Auton Robot (2011) 30: 351–368
DOI 10.1007/s10514-011-9223-6

Autonomous topological modeling of a home environment and topological localization using a sonar grid map

Jinwoo Choi · Minyong Choi · Sang Yep Nam · Wan Kyun Chung

Received: 4 February 2010 / Accepted: 17 February 2011 / Published online: 9 March 2011

Abstract This paper presents a method of autonomous topological modeling and localization in a home environment using only low-cost sonar sensors. The topological model is extracted from a grid map using cell decomposition and normalized graph cut. The autonomous topological modeling involves the incremental extraction of a subregion without predefining the number of subregions. A method of topological localization based on this topological model is proposed wherein a current local grid map is compared with the original grid map. The localization is accomplished by obtaining a node probability from a relative motion model and rotational invariant grid-map matching. The proposed method extracts a well-structured topological model of the environment, and the localization provides reliable node probability even when presented with sparse and uncertain sonar data. Experimental results demonstrate the performance of the proposed topological modeling and localization in a real home environment.

Keywords Topological modeling · Topological localization · Sonar sensors · Grid map · Grid-map matching · Home environment

J. Choi · M. Choi · W.K. Chung (✉)
Dept. of Mechanical Engineering, Pohang University of Science and Technology (POSTECH), Pohang, Korea
e-mail: wkchung@postech.ac.kr

J. Choi
e-mail: shalomi@postech.ac.kr

M. Choi
e-mail: minyong@postech.ac.kr

S.Y. Nam
Department of Information and Communication Engineering, Kookje College, Kyeongki-do, Korea
e-mail: r13337@unitel.co.kr

1 Introduction

A mobile robot requires an environmental modeling and localization capability to be able to navigate autonomously. Researchers have developed various robot mapping and localization methods for this purpose, including simultaneous localization and mapping (SLAM) algorithms (Thrun 2002; Thrun et al. 2001; Leonard and Durrant-Whyte 1991; Gutmann and Konolige 1999).

In general, internal representations of the environment recognized by robotic sensors can be classified into metric-based and topological approaches. The occupancy grid map (Elfes 1989) and feature-based methods (Leonard and Durrant-Whyte 1991) are typical examples of the metric-based approach. The metric map represents geometric entities of the environment as exact locations with respect to a reference frame, and localization is achieved by obtaining the location of the mobile robot as accurate measures. These exact representations help the mobile robot perform elaborate tasks that require high accuracy, such as placing a cup in an exact location.

On the other hand, the topological approach represents environmental entities as a graphical model consisting of nodes and edges (Choset and Nagatani 2001; Remolina and Kuipers 2004). The nodes are extracted from some meaningful places such as junctions, and the edges are determined from the connectivity of the nodes. In this approach, localization is performed by finding a node in which the robot is currently located. The topological approach has the advantage of being able to handle a large amount of data because of its compact and abstracted scheme. Moreover, it is suitable for human-robot interaction because the representation methodology is similar to the environmental perception of human beings.

In this paper, we focus on the topological approach using only sonar sensors in a home environment. Sonar sensors, which give relatively accurate range readings, are the most commonly used type of range sensor. They are cost effective and useful for mobile robot applications such as obstacle avoidance. However, because of their large beamwidth, sonar sensors suffer significantly from angular uncertainty and specular reflection, both of which make it difficult to reliably apply the sensors to feature-based approaches. Even though several methods have been developed that produce successful results in structured indoor environments (Tardós et al. 2002; Choi et al. 2005; Yap and Shelton 2009), the feature-based approach using sonar sensor always has the potential for failure due to the high uncertainty of the sonar sensor. Making a robot system work effectively using this type of sensor requires two things: improving the sensor performance by reducing the uncertainty, and enhancing the mapping and localization capability in the presence of uncertain sensor data. Enhancing the performance of the sonar sensor is not the topic of this paper; instead, we propose a robust topological approach based on a sonar grid map. The creation of the environmental model involves partitioning the grid map into several subregions, and localization is performed by comparing the current local grid map with the grid map constructed during the modeling procedure.

The proposed method is composed of two phases: topological modeling in which subregions are extracted from the sonar grid map, and topological localization in which the local grid map is compared with the original grid map.

First, we propose an efficient environmental modeling method for dividing the entire environment into several subregions to extract the topological model. An offline modeling process is first proposed to create the environmental model (Choi et al. 2009a), and then this offline method is improved to create an autonomous topological modeling method (Choi et al. 2009b). The offline topological modeling comprises three steps: generating the grid map for the whole environment, applying cell decomposition, and extracting the topological representation using normalized graph cut. Even though the offline method provides reliable modeling results suitable for a home environment, it requires that the number of subregions is predefined. To overcome this limitation, the autonomous topological modeling method was developed to extract new subregions from the current grid map incrementally. The extraction of a new subregion is performed in a manner that guarantees its convexity. This convexity criterion results in spatial recognition similar to perception in humans because a convex region generally involves similar spatial characteristics.

Second, localization based on the extracted topological model is achieved by matching the current local grid map with the original grid map generated during the modeling procedure (Choi et al. 2009a). The proposed localization is composed of four processes: extracting a template grid map by filtering out noisy data in the local grid map, selecting candidate locations using rotational invariant grid-map matching, calculating a prior node probability using a robot motion model, and obtaining posterior node probability. A local grid map always contains noisy data because sufficient sonar data cannot be accumulated. Therefore, as the first step in localization, the template grid map is extracted from the local grid map by filtering out the noisy data, and the grid-map matching is performed using the template grid map. Then, the node probabilities are calculated from the rotational invariant grid-map matching process and the prior probability, which is obtained from the previous node probability and the robot motion model. Here, a relative distance and a relative angle are used as the robot motion model to prevent accumulation of odometry error. The size of the local grid map used in these processes is determined adaptively using a test of the node probability entropy. Using this entropy test, the mobile robot can determine whether the current local grid map should be expanded by accumulating more sensor data.

The proposed method results in a well-structured topological representation of the environment and reliable localization performance using only sonar sensors. The proposed topological modeling and localization methods have several advantages.

For the proposed topological modeling,

- cell decomposition can systematically extract empty regions in the grid map and produce a roughly modeled graph structure for the empty environment,
- normalized graph cut produces an effective clustering result by maximizing the similarity within clusters; this has low computational burden because of the cell decomposition process, and
- the topological map can be constructed without defining the number of subregions in advance.

For the proposed localization,

- a successful grid map match is possible even with noisy data in the local grid map,
- odometry errors do not accumulate because only temporary relative robot motion is used to calculate the node probability, and
- the proposed localization guarantees convergent and reliable node probability using the entropy test.

Furthermore, the proposed topological method can be integrated with high-level semantic information. By providing semantic information such as “bedroom” or “kitchen” to the extracted subregions, the proposed method would be very useful for human-robot interaction in the high-level planning stage.

The remainder of this paper is organized as follows. Section 2 presents related topological approaches. Section 3 describes the method of topological modeling using the sonar grid map. Section 4 presents the localization procedure using local grid-map matching. Section 5 describes experimental results, and Sect. 6 presents the conclusion.

2 Related works

Several topological approaches have been developed for reliable environmental modeling and localization in indoor environments. The generalized Voronoi graph (GVG), one of the best-known topological approaches, detects nodes and edges by investigating equidistant obstacles. Geometric characteristics are used to recognize nodes for localization purposes. The GVG has been successfully implemented as topological SLAM in a corridor environment (Choset and Nagatani 2001), and Beeson et al. (2005) developed the extended Voronoi graph that enabled the Voronoi graph to be applied to non-corridor environments. Similarly, Doh et al. (2009) developed a topological SLAM algorithm under semi-permanent dynamics induced by door opening and closing. Lee et al. (2006) categorized entire environments into node and edge regions using the eigenvalue ratio (EVR) of sonar sensor data.

Using a different approach, other researchers developed topological representations by partitioning the environment into several node regions. Thrun (1998) extracted topological graphs from grid maps based on Voronoi diagrams for the efficient planning method. Buschka and Saffiotti (2002) obtained room-like spaces from grid maps using fuzzy morphological openings and watershed segmentation. Similarly, the graph partitioning method has been applied to grid maps to extract node regions (Zivkovic et al. 2006; Brunskill et al. 2007). Mozos and Burgard (2006) applied the AdaBoost learning algorithm to laser-range data to classify door, room, and corridor regions. As an example of the use of vision sensors, Tapus and Siegwart (2006) used the fingerprints of places to generate an appearance-based topological model.

These methods cannot easily be directly applied to our system. Some are suitable only in corridor environments where few meaningful locations such as corner points and crossing points are used as nodes, and localization is only performed when the robot arrives at these node points (Choset and Nagatani 2001; Doh et al. 2009; Lee et al. 2006). In a home environment, on the other hand, nodes should be segmented as spaces such as rooms. Several of the existing approaches do segment the environment into several subregions. However, some previous methods used only narrow passages, such as doorways, to extract nodes (Buschka and Saffiotti 2002; Mozos and Burgard 2006), and others could not provide reliable topological localization in a home environment using only sparse sonar sensors, because they did not consider the localization problem (Beeson et al. 2005; Zivkovic et al. 2006) or they relied on laser range finders or vision sensors for the localization (Brunskill et al. 2007; Tapus and Siegwart 2006; Mozos and Burgard 2006).

Unlike the existing approaches, this paper describes a novel topological algorithm consisting of environmental modeling and localization using only low-cost sonar sensors to achieve reliable results in a home environment.

3 Topological environmental modeling in a home environment

Environmental modeling using a topological representation should extract the nodes and edges from the complete environment. In a home environment, extracting meaningful points such as junctions, in the case of corridors, is difficult, because almost all locations are potentially meaningful for mobile robot tasks. Therefore, extracting nodes as spaces such as rooms is more appropriate than acquiring them as specific points, and edges should be defined as the connections between those spaces. For this purpose, a navigable region for the mobile robot is extracted from the environment, and the navigable region is divided into several subregions. These subregions are then regarded as nodes.

As previously mentioned, an offline topological modeling process is first proposed, and an autonomous topological modeling algorithm is then developed by improving the offline method.

