Topological SLAM Based on Voronoi Diagram and Extended Kalman Filter

Chang-Hyuk Choi *, Jae-Bok Song **, Munsang Kim *** and Woojin Chung ***

*Intelligent Robotics Lab., Dept. of Mechanical Eng., Korea Univ., Anam-dong, Sungbuk-ku, Seoul, 136-701, Korea
(Tel : 82-2-929-8501; Fax : 82-2-3290-3757 ; E-mail: vision2@korea.ac.kr)
**Dept. of Mechanical Eng., Korea Univ., Anam-dong, Sungbuk-ku, Seoul, 136-701, Korea
(Tel : 82-2-3290-3363; Fax : 82-2-3290-3757 ; E-mail: jbsong@korea.ac.kr)
***Advanced Robotics Research Center, KIST, Hawolgok-dong 39-1, Sungbuk-gu, Seoul, 136-791, Korea
(Tel : 82-2-958-5623; Fax : 82-2-958-5629 ; E-mail: {munsang, wjchung}@kist.re.kr)

Abstract: Through the simultaneous localization and map building (SLAM) technique, a robot can create maps about its unknown environment while it continuously localizes its position. Grid maps and feature maps have been widely used for SLAM together with application of probability methods and POMDP (partially observed Markov decision process). But this approach based on grid maps suffers from enormous computational burden. Topological maps, however, have drawn more attention these days because they are compact, provide natural interfaces, and are easily applicable to path planning in comparison with grid maps. Some topological SLAM techniques like GVG (generalized Voronoi diagram) were introduced, but it enables the robot to decide only whether the current position is part of GVG branch or not in the GVG algorithm. In this paper, therefore, to overcome these problems, we present a method for updating a global topological map from the local topological maps. These local topological maps are created through a labeled Voronoi diagram algorithm from the local grid map built based on the sensor information at the current robot position. And the nodes of a local topological map can be utilized as the features of the environment because it is robust in light of visibility problem. The geometric information of the feature is applied to the extended Kalman filter and the SLAM in the indoor environment is accomplished. A series of simulations have been conducted using a two-wheeled mobile robot equipped with a laser scanner. It is shown that the proposed scheme can be applied relatively well.

Keywords: Topological maps, SLAM (simultaneous localization and map building), Voronoi diagrams, CA architecture, Extended Kalman filter (EKF)

1. INTRODUCTION

Most tasks of a mobile robot are performed in the unknown environment. Even if the model of the environment is available, a robot usually faces unpredicted situations such as obstacles in its path. Therefore, it is important to plan the appropriate motion for a given situation [1]. The underlying task for motion planning is to model the robot's environment correctly and compactly, which is called map building. The environment model is used not only in robot motion planning but also in estimating the robot's current configuration (i.e., localization). Thus the model should be continuously updated to reflect the change in the environment. The above tasks are generally termed navigation of a mobile robot.

It is assumed in most map building tasks that the position and orientation of a robot are accurately known. However, localization by dead reckoning usually has several sources of errors such as slippage and unevenness of the ground, etc. Furthermore, this positioning error is accumulated over time. Therefore, it is necessary to perform both map building and localization simultaneously when a map is built in the unknown environment. This task is called SLAM (Simultaneous Localization And Mapping) or CML (Concurrent Mapping and Localization).

SLAM is divided into 3 paradigms according to its underlying map building methods (e.g., grid maps, topological maps, and hybrid maps). Since grid maps can model the environment comparatively simply and exactly, various methods like probabilistic methods are applicable. But a large amount of memory and long computational time are required. Thrun [2] proposed a SLAM method based on grid maps and the probabilistic maximum likelihood estimation method. Simmons and Koenig [3] implemented SLAM in the structured environments using POMDP (partially observed Markov decision process). They classified the robot's states into 6 and determined the current state using connectivity between these states. It is, however, difficult to define all states of the robot in the unstructured environments.

