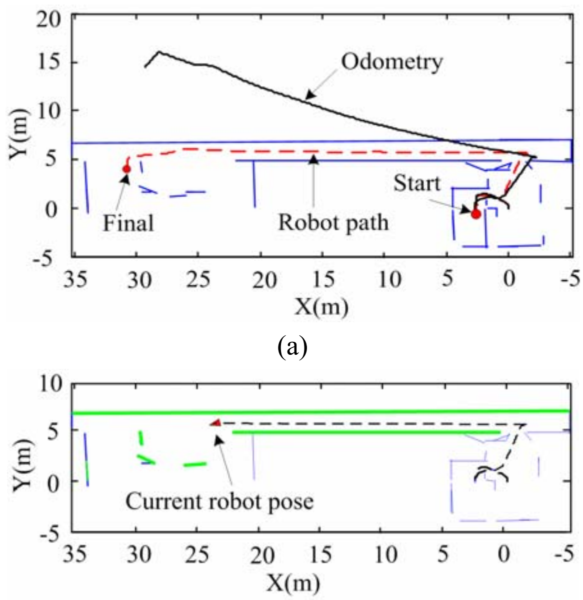
Fig. 10. Environment of office; (a) CAD data, and (b) local grid map constructed by EKF-SLAM/TI.



(a)

(b)

Fig. 11. Environment of building floor; (a) CAD data, and (b) global grid map constructed by EKF-SLAM/TI.



(a)

(b)

Fig. 12. Estimates of robot path; (a) conventional EKF-SLAM, and (b) the proposed EKF-SLAM/TI.

conventional EKF-SLAM algorithm. During the navigation from the start to the final position, the robot extracted 33 line features in total. On the other hand, Fig. 12(b) is the path estimated by the EKF-SLAM/TI. In order to update the current robot pose, the EKF-SLAM/TI requires only 7 line features denoted as thick lines in Fig. 12(b), whereas the EKF-SLAM should use more than 25 line features (assuming the same robot pose with Fig. 12(b)), most of which has little effect on the update of the current pose. The robot paths for the two methods are almost identical. Furthermore, as shown in Figs. 10 and 11, the maps constructed using the EKF-SLAM/TI are similar to the accurate map drawn using the CAD data.

5.2. Processing time

Fig. 13 compares the processing time of the conventional EKF-SLAM with that of the proposed EKF-SLAM/TI. As shown in the figure, the processing time of the EKF-SLAM increases almost linearly with the number of features. Moreover, its computation load usually increases rapidly as the number of features exceeds about 100. Therefore, the EKF-SLAM involving a large number of features cannot be successfully operated in real-time. However, the EKF-SLAM/TI shows nearly a constant processing time (in this case, 30-40msec) regardless of the number of features because this scheme uses only the features that are located near the current robot position. This means that the EKF-SLAM/TI can conduct the mapping and localization of a mobile robot in real-time even for environments with a lot of features.

Fig. 14 shows the process of building a map of a large environment by applying the EKF-SLAM/TI. The environment shown in Fig. 14 is a 70m x 10 m building floor consisting of many features. The robot extracted 88 features as it moved at a speed of 0.2m/sec for 20 minutes. As shown in Fig. 13, the EKF-SLAM/TI can simultaneously build a map and localize itself relative to the map in real time, which cannot be achieved with the conventional EKF-SLAM. In Fig. 14, the left column represents the global grid maps and the right column the local grid maps of 10m x 10m. The capital letters of Fig. 14 indicate the corresponding robot pose. In the experiments, the errors of the estimated robot pose were about ±20cm and ±5°.

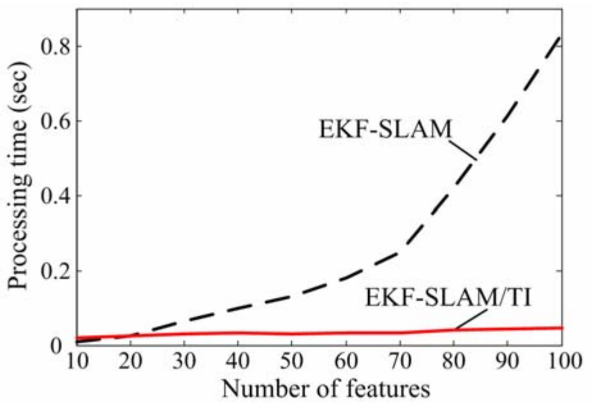


Fig. 13. Processing time of EKF-SLAM (solid line) and EKF-SLAM/TI (dash line).

