Mobile Robot Exploration in Indoor Environment Using Topological Structure with Invisible Barcodes

Jinwook Huh, Woong Sik Chung, Sang Yep Nam, and Wan Kyun Chung

This paper addresses the localization and navigation problem in the movement of service robots by using invisible two dimensional barcodes on the floor. Compared with other methods using natural or artificial landmarks, the proposed localization method has great advantages in cost and appearance since the location of the robot is perfectly known using the barcode information after mapping is finished. We also propose a navigation algorithm which uses a topological structure. For the topological information, we define nodes and edges which are suitable for indoor navigation, especially for large area having multiple rooms, many walls, and many static obstacles. The proposed algorithm also has the advantage that errors which occur in each node are mutually independent and can be compensated exactly after some navigation using barcodes. Simulation and experimental results were performed to verify the algorithm in the barcode environment, showing excellent performance results. After mapping, it is also possible to solve the kidnapped robot problem and to generate paths using topological information.

Keywords: Invisible barcode, mobile robot, navigation, topological mapping.

Manuscript received Mar. 18, 2006: revised Dec. 25, 2006.

This work was supported in part by the Ministry of Health and Welfare, Republic of Korea, under grant (02-PJ3-PG6-EV04-0003), by the International Cooperation Research Program (M6-0302-00-0009-03-A01-00-004-00) of the Ministry of Science and Technology, Republic of Korea, and by the National Research Laboratory (NRL) Program (M1-0302-00-0040-03-J00-00-024-00) of the Ministry of Science and Technology, Republic of Korea.

Jinwook Huh (phone: +82 42 821 2837, e-mail: jwhuh@postech.ac.kr) is with 1st R&D Center, Agency for Defense Development, Daejeon, Korea.

Woong Sik Chung (e-mail: newkid72@microrobot.co.kr) is with the Research Center, Microrobot Co., Ltd.

Sang Yep Nam (email: r13337@unitel.co.kr) is with Kook Je College, Gyeonggi-do, Korea.

Wan Kyun Chung (e-mail: wkchung@postech.ac.kr) is with the School of Mechanical Engineering, Pohang University of Science and Technology, Pohang, Korea.

I. Introduction

In recent years, research and development for service robots has been actively pursued. For example, many commercial vacuum cleaning robots have been introduced [1]. However, commercial vacuum cleaning robots are still inefficient, because their algorithms do not solve the localization problem perfectly. Over the past decades, a considerable number of algorithms for navigation and path planning such as the extended Kalman filter (EKF) or the particle filter in simultaneous localization and mapping (SLAM) have been proposed [2]-[4]. However, these algorithms are not easy to implement in commercial robots because commercial robots use imprecise sensors for economic reasons. Moreover, commercial robots cannot process these algorithms which require large computational burdens. To solve localization problems, some robots use artificial landmarks [5], [6]. Using artificial landmarks is more accurate and creates less computation burden than using natural landmarks. However, it requires special treatment of the environment in the walls or ceilings, and there is additional cost. Hernaóndez [6] suggested a system which was designed for a closed location with a maximum size of 20 m × 20 m and has a maximum error of 5 cm in the robot position and 1 degree in the orientation. However, this system is expensive and the height of the receiving device must be around the position of the emitter.

Evolution Robotics [7] recently developed NorthStar. The NorthStar detector uses triangulation to measure position and heading in relation to IR light spots which can be projected onto the ceiling. It is not expensive; its cost is about 80 US dollars. The heading error is less than 2 degrees and the position error is less than 15 cm, but the detector can localize only when it detects light spots on the ceiling. For this reason, it

is hard to localize in a room where there is no projector.

In this paper, we propose artificial landmarks which are suitable for an indoor environment. Our artificial landmarks are invisible 2-dimensional barcodes on the flooring surface. Our barcode system is more accurate and cheaper than the Northstar system. Although there is the additional cost for printing barcodes on the flooring surface, it is very cheap. The cost of ink is several cents for approximately 10 m of flooring and one major floor manufacturing company in Korea provides the barcoded floor. The barcodes are not normally visible, but can be seen under ultraviolet (UV) illumination. The barcodes give information regarding position and the relative angle between the robot and barcodes. The maximum error is 3.75 cm in the robot position and 1 degree in the orientation, since the barcodes exist at every 3.75 cm in x and y directions. As expected, the proposed system shows remarkable results in solving localization problems.

Based on accurate localization information, we can develop an algorithm which improves the performance of previous navigation algorithms due to the excellent performance of the localization method. In general, an indoor domestic environment has one living room and several other rooms. The rooms may be divided by walls or other static partitions. There are several obstacles such as furniture in most rooms. In particular, a robot must pass a small area such as a doorway to move to other regions; therefore, we need to divide the indoor environment into several regions. However, most previous studies have handled the indoor environment as just one region. There have been several attempts to decompose the indoor environment [8], [9] or to make a topological map [10]. However, the decomposition in these algorithms was dependent on obstacle shapes, and these algorithms are a little inefficient due to the fact that the indoor environment is decomposed into many regions. Many other classical methods for the topological representation of the environment have been proposed [11], including splitting the global reference frame into several local frames [12] and hybrid representation [13]. However, these algorithms use laser sensors; therefore, they cannot be applied to commercial robots.