A topological map represents the environment with a graph (or roadmap) which consists of nodes (e.g., some regions or feature points) and arcs (i.e., connectivity of nodes). It is the basis of relative navigation [4], in which all robot motions are expressed with respect to objects (or distinct places, landmarks). The topological approach provides more efficient and compact maps requiring much less memory than other approaches. Choset implemented SLAM using GVG (generalized voronoi diagram). In this scheme, however, the robot can decide only whether the current position is a part of GVG branch or not. In some sense, the GVG is not applicable to SLAM because the robot needs to know its position at every instant in SLAM applications. On the other hand, Durrant-Whyte and Nebot [6] implemented SLAM using artificial landmarks and a laser scanner to sense them.

A hybrid map proposed by Tomatis [7] combines a metric map and a topological map. The main idea is to link the local metric map to the global topological map. The local metric map, feature map, is built by extracting the line segments from the sensor data of a laser scanner. A hybrid map can take advantages of compact environment modeling of the topological map and accuracy and robustness of the grid map. However, it is difficult to determine which map paradigm is suitable to a given environment.

In this paper, a local Voronoi diagram is constructed by evolution of the CA architecture, one of the distance transform methods. Contrary to the conventional CA algorithm using a global grid map to build a global topological map, an incremental construction method is proposed. In this method, given sensor measurement, the local topological map is built and updated to the global one. And the so-called topological SLAM is conducted by applying the extended Kalman filter to the topological map.

The paper is organized as follows. Section 2 presents topological map building using CA expansion including visibility analysis to verify the feasibility of the algorithm. Section 3 deals with the method to apply the extended Kalman filter to the topological SLAM. Simulation results are discussed in Section 4.

2. CONSTRUCTION OF VORONOI DIAGRAM

One method for constructing a topological map (or roadmap) for the environment is skeletonization (or retraction) of the configuration space. The points on the resulting skeleton have at least two closest boundary points of the nearby objects. Skeletonization is done by the distance transform method or the Voronoi diagram approach, and the latter is commonly used.

2.1 CA architecture and Voronoi diagram

Construction of a Voronoi diagram in this research is based on CA (cellular automata) architecture [10]. It is composed of grids of local linkage and evolves with the lapse of time. Each grid has a specific value and is affected by the value of the source which is the center of evolution. Therefore, the neighboring grids have the same value of the source value in the next time step. During the evolution, the Voronoi diagram is constructed in which the 2 different values meet because it corresponds to the equidistant line from the 2 different source grids. The local linkage between the grids is limited to 4-nn (4-nearest neighbors) as shown in Fig. 1, *cell*(i-1, j), *cell*(i+1, j), *cell*(i, j-1), and *cell*(i, j+1) for the central cell *cell*(i, j).

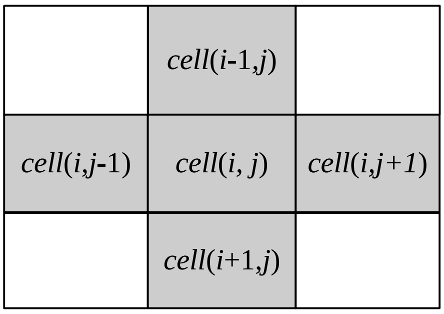
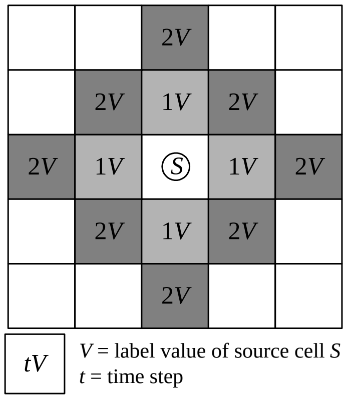
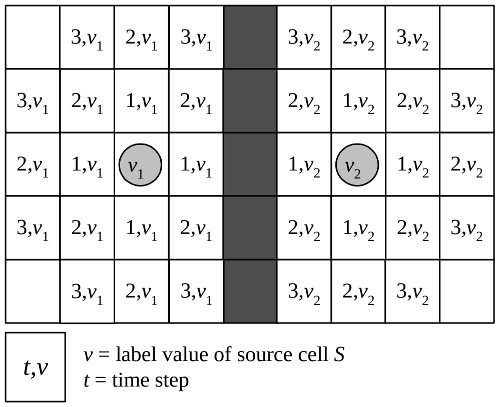


Fig. 1 Description of 4-nn.