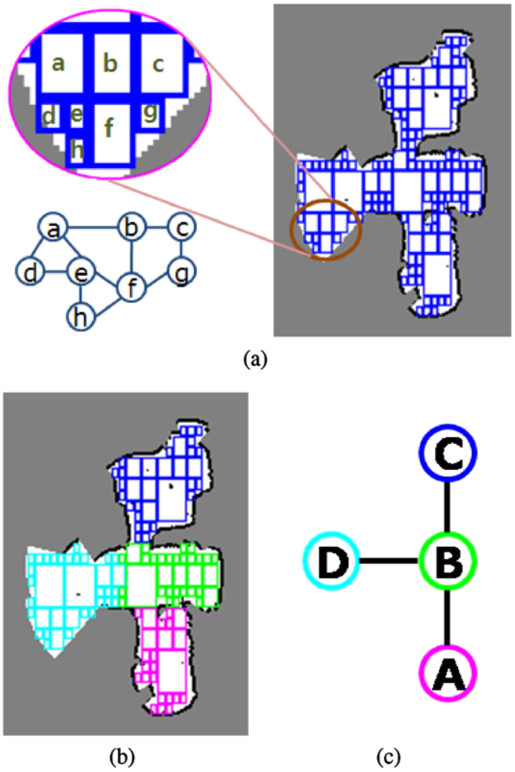
3.1 Offline topological modeling

The first step in offline topological modeling is to generate an occupancy grid map for the entire environment. The robot pose is given by calibrated odometry data while generating the grid map (Yun et al. 2008). Many occupancy grid map generation methods exist, but the improvement of the grid map performance is outside the scope of this paper. Therefore, we used the existing grid mapping method (Lee and Chung 2009).

3.1.1 Cell decomposition

The generated binary grid map consists of empty grids $m(x, y) = 0$, which represent free space, and occupied grids $m(x, y) = 1$, which contain obstacles. Cell decomposition of the generated grid map is used for topological modeling of the navigable free space to extract empty grids systematically.

Cell decomposition, also known as quadtree cell decomposition, divides a square cell into four smaller square cells of the same size if the original cell is composed of both free and obstacle spaces (Katevas et al. 1998). This process



(a)

(b)

(c)

Fig. 1 Example of offline topological modeling: (**a**) initial draft topological model, (**b**) normalized graph cut in which each cluster is represented as a different color, and (**c**) extracted topology

is executed recursively until every cell is decomposed into separate free and obstacle spaces. Cell decomposition effectively extracts empty regions in the grid map, and the empty regions can be modeled as squares of various sizes. A large empty space would be modeled mainly as a few large squares with a few relatively small squares, as shown in Fig. 1(a).

Cell decomposition provides an initial draft model of the topological representation of the environment. Each extracted cell becomes a node of the draft topological model, and the connecting edge is determined by the adjacency of two cells (Fig. 1(a)).

3.1.2 *Normalized graph cut*

The draft topological model constructed using cell decomposition provides a connected graph structure for the empty regions of the environment. However, many small cells should be merged with large cells that could be considered to be part of the same region. Normalized graph cut is applied to the draft topological model for effective clustering.

Normalized graph cut, a clustering method using graph partitioning, was proposed by Shi and Malik (2000), who applied it to image segmentation. It uses a graph structure $G(V,E)$ composed of a set of vertices (nodes) $V = \{V_1, V_2, \cdots, V_n\}$ and a set of edges $E = \{E_1, E_2, \cdots, E_m\}$. Each edge has a weight w_{ij} that represents the similarity between V_i and V_j. To segment the graph, the normalized cut (*Ncut*) is defined to measure the similarity between two clusters that should be segmented. *Ncut* between two clusters C_1 and C_2 can be obtained by

$$Ncut = \frac{\sum_{i\in C_1, j\in C_2} w_{ij}}{\sum_{i\in C_1, j\in V} w_{ij}} + \frac{\sum_{i\in C_1, j\in C_2} w_{ij}}{\sum_{i\in C_2, j\in V} w_{ij}}. \tag{1}$$

We can obtain two segmented clusters with minimum similarity by minimizing *Ncut*. The result also maximizes a measure of similarity within clusters C_1 and C_2 because the sum of (1) and (2) is a constant value.

$$Nassoc = \frac{\sum_{i\in C_1, j\in C_1} w_{ij}}{\sum_{i\in C_1, j\in V} w_{ij}} + \frac{\sum_{i\in C_2, j\in C_2} w_{ij}}{\sum_{i\in C_2, j\in V} w_{ij}}. \tag{2}$$

Unfortunately, finding the minimum *Ncut* is an NP-hard problem. Therefore, spectral clustering is generally used as an approximate solution. Spectral clustering to minimize *Ncut* is performed in the following steps:

1. Construct a neighborhood graph with a corresponding $n \times n$ affinity matrix $W(i,j) = w_{ij}$.
2. Compute the normalized graph Laplacian $L = D^{-1/2}(D-W)D^{-1/2}$, where $D = \mathrm{diag}\{d_1, \cdots, d_n\}$ and $d_i = \sum_j W_{ij}$.
3. Find the k smallest eigenvectors $u_1, \cdots, u_k$ of L and form the matrix $U = [u_1 \cdots u_k] \in \mathbb{R}^{n\times k}$.
4. Form matrix $\tilde{U}$ from U by re-normalizing each row of U to have unit norm, i.e., $\tilde{U}_{ij} = U_{ij}/(\sum_j U_{ij})^{1/2}$.
5. Treat each row of $\tilde{U}$ as a point in $\mathbb{R}^k$, and segment them into k groups using the k-means algorithm.
6. Assign V_i to cluster j if and only if row i of $\tilde{U}$ is assigned to cluster j.

Based on the result of cell decomposition, each cell becomes node V_i and the weight value between two cells is defined to calculate the affinity matrix W using (3).

$$w_{ij} = \begin{cases} \sharp \text{ of adjacent grids} & \text{if } V_i \text{ and } V_j \text{ are adjacent} \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Using the draft graph model and the affinity matrix, k clusters can be segmented with a predefined variable k. Figure 1(b) shows the result of the normalized graph cut using the draft topology model of Fig. 1(a). We segmented four clusters using the normalized graph cut, and the segmented clusters can be represented as graphical structures, as shown in Fig. 1(c). Four clusters were extracted successfully based on the spatial geometry. Two rooms were segmented into nodes A and C, and the remaining space was divided into nodes B and D.

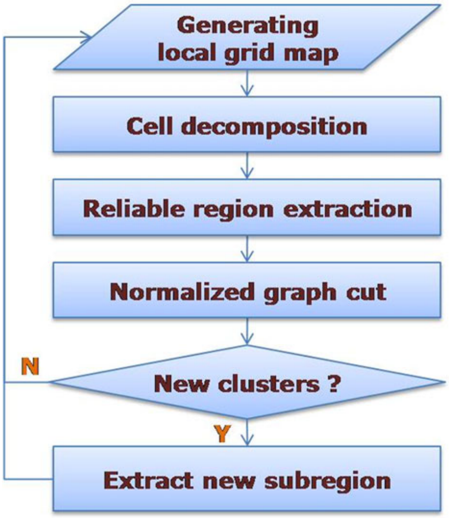


Fig. 2 Flowchart for the autonomous topological modeling using sonar grid map

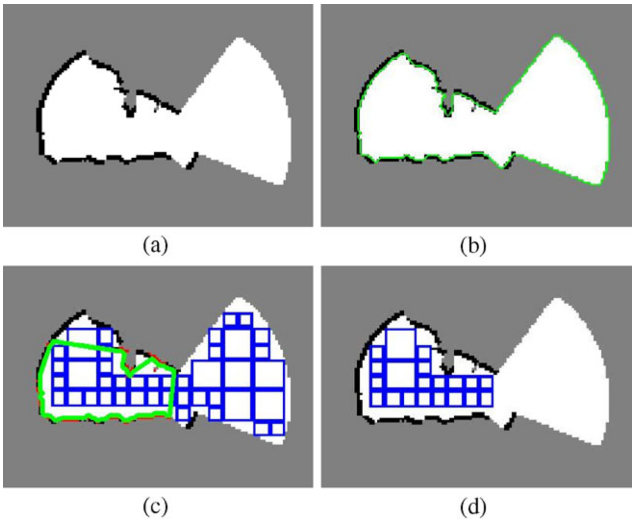


Fig. 3 Obtaining reliable regions in the local grid map: (**a**) noisy local grid map, (**b**) boundary tracing, (**c**) contour for reliable region, and (**d**) reliable cells

3.2 Autonomous topological modeling

The offline method can successfully extract the topological representation from the grid map. However, the number of clusters k must be manually predefined, and the topological model should be extracted when the grid map is generated for the entire environment. If the robot navigates unexplored areas after extracting the topological model, the topological model must be reconstructed starting with the generation of the grid map. To overcome these limitations, the offline method is improved into an autonomous topological modeling process.

Autonomous topological modeling involves incremental extraction of subregions. A local grid map is generated around the robot as it navigates the environment, and the subregions are extracted using the local grid map. During this process, the subregion can be extracted without predefining the number of subregions.

Figure 2 shows a flowchart of the autonomous topological modeling method. Most processes are similar to those in the offline method. The major differences are the extraction of a reliable region in the local grid map and extraction of new subregions.

3.2.1 Obtaining reliable regions in the local grid map

Autonomous topological modeling should be performed using an incomplete local grid map. The local grid map always contains noisy data because sufficient sensor data cannot be accumulated to filter out spurious sonar data (Fig. 3(a)). To reduce the effects of the noise, reliable regions in the local grid map should be obtained before subregion extraction takes place.

As the first step in obtaining reliable regions, a boundary-tracing technique is used to find boundaries between occupied and empty regions (Fig. 3(b)). A binary grid map is acquired from the local grid map, and all boundary grids between occupied and empty grids are obtained. Then, a sequence of the connected boundary grids is obtained by tracing the boundary grids in a certain direction, clockwise or counterclockwise (Gonzalez and Woods 2002). The boundary-tracing method finds a contour that encloses the empty regions in the local grid map.

A sonar sensor model is then used to measure a confidence level for each occupied grid. A sonar beam S generated by a transmitter has a sound pressure function $P_S(r,\theta)$ defined as

$$P_S(r,\theta)=\frac{\beta f a^4}{r^2}\left(\frac{2J_1(ka\sin\theta)}{ka\sin\theta}\right)^2, \tag{4}$$

where r is the distance from the transmitter, and θ is the angle with respect to the transmitter direction. Detailed explanations for the other variables can be found in Kleeman and Kuc (2008).

