Full paper

Thinning-Based Topological Exploration Using Position Possibility of Topological Nodes

Tae-Bum Kwon and Jae-Bok Song *
Korea University, Anam-dong, Sungbuk-gu, Seoul, South Korea
Received 27 July 2007; accepted 4 September 2007

Abstract
A grid map can be efficiently used in navigation, but this type of map requires a large amount of memory in proportion to the size of the environment. As an alternative, a topological map can be used to represent the environment in terms of discrete nodes with edges connecting them. It is usually constructed by Voronoi-like graphs, but in this paper the topological map is built based on the local grid map by using a thinning algorithm. This new approach can easily extract the topological information in real-time and be robustly applicable to the real environment, and this map can be autonomously built by exploration. The position possibility is defined to evaluate the quantitative reliability of the topological map and then a new exploration scheme based on the position possibility is proposed. From the position possibility information, the robot can determine whether or not it needs to visit a specific end node, which node will be the next target and how much of the environment has yet been explored. Various experiments showed that the proposed map-building and exploration methods can accurately build a local topological map in real-time and can guide a robot safely even in a dynamic environment.

Keywords
Mobile robot, topological map building, topological exploration, thinning, position possibility

1. Introduction
Mapping is the task of modeling a robot's environment and is important in the navigation of a mobile robot. Two major paradigms have been used for mapping the indoor environment: an occupancy grid map and a topological map. The former divides the environment into evenly divided cells and calculates the occupancy possibility of each cell; it can construct an accurate metric map simply, but requires a large amount of memory and might be inefficient. The latter, on the other hand, models the environment with nodes and with edges representing the connectivity...
of the nodes. It is an efficient and compact Voronoi graph-like map, requiring much less memory, but it is difficult to apply to accurate localization because it lacks information [1].

Several ways of building a topological map have been proposed. One of the most common methods for building a topological map is the generalized Voronoi graph (GVG) [2]. The GVG is formed by a set of points equidistant to the obstacles in the n-dimensional space. Using this approach the robot can collect topological information with low-cost sonar sensors, but in this case the robot has to move on the GVG edges sufficiently, and some types of nodes and edges such as weak meet points and boundary edges tend to be extracted due to sensor noise and environmental complexity. Therefore, the reduced GVG (RGVG) was proposed to eliminate some of these unwanted nodes and edges [3]. Some researchers have also employed the Voronoi-like graph used by the GVG to extract topological information [4, 5].

In order to build a map, a robot should be able to autonomously explore its unknown environment. To accomplish this task, a robot requires an exploration strategy in which the robot is guided to cover the environment. The requirements for exploration are its completeness and efficiency. Completeness means that it should cover the entire environment without missing any portion of the environment. Efficiency means how fast the robot can finish the task of mapping by minimizing the travel distances to cover the entire environment.

Several exploration algorithms have been proposed so far. In a frontier-based exploration strategy, a robot detects the frontiers, i.e. the regions on the boundary between the unexplored and the open space. The robot then moves to the new frontiers to explore the unknown environment until the entire environment has been explored [6–9]. Although frontier-based exploration generally shows good performance, this scheme has some drawbacks of not using the information about the obstacles which can serve as a guide for a robot’s moves and for correcting its localization errors. To overcome this problem, an autonomous exploration method via the regions of interest was proposed [10]. In this research, the next best view was searched to get the next sensor data which could be used to improve the quality of the map under construction. However, little research has been done to improve the efficiency of exploration.

Another main strategy for exploration is a topological exploration based on the GVG representation. In the topological simultaneous localization and mapping (T-SLAM) developed for exploration of the environment, the robot traces all GVG edges, and visits all meet points and boundary points [11]. Thus, the extended Voronoi graph (EVG) was proposed [12]. In this research, both mid-line following and wall following were used to control the robot motion and to model the environment. In these cases, it is not easy to adjust the trade-off between efficiency and completeness.

The contribution presented in this paper is to propose a novel topological exploration scheme. We propose an approach to build a topological map in real-time by applying the thinning algorithm to the occupancy grids which are probabilistically
Figure 1. Experimental environments, local grid map and local topological map built by TTM and TTE.