Consider the evolution of the CA architecture. In Fig. 2(a), the evolution is conducted from a single source. Suppose that the source grid is assigned a value v at time t. At the next time t+1, the values of the 4-nn cells take on the same value of the *cell*(i, j) at time t, thus leading to a diamond-shaped evolution with time. In Fig. 2(b), two source grids are assigned the different values v_1 and v_2, respectively. In this case each source of different value evolves simultaneously, and each evolution meets at the points equidistant to these source grids. The Voronoi diagram is constructed by connecting these meet points. Note that the meet points created by the grids with identical source values are not considered as part of the Voronoi diagram since they result from either evolution of the identical source or evolution of the grids belonging to the same boundary of the object. The Voronoi diagram construction based on the CA architecture needs no distance calculation or complex mathematical computation, thus leading to fast execution.



(a) CA expansion from a single source.



(b) CA expansion from double sources.

Fig. 2 CA expansions

2.2 Boundary detection and edge code assignment

To construct a Voronoi diagram using the CA architecture, a local grid map from the measured sensor data is built in the beginning. Conventional construction of a Voronoi diagram based on the CA architecture required a complete global grid map, so boundary extraction of each object was needed. In this paper, however, incremental construction of a Voronoi diagram is adopted, in which the grids obtained from the local sensing data directly correspond to the object boundary, so the boundary extraction algorithm is not needed. .

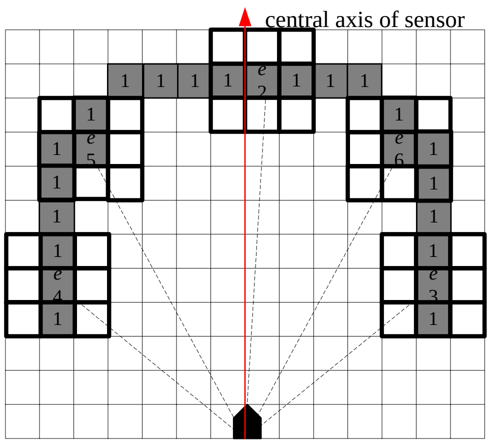


Fig. 3 Measured sensor data (gray) and assigned edge codes (e_n) in accordance with the condition of the edge.

Edge codes are assigned according to the conditions (e.g., direction) of each boundary edge. In the case of the object like Fig. 4, regions 1 and 2 belong to the identical object, but the direction of each region is perpendicular to each other. In this case the Voronoi diagram is generated at the equidistant points from both regions.

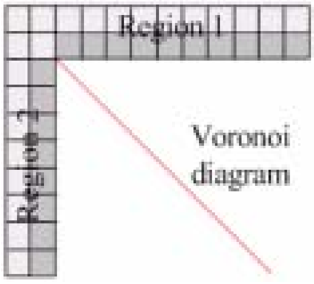


Fig. 4 Detected edges and the corresponding Voronoi diagram.

In the actual algorithm, edge directions can be determined by investigating the occupancy state of the 4-nn cells of the source grid which is assigned the edge codes shown in Fig. 5. The 8-nn (all neighboring grids of the mask) cells should be considered to express all possible states of the edge, but investigation of only 4-nn cells can offer computationally efficient results with little effect on the overall performance [10]. In the original edge code assignment algorithm for CA evolution, each edge is classified into 12 with respect to its occupancy pattern. In this research, however, the laser rangefinder scans only in the front of the robot, and thus only 4-nn based 6 edge codes indicated in Fig. 5 are sufficient to represent all possible states.

In Fig. 5, edge code 1 is the salt-and-pepper noise, and the corresponding grid is not used as a source. Edge codes 2 to 6 represent the boundary points facing on various directions. Practically, range sensing can give only a 1-cell wide trace. In Fig. 5, however, each edge code has 2-cell wide thickness to easily classify edge directions. To distinguish between *e*3 and *e*4 in the local range measurement, the relative angle to the sensor axis is needed. That is, edge code *e*3 (or *e*4) is assigned to the grids which are in the CW (or CCW) direction with respect to the central axis of the laser scanner.