The sound pressure function can be used to measure the amount of sensor information. The confidence for each occupied grid $m(x,y)$ is evaluated using the sensor model as

$$Conf(x,y)=\sum_{m(x,y)\in Occ.(S)} P_S(r,\theta), \tag{5}$$

where $Occ.(S)$ is a set of grids determined to be occupied according to sensor data S. The confidence value is used to assess the reliability of each occupied grid. A grid with a higher confidence value is more reliable than a grid with a lower confidence value, because the confidence reflects the amount of sensor information that determines the current status of each grid. The occupied grids that have confidence values greater than the average confidence value are then classified as confident grids (6).

$$Conf(x, y) > avg(Conf), \tag{6}$$

where $avg(Conf)$ is the average confidence value of all occupied grids in the local grid map. The grids classified as confident are regarded as truly occupied grids, whereas all other grids are considered to be noisy data.

Using the obtained confident grids, the reliable region can be found by obtaining a contour that connects those confident grids (Fig. 3(c)).

Finally, reliable regions in the local grid map can be obtained by removing the decomposed cells that are outside the contour (Fig. 3(d)). A reliable region is obtained from the noisy local grid map using these processes, and the remaining reliable cells are applied to the normalized graph cut to extract subregions.

3.2.2 Extracting a new subregion

The autonomous topological modeling method involves extracting a new subregion from the obtained reliable cells. To extract the new subregion, the proposed method tentatively divides these cells into two clusters using the normalized graph cut and determines whether the cells should be divided into two clusters. If the cells should be divided into two clusters, one of the two divided clusters is extracted as a new subregion. Otherwise, the robot continues to generate the grid map.

The convexity of the subregion is used as a criterion for determining the division of the cells. In other words, the subregion is extracted in such a way as to guarantee its convexity. More details of this procedure follow.

1. Evaluate a measure of convexity, considering the reliable cells to be one cluster.

$$\mathcal{C}_{1\text{cluster}} = \frac{\sharp \text{ of } occ. \text{ grids} \in CH1}{\sum \text{size of Cell}}, \tag{7}$$

   where $CH1$ is a convex hull of all the reliable cells (Fig. 4(a)).
2. Divide the reliable cells into two clusters using the normalized graph cut.
3. Evaluate the convexity measure, considering the reliable cells to be two clusters.

$$\mathcal{C}_{2\text{clusters}} = \frac{\sum_{i=1}^{2} \sharp \text{ of } occ. \text{ grids} \in CH2(i)}{\sum \text{size of Cell}}, \tag{8}$$

   where $CH2(i)$ represents a convex hull of all cells corresponding to the i^{th} cluster (Fig. 4(b)).

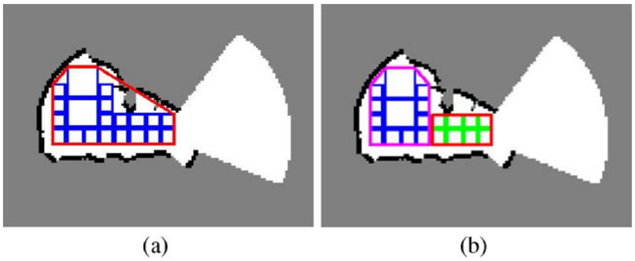


Fig. 4 Convex hulls for (**a**) one cluster and (**b**) two clusters. The *blue* and *green cells* represent two different clusters obtained from the normalized graph cut

4. Extract a new subregion if the following conditions are satisfied.

$$\mathcal{C}_{1\text{cluster}} > c_t \quad \& \quad \mathcal{C}_{2\text{clusters}} < 0.5 \times \mathcal{C}_{1\text{cluster}}, \tag{9}$$

   where c_t is a threshold value.

The convexity measures $\mathcal{C}_{1\text{cluster}}$ and $\mathcal{C}_{2\text{clusters}}$ are the ratios of occupied grids in the convex hulls $CH1$ and $CH2$, respectively, to the total size of reliable cells. A smaller convexity measure indicates that the region is more convex. The former condition in (9) means that the reliable cells cannot be regarded as a convex region using the allowable threshold value c_t, and they should be divided into two regions to extract a convex subregion. The threshold value, c_t, was determined by various experiments to be 0.2. The physical meaning of this value is that, at most, 20% of the grids in a subregion are allowed to be occupied for the subregion to be considered a convex region. The latter condition in (9) ensures that the normalized graph cut provides effective clustering to guarantee the divided clusters as convex regions.

When the conditions are satisfied, the divided cluster that is farther from the current robot location is extracted as a new subregion. The grid map around the cluster that is closer to the current robot location might contain undetermined grids, which should be updated by the subsequent sensor data. On the other hand, the grid map around the cluster that is farther from the current robot location has already been updated sufficiently to be temporarily considered static, as it is not affected by current sensor data. So, the cluster that is farther from the current robot location is extracted as the new subregion. The subregion extracted this way is not considered for any subsequent segmentation unless the robot navigates back to the same subregion again.

Figure 5 shows the process of extracting new subregions incrementally. The cells filled with horizontal and vertical line patterns represent the two current tentative clusters. During the first two steps, shown in Figs. 5(a) and 5(b), new subregions are not extracted, and the obtained cells are considered to be in one subregion because they do not satisfy the dividing conditions in (9). The conditions are satisfied in Fig. 5(c), so the cells filled with a vertical line pattern, which

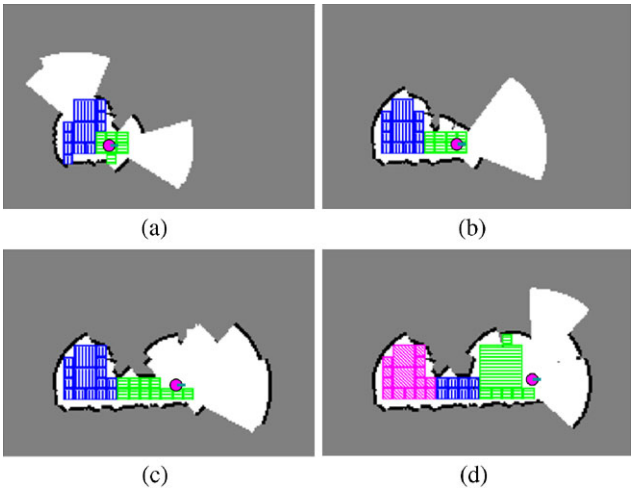


Fig. 5 Example of autonomous topological modeling using a sonar grid map. Cells filled with vertical and horizontal line patterns represent the two current tentative clusters. An extracted subregion is represented by cells filled with a diagonal line pattern

are farther from the current robot location than the cells filled with a horizontal line pattern, are extracted as a new subregion. The newly extracted subregion is represented as cells filled with a diagonal line pattern in Fig. 5(d), and they are not reconsidered for subsequent extraction of another new subregion. In other words, the remaining spaces (i.e., the cells filled with horizontal and vertical line patterns), except the extracted new subregion (i.e., the cells filled with a diagonal line pattern), are used to extract another new subregion in Fig. 5(d). A successful topological model can be achieved autonomously from the sonar grid map using this process.

4 Topological localization in a home environment

In the topological representation, localization of a mobile robot can be performed by finding the node where the robot is currently located. Node classification using laser range finders or vision sensors would be relatively easy. However, node classification using sonar sensors is difficult because of the sparseness and uncertainty of the data they produce. When sonar sensors are used, sonar data should be accumulated to overcome the limitations of the sensor performance.

In this paper, a local grid map around the current robot position is generated to accumulate the sensor data. The robot can determine its own location by comparing the local grid map to the original grid map generated during the modeling procedure. Figure 6 shows a flowchart of the proposed topological localization procedure. We first extract a template grid map by filtering out the uncertain data because of the uncertain information contained in the local grid map. Candidate locations in each node are then selected by applying rotational invariant matching to the extracted template grid map. Finally, node probabilities are calculated from the prior node probability and the grid-map matching probability.

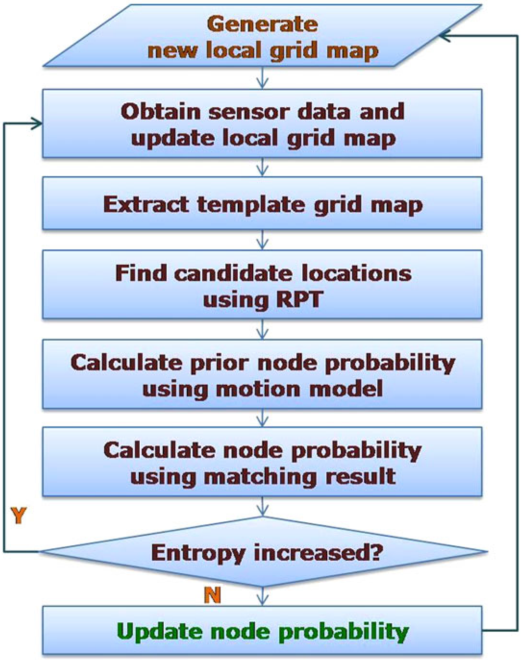

Generate new local grid map
Obtain sensor data and update local grid map
Extract template grid map
Find candidate locations using RPT
Calculate prior node probability using motion model
Calculate node probability using matching result
Y
Entropy increased?
N
Update node probability

Fig. 6 Flowchart of the proposed topological localization using sonar grid-map matching

A test of the node probability entropy is also used to generate a reliable local grid map. Using the entropy test, the size of the local grid map can be determined adaptively for successful grid-map matching.

4.1 Extracting the template grid map

The template grid map is extracted using the confidence measure of (5) which was proposed to obtain the reliable region in Sect. 3.2. A grid has high confidence if it is repeatedly determined to be occupied using high-pressure sound data. On the other hand, a grid detected by low-pressure sound data only a few times has a low confidence value. High-confidence grids can be regarded as reliable data, and low-confidence grids can be considered as noisy data. The noisy data can be filtered out of the local grid map for reliable grid-map matching using the confidence level.

Figure 7(a) shows an example of a local grid map. As previously mentioned, the local grid map contains a large number of uncertain data. A confidence map of the local grid map is obtained as shown in Fig. 7(b) for reliable matching. The template grid map is obtained as shown in Fig. 7(c) using the confidence measure and indicates that the extraction of the template grid map was successfully achieved by removing the uncertain data.