There have also been some attempts to compose topological structures using cheap sonar sensors [14], [15]. In this paper, topology-based maps can be automatically extracted from the data of an occupancy grid built from 16 Polaroid ultrasonic sensors using techniques borrowed from the image processing field. However, this topology-based map can only be obtained after building an occupancy grid map. Therefore, this algorithm differs from ours, in which topological information is automatically obtained when navigation and mapping are performed.

We propose a new navigation rule and definitions of *node* and *edge* which are suitable for topological representation of the indoor environment. They can also be applied to commercial robots, because the algorithm works well using cheap distance sensors. In comparison with previous algorithms, our navigation algorithm has several advantages including the easy solution of the kidnap problem, as will be discussed later.

The remainder of this paper is organized as follows: In section II, we explain the overall system. Section III describes the localization method of our system. In section IV we suggest an algorithm to improve navigation performance in indoor environments, the experimental results are explained in section V, and the conclusion follows.

II. System Description

Although carpets are common in indoor environments in Europe and America, hard flooring is popular in many countries. Our system can only be used in hard-flooring environments.

1. Flooring

1) The appearance of the flooring used in our system is exactly the same as common flooring when viewed under normal lighting conditions (Fig. 1(a)). However, when the barcodes are illuminated by UV light, they are revealed on the surface as shown in Fig. 1(b).
2) The flooring is composed of 7.5 cm × 90 cm pieces produced by Hanwha L&C Corp. Each piece has barcodes which are placed in an array of two columns and twenty four rows.
3) Each barcode has information of x value and y value. The x value in the first column is a random even value, and the x value in the second column is the value of one added to the x value in the first column. The y value monotonically increases or decreases from 0 to 23 in the same piece. When a robot reads a barcode, it knows its relative heading angle with respect to a given barcode (θ) and its relative distance from the barcode.
4) There is no rule for the placement of each piece except that the pieces are laid with a y-directional interval of three barcodes. Inevitably, it is possible for identical barcodes to exist in nearby regions.

Barcoded flooring has the following advantages. First, it does not incur additional cost except the cost of the ink to print the barcodes. Second, it is invisible, so it has the natural appearance of wood unlike other artificial landmarks. Third, it provides enormous information for localization, because barcodes exist at

Fig. 1. System description : (a) robot and flooring and (b) barcodes made visible by UV illumination.

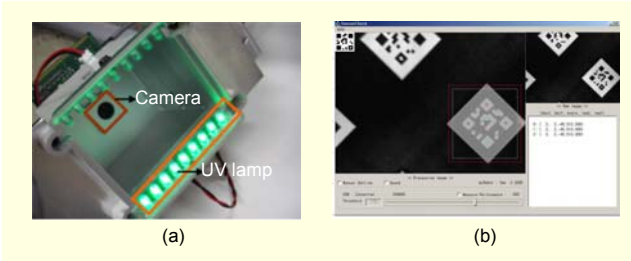

Fig. 2. (a) Barcode reader system and (b) captured image.

every 3.75 cm. However, the barcodes do not give the absolute location of the robot because the flooring pieces are randomly placed at the time of installation (except that the pieces are laid with a y-directional interval of three barcodes). For this reason, we need mapping and localization to provide relations between pieces. However, we need to point out here that the barcodes can be damaged or the barcode reader can give a noisy signal. To handle these situations, we need a robust localization algorithm.

2. Barcode Reader

As shown in Fig. 2(a), the barcode reader has a CMOS camera and a UV lamp. When the barcodes on the flooring are illuminated by UV light they are revealed. A camera reads the revealed barcode information: the x and y values. The robot can calculate the relative heading angle between its heading direction and the barcode center line; it can also read the relative distance between the barcode center point and the image center point (Fig. 2(b)).

3. Platform and Sensor

Our platform has two wheels, a bumper sensor, a UV lamp, a barcode reader, and Sharp IR distance sensors (five GP2D120 and two GP2YOA02YK) as shown Fig. 1(a). The GP2D120 and GP2YOA02YK sensors use triangulation to detect distance. Their measurement distances are 4 to 30 cm and 20 to 150 cm, respectively. They cost less than $10 each.

III. Localization

For localization, we use barcodes and odometry data. There are two steps in the localization process, prediction and correction. The prediction step is the procedure to predict the current barcode value only using odometry data. The correction step is the procedure to correct the current robot position using the predicted current barcode value and the real current barcode value as shown in Fig. 3. The basic terms used in this section are the following:

(O_x, O_y): odometry data
(B_{px}, B_{py}): previous barcode x, y value
(B_{cx}, B_{cy}): current barcode x, y value
D_{Gx}, D_{Gy}: interval between barcodes (3.75 cm)
N_{px}, N_{py}: predicted number of passed barcodes in x, y direction
(d_{px}, d_{py}): distance from previous barcode to previous robot center position
(d_{cx}, d_{cy}): distance from current barcode to current robot center position
L_x, L_y: distance between previous barcode and predicted barcode
M_{px}, M_{py}: predicted current barcode mod
M_{cx}, M_{cy}: real current barcode mod
(x_p, y_p): previous robot position
(x_c, y_c): corrected current robot position

Both N_{px} and N_{py} are integer values. The greatest integer that is less than or equal to x is denoted as $\lfloor x \rfloor$. The modulo operation denoted as a mod m returns the remainder on division of a by m.