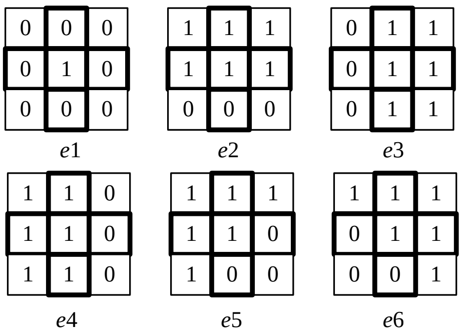
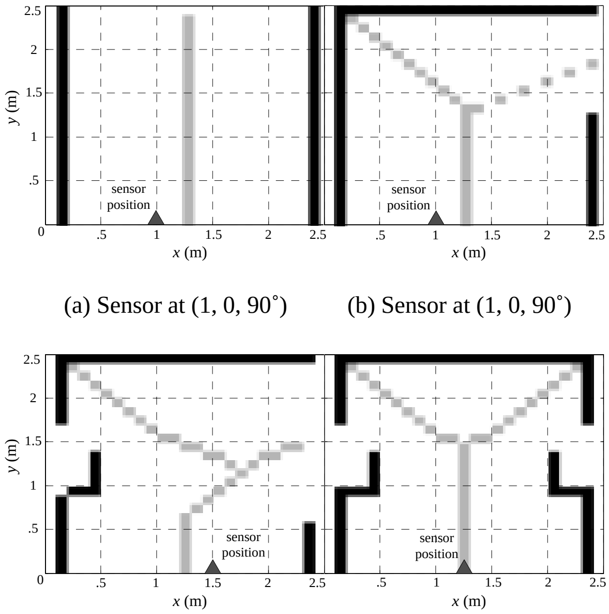


Fig. 5 Possible edge forms and edge code values.

2.3 Construction of Voronoi diagram

A Voronoi diagram is constructed by evolving the sources assigned to each boundary point with time. The values of the 4-nn cells of the current grid are compared with each other at each time step. If 2 or more values are not identical, the current grid is the branch of the Voronoi diagram. To construct the Voronoi diagram with this algorithm, 2 steps are required. The first step is to detect the edge of each object and to assign the source value to each edge. The second step is to perform CA evolution with this edge as an initial source, and construct the Voronoi diagram at the points where 2 or more evolving sources meet.

Various experiments on construction of the Voronoi diagram have been conducted to verify whether robust and continuous Voronoi diagrams can be built under a variety of environments. A Sick laser scanner provides the range data with angular resolution of 0.5°. Then a local grid map composed of 10cm×10cm grids is built and the local Voronoi diagram is constructed based on this local grid map. Fig. 6 shows Voronoi diagrams under various environments.



(a) Sensor at (1, 0, 90˚) (b) Sensor at (1, 0, 90˚)

(c) Sensor at (1.5, 0, 90˚) (d) Sensor at (1.3, 0, 90˚)

Fig. 6 Local Voronoi diagrams for various environments

In Fig. 6(a) the Voronoi diagram is formed in the middle of two parallel walls, while in Fig. 6(b) the Voronoi diagram is modified to reflect the open portion of the right wall. Fig. 6(c) represents the Voronoi diagram when there exists a column on the left wall. It is observed from Fig. 6(d) that the Voronoi diagram based on the CA architecture is robust in that its construction is not affected by small changes (noise).

2.4 Incremental construction of global Voronoi diagram

The environment for the test is about 6mx5m large. The laser range data are gathered automatically at each location. Fig. 7 is a laser-based global grid map of the environment under consideration and the path of a robot. Visibility (or occlusion) problems can occur. Note that this global grid map is shown only for better understanding of the environment, and is not used to construct the global topological map.