Figure 1 shows an overall view of the experimental environment, the underlying local grid map and TTM. This map was autonomously built by the proposed mapping and exploration scheme. This paper only investigates the topological map-building and autonomous exploration method. For various applications, the map built by the proposed method can be used with other topological techniques, such as topological place detection [4, 12].

The remainder of this paper is organized as follows. Section 2 reviews the TTM. Section 3 discusses the concept of position possibility. Section 4 describes how the edge of the topological map can be used as a path for exploration in dynamic environments. Section 5 represents the proposed exploration scheme and experimental results. Finally, Section 6 presents conclusions.

2. TTM

In this paper, a novel method for topological map building is used. In this method, a local topological map is constructed by applying a thinning algorithm to a local occupancy grid map, which models the robot’s neighboring environment. A TTM built in this way requires a simpler process than other methods based on the construction of the Voronoi graph, but can provide similar performance. The thinning
The algorithm adopted for the TTM is a type of image processing algorithm which has been used to detect the skeletons of images [13].

Figure 2 shows an example of TTM construction for a given environment. The empty space, to which the thinning process is applied, is selected. Then, this empty space is contracted from both the outside of the objects and the inside of the wall boundary. This process is repeated until a skeleton corresponding to the thinnest line for the free space is extracted. Note that connectivity of the empty space is still preserved even with thin lines. More detailed information on the edge extraction using the thinning process is described in Ref. [14].

After the edges are extracted through the thinning process, three types of nodes can be extracted, as shown in Fig. 3a. 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, at which more than three edges meet, represents a junction (e.g., intersection of corridors). A corner node denotes the point at which the slope of an edge varies significantly. For example, without introducing the corner node N₃ in Fig. 3, edge A that can detour the object cannot be reproduced from only the information on the connectivities of nodes N₁ and N₂.

These nodes can be extracted by counting the number of cells comprising the edges. As shown in Fig. 3b, a cell with only one neighboring cell is identified as an end node and a cell with more than three neighboring edges as a branch node. On the other hand, the corner node is extracted as follows. The parameter $l_{\text{line}}$ is defined...
as the length of the virtual line between nodes A and B, and \( d_{\text{max}} \) as the distance from the virtual line to the most distant cell. The corner node is then generated at the cell that has the maximum distance from the line provided \( d_{\text{max}} > \alpha_{\text{line}} \) (i.e., \( \alpha = 0.3 \) in this research).

The position of a corner node is somewhat unstable during exploration, as shown in Fig. 3c. As a robot explores the environment, the nodes may change their positions. However, a corner node is extracted mainly for planning a path that detours an object, whereas the branch and end nodes are exploited in other navigation techniques such as localization, exploration, etc. Therefore, a slight change in the position of a corner node is not important in the proposed method because even the approximate position of a corner node is sufficient to plan a path (\( N_2 \rightarrow N_3 \rightarrow N_1 \)) that guides the robot to the end node with obstacle avoidance, as shown in Fig. 3c.

The thinning process is performed on a specific region of a predefined size (e.g., 10 m \( \times \) 10 m in this research). The rectangular region having a robot at its center is defined as the thinning window, which indicates the portion of the grid map to which the thinning process is applied. The topological map built in this thinning window is called a local TTM. As the robot travels, the environment covered by the map continues to change and thus a new local TTM is constructed continuously. Figure 4 shows an example of grid map construction and the corresponding TTM during navigation in a real environment. The computation time to extract topological nodes and edges was about 30 ms with a P4 notebook computer. More detailed information on the building of a TTM and its extension is described in Ref. [14].

In a strict sense, a topological map like a subway map has no geometric information. However, the node of a TTM can have geometric information as well as...
topological information because it is extracted from the grid map. It is natural for mobile robot navigation and other studies about the topological map also exploit geometric information. For example, the GVG-based scheme not only keeps the adjacency relationships among nodes, but also the lengths of each edge and the departure angle of the edges, which can be considered as the geometric information.