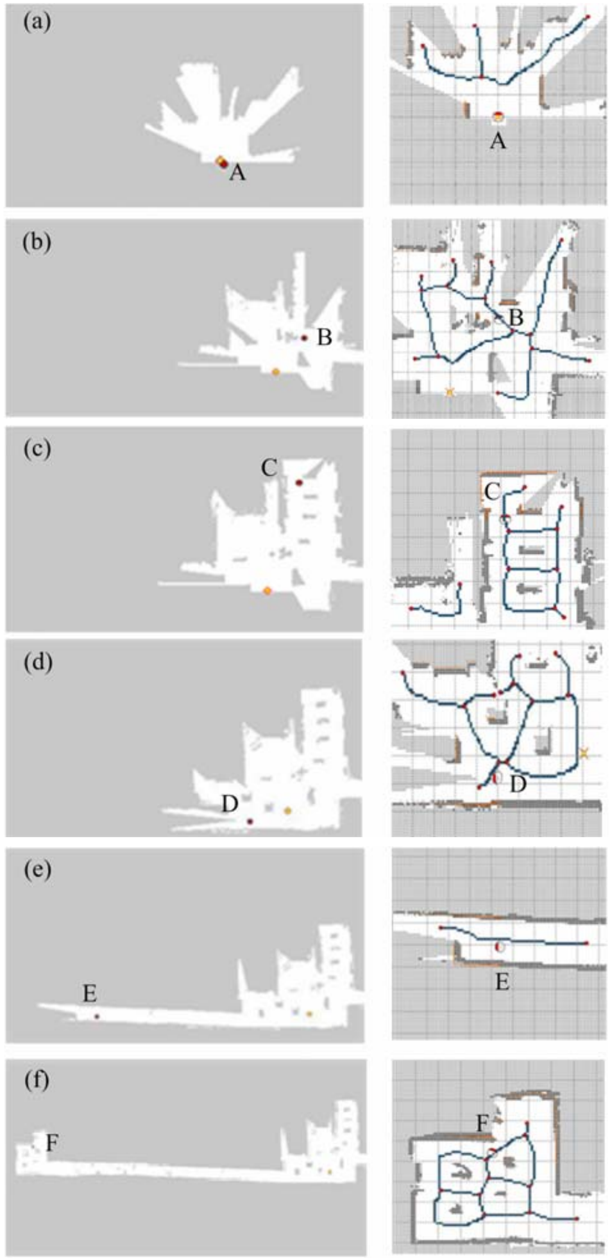


Fig. 14. Process of building a map using the EKF-SLAM/TI.

6. CONCLUSIONS

This paper proposes an EKF-SLAM/TI scheme that exploits the topological information based on a thinning process to improve the efficiency of conventional EKF-SLAM. The SLAM/TI was verified by various experiments. From this research, the following conclusions have been drawn.

1. The EKF-SLAM/TI can detect only the features needed to update the robot, which leads to a reduction in the computational burden. Therefore, mapping of a large environment can be conducted in real-time.
2. The accuracy of the EKF-SLAM/TI is comparable to that of the conventional EKF-SLAM that attempts to match all the features at the cost of an increase in computational time.

REFERENCES

[1] J. J. Leonard and H. F. Durrant-Whyte, *Directed Sonar Sensing for Mobile Robot Navigation*, *Kluwer Academic Publishers*, Boston, MA, 1992.

[2] S. B. Williams, H. F. Durrant-Whyte, and T. Baily, "Map management for efficient simultaneous localization and mapping (SLAM)," *Autonomous Robots*, vol. 12, no. 3, pp. 267-286, May 2002.

[3] M. Montemerlo and S. Thrun, "Simultaneous localization and mapping with unknown data association using FastSLAM," *Proc. of IEEE International Conference on Robotics and Automation*, pp. 1985-1991, September 2003.

[4] 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, June 2001.

[5] S. S. Lee, S. H. Lee, and D. S. Kim, "Recursive unscented Kalman filtering based SLAM using a large number of noisy observations," *International Journal of Control, Automation, and Systems*, vol. 4, no. 6, pp. 736-747, December 2006.

[6] S. Thrun, W. Burgard, and D. Fox, *Probability Robotics*, MIT press, Cambridge, MA, 2005.

[7] B. Y. Ko and J. B. Song, "Real-time building of thinning-based topological map with metric features," *Proc. of Int. Conf. on Intelligent Robots and Systems*, pp. 1524-1529, 2004.

[8] T. Baily, E. M. Nebot, J. K. Rosenblatt, and H. F. Durrant-Whyte, "Data association for mobile robot navigation: A graph theoretic approach," *Proc. of Int. Conf. on Robotics and Automation*, pp. 2512-2517, 2000.

[9] L. Zhang, "Line segment based map building and localization using 2D laser rangefinder," *Proc. of Int. Conf. on Robotics and Automation*, pp. 2538-2543, 2000.

Yong-Ju Lee received the B.S. degree in Mechanical Engineering from Korea University in 2004. He is now a Student for Ph.D. of Mechanical Engineering from Korea University. His research interests include mobile robotics.

Tae-Bum Kwon received the B.S. degree in Mechanical Engineering from Korea University in 2003. He is now a Student for Ph.D. of Mechanical Engineering from Korea University. His research interests include mobile robotics.

Jae-Bok Song received the B.S. and M.S. degrees in Mechanical Engineering from Seoul National University in 1983 and 1985, respectively. Also, he received the Ph.D. degree in Mechanical Engineering from MIT in 1992. He is currently a Professor of Mechanical Engineering, Korea University, where he is also the Director of the Intelligent Robotics Laboratory from 1993. His current research interests lie mainly in mobile robot, design and control of intelligent robotic systems, and haptics.