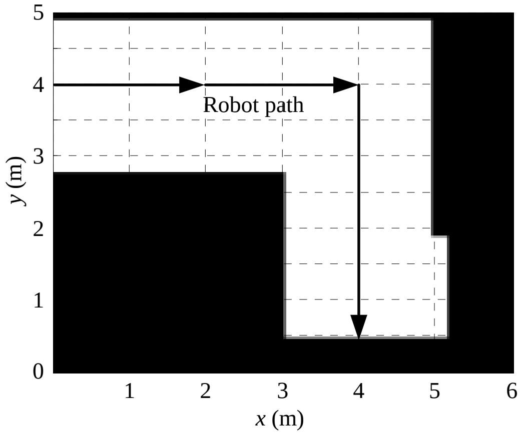
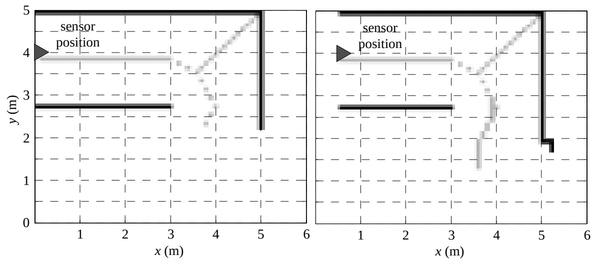


Fig. 7 Global grid map of environment under consideration.

Local Voronoi diagrams have been constructed at the 10 locations on the robot's path in the environment with a laser scanner. These diagrams are continuously incrementally updated to the global one. Fig. 8 represents the local Voronoi diagram at each location and shows that the medial line can be extracted in the Voronoi diagram constructed locally. And visibility problem does not occur and the local Voronoi diagrams can be directly updated to the global one. Contrary to the GVG method, more branches can be extracted and nodes can be seen everywhere, so navigation performance (localization and path planning) can be improved. Fig. 9 illustrates the final Global Voronoi diagram updated from all local Voronoi diagrams. Each grid belonging to the global Voronoi diagram is linked in the list structure and updated assuming that current sensor measurement is reliable.



(a) Sensor position at (0, 4, 0˚) & (0.5, 4, 0˚)

Fig. 8 Local Voronoi diagrams at each location.

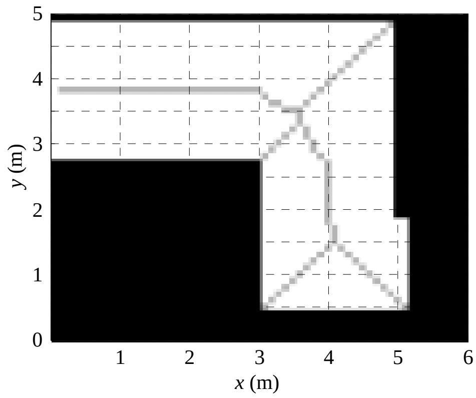


Fig. 9 Final global topological map.

3. MODELS FOR EXTENDED KALMAN FILTER

3.1 Process model

The state of the robot system in navigation consists of the robot's position and orientation and the location of nodes in the topological map. The model of the system is given by

$$\mathbf{x}(k) = [\mathbf{x}_v(k)\ \mathbf{p}(k)]^T \tag{1}$$

$$\mathbf{x}_v(k) = [x(k)\ y(k)\ \theta(k)]^T \tag{2}$$

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k)) + \mathbf{u}(k+1) + \mathbf{v}(k+1) \tag{3}$$

$$\mathbf{v}(k) \sim N(0, \mathbf{Q}(k)) \tag{4}$$

where $\mathbf{x}(k)$ is the robot state, $\mathbf{p}(k)$ is the node position information, $\mathbf{f}(k)$ the state transition matrix, $\mathbf{u}(k)$ the control input vector, and $\mathbf{v}(k)$ the noise vector with zero mean and covariance $\mathbf{Q}(k)$. $\mathbf{p}_i$ is the ith node and the location of the node is assumed to be invariant. It is defined by

$$\mathbf{p}_i(k+1) = \mathbf{p}_i(k) \tag{5}$$

The state of the robot (localization) and the location of the nodes (map building) can be gathered simultaneously from Eq. (3) and SLAM can be performed.