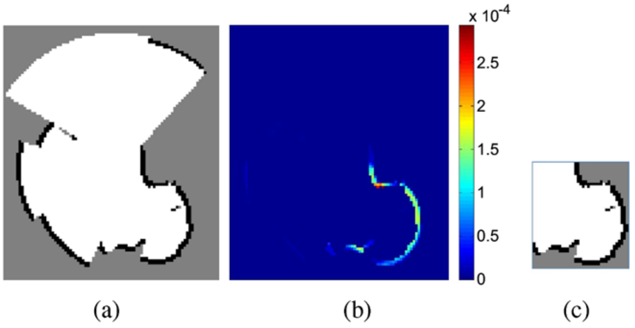


Fig. 7 Extracting the template grid map: (**a**) uncertain local grid map, (**b**) confidence map, and (**c**) extracted template grid map

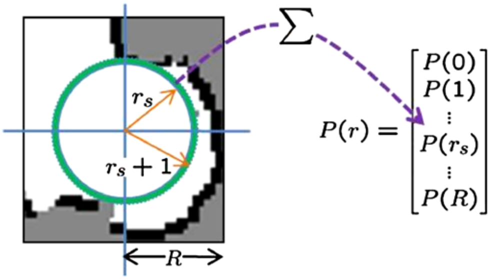


Fig. 8 Ring projection transformation

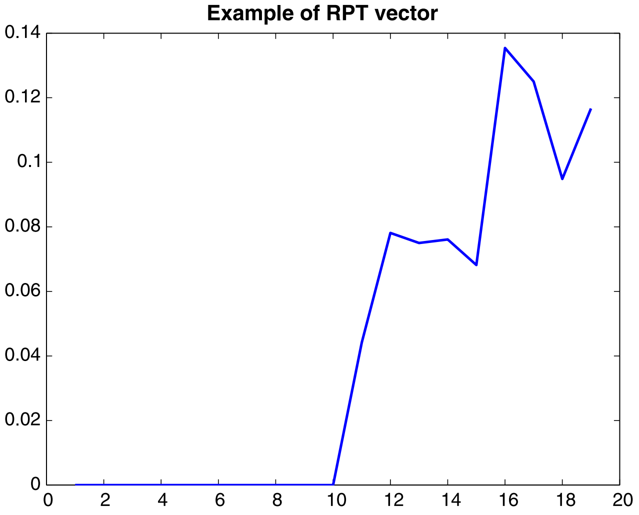


Fig. 9 Ring projection transformation vector

4.2 Rotational invariant grid-map matching

Candidate locations are acquired by matching the template grid map with the original grid map. One candidate location is obtained from each node, and the candidate location is assigned to the node as the location of a grid within the corresponding subregion in the original grid map. The selected candidate locations then become the representative locations of the nodes for the current observation. Moreover, the matching probabilities are obtained to calculate the node probabilities.

4.2.1 Selecting candidate locations

The candidate locations are selected as the most similar location within each node of the template grid map. The matching method for obtaining the candidate locations should be rotational invariant because a difference in rotation exists between the template grid map and the original grid map. Ring projection transformation (RPT) is used as the rotational invariant grid-map matching method.

RPT is a data-reduction method that transforms two-dimensional data to a one-dimensional vector (Lin et al. 2006). It transforms all data at a distance of $[r, r+1)$ to vector $P(r)$, as shown in Fig. 8. For a grid map $m(x, y)$ whose center point is (x_c, y_c), the RPT vector $P(r)$ is obtained as follows:

$$P(r) = \frac{1}{S_r} \sum_{r_{xy} \in [r, r+1)} m(x, y), \tag{10}$$

where r is an integer value from 0 to R, S_r is the total number of grids in $[r, r+1)$, and $r_{xy} = \sqrt{(x-x_c)^2 + (y-y_c)^2}$. Here, the size of the RPT vector, R, is determined as half of the shortest dimension of the template grid map. Figure 9 shows an example of the RPT vector for the template grid map in Fig. 7(c). The RPT vector represents a ratio of occupied grids in a range of $[r, r+1)$. The RPT vectors for a template grid map $P_T(r)$ and original grid map $P_O(r)$ can be obtained from (10), and the rotational invariant matching can be achieved by calculating the normalized correlation between two vectors using

$$\rho = \frac{\sum_{r=0}^{R} \{P_T(r) - \mu_T\}\{P_O(r) - \mu_O\}}{\left(\sum_{r=0}^{R} \{P_T(r) - \mu_T\}^2 \cdot \sum_{r=0}^{R} \{P_O(r) - \mu_O\}^2\right)^{1/2}}, \tag{11}$$

where $\mu_T = \frac{\sum_{r=0}^{R} P_T(r)}{1+R}$, and $\mu_O = \frac{\sum_{r=0}^{R} P_O(r)}{1+R}$.

The most similar location within each node can be obtained by acquiring the maximum normalized correlation for each node. In other words, k candidate locations are obtained from k extracted nodes.

4.2.2 Calculating matching probability

The matching probability can be regarded as the correlation value of each candidate location. However, we used an additional matching process to obtain a more distinct matching probability because the RPT reduces the data too much.

To achieve more distinctiveness, we compared distances from the candidate location to the closest obstacle for every degree in 360°. A distance vector for a candidate location ($L(x_i, y_i)$) can be obtained as

$$D_i(\theta) = \min \mathrm{Dist}(Occ(x, y) \text{ from } L(x_i, y_i) \text{ in } \theta \text{ direction}), \tag{12}$$

where θ is an integer in the range 1–360, and $Occ(x, y)$ is a set of occupied grids in the grid map. The distance vector for i^{th} node D_i can be obtained with respect to the candidate location L_i, and another distance vector for the template grid map D_T is obtained from the center point (x_c, y_c) of the template grid map. A measure of dissimilarity between two distance vectors is then calculated as

$$\triangle D_i = \arg\min_{\theta_c} \sum_{\theta=1}^{360} |D_T(\theta) - D_i(\theta - \theta_c)|, \tag{13}$$

where $D(\theta) = D(\theta + 360)$ for $\theta < 0$.

After calculating $\triangle D$ for every node, the similarity of the distance vector for i^{th} candidate location is obtained by

$$P_D(i) = \frac{1/\triangle D_i}{\sum_{j=1}^{k} 1/\triangle D_j} \tag{14}$$

where k is the number of nodes.

Using this similarity measure, the matching probability of each node is obtained by multiplying by the normalized correlation calculated from RPT vectors.

$$P_{match}(i) = P_D(i) \times \rho(x_i, y_i) \tag{15}$$

4.3 Calculating node probability

The node probability can be calculated from observation and prior information. The grid-map matching probability can be regarded as the observation, and the prior information should be obtained from the previous node probability and the robot motion model. In this paper, the relative distance and relative angle are used as the motion model for acquiring the prior node probability.

4.3.1 Basic equations

The node probability using the motion model and observations can be calculated by

$$\begin{aligned} &P(N_t = N_i | u_{1:t}, z_{1:t}) \\ &\quad = \eta_1 P(z_t | N_t = N_i, u_{1:t}, z_{1:t-1}) P(N_t = N_i | u_{1:t}, z_{1:t-1}) \\ &\quad = \eta_1 P(z_t | N_t = N_i) P(N_t = N_i | u_{1:t}, z_{1:t-1}) \end{aligned} \tag{16}$$

where η_1 is a normalizing factor, N_t is the node where the robot is located at time t, N_i is the i^{th} node, and $u_{1:t}$ and $z_{1:t}$ are the robot motions and observations up to time t, respectively.

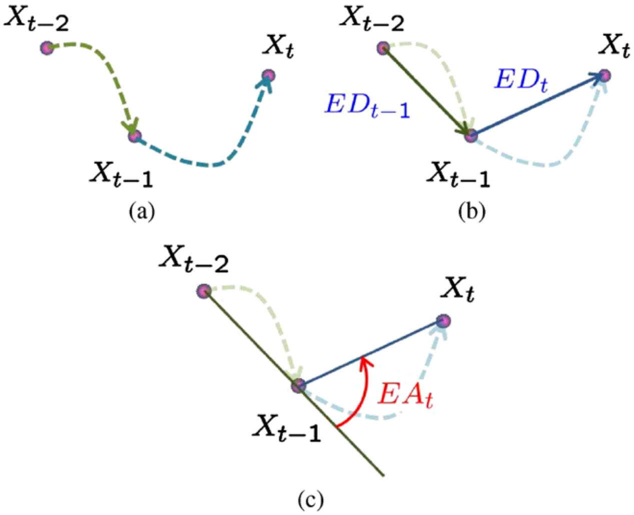


Fig. 10 Robot motion model: (**a**) actual robot motion, (**b**) effective distance (*ED*), and (**c**) effective angle (*EA*)

The first part of (16) is a likelihood and the last part is the prior information. The likelihood can be obtained from the grid-map matching probability in (15). The prior information can be derived from the previous node probability and the robot motion model using

$$\begin{aligned} &P(N_t = N_i | u_{1:t}, z_{1:t-1}) \\ &\quad = \sum_j P(N_t = N_i | N_{t-1} = N_j, u_{1:t}, z_{1:t-1}) \\ &\qquad \times P(N_{t-1} = N_j | u_{1:t-1}, z_{1:t-1}). \end{aligned} \tag{17}$$

4.3.2 Effective distance as a motion model

The robot motion model is obtained from odometry data corresponding to the robot path that generates the current local grid map. Because the candidate locations are selected with respect to a center point of the template grid map, the relative distance is measured as a distance between the center points of the previous template grid map and the current template grid map. The straight-line distance between two points is referred to as the effective distance (*ED*) and, it is used as the robot motion model even though the robot does not necessarily move in a straight line.

For example, when a robot moves as shown in Fig. 10(a), only the straight line is considered for the *ED* even though the robot actually moves along the dashed line. Therefore, the *ED* for the robot path in Fig. 10(a) becomes the length of the straight line shown in Fig. 10(b).

The prior node probability is calculated by the following procedures using the *ED*. First, a predicted *ED*, $\hat{d}$, is obtained from odometry. Estimated *ED*s are then acquired by calculating distances between candidate locations of the previous and current template grid maps. In other words, an estimated *ED*, d_{ij}, is calculated as the distance between a candidate location $L_{t-1}(x_j, y_j)$ in the j^{th} node at time $t-1$ and another candidate location $L_t(x_i, y_i)$ in the i^{th} node at time t. Finally, the first part of the prior node probability in (17) is derived as

$$\begin{aligned}
&P(N_t = N_i | N_{t-1} = N_j, ED_{1:t}, z_{1:t-1}) \\
&\quad = P(N_t = N_i | N_{t-1} = N_j, ED_t) \\
&\quad = \mathcal{N}(ED; \hat{d}, \sigma_d^2)|_{ED=d_{ij}} \qquad (18)
\end{aligned}$$

where $\mathcal{N}(\cdot)$ is the normal distribution function. Node probabilities for every node can be calculated using this approximate motion model.

4.3.3 *Effective angle as a motion model*

The relative angle of robot motion is used to complement the *ED*. The direction of ED_{t-1} is used as the reference for the relative angle at time t. As shown in Fig. 10(c), the effective angle (*EA*) at time t is defined as the relative angle of ED_t with respect to the direction of ED_{t-1}.