Figure 5 shows some data for the comparison of a TTM with one of the well-known previous methods introduced in Ref. [1]. In Ref. [1], the Voronoi diagram is extracted from the occupancy grid map and the free space is partitioned by the critical lines. Then, the partitioned regions are mapped into a topological graph, which corresponds to a topological map. These processes are depicted in Fig. 5a, which is cited from Ref. [1]. Figure 5b shows the topological map of the same environment built by the proposed TTM scheme. Two topological maps are almost identical topologically. Both the end nodes (dead-ends) and branch nodes (intersections), which are important in a topological map, are exactly identical, whereas other types of nodes are different (as marked by the arrows). However, the proposed thinning-based scheme is much easier than the method in Ref. [1]. Figure 5c and 5d shows examples for another environment and the proposed scheme builds a topologically identical map as compared with the method in Ref. [1].

3. Position Possibility

3.1. Characteristics of End Nodes of a TTM

The nodes of a TTM have their own characteristics. For example, the position of the branch node, at which more than three edges meet, is relatively robust to changes in the environment and sensing methods. On the other hand, the position of the end node, which represents the end of an edge, is likely to change according to the degree to which the environment is mapped in the grid map. That is, the end node position continuously changes until the environment around the end node is sensed completely. Therefore, the observation of the end node position may indicate how much of the environment around the end node has been explored.

Figure 6 shows a topological map constructed in real-time as the robot navigates through an unknown environment. In Fig. 6, the circle with the red mark denotes
Figure 5. (a and c) Topological maps built by the method in Ref. [1]. (b and d) Topological map built by the TTM scheme.

Figure 6. Change in end node position during exploration.
the mobile robot and the red solid mark points to the robot’s heading. In Fig. 6a, the robot scans the unexplored frontal area using a laser scanner, which provides 181 range readings with a resolution of $1^\circ$. Based on these range data, the local occupancy grid map and then its local TTM are constructed, as shown in Fig. 6a. No information on the area to the left of node A is available with the current sensor location. However, as the robot traverses towards end node A and the unseen area around node A is exposed to the range sensor, node A continues to change its position through Fig. 6b and 6c until the robot has thoroughly explored. Note that the branch nodes maintain their positions unlike the end nodes.

3.2. Position Possibility of End Nodes

During topological mapping, the reliability of the topological features (e.g., nodes and edges) needs to be evaluated. To this end, we introduce the position possibility, which indicates how much of the environment around the end node under consideration has been explored. This possibility can be obtained by investigating the behavior of the end node in situations shown in Fig. 7.

In this context, a reliable node position means that the node under consideration will remain at the current location even after its neighboring environment has been completely mapped. For example, if the end node does not change its position even after the robot has explored its neighboring environment, then the position possibility of this end node staying at this current position is regarded as 1.0. Two types of position probabilities are defined in following sections.

3.2.1. Distance-Based Position Possibility

The first type of position possibility is associated with the distance between the node and the robot, as shown in Fig. 7a. The closer the robot approaches a node, the more completely the region around the node is mapped. It is defined based on the following two assumptions.

Assumption 1. If all parts of the environment which can affect the end node position are thoroughly sensed, then the end node does not change its position.

![Figure 7. A robot can scan the environment around the node at different distances (a) and from different angles (b).](image-url)
Assumption 2. When the robot scans a node at the maximum distance, $d_{\text{max}}$, at which it can continuously sense all cells occupied by the obstacle, the possibility of fully exploring the environment around that node is assigned as 0.5. If the robot scans a node closer (or further) than this distance, the reliability becomes higher (or lower).

Based on these two assumptions, the distance-based position possibility (DPP) for node $n$ is defined as:

$$ D_{n,t}(X_{n,t}, X_{r,t}) = \frac{d_{\text{max}}}{d_n + (d_{\text{max}} - d_{\text{min}})}, $$

where

$$ d_n = \begin{cases} d_{\text{min}}, & \text{if } (\min_i \|X_{n,i} - X_{r,i}\| < d_{\text{min}}) \\ \min_i \|X_{n,i} - X_{r,i}\|, & \text{else} \end{cases} $$

(i = 1, \ldots, t),

(2)

where $X_{n,t}$ and $X_{r,t}$ are the absolute positions of node $n$ and the robot at time $t$ with respect to the global reference frame, respectively, and $d_{\text{min}}$ is a minimum distance at which the robot can sense the location of node $n$ in consideration of the radius of a robot and so on. In (2), $d_n$ represents the minimum distance between node $n$ and the robot center measured during exploration up to time $t$. Any node out of the scanning range or blocked by obstacles is excluded in the computation of (2). By Assumption 1, the DPP of a node is 1 when $d_n$ is less than $d_{\text{min}}$, which is the approximate radius of the robot used in the experiment, because the environment is highly likely to be closely examined at this distance. It is assumed from Assumption 2 that the robot is not likely to reasonably sense the environment when the distance is greater than $d_{\text{max}}$, thus, the DPP is assigned as 0.5 when $d_n$ is $(d_{\text{min}} + d_{\text{max}})$ considering the offset between the center of the robot and the laser scanner. Between these two limits, the DPP is designed to decrease inversely proportional to the distance between the node and the robot. For computation of the DPP, the minimum distance from the node to the robot is stored and compared with the distance at that time whenever the robot sees that node.

In this experiment the laser scanner provides 181 range readings with a resolution of $1^\circ$ and the size of each cell in the grid map is 10 cm × 10 cm. In this case, the robot can continuously sense all successive cells when the distance to the cell is within a certain range. In region A of Fig. 8, the direction from the sensor to the cells is perpendicular to the direction of the arrangement of cells. In this case, the distance between two neighboring cells is 10 cm and all successive cells can be sensed (with no cell missed) when the distance from the sensor to the cells is shorter than $r_1 = 573$ cm. However, in region B, the direction from the sensor to the cells is at an angle of $45^\circ$ with respect to the direction of the arrangement of cells. In this case, the distance from the sensor to the cells should be shorter than $r_2 = 405$ cm to sense all successive cells with a resolution of $1^\circ$. Therefore, $d_{\text{max}}$ was chosen as
4.0 m and $d_{\text{min}} = 0.5$ m was derived from the approximate radius of a robot used in the experiments.

3.2.2. Angle-Based Position Possibility

The second type of position possibility is associated with the angle between sensing positions, as shown in Fig. 7b. Let us define the angle difference $\alpha$ as:

$$\alpha_n = \max_{i,j} |\angle(X_{n,i} - X_{r,i}) - \angle(X_{n,j} - X_{r,j})| \quad (i, j = 1, \ldots, t),$$

(3)

where $\angle(X_{n,i} - X_{r,i})$ is the angle of the vector from the robot center to the node measured counterclockwise relative to the $X$-axis of the global reference frame. Any node out of the scanning range (180°) or blocked by obstacles is excluded in the computation of (3). Note that the angle difference $\alpha_n$ is the maximum of the angle differences $|\alpha_1 - \alpha_2|$, $|\alpha_2 - \alpha_3|$ and $|\alpha_3 - \alpha_1|$ in the case of Fig. 7b, so $\alpha_n = |\alpha_3 - \alpha_1|$. To calculate this, each node stores all the angles from the $X$-axis of the global reference frame to the vector from the robot center to the node whenever the robot sees that node.

The angle-based position possibility (APP) of a node remaining at the current location is likely to increase when the robot sees the end node from various angles. It is defined based on the following assumption:

**Assumption 3.** If the robot sees an end node at two places with an angle difference of 90° between the lines connecting each place with the end node and the environment around the end node is sufficiently sensed, the node does not change its position.

In Fig. 9a, the robot scans the environment and one end node is generated at the corner. After some movement, the robot scans the environment at another place.
which is 90° apart. In this case, a change in the end node position does not occur because the environment can be sufficiently sensed at the first place. On the other hand, in Fig. 9b, the end node changes its position after scanning a new environment because the environment could not be fully sensed at the first scanning position. In this case, the end node continues to change its position until the whole environment around this end node is sensed. Figure 9c shows the worst case. At the first place, the robot cannot see the opening of the wall and the environment is gradually sensed as a robot sees around the corner at other places. However, the edge and node extracted by the thinning-based method cannot change their shape and position because the width of the opening is about the minimum value required for thinning, which is related to the diameter of a robot. In this case, region A cannot generate the thinning-based edge and node because it is considered that the opening is too narrow for a robot to go through it [14]. After exploring the environment beyond the opening, the width of region B is found to be greater than the minimum width, and the edge and node can be extracted. Therefore, the position of the node will not change unless it is seen at an angle difference of about 90° as shown in the
right figure of Fig. 9c. It means that a robot should observe a node at various places, which are 90° apart in order to be confident that the position of the node is stable even in the worst case. This justifies Assumption 3.

Based on this assumption, the APP for node \( n \) is defined by:

\[
A_{n,t}(X_{n,t}, X_{r,t}) = \begin{cases} 
\frac{\alpha_n}{90} & (\alpha_n \leq 90^\circ) \\
1 & (\alpha_n > 90^\circ).
\end{cases}
\] (4)

Note that the APP is so designed such that \( A_{n,t} = 1.0 \) when \( \alpha_n \geq 90^\circ \) and \( A_{n,t} = 0 \) when \( \alpha_n = 0^\circ \). Since the laser scanner has a scanning range of 180°, \( \alpha_n = 180^\circ \) means that the neighboring environment of the node of interest has been fully scanned. However, the end nodes are usually placed in a corner, as shown in Fig. 10, so the environment around the end node can be thoroughly examined even when \( \alpha_n \approx 90^\circ \).

### 3.2.3. Overall Position Possibility

The end nodes can be generated in two cases as shown in Fig. 10. In Fig. 10a, the end node is generated at a corner where two walls meet. In this case, the robot does not need to visit the end node because an examination of the environment around the node from various angles (even at a great distance) is sufficient to thoroughly model the environment. In this case, the APP is more preferable to examine the reliability of the end node. In Fig. 10b, the end node is generated at the dead end of a narrow hallway. It is difficult to scan the environment from various angles, so the robot needs to approach the end node sufficiently closely. In this case, DPP is more preferable to examine the reliability of the end node. In order to consider both cases, the overall position possibility, \( P_{n,t} \), is defined as:

\[
P_{n,t}(X_{n,t}, X_{r,t}) = \max(D_{n,t}, A_{n,t}),
\] (5)

where \( n \) is the node number.

Figure 11 illustrates how the position possibility is calculated in a real environment similar to Fig. 10a. In Fig. 11a, the DPP of node A is computed as 0.51 by (1), once the edges and nodes have been generated. In Fig. 11b the distance from the robot to node A is \( r = 2.5 \) m, so the DPP of end node A is computed as 0.66 and the overall position possibility becomes \( P = 0.66 \) according to (5). Then, the

Figure 10. Two cases in which end nodes are generated.
robot moves along the path and all angle differences between all pairs of the previous robot positions are checked. The maximum angle difference for end node A is obtained as $\alpha = 74^\circ$ between robot poses A and B, so the APP is computed as 0.82 by (4) and the overall position possibility becomes 0.82. Figure 12 shows how the position possibility is calculated in a real environment similar to Fig. 10b. In this environment composed of a narrow hallway, the position possibility increases as the robot approaches the end node. These experimental results show that (1)–(5) can adequately compute the position possibility of the end node for the two cases in Fig. 10.

3.3. Continuous Updates of Position Possibility

As shown in Figs 11 and 12, during mapping, topological information is continuously changed in real-time. In some cases, end nodes appear or disappear and the position of the end node changes slightly or significantly. In this case, the robot should determine whether a new end node is generated or an existing end node changes its position. If the amount of the position change is smaller than the threshold, a simple Euclidean distance, the node maintains its prior information.
Node A in Fig. 13 shows an example. In Fig. 13b, the position possibility of node A is 0.63. In Fig. 13c and 13d, node A of Fig. 13b is out of the thinning window, but node A is considered as a virtual node and information is conserved. Once the virtual node is again within the thinning window, the robot attempts to find the similar node and it is matched to the real end node, as shown in Figure 13e. Its position possibility is determined as 0.63 by all information obtained from Fig. 13b–e. After some movement, the environment around node A is fully explored and it disappears because the end node cannot be generated in such an environment.

4. Path Following and Obstacle Avoidance Using Thinning-Based Topological Edge

The edges of a TTM generated in real-time during exploration can serve as the safest paths of travel, since they are the medial lines between obstacles. The medial line does not mean the optimal path. However, a robot can sense the environment more widely from the medial line than the optimal path, which tends to be closer to the obstacles. Moreover, the proposed exploration scheme (TTE) does not require an additional path planning method, such as A* search, gradient method, etc.

Figure 14 shows the experimental result for TTM-based path following during exploration. After the edge to follow is determined, a robot attempts to track the target point set on this edge by motion control which minimizes the error between the target and current position [15]. The robot travels on the previous edge unless the difference between the new and previous edges is not large. That is why the target points are sometimes slightly placed off the edge, as shown in Fig. 14.
Figure 14. Path following using thinning edges. The dot in front of the robot is extracted from the thinning edge and used as reference position of motion control.

Figure 15. Obstacle avoidance using the thinning edge.

The edge of a TTM can be also used as the exploration path in the environment in which moving obstacles exist. If an obstacle is located on the topological edge, either a new edge free of moving obstacles is extracted or no edge is extracted in a narrow or small region. Figure 15 shows how the robot avoids a moving obstacle on the thinning-based edges extracted in real-time. In Fig. 15a, the edge extracted from the map of a narrow hallway does not include the obstacle. As the robot approaches the dynamic obstacle, a new edge is extracted to take the obstacle into account, as shown in Fig. 15b. As illustrated in Fig. 15c and 15d, a new target point is then set for the robot’s motion control.

5. Autonomous Exploration

In order to autonomously map the environment, the robot should solve three problems in addition to carrying out the basic navigation elements such as mapping and
localization. That is, a robot should be able to choose the target points leading to the unknown environment, generate an exploration path in the unknown area and determine how much of the environment has been explored to complete the exploration process. These three problems are specific only to the task of exploration. In this research, these problems were resolved by use of a TTM and the position probabilities of the end nodes of a TTM.

Without additional path planning, the robot can safely explore an unknown area while avoiding obstacles by the method explained in Section 4. During exploration, the position probabilities of the end nodes are continuously computed by (5) and the low values of position probabilities are increased by navigation. In order to select the best target node, the cost of each end node is calculated as follows:

\[ C_n = \alpha \cdot |X_n - X_r| + \beta \cdot |\angle(X_n - X_r) - \angle X_r| + \gamma \cdot P_n, \]  

where \( C_n \) is the cost associated with node \( n \), \( |X_n - X_r| \) is the distance between node \( n \) and the robot center, \( |\angle(X_n - X_r) - \angle X_r| \) is the angle between the robot heading and node \( n \), \( P_n \) is the position possibility of node \( n \), respectively. The weights of these three terms are adjusted by the parameters \( \alpha \), \( \beta \) and \( \gamma \). By (6), the cost decreases as a robot comes closer to the end node, the node is nearer to the direction of the robot heading and the position possibility of the end node is lower. Then, the node with the lowest cost is selected as the next target. The target node can disappear during exploration and/or its position possibility exceeds the threshold (e.g., 0.9) as the environment around the node is sufficiently explored. In such a case, the robot reselects the target node among the end nodes by using (6) and then continues to explore the environment. The whole environment has been fully explored when the position probabilities of all end nodes exceed a prescribed threshold.

The proposed TTE has been tested in various environments using the Pioneer DX2 robot. One laser scanner (SICK LMS200) was installed on the robot. Figure 15 shows the exploration process in a real environment. Each cell of the grid map is 10 cm × 10 cm in size and the environment is 10 m × 10 m in size. The figures beside each end node denote the end node number and its position possibility computed by (5) at the current time. The robot moved at a velocity of 20 cm/s and the total time required for completion of exploration was less than 2 min.

The robot is placed in the unknown environment at the beginning. In Fig. 16a, end node 1 is selected as a target node because it has the smallest cost among all end nodes. As the robot traverses toward end node 1 along the topological edge, end node 1 changes its position and its position possibility continues to increase. If the position possibility exceeds 0.9, then the end node is believed to maintain its position with confidence enough to guarantee that the robot does not need to visit this node to complete the modeling. In Fig. 16c, the robot starts to head for another end node without directly visiting end node 1 and end node 3 is then selected as the next target node. After the position possibility of end node 2 increases, end node 3 is selected as a new target node and the robot moves to end node 3. In Fig. 16g, a new environment is detected and consequently the target node keeps moving, as
shown in Fig. 16f and 16g. In the end, end node 3 disappears in Fig. 16h. Then, the robot heads for another end node 3 and its position possibility is calculated as 0.90 because the end node has been observed from various angles through Fig. 16d–h. Therefore, the robot stopped its exploration after the position probabilities of all end nodes exceed 0.9. Figure 16i shows the map of the environment.