3.2 Observation model

The robot is equipped with a range sensor that can sense the relative location of the nodes (laser scanner) and encoders that can estimate the position and orientation of the robot. The observation model is given by

$$\mathbf{z}(k) = \mathbf{h}(\mathbf{x}(k)) + \mathbf{w}(k) \tag{6}$$

$$z_i(k) = \begin{bmatrix} r_i(k) \\ \theta_i(k) \end{bmatrix} = \begin{bmatrix} \sqrt{(x_i(k) - x(k))^2 + (y_i(k) - y(k))^2} \\ \tan^{-1}\left(\dfrac{y_i(k) - y(k)}{x_i(k) - x(k)}\right) + \theta(k) \end{bmatrix} + \begin{bmatrix} w_r(k) \\ w_\theta(k) \end{bmatrix} \tag{7}$$

$$\mathbf{w}(k) \sim N(0, \mathbf{R}(k)) \tag{8}$$

where $z_i(k)$ is the output of the sensor, h_i is the observation function, $r_i(k)$ is the relative distance to the ith node, $x_i(k)$ the x coordinate, $y_i(k)$ the y coordinate, and $\theta_i(k)$ the relative angle of the ith node from the robot coordinate. $\mathbf{w}(k)$ is a vector of observation noise with zero mean and covariance $\mathbf{R}(k)$.

3.3 Estimation process

The Kalman filter algorithm consists of 3 stages of prediction, observation and update. In the prediction stage, the estimate $\hat{\mathbf{x}}(k \mid k)$ of the state $\mathbf{x}(k)$ and the estimate of the covariance $\mathbf{P}(k|k)$ is computed using Eqs. (1) and (7) by following equations:

$$\hat{\mathbf{x}}(k+1 \mid k) = f(\hat{\mathbf{x}}(k) + \mathbf{u}(k)) \tag{9}$$

$$\hat{\mathbf{z}}_i(k+1 \mid k) = \mathbf{h}_i(\hat{\mathbf{x}}(k+1 \mid k)) \tag{10}$$

$$\mathbf{P}(k+1 \mid k) = \nabla f\ \mathbf{P}(k \mid k) \nabla f^T + \mathbf{Q}(k) \tag{11}$$

In the observation stage, an innovation $\mathbf{v}_i(k+1)$ which is the difference between the estimated observation and the actual observation and the covariance of this innovation $\mathbf{S}_i(k+1)$ are computed by

$$\mathbf{v}_i(k+1) = \mathbf{z}_i(k+1) - \hat{\mathbf{z}}_i(k+1\,|\,k) \tag{12}$$

$$\mathbf{S}_i(k+1\,|\,k) = \mathbf{h}_i\mathbf{P}(k+1\,|\,k)\,\mathbf{h}_i^T + \mathbf{R}_i(k+1) \tag{13}$$

Finally, in the update stage the state estimate and the corresponding state estimate covariance are then updated by the following equations:

$$\hat{\mathbf{x}}(k+1\,|\,k+1) = \hat{\mathbf{x}}(k+1\,|\,k) + \mathbf{W}_i(k+1)\mathbf{v}_i(k+1) \tag{14}$$

$$\mathbf{P}(k+1\,|\,k+1) = \mathbf{P}(k+1\,|\,k) - \mathbf{W}_i(k+1)\mathbf{S}(k+1)\mathbf{W}_i^T(k+1) \tag{15}$$

$$\mathbf{W}_i(k+1) = \mathbf{P}(k+1\,|\,k)\mathbf{h}_i^T(k)\mathbf{S}_i^{-1}(k+1) \tag{16}$$

4. SIMULATION RESULTS

To verify validity of the proposed algorithm, various simulation tests on the topological map building and localization (SLAM) have been conducted. In the conventional topological SLAM, artificial landmarks are installed in the environment and the SLAM is performed using these landmarks. In the proposed algorithm, however, the natural landmarks (the nodes of the topological map) obtained with CA expansion are used in the implementation of SLAM.

The environment under consideration is shown in Fig. 10. This grid map is shown only for the purpose of better understanding of the environment, and is not used for navigation. Fig. 11 shows the result of global topological mapping and the path where the robot navigates and performs SLAM. In the figure, the letters *S* and *F* indicate the start and final nodes, respectively. Note that there are many reachable paths from *S* to *F*.