The prior node probability using the *EA* can be calculated similar to (18). However, the *EA* is related not only to the state of time step $t-1$ but also to that of time step $t-2$. The equation for the prior node probability should be expanded to time step $t-2$. The following derivation assumes that *ED* and *EA* are independent.

$$\begin{aligned}
&P(N_t = N_i | N_{t-1} = N_j, ED_{1:t}, EA_{1:t}, z_{1:t-1}) \\
&\quad = \eta_2 P(N_t = N_i | N_{t-1} = N_j, ED_{1:t}, EA_{1:t-1}, z_{1:t-1}) \\
&\qquad \times P(N_t = N_i | N_{t-1} = N_j, ED_{1:t-1}, EA_{1:t}, z_{1:t-1}) \\
&\quad = \eta_2 P(N_t = N_i | N_{t-1} = N_j, ED_t) \\
&\qquad \times \sum_k P(N_t = N_i | N_{t-1} = N_j, N_{t-2} = N_k, \\
&\qquad ED_{1:t-1}, EA_{1:t}, z_{1:t-1}) \\
&\qquad \times P(N_{t-2} = N_k | N_{t-1} = N_j, ED_{1:t-1}, DA_{1:t}, z_{1:t-1}) \\
&\quad = \eta_2 P(N_t = N_i | N_{t-1} = N_j, ED_t) \\
&\qquad \times \sum_k P(N_t = N_i | N_{t-1} = N_j, N_{t-2} = N_k, EA_t) \\
&\qquad \times P(N_{t-2} = N_k | N_{t-1} = N_j, u_{1:t-1}, z_{1:t-1}) \qquad (19)
\end{aligned}$$

In (19), the first part is same as (18), and the last part can be obtained from the previous motion model. The middle part of (19) can be calculated as

$$\begin{aligned}
&P(N_t = N_i | N_{t-1} = N_j, N_{t-2} = N_k, EA_t) \\
&\quad = \mathcal{N}(EA; \hat{a}, \sigma_a^2)|_{EA=a_{ijk}} \qquad (20)
\end{aligned}$$

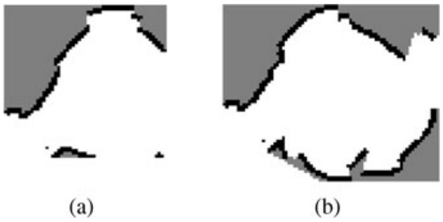


Fig. 11 Template grid maps: (**a**) inadequate template grid map for reliable grid-map matching and (**b**) template grid map expanded by accumulating more sensor data

where $\hat{a}$ is a predicted *EA* obtained from odometry and a_{ijk} is an estimated *EA* calculated from candidate locations $L_{t-2}(x_k, y_k)$, $L_{t-1}(x_j, y_j)$ and $L_t(x_i, y_i)$.

The prior node probability can be obtained efficiently using the *ED* and *EA* simultaneously. Moreover, the proposed method has an advantage in that the accumulation of odometry error is bounded within two time steps because it is based on relative motion models.

Topological localization can be therefore achieved successfully by calculating the posterior node probability using the prior node probability and the grid-map matching probability.

4.3.4 *Entropy test of node probability*

The size of the local grid map might be determined by a constant traveling time or a constant traveling distance. Unfortunately, the constant time and traveling distance criteria are generally insufficient for the sparse sonar grid map. For example, the two template grid maps in Fig. 11 were obtained from the same room, node C in Fig. 1. The template grid map in Fig. 11(a) is difficult to match to the original grid map because of insufficient information. Therefore, the candidate locations using the template grid map in Fig. 11(a) may be incorrect. In this case, the template grid map should have more information obtained by accumulating more sensor data, as in Fig. 11(b). Even though the template grid map in Fig. 11(b) also contains noisy data, this template could be matched to the original grid map to find reliable candidate locations.

For this purpose, an entropy test is used to determine whether more sensor data should be accumulated:

$$H(P) = \sum_{i=1}^{n} -P(i) \log_n P(i) \qquad (21)$$

where n is the number of nodes, and $P(i)$ is the node probability for the i^{th} node. Without any loss of generality, we can say that a successful observation should result in a convergence of the node probability, and the convergent result would reduce the entropy value of the node probability. Therefore, increasing entropy means that the information in the template grid map is not sufficient to update the node probability reliably. In the proposed entropy test, we use the increasing entropy of node probability as an indicator to accumulate more sensor data to construct the local grid map. In other words, when the entropy measure increases, the node probability is not updated, and the robot continues to generate the local grid map by accumulating more sensor data. This entropy test was tested in a real experiment with some heuristics. It was applied to the localization process when the maximum node probability was less than some threshold value. The size of the local grid map can be determined adaptively using the entropy test. The obtained local grid map could be then used to update the node probability reliably.

The proposed topological localization works well principally in a static environment, which means that the environment in the localization procedure is same as in the modeling procedure. In addition, it can also handle small dynamic changes on the order of less than tens of centimeters. General environmental changes in a home environment include moved tables and chairs, an open door that has been closed, and moving people. If these kinds of dynamic changes occur, localization would result in a temporary unreliable node-probability update due to the dynamic changes. However, the proposed method recovers reliable results using the entropy test and the subsequent Bayesian node probability updates (16).

In a dynamic environment, the candidate locations cannot be obtained accurately when the dynamic parts of the environment dominate the proposed grid map matching. A failure of the grid map matching induces a divergent node probability, which increases the entropy measure of the node probability. This increased entropy is detected by the proposed entropy test procedure, enabling the topological localization method to exclude the calculated node probability. The robot, therefore, does not update the node probability and continues to expand the local grid map by accumulating more sensor data. Through repetitions of this process, the robot obtains sensor data in which a greater portion of the data corresponds to static parts of the environment. After accumulating enough sensor data, the robot obtains the candidate location accurately. Then, the proposed localization method can obtain a reliable convergent node probability by simultaneously using both the robot motion model and the observation model in the Bayesian update process. The performance of the proposed topological localization in dynamic environments will be verified by experiments described in Sect. 5.2.2.

5 Experimental results

This section presents the experimental results of the proposed topological modeling and localization method. The experiments were conducted using a Pioneer 3-DX differential drive robot (Fig. 12(a)) equipped with 12 Murata MA40B8 sonar sensors in a 11.4 m × 8.7 m home environment of several rooms containing items of furniture (Fig. 12(b)). The robot was manually guided along an arbitrary path at an average speed of about 0.15 m/s while acquiring sensor data at a rate of 4 Hz. The grid size of the generated grid map was 5 cm × 5 cm.

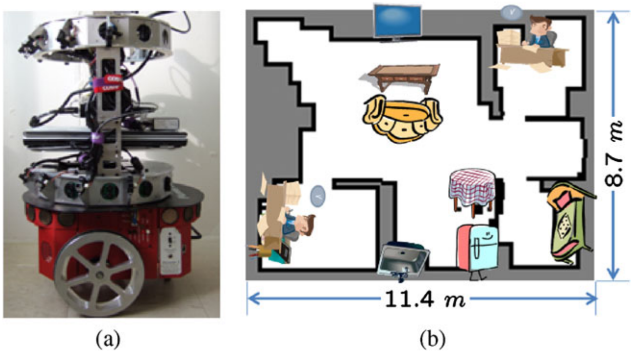


Fig. 12 Experimental setup: (**a**) Pioneer 3-DX with 12 Murata sonar sensors and (**b**) experimental environment

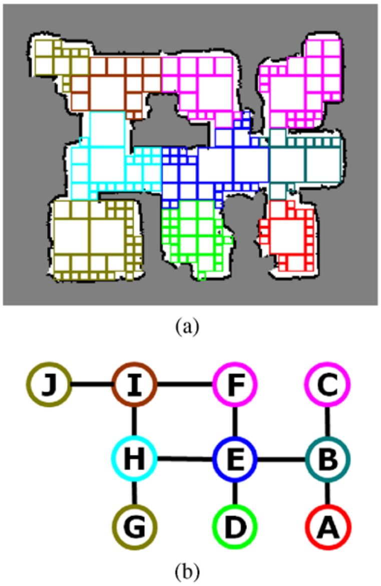


Fig. 13 Experimental result of the offline environmental modeling process: (**a**) clustering result (each cluster is a different color) and (**b**) extracted topology

5.1 Experimental results of topological modeling

Figure 13 shows the experimental results of the offline topological modeling process. The robot visited the complete environment and generated a grid map, dividing the environment into ten subregions corresponding to those shown in Fig. 13(a). The number of subregions was predefined manually in the offline modeling method. The proposed offline

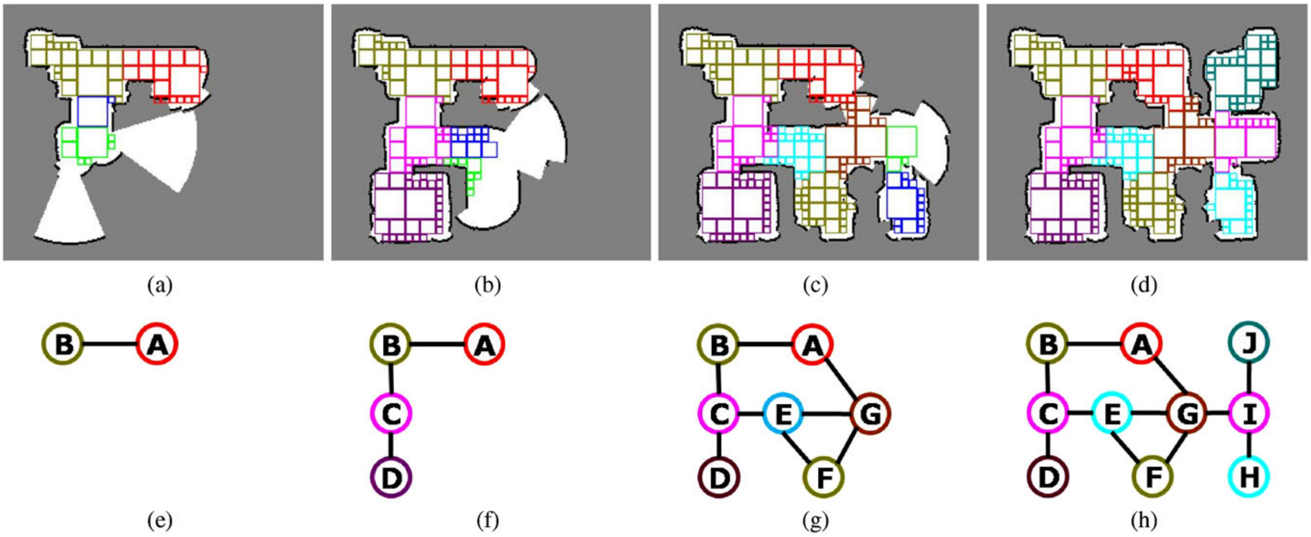


Fig. 14 Experimental results of the autonomous topological modeling process: (**a**)–(**d**) autonomous subregion extractions (each subregion is a different color) and (**e**)–(**h**) the corresponding topological models. The *blue* and *green cells* in (**a**)–(**c**) were not included in the corresponding topological models because they were not yet extracted as new subregions

topological modeling process provided a successful topological representation of the home environment. Three rooms were detected as subregions (nodes A, C, and G) by segmenting around doorways effectively, and the kitchen was modeled as a separate subregion (node D). The remaining area was also divided into several subregions reflecting the geometry of the environment. Figure 13(b) shows a graphical model of the extracted subregions.