In this approach a trade-off between careful exploration (completeness) and fast exploration (efficiency) can be adjusted by the threshold. If the threshold is set to a very high value, the robot visits every end node very close. In this case, the robot tends to excessively observe the environment without any consideration of the sensing ability. It is necessary, therefore, to select a certain threshold value and the robot can explore more efficiently. In this case some parts of the environment may not be properly mapped.

Figure 17 represents the flowchart for the proposed TTE scheme. TTE consists of four major parts. Each part functions as follows.
Figure 17. Flowchart for the entire TTE scheme.

(i) The ‘mapping and position possibility’ part builds a grid map and then a local TTM using the range data. It then calculates the position possibility of each end node.

(ii) If the start and target nodes are set and the position possibility of the target node is smaller than the threshold, the ‘trajectory generation and motion control’ part determines which edges the robot should follow to reach the target node and plans a trajectory. The robot is then controlled to follow this trajectory.

(iii) When the ‘node selection’ part is initiated, the costs of all end nodes are calculated and the target node is selected as the end node with the lowest cost. The robot then selects a suitable start node on the nearest edge within the sensing range.

(iv) During exploration, the ‘exploration completion check’ part is always in operation. This part checks whether the position possibility of the target node exceeds the prescribed threshold. If the position possibility is greater than the threshold, the ‘node selection’ part selects another target node. If the position probabilities of all end nodes exceed the threshold, the exploration process ends.
Figure 18. Grid and topological maps created using the proposed TTE scheme and robot paths illustrated during exploration: (a) office and (b) entire floor of building.

Figure 18a and 18b shows the map constructed by applying the proposed TTE strategy for other environments. Here, paths 1, 2, 3, 4 and 5 illustrate the actual paths along which the robot traveled in succession. The robot did not explore every nook and corner of the environment, but could obtain only the relatively accurate environment model. Figure 18b shows the map for the large-scale environment. This environment was 40 m $\times$ 12 m in size and consisted of a room, hallway, doors, a hall, etc. This map demonstrates that TTE can also effectively explore even large-scale environments.

6. Conclusions

In this paper, an efficient exploration scheme called a TTE which uses a TTM was proposed. To determine the level of thoroughness of exploration for a specific region of an environment, the concept of position possibility was introduced to evaluate the reliability of end node positions. With TTM and TTE operating together, the robot can autonomously model the unknown environment. This exploration strategy was validated by a series of experiments. From this research, the following conclusions have been drawn:

(i) The edge of a topological map can serve as a navigation path which is free of static and dynamic obstacles. Therefore, the proposed TTE scheme does not require additional path planning for exploration of an unknown environment.

(ii) The position possibility was useful for selecting the target point during exploration and judging the level of thoroughness of exploration for a specific region of an environment.

(iii) The proposed TTE based on position possibility can efficiently model the environment because the robot does not have to visit all end nodes directly to complete exploration.

The proposed method has some drawbacks. For example, it does not work well in open space. When the environment is very large and the obstacles are beyond the sensing range, the robot cannot generate useful thinning-based topological edges and nodes. In this unusual case, the EVG would work better as it would start wall
following. A major concern of this research is how to explore the environment for autonomous mapping. However, the purpose of exploration is to generate an accurate map of an unknown environment. Therefore, the localization accuracy is very important for obtaining a quality map. In this research, the robot estimates its accurate pose using the extended Kalman filter with line features extracted from the laser scanning data. Currently, research on localization using topological information is under way.

Acknowledgments

This paper was produced 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


About the Authors

**Tae-Bum Kwon** received the BS degree in Mechanical Engineering from Korea University, in 2002. He is currently a PhD candidate in Mechanical Engineering at Korea University. His research interests include mobile robotics.

**Jae-Bok Song** received the BS and MS degrees in Mechanical Engineering from Seoul National University, in 1983 and 1985, respectively, and the PhD 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 the Intelligent Robotics Research Center at Korea University. He is also an Editor for the International Journal of Control, Automation and Systems. His current research interests include mobile robot navigation, design and control of safe manipulators, and haptics.