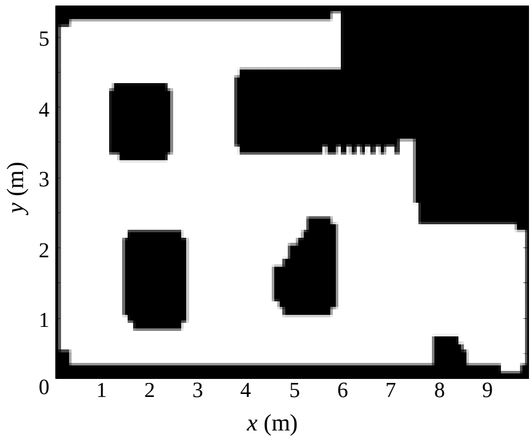


Fig. 10 Global grid map of the environment under consideration.

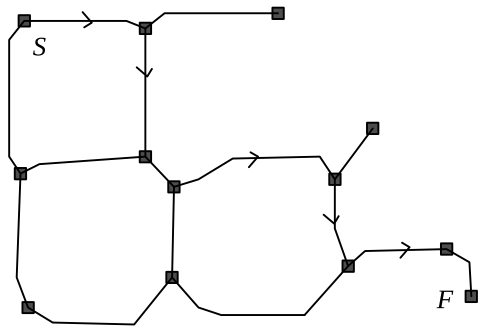


Fig. 11 The caption should be placed after the figure.

The global topological map contains geometrical information of the nodes and connectivity between nodes from the local one. In Fig. 12, the robot navigates from *S* to *F* and continuously updates the recognizable node information (the red dots means that the robot detected this node). And this node information is used as a reference for localization. In this figure, the red solid line is the path obtained with localization and the green dotted line is the actual path. The errors of the position and the orientation are shown in Fig. 13. Fig. 13(a) is the orientation error, Fig. 13(b) is the *y*-direction error, and Fig. 13(c) is the *x*-direction error. All these error converge into the constrained ranges.

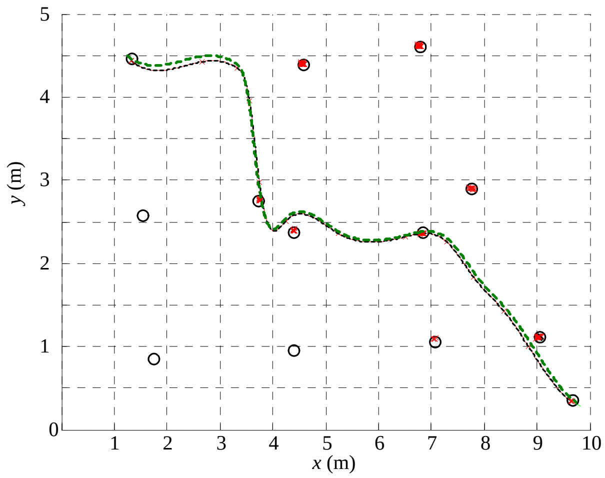
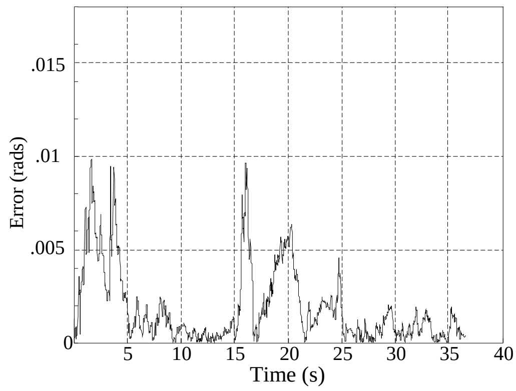
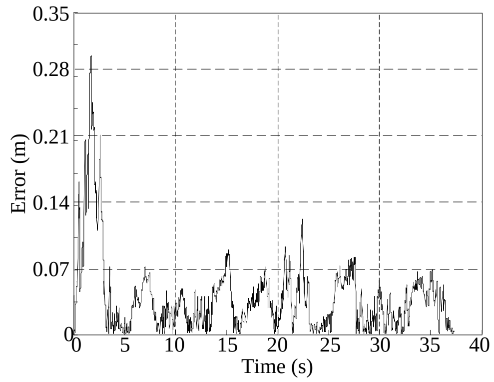