The autonomous topological modeling was used under the same experimental conditions. As mentioned, the threshold value c_t in (9) is used as 0.2, and this threshold means that 20% of the occupied grids in the subregion were allowed to extract a convex region. Figure 14 shows the processes of the proposed autonomous topological modeling method. Figures 14(a)–14(d) show the results of obtaining reliable regions and extracting subregions, and Figs. 14(e)–14(h) show the corresponding topological models. The blue and green cells in Figs. 14(a)–14(c) represent tentative subregions, which are not represented in the topological models in Figs. 14(e)–14(g) because they are not yet extracted as new subregions. The experimental results show that the reliable cells in the reliable region were successfully obtained by filtering out the noisy data, and the topological models were constructed effectively by extracting subregions from the grid map as the robot moved.

As a result, the environment was partitioned into ten subregions (Fig. 14(d)) by incremental modeling, and the subregions were successfully assembled into a topological model as shown in Fig. 14(h). The topological model properly represents the environment, as did the offline modeling method. The three rooms were allocated to three different subregions (nodes D, H, and J), and the kitchen was extracted as node F. Areas in the living room were segmented into several subregions because of the sofa and table located in the center of the room.

A comparison of Fig. 14 with Fig. 13 shows that the autonomous modeling method resulted in different subregion extractions than the offline modeling method did, even though the same number of subregions was extracted in both methods. This difference is a natural result of the differing ways in which the two methods determine the minimum normalize cuts. In the offline modeling method, the number of subregions is predefined; because the environment was divided into ten subregions at once, the minimum normalized cuts were determined based on the entire environment, whereas in the autonomous method, each minimum normalized cut was found in the local area incrementally. The normalized cut (N_{cut}) is calculated from the weights within clusters as well as the weights between clusters (1). Therefore, the clustering result is affected by the grid map used in the modeling method. As a result, the subregion extraction of the autonomous method, which uses a local grid map, cannot be identical to that of the offline method, because the autonomous method extracts subregions before generating a grid map of the complete environment. Moreover, the autonomous method also considers the convexity criterion (9), which can account for differences between the offline and autonomous methods, as the convexity criterion determines when new subregions should be extracted regardless of the normalized cut in the autonomous modeling method.

Figure 15 shows the measure in (7) for each extracted node to evaluate the convexity of the extracted subregion. All the subregions had convexity measures less than the threshold value $c_t = 0.2$, and the results confirmed that

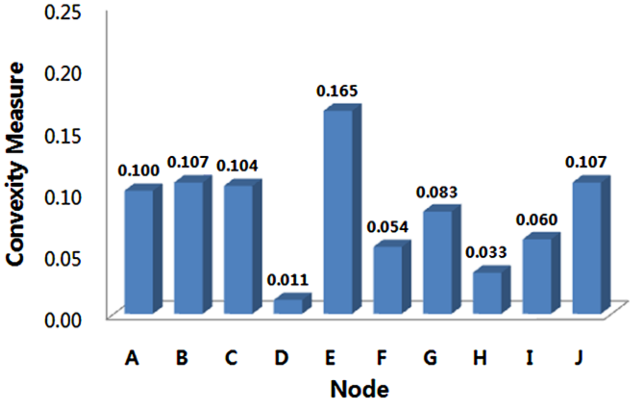


Fig. 15 Convexity measures for extracted subregions

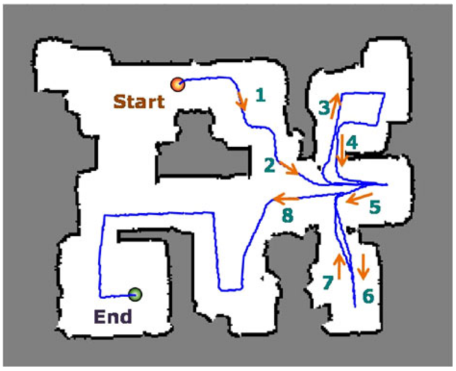


Fig. 16 Robot path for the topological localization in the static environment. The numbers represent the path sequence

the proposed autonomous topological modeling method extracted subregions while guaranteeing the convexity.

Therefore, the proposed method provides a successful topological model using sonar grid maps by extracting subregions autonomously.

5.2 Experimental results of topological localization

Two kinds of experiments were conducted to verify the proposed topological localization method based on the topological model shown in Fig. 14. The values for the standard deviations, σ_d and σ_a, used in (18) and (20), were 50 cm (10 grids) and 15°, respectively.

5.2.1 Static environment

The first experiment was performed without changing the environment after the modeling procedure. The robot navigated the path shown in Fig. 16 while acquiring sensor data around the current location and generated local grid maps to perform the topological localization.

Figure 17 shows the local grid map that was generated and the corresponding template grid map generated from node A at the first time step. The template grid map was successfully extracted by filtering out the noisy data in the local grid map using the algorithm described in Sect. 4.1. Figure 18 shows the ten candidate locations obtained from nodes A–J using rotational invariant grid-map matching with the extracted template grid map. Of all the candidate locations, Fig. 18(a), which is the true hypothesis because the template grid map was generated in node A, has the largest number of geometric shapes in common with the template grid map.

Figure 19 shows the experimental results of topological localization with six data sets from among all those captured. The upper row in Fig. 19 shows the generated template grid maps and the matching results. The matched grid maps were obtained from the candidate locations that had maximum node probability at each time step. The matching results show that the proposed method provided reliable grid-map matching even in the presence of noisy sensor data.

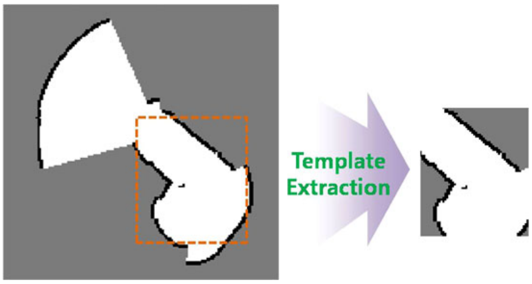


Fig. 17 Local grid map and corresponding template grid map at the first time step

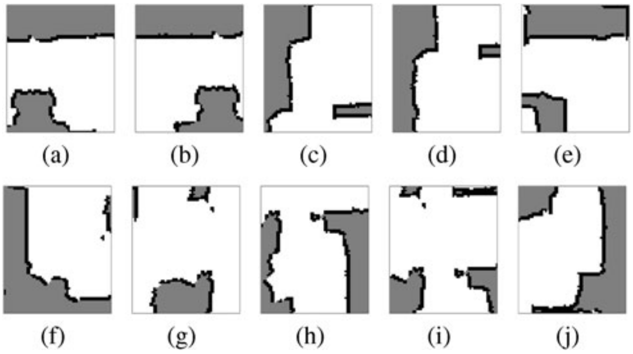
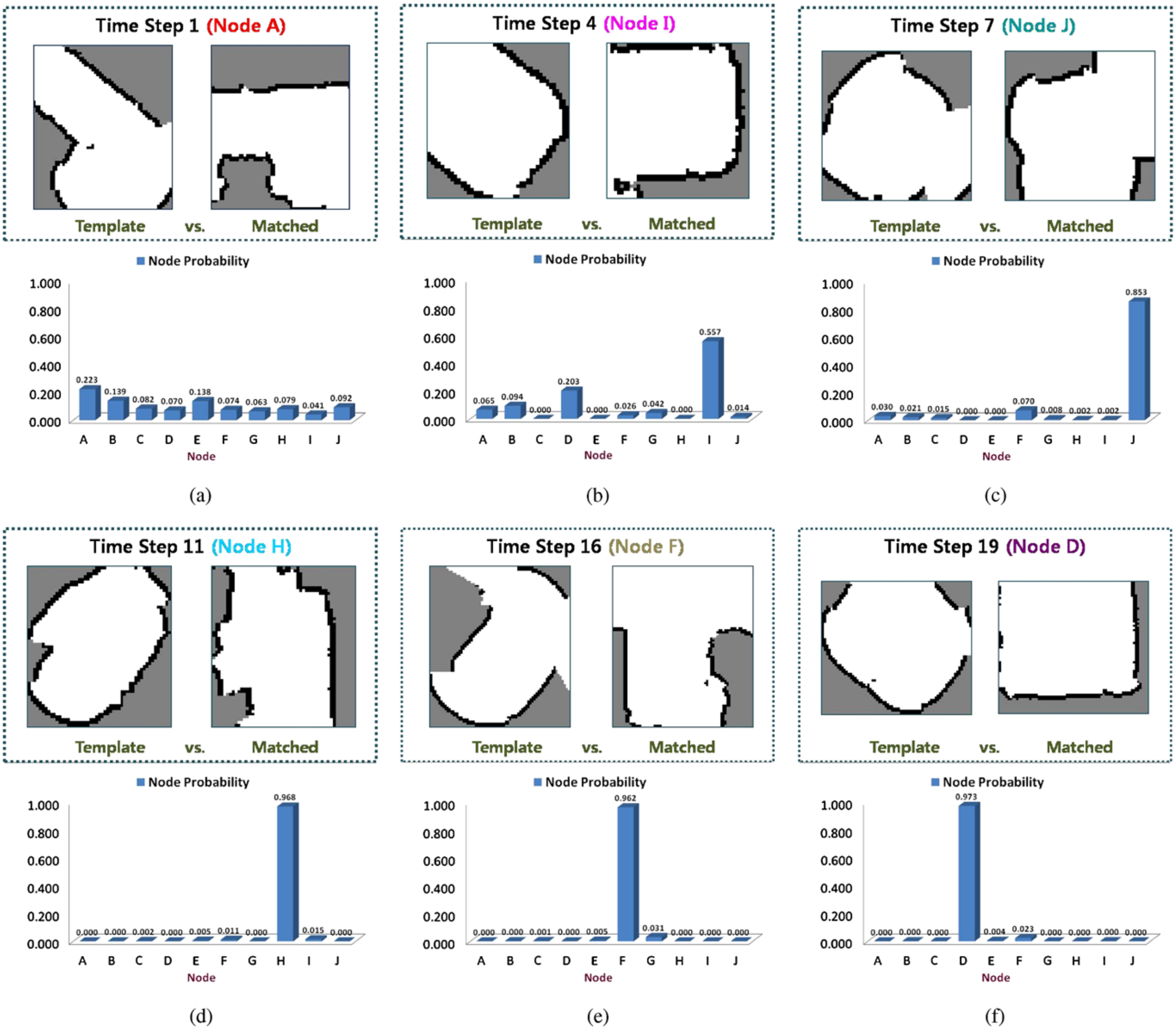


Fig. 18 Ten candidate locations for the template grid map of Fig. 17

The lower row in Fig. 19 shows the corresponding node probabilities. Because no prior information was available at time step 1, the node probability was obtained using only the grid-map matching probability as shown in Fig. 19(a). As the robot moved, the node probability was updated using the proposed motion model and grid-map matching. As a result, the proposed topological localization exhibited a convergent node probability as the node probability was updated.



(a) (b) (c)

(d) (e) (f)

Fig. 19 Experimental topological localization results in the static environment. The grid-map matching result and the corresponding node probability are shown for each time step

Figure 20 shows the maximum node probability at each time step to evaluate the convergence of the node probability. The maximum node probability tends to increase as the node probability is updated. At time steps 5 and 12, the local grid maps could not provide successful observations because of an insufficient accumulation of sensor data. The entropy increases as shown in Fig. 21 because of the inadequate local grid maps. At those time steps, the node probability was not updated, and the local grid map was simply expanded by accumulating more sensor data. In Fig. 20, the node probabilities that were not updated are represented by a different color. Except for time steps 5 and 12 when the node probability was not updated, the maximum node probability increased and the entropy decreased as the node probability was updated. These results confirm the conver-

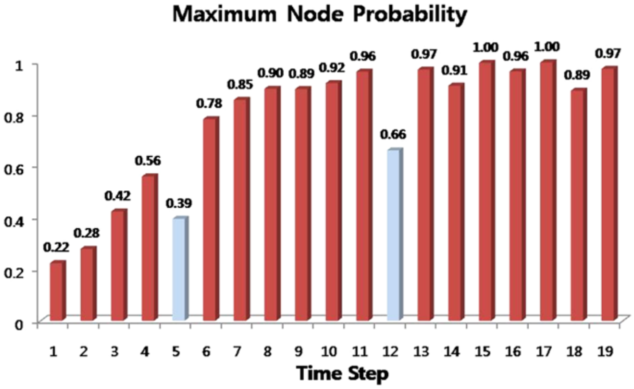


Fig. 20 Maximum node probability in the static environment. The node probability at time steps 5 and 12, represented by a different color, is not updated because the entropy increases

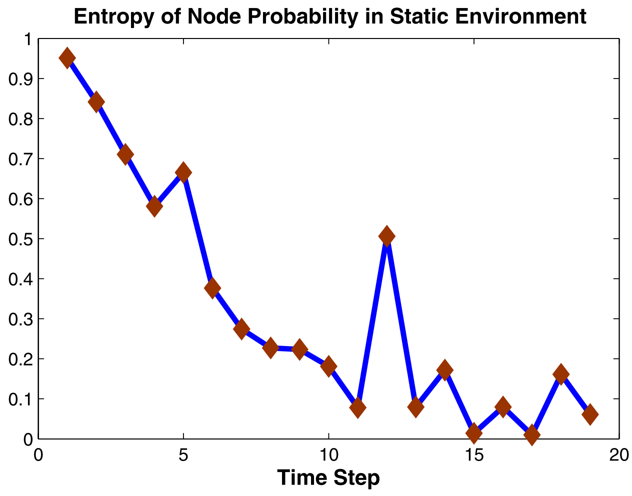


Fig. 21 Entropy of node probability in the static environment

gence of the node probability in the proposed localization method.

Therefore, the experimental results of topological localization show that the proposed method provided reliable grid-map matching and node-probability update.

5.2.2 *Dynamic environment*

The second experiment was performed after making small environmental changes once the modeling procedure was complete, as shown in Fig. 22: a 70 cm × 70 cm sofa was removed, an open door was closed, and a chair was removed. The robot navigated the path shown in Fig. 23 in this changed environment.

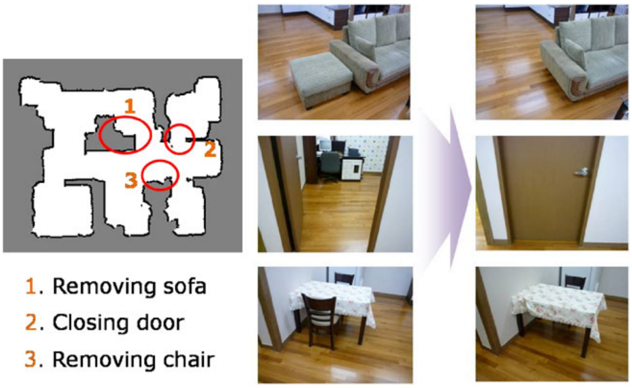


Fig. 22 Dynamic changes in the second topological localization experiment

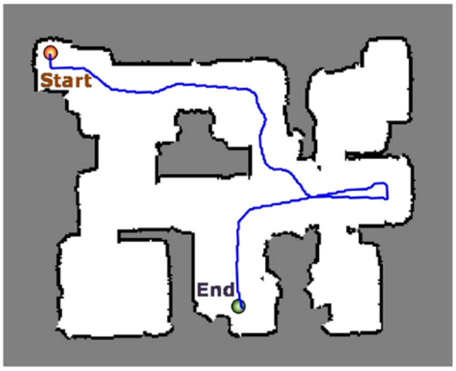
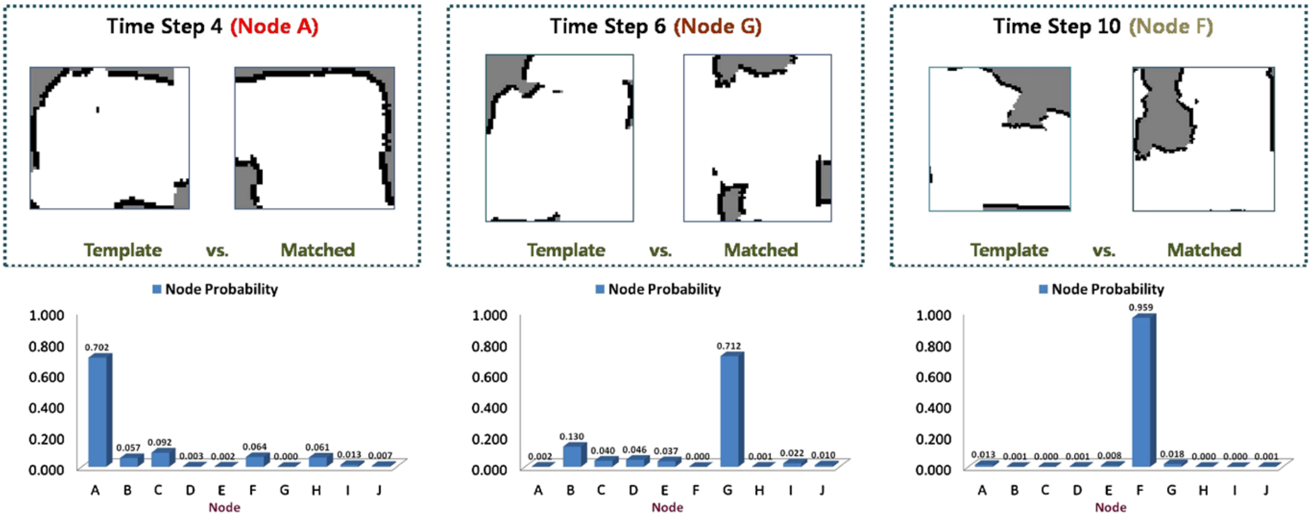


Fig. 23 Robot path for the topological localization in the dynamic environment

Figure 24 shows the localization results for three sampled data sets. Figure 25 and Fig. 26 show the node probability of



(a) (b) (c)

Fig. 24 Experimental results of topological localization in the dynamic environment. The grid-map matching result and the corresponding node probability are shown for each time step

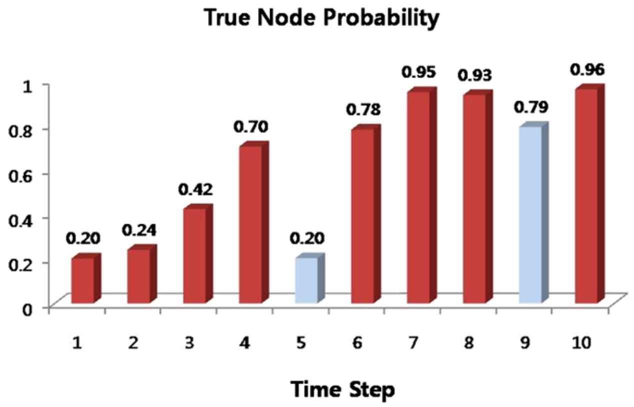


Fig. 25 Node probability of true hypothesis in the dynamic environment. The node probability at time steps 5 and 9, represented by a different color, was not updated because the entropy increased

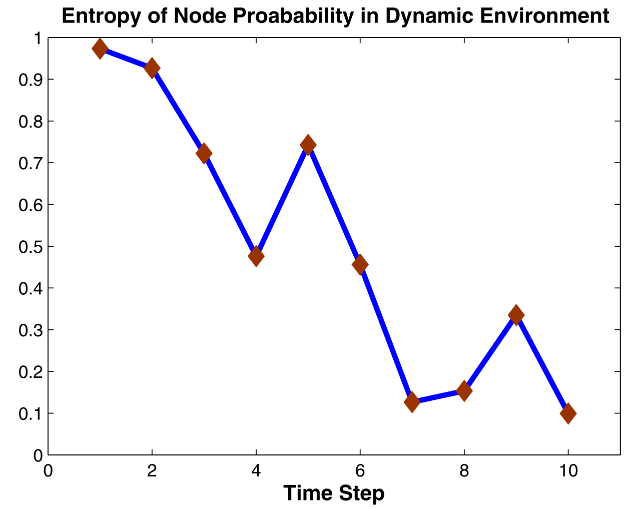


Fig. 26 Entropy of node probability in the dynamic environment

the true hypothesis and the entropy of the node probability, respectively, for all data sets. The entropy increased at time steps 5 and 9, and these increasing entropy values were due mainly to the changes in the environment. These changes affected the local grid maps for those time steps as shown in Fig. 27. Because of the sofa and chair that had been removed, the candidate location for the true hypothesis could not be obtained accurately, resulting in a dramatic decrease of the true node probability.

Even though the node probability was affected by the dynamic changes, the proposed localization method can recover the reliable node probability in subsequent updates. The false updates were filtered out using the entropy test, and the node probability was updated using more reliable local grid maps. As a result, the node probability can provide more reliable results, as shown in Fig. 24(b) and 24(c).

The experimental results confirmed the good performance of the proposed topological localization method in a real home environment. The method works well in a static environment and can also handle small dynamic changes through the Bayesian update using the approximate motion model, grid-map matching, and entropy test.

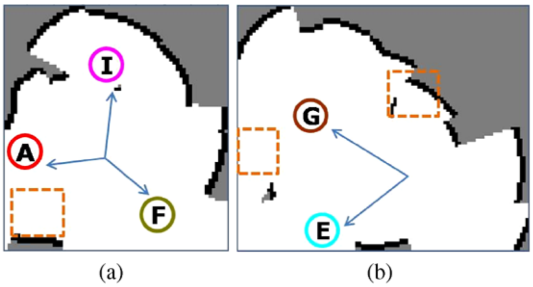


Fig. 27 Local grid maps around (**a**) node G at time step 5 and (**b**) node F at time step 9. The *dashed boxes* represent the sofa and chair that had been removed

6 Conclusions

This paper described a method for autonomous topological modeling and localization using only low-cost sonar sensors. The proposed topological approach was developed to be appropriate for a home environment. The proposed method provided reliable modeling and localization results using sparse and noisy sonar data.

The autonomous topological modeling method generates a topological model that reflects the spatial geometry of the environment. By applying cell decomposition and normalized graph cut, the topological model can be acquired by autonomously partitioning the grid map into several subregions without predefining the number of subregions. Furthermore, the convexity criterion guarantees that the extracted subregions can be regarded as convex regions.

The topological localization method provides reliable results even in the face of the noisy sonar data and small dynamics. The proposed localization gives a convergent node probability, using rotational invariant grid-map matching and the approximate relative motion model, which prevents an accumulation of odometry error. Moreover, the size of the local grid map is determined adaptively by checking the entropy of the node probability. The entropy test also guarantees convergent localization results even in the face of small dynamic changes.

The proposed method can be used as a basic framework for human-robot interaction to use high-level semantic information in mobile robot navigation. By providing semantic information to the extracted subregions, the robot may recognize the environment as a human would.

Although the proposed method was developed for sonar sensors, it can also be applied to any type of sensor that generates grid maps (e.g., laser range finders or stereo vision sensors).

Acknowledgements This work was supported in part by the Acceleration Research Program of the Ministry of Education, Science and Technology of the Republic of Korea and the National Research Foundation of Korea [R17-2008-021-01000-0], in part by the Agency for Defense Development and by Unmanned Technology Research Center (UTRC), Korea Advanced Institute of Science and Technology, and in part by the IT R&D program of MKE/IITA [2008-F-038-1, Development of Context Adaptive Cognition Technology].

References

Beeson, P., Jong, N. K., & Kuipers, B. (2005). Towards autonomous topological place detection using the extended Voronoi graph. In *Proc. of IEEE international conference on robotics and automation* (pp. 4373–4379).

Brunskill, E., Kollar, T., & Roy, N. (2007). Topological mapping using spectral clustering and classification. In *Proc. of IEEE/RSJ international conference on intelligent robots and systems* (pp. 3491–3496).

Buschka, P., & Saffiotti, A. (2002). A virtual sensor for room detection. In *Proc. of IEEE/RSJ international conference on intelligent robots and systems* (pp. 637–642).

Choi, J., Ahn, S., & Chung, W. K. (2005). Robust sonar feature detection for the SLAM of mobile robot. In *Proc. of IEEE/RSJ international conference on intelligent robots and systems* (pp. 3415–3420).

Choi, J., Choi, M., Lee, K., & Chung, W. K. (2009a). Topological modeling and classification in home environment using sonar gridmap. In *Proc. of IEEE international conference on robotics and automation* (pp. 3892–3898).

Choi, J., Choi, M., & Chung, W. K. (2009b). Incremental topological modeling using sonar gridmap in home environment. In *Proc. of IEEE/RSJ international conference on intelligent robots and systems* (pp. 3582–3587).

Choset, H., & Nagatani, K. (2001). Topological simultaneous localization and mapping (SLAM): Toward exact localization without explicit localization. *IEEE Transactions on Robotics and Automation*, *17*(2), 125–137.

Doh, N. L., Lee, K., Chung, W. K., & Cho, H. (2009). Simultaneous localisation and mapping algorithm for topological maps with dynamics. *IET Control Theory and Applications*, *3*(9), 1249–1260.

Elfes, A. (1989). Using occupancy grids for mobile robot perception and navigation. *IEEE Computer*, *22*(6), 46–57.

Gonzalez, R. C., & Woods, R. E. (2002). *Digital image processing* (2nd ed.). New Jersey: Prentice Hall.

Gutmann, J.-S., & Konolige, K. (1999). Incremental mapping of large cyclic environments. In *Proc. of IEEE international symposium on computational intelligence in robotics and automation* (pp. 318–325).

Katevas, N. I., Tzafestas, S. G., & Pnevmatikatos, C. G. (1998). The approximate cell decomposition with local node refinement global path planning method: Path nodes refinement and curve parametric interpolation. *Journal of Intelligent and Robotic Systems*, *22*(3–4), 289–314.

Kleeman, L., & Kuc, R. (2008). Sonar sensing. In B. Siciliano & O. Khatib (Eds.), *Handbook on robotics*. Berlin: Springer.

Lee, K., Cho, N., Chung, W. K., & Doh, N. L. (2006). Topological navigation of mobile robot in corridor environment using sonar sensor. In *Proc. of IEEE/RSJ international conference on intelligent robots and systems* (pp. 2760–2765).

Lee, K., & Chung, W. K. (2009). Effective maximum likelihood grid map with conflict evaluation filter using sonar sensors. *IEEE Transactions on Robotics*, *25*(4), 887–901.

Leonard, J. J., & Durrant-Whyte, H. F. (1991). Simultaneous map building and localization for an autonomous mobile robot. In *Proc. of IEEE/RSJ international conference on intelligent robots and systems* (pp. 1442–1447).

Lin, Y., Chen, C., & Wei, C. (2006). New method for subpixel image matching with rotation invariance by combining the parametric template method and the ring projection transform process. *Optical Engineering*, *45*(6), 067 202(1-9).

Mozos, O. M., & Burgard, W. (2006). Supervised learning of topological maps using semantic information extracted from range data. In *Proc. of IEEE/RSJ international conference on intelligent robots and systems* (pp. 2772–2777).

Remolina, E., & Kuipers, B. (2004). Towards a general theory of topological maps. *Artificial Intelligence*, *152*, 47–104.

Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(8), 888–905.

Tapus, A., & Siegwart, R. (2006). A cognitive modeling of space using fingerprints of places for mobile robot navigation. In *Proc. of IEEE international conference on robotics and automation* (pp. 1188–1193).

Tardós, J. D., Neira, J., Newman, P. M., & Leonard, J. J. (2002). Robust mapping and localization in indoor environments using sonar data. *International Journal of Robotic Research*, *21*(4), 311–330.

Thrun, S. (1998). Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, *99*(1), 21–77.

Thrun, S., Fox, D., Burgard, W., & Dellaert, F. (2001). Robust Monte Carlo localization for mobile robots. *Artificial Intelligence*, *128*(1–2), 99–141.

Thrun, S. (2002). Robotic mapping: A survey. In G. Lakemeyer, & B. Nebel (Eds.), *Exploring artificial intelligence in the new millennium*. San Mateo: Morgan Kaufmann.

Yap, T. N., & Shelton, C. R. (2009). SLAM in large indoor environments with low-cost, noisy, and sparse sonars. In *Proc. of IEEE international conference on robotics and automation* (pp. 1395–1401).

Yun, Y., Park, B., & Chung, W. K. (2008). Odometry calibration using home positioning function for mobile robot. In *Proc. of IEEE international conference on robotics and automation* (pp. 2116–2121).

Zivkovic, Z., Bakker, B., & Krose, B. (2006). Hierarchical map building and planning based on graph partitioning. In *Proc. of IEEE international conference on robotics and automation* (pp. 803–809).

Jinwoo Choi received his B.S. and M.S. degree in Mechanical Engineering from Pohang University of Science and Technology (POSTECH), Korea, in 2003 and 2005, respectively. He is currently a Ph.D. candidate in Mechanical Engineering from POSTECH, Korea. His current research interests are mainly concentrated on mapping, localization, SLAM, and robust sonar features.

Minyong Choi received his B.S. and M.S. degree in Mechanical Engineering from Pohang University of Science and Technology (POSTECH), Korea, in 2003 and 2005, respectively. He is currently a Ph.D. candidate in Mechanical Engineering from POSTECH, Korea. His current research interests are mainly concentrated on mapping, localization, and visual SLAM.

Sang Yep Nam received the B.S., M.S., and Ph.D. degrees in electronics engineering from Dankook University, Korea, in 1982, 1984, and 2002, respectively. He is a Professor of the Department of Information & Communication of Kook Je College (KJC), Korea. He has an interest in embedded systems, RFID/USN, speech recognition systems, medical image processing systems, factory automation, and ubiquitous robotic companions.

Wan Kyun Chung received his B.S. degree in Mechanical Design from Seoul National University in 1981, his M.S. degree in Mechanical Engineering from KAIST in 1983, and his Ph.D. in Production Engineering from KAIST in 1987. He is a professor in the school of Mechanical Engineering, POSTECH (he joined the faculty in 1987). In 1988, he was a visiting professor at the Robotics Institute of Carnegie-Mellon University. In 1995 he was a visiting scholar at the university of California, Berkeley. His research interests include the localization and navigation of mobile robots, underwater robots and development of robust controllers for precision motion control. He is a director of the National Research Laboratory for Intelligent Mobile Robot Navigation. He is serving as an Editor for IEEE Trans. on Robotics, and he is on the international editorial board for Advanced Robotics.