First, we calculate N_{px} and N_{py} only using odometry data as

$$N_{px} = \left\lfloor \frac{O_x + d_{px}}{D_{Gx}} \right\rfloor, \quad N_{py} = \left\lfloor \frac{O_y + d_{py}}{D_{Gy}} \right\rfloor. \tag{1}$$

If N_{px} is a negative value, it means that the robot moves in the negative direction. At each step, O_x measures the odometric displacement in the x direction; therefore, if the robot moves in the negative direction, O_x is a negative value and N_{px} is also negative value.

Because the flooring pieces are laid with a y-directional interval of 3 barcodes as shown in Fig. 3, the y value of the predicted current barcode increases or decreases three values whenever crossing a border. Thus, the predicted current barcode mod is

$$M_{px} = (B_{px} + N_{px}) \bmod 2. \tag{2}$$

In case B_{px} mod 2 = 1,

$$M_{py} = \left(B_{py} + N_{py} + 3 \times \left\lfloor \frac{N_{px} + 1}{2} \right\rfloor \right) \bmod 24. \tag{3}$$

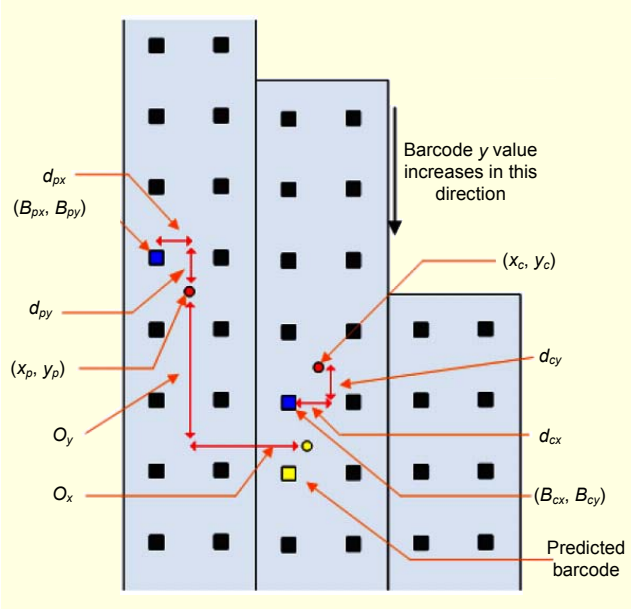

Fig. 3. Correction of the robot position.

In case B_{px} mod 2 = 0,

$$M_{py} = \left(B_{py} + N_{py} + 3 \times \left\lfloor \frac{N_{px}}{2} \right\rfloor \right) \bmod 24. \quad (4)$$

Next, the correction step corrects the current robot position using the predicted current barcode value and the real current barcode. First, we calculate the distance between the previous barcode and the predicted current barcode using d_{px}, d_{py} and odometry data.

$$L_x = \left\lfloor \frac{O_x + d_{px}}{D_{Gx}} \right\rfloor \times D_{Gx}, \quad L_y = \left\lfloor \frac{O_y + d_{py}}{D_{Gy}} \right\rfloor \times D_{Gy}. \quad (5)$$

The barcode reader gives the x, y value of the current barcode and d_{cx}, d_{cy} which are the distances from the current reading barcode to the robot center. The current barcode *mod* is obtained as

$$M_{cx} = B_{cx} \bmod 2, \quad (6)$$

$$M_{cy} = B_{cy} \bmod 24. \quad (7)$$

Now, we can calculate corrected robot position:

$$x_c = x_p + L_x + (d_{cx} - d_{px}) + \alpha, \quad (8)$$

$$y_c = y_p + L_y + (d_{cy} - d_{py}) + \beta. \quad (9)$$

In (8) and (9), α and β compensate the difference between the predicted barcode and the real barcode. Table 1 shows the values of α and β.

In the case that some area of the floor is occupied by paper or carpet, when the robot passes onto the occupied floor, the robot accumulates odometry information without updating its current position.

Table 1. Values α and β for updating robot position.

$M_{px} - M_{cx}$	$M_{py} - M_{cy}$	α	β
0	0	0	0
	1	0	D_y
	-1	0	$-D_y$
1 ($M_{px} = 0$)	0	D_x	0
	1	D_x	D_y
	-1	D_x	$-D_y$
	-3	$-D_x$	0
	-4	$-D_x$	$-D_y$
	-2	$-D_x$	D_y
1 ($M_{px} = 1$)	3	D_x	0
	4	D_x	D_y
	2	D_x	$-D_y$
	-1	$-D_x$	$-D_y$
	0	$-D_x$	0
	1	$-D_x$	D_y

In this environment, mapping is necessary to describe the relationship between flooring pieces. After mapping, the robot has barcode information of the whole region, thus the robot can recover from having been kidnapped. It can also represent the indoor environment using the topological structure. Mapping can be done using the algorithm which will be explained in the next section.

IV. Navigation

For an indoor environment having several rooms, each room is enclosed with walls; thus, the robot should pass through a doorway to move to other rooms. For this reason, we create a topological map which includes information about the relative location of rooms. Many papers explain the combination of topological and metric mapping. Some researchers [14], [15] have tried to use the features of a room. However, in navigation, there has been no trial to pass a robot through a doorway to move to another room. We offer a new edge definition that is suitable for this function. This topological representation of the indoor environment was an advantage in computation. We only need to load the local map instead of the full map during navigation. In this section, we explain how to define nodes and edges and how to compose a topological map and a local map. We also explain how to obtain barcode information of the whole area. Because the robot has barcode information of the whole area after mapping, the robot can

localize itself in any location.

1. Definition of Node and Edge

For mapping and navigation, we assume the following:

Assumption 1. Navigation and Mapping

1) The robot can move between rooms: There can be no doorsill or closed door.
2) The flooring of the room is covered with invisible barcodes: Localization is performed using odometry and barcode data.
3) The size of the region to build a map is limited to an indoor environment.
4) Obstacles cannot move while robot is doing mapping: Obstacles are allowed to move after mapping, but there can be no dynamic obstacle while building a map.
5) There is no case in which the robot is unable to cross into another region because of obstacles.

In a general topological structure, an edge is a connection of nodes which are distinct places. However, we will recharacterize what constitutes an edge first, because edges are more important than nodes in our topological map. An edge is a way to other nodes. The explicit characteristics of an edge are as follows.

Definition 1. Edge

An edge is the position between static obstacles which connects two regions. This corresponds to, for example, the location of a doorway connecting two regions. More specifically, an edge can be defined as

1) the empty area between static obstacles where a robot can pass,
2) the width W between static obstacles:

$$\frac{3}{2}R < W < L, \quad (\text{R : radius of robot, L : door width}), \text{ and}$$

3) the connection between nodes.

Door width is standardized in countries throughout the world, and it is between 0.7 m and 1.5 m in most countries; therefore we choose 1.5 m for the value of L. There are two types of edges. Type A exists to the left or right when a robot follows a wall. Type B exists in front of a robot (Fig. 4). The data of edges includes indexes of the start node and end node, the starting position and ending position, and the relative position between the start node and end node. The edge defined above has the following properties:

Property 1. Edge

1) Both terminations of an edge are nodes: The termination of an edge must be linked to a node and nodes linked to the edge must not be the same.
2) An edge which has the same indices of linked nodes and the same position is the same edge.
3) There can be many edges which link two regions.

Definition 2. Node

A node is a region linked to other node by edges and surrounded by static obstacles or walls.

The data of a node includes the node index, the data of linked edges and the local grid map. A node can be considered as room, thus an indoor space can have several nodes. The reason each node has its own grid map is to prevent the accumulation of errors in each node. If a robot has one total grid map, errors affect the whole region. On the other hand, when the robot has a grid map of each node separately, errors in each node are mutually independent. A node has the following properties.

Property 2. Node

The region of a node does not overlap with other nodes. Property 2 can be used to solve the cyclic problem.

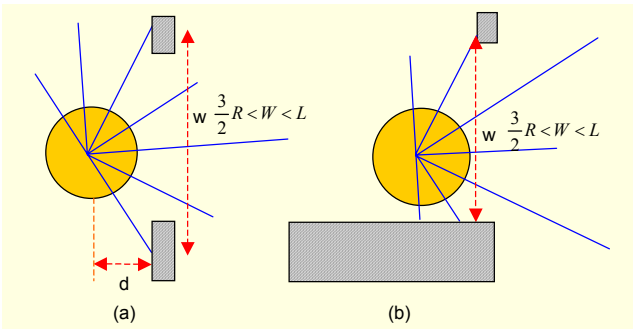

Fig. 4. Edge types: (a) type A (b) type B.

2. Finding a New Edge

Because the measurement distances of sensors using IR are not long, a robot would not be able to sense the whole area simultaneously. For this reason, a robot should basically follow walls at first. While a robot is following the walls it can detect edges.

3. Creation, Elimination, and Merging of Edges

We defined nodes and edges as in previous section; however, there are many cases in which the edge definition does not clearly apply (Fig. 5).

Figure 5(a) illustrates a case in which there are two edges to cross to another node. For example, in some cases there are two doors between two rooms. In such a case, the robot creates two edges. In Fig. 5(b), there is one door between rooms, but the robot detects two edges because of an obstacle. This case is not a problem, because the robot can cross to another node using any

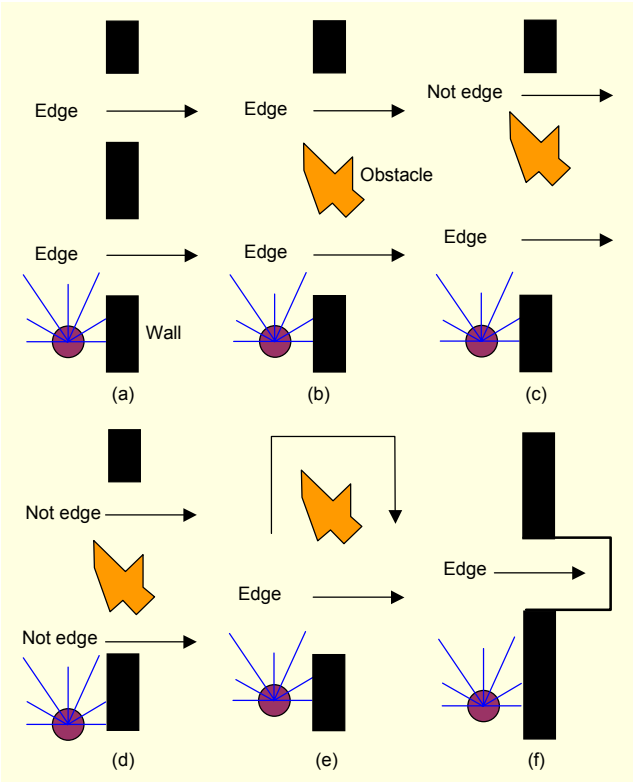

Fig. 5. Exceptional cases in edge mapping.

edge. In Fig. 5(c), there is one door, and there is no difference except a slight change in the starting position of the edge. In Fig. 5(d), there is one door, but the robot can not detect the edge because of an obstacle. In assumption 1, we stated that there cannot be a case in which a robot is unable to cross to another region because of obstacles. In Fig. 5(d), the robot cannot move to the other node. This contradicts assumption 1, as the robot can not move to the other node; therefore a case such as that shown in Fig. 5(d) should not occur during mapping.

Figure 5(e) and (f) shows exceptional cases which occur in mapping. The case shown in 5(e) is regarded as having edges between the obstacle and the wall. We define these edges as fusion edges which should be eliminated. Fusion edges are one of a pair, and nodes linked to the fusion edge and the other edge are the same. As we have defined the properties of an edge, the nodes linked to edges must not be the same. Therefore, fusion edges are not real edges. When a robot crosses to a new node using the unexplored edge (fusion edge), the robot can become aware that the same edge exists in the current node. When two edges are the same as each other in the same node, the fusion edges are eliminated.

Figure 5(f) does not show an edge, but the robot believes that there is an edge in pre-mapping. However, when crossing using this edge, a robot is aware that the edge is not connected to another node. As we have defined the properties of an edge, termination of an edge must be linked to a node. Therefore, this edge is not a real edge. We define such an edge as a dead-end edge and it is eliminated from the topological map.

The last exceptional case is the cyclic case. A cyclic case is illustrated in Fig. 5(a) and (b). As we have defined the properties of a node, the region of a node does not overlap another node. When a robot finds a new node, it examines whether the new node matches previous nodes. The robot changes its current position to relative positions of each node. If the changed relative position belongs to any node, the new node is matched to that node. If the new node matches a previous node, it is not added to the topological map, and the current edge is also merged into the same edge which existed in the previous node.

4. Composition of Topological Map and Local Map

Each node has three kinds of data: the node index, the data of linked edges, and the local grid map of the node. The node index is an integer value. The edge data comprises three kinds of information: indexes of the start and end nodes, the starting and ending positions, and the relative position between the start node and end node. The local grid map of a node is a kind of cell array. Each cell has barcode information (x, y) and probability of occupancy. There are hundreds of cells in the local grid map of a node, and there are one or two nodes per room. However, because each node has a local grid map, we can have the barcode information and probability for the whole region. In each node, the robot uses only the local grid map which corresponds to the current node. In the next section, we explain the exploration procedure used to compose the map.

5. Exploration Procedure

Using the previous definitions and edge handling, the overall procedure of navigation for exploration is shown in Fig. 6. We solve the localization problem using the method described in section II. If the robot has barcode information of the whole area, the robot can localize itself in any position. Therefore, the robot should navigate to make a whole map. The navigation procedure has several steps.

A. Pre-Mapping

In this step, a robot basically follows the walls. When the robot follows the walls, it can detect candidate edges and it should not cross edges. After pre-mapping, the robot knows the number of candidate edges, the start positions of the candidate edges and the outline of the node.

B. Full Mapping

Full mapping is the step in which the robot explores the whole region of the node. We use a zigzag algorithm, but this algorithm

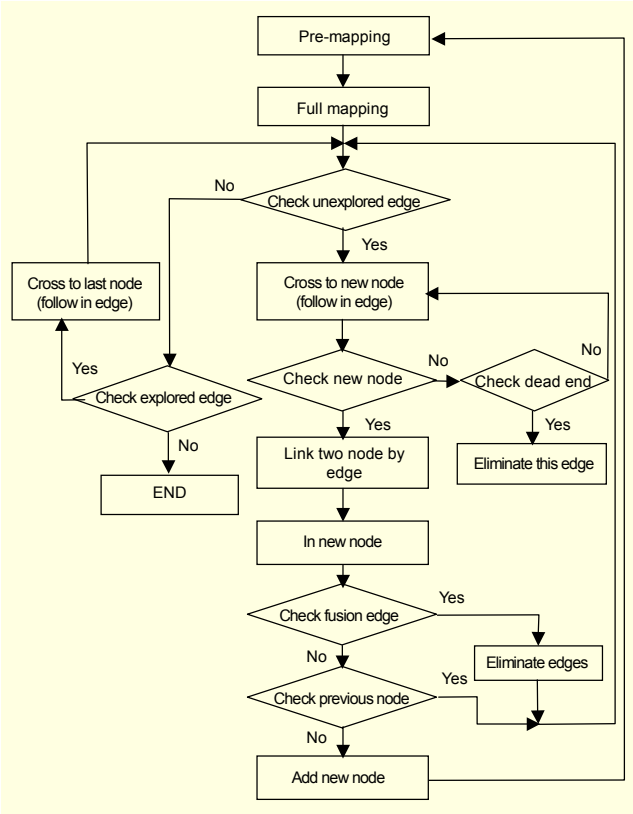

Fig. 6. Overall procedure of exploration.

is not efficient. Other coverage algorithms could be used in this step to fill one node. However, that is not the main focus of this paper. In future work, we will improve the full-mapping mode.

C. Move to a New Node

After full-mapping, the robot checks whether any unexplored edge exists in the current node. If unexplored edges exist in the current node, the robot moves to the starting position of an unexplored edge using the A* algorithm. After reaching the starting position of the edge, the robot moves through the edge region until perceiving a new node. In the case shown in Fig. 5(f), the robot is not able to find a new node an reaches a dead-end. The robot eliminates the dead-end edge in this case.

In the case of a new node, the robot checks for the type of case shown in Fig. 5(e). If the ending position of the current edge is adjacent to the start position of the other edge, they are fusion edges as in Fig. 5(e); therefore, the two edges are eliminated.

Finally, the robot checks whether the new node has been explored previously. This is to prevent the occurrence of a cyclic case as shown Fig. 5(a) and (b).

When the node is not a cyclic case, the robot makes a new node and adds this node to the topological map. In the new node, the robot resets its current position as (0, 0), makes a new local grid map, and executes pre-mapping.

D. Move to Last Node

When no unexplored edge is found, the robot moves to the last node (commonly called the parent node). In this step, the moving algorithm of the robot is the A* algorithm which was used in moving to a new node. In the last node, the robot checks whether an unexplored edge exists. If an unexplored edge exists, the robot moves to the new node. However, if there is no remaining unexplored edge in the node, the robot moves to the parent node of the current node. Consequently, the robot comes back to the first node. When there is no unexplored edge in any node, the mapping procedure is finished. Commercially, once the mapping is done, a user can command the robot to clean a specific room using the node index.

6. Advantages of Proposed Navigation Algorithm

Map building with topological information using our definition of edge and node has several advantages in comparison with map building using other topological information. First of all, when making a path to a certain position in another node, instead of scanning the whole map, we can search indexes connecting the start node with the target node and paths to the starting positions of edges in each node very simply without computational burden. Secondly, the robot moves to another node after completely covering or exploring one node. It is very efficient, because there are many walls and obstacles in an indoor environment. When a coverage and exploring algorithm is applied to the whole environment, walls and obstacles will disturb the algorithm. We will analyze efficiency using time and distance of path in a future study. Finally, errors that occur in each node are mutually independent, so this limits errors in mapping.

V. Results

1. Simulation Results

We performed simulations to validate the navigation performance of the proposed system using topological information in an environment closely resembling a real indoor environment as shown in Fig. 7(a). Figure 7(b) shows the topological structure of the environment. The robot radius is 210 mm, the dimensions of the environment are 8 m×12 m and the wall thickness is 100 mm. We assumed that a robot is unable to go into the bathroom or the utility room.

After the simulation, the indoor environment in Fig. 7(a) was divided into six nodes. Case 1 in Fig. 7(a) is an example of the case shown in Fig. 5(c); an edge is created between the wall and a pot. Case 2 is an example of the case shown in Fig. 5(f). In case 2, the irregularities of the wall were treated as edges at first, but these were not connected to any other node, so when crossing to a new node, these were eliminated. Case 3 is an example of the case shown in Fig. 5(e). In this case, the robot thought that two edges existed at first. When crossing into the new node, the robot could

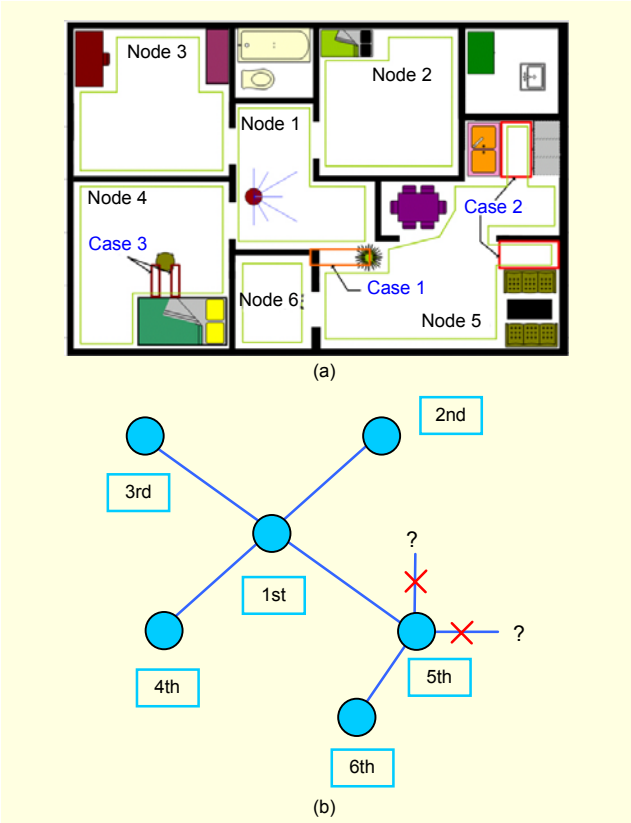

Fig. 7. Indoor environment in simulation: (a) simulation environment and (b) topological structure.

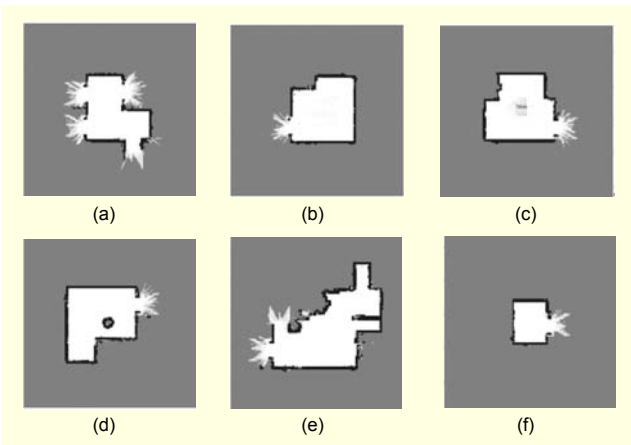

Fig. 8. Local grid map: (a) node 1, (b) node 2, (c) node 3, (d) node 4, (e) node 5, and (f) node 6.

know that these were the same. Because these edges are fusion edges, both edges were eliminated. Figure 8 shows the local grid maps of each node after mapping. From these results, we know that in cases 2 and 3 irregularities and obstacles were not treated as edges. Node history was as follows: node 1 → node 2 → node 1 → node 3 → node 1 → node 4 → node 1 → node 5 → node 6 → node 5 → node 1.

Table 2. Final robot position.

	$x_{position}$ (cm)	$y_{position}$ (cm)
Real position	1.0	3.6
Odometry	79.1	-62.3
Corrected position	0.1	0.2
Odometry error	78.1	-65.9
Corrected position error	-0.9	-3.4

2. Localization Experiment

Two sets of experiments were performed to validate localization performance using the robot shown in Fig. 1(a). This robot does not have good encoders (300 pul. / rev.) and it moves on wheels; therefore, the odometry data is vastly inaccurate. In this section, we show that localization is possible even using this inaccurate odometry and barcodes.

The first experiment followed a 1.5 m × 1.5 m rectangular path. The robot autonomously navigated the path eleven times at a speed of 0.15 m/s and returned to the initial position. The total navigated length was 66 m. Figure 9(a) shows the results of localization using barcodes. The red line represents the odometry path and the blue line shows the path corrected by barcodes. As shown in Fig. 9(a), the odometry error is unbounded, but the corrected data using barcodes is definitely accurate. Table 2 shows the final robot position. The error of odometry with respect to the real position is (78.1, -65.9) cm and the error of the corrected robot position is (-0.9, -3.4) cm. The error in the corrected robot position is caused by d_{cx} and d_{cy}, the distance from the current reading barcode to robot center position. Because they are calculated from images obtained by the barcode reader, we need precise calibration for accurate calculation; however, this error is bounded and does not exceed the distance of the intervals between barcodes. This experiment shows good performance in localization.

In the first experiment, the robot trajectory could not be shown to follow a rectangular path accurately because we did not know the real path of the robot except the start and end positions.

The second experiment was performed in a 340 cm×270 cm room. The robot followed the wall using GP2D120 sensors. The robot autonomously navigated five times at a speed of 0.15 m/s. Figure 9(b) shows the results of the robot following the walls. The red line shows the odometry data, the blue line shows the corrected robot path, and yellow dots show the position of the walls as calculated by sensor distance and corrected robot position. The yellow dots and blue line show good agreement with the locations of the real walls and the real path. In contrast, the odometry error gradually increases.

These two sets of experiments show excellent localization

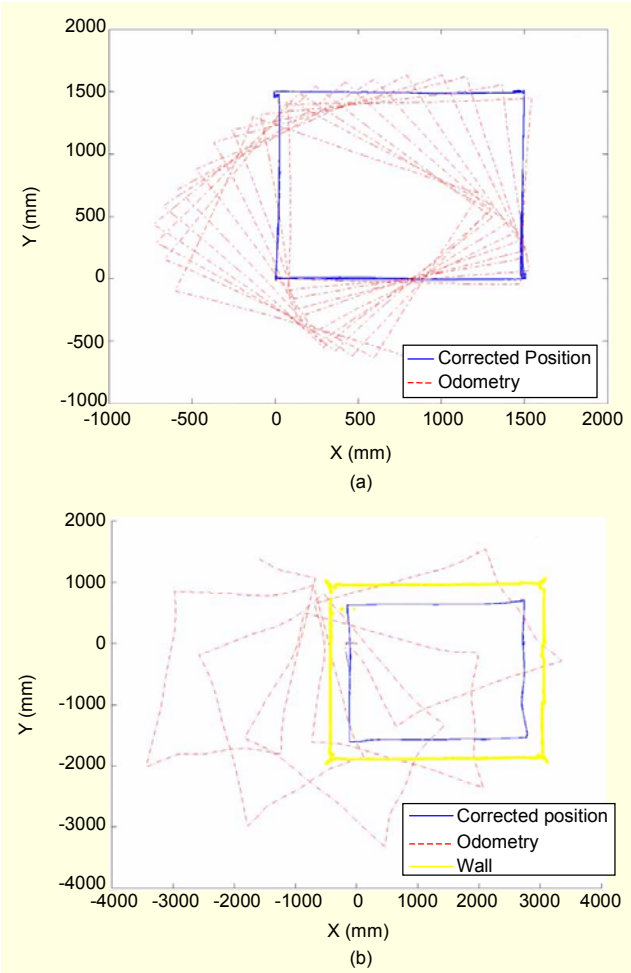

Fig. 9. Experiment results of localization: (a) result of rectangular path navigation and (b) result of robot following the walls.

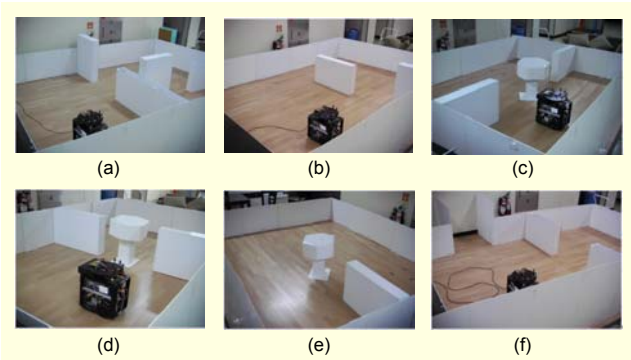

Fig. 10. Experimental environment.

performance.

3. Navigation Experiment

We performed experiments to validate the navigation performance using topological information in the various environments (see Fig. 10). Navigation experiments were carried out using our BSR-I differential-drive robot equipped with seven IR sensors (GP2D120 and GP2YOA02YK). When the robot receives distance information from a position sensitive detector (PSD) sensor, it updates the probability values of cells. We use basic Bayes filters using an inverse sensor model to update the values of cells [16]. The robot autonomously navigates and builds a map of the unknown environment. In this section, we explain the experimental results which demonstrate how to manage the various edges illustrated in Fig. 5.

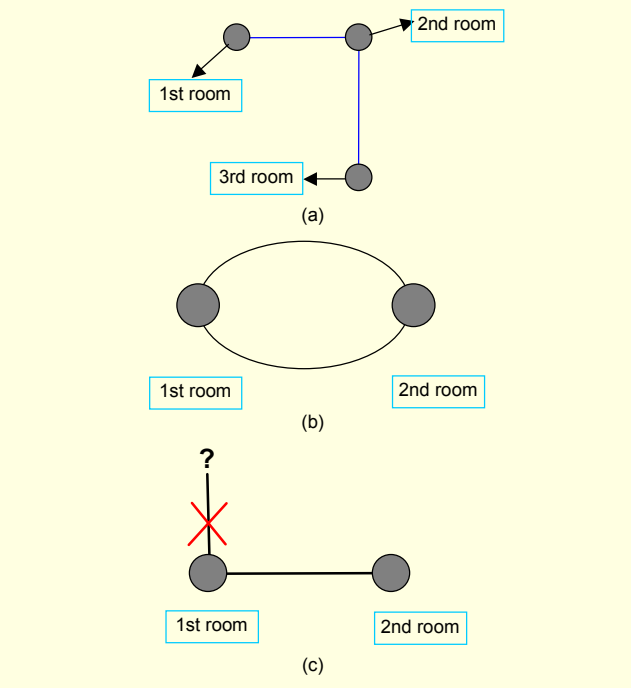

Fig. 11. Topological structure of environments.

A. Composition of Topological Map

First, we performed an experiment in an environment comprising three nodes as shown in Fig. 10(a). Figure 11(a) shows the generated topological structure of the environment. After building a map of the first node, the robot knew that the first node had one unexplored edge. It then moved to the second node using the unexplored edge. After mapping the second node, the robot knew that the second node had one unexplored edge and one explored edge. After mapping the third node, there was no unexplored edge. The robot finally moved again to the first node. Figure 12 shows the local grid map of each node.

B. Experiment Result of Case 1 (Fig. 5(a))

In a home environment, it is possible to for several edges to exist which link two rooms. In Fig. 10(b), there are two edges between two nodes. Figure 11(b) shows the topological structure of the