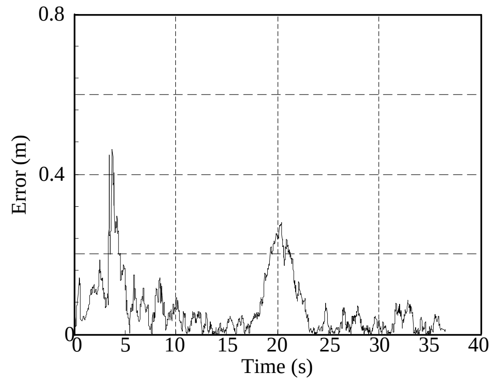
Fig. 12 Navigation for SLAM and updated nodes.



(a) Orientation error.



(b) Position error in *x*-direction



(c) Position error in *y*-direction

Fig. 13 Errors occurred during navigation.

5. CONCLUSIONS

In this research, fast and simple approach to global topological map building and SLAM method are proposed for a mobile robot equipped with a laser scanner. A robot gathers sensor data on the path and a local topological map is constructed using the CA expansion. The global topological map is then built from local topological maps. Using this map building method, topological SLAM can be performed. The following conclusions are drawn.

1. The global topological map using CA expansion does not cause the visibility problem.
2. Node information can be gathered everywhere the robot is.
3. Topological SLAM using natural landmarks is performed relatively well in the structured environments.

ACKNOWLEDGMENTS

This research was supported in part by the Service Robot Technology Development project, Ministry of Science and Technology in Korea.

REFERENCES

[1] H. Choset and J. B. Kondo, "Sensor-Based Exploration: The Hierarchical Generalized Voronoi Graph," *The Int. J. of Robotics Research*, 2000, Vol. 19, No. 2, pp. 97 – 125.

[2] S. Thrun, W. Burgard, and D. Fox, "A Probabilistic Approach to Concurrent Mapping and Localization for Mobile Robots." *Machine Learning and Autonomous Robots (joint issue)*, 1998, pp. 1-25.

[3] S. Thrun, "Learning Maps for Indoor Mobile Robot Navigation," *Artificial Intelligence*, 1999, Vol. 1, pp. 21 - 71.

[4] R. G. Simmons and S. Koenig, "Probabilistic Robot Navigation in Partially Observable Environments," *Proc. of the 14th Int. Joint Conf. on Artificial Intelligence*, 1995, pp. 1080-1087

[5] H. Choset, Member and K. Nagatani , "Topological Simultaneous Localization and Mapping (SLAM): Toward Exact Localization Without Explicit Localization," *IEEE Trans. on Robotics and Automation*, Vol. 17, No. 2, 2001, pp. 125-137

[6] M. W. M. Gamini Dissanayake and H. F. Durrant-Whyte, "A Solution to the Simultaneous Localization and Map Building (SLAM) Problem," *IEEE Trans. on Robotics and Automation*, Vol. 17, No. 3, 2001, pp. 229~241

[7] N. Tomatis, I. Nourbakhsh, and R. Siegwart, "SLAM: a global topological model with local metric maps," *IEEE Int. Conf. on Intelligent Robots and Systems*, 2001, pp. 421-426

[8] J. O'Rourke, "Computational Geometry in C," *Cambridge University Press*, 1994.

[9] D. V. Zwynsvoorde, T. Simeon and R. Alami, "Incremental Topological Modeling using Local Voronoi-like Graphs," *IEEE Int. Conf. on Intelligent Robots and Systems*, 2001, pp.897-902.

[10] P. G. Tzionas, A. Thanailakis and P. G. Tsalides, "Collision-Free Path Planning for a Diamond-Shaped Robot Using Two-Dimensional Cellular Automata," *IEEE Trans. on Robotics and Automation*, 1997, Vol. 13, No. 2, pp.237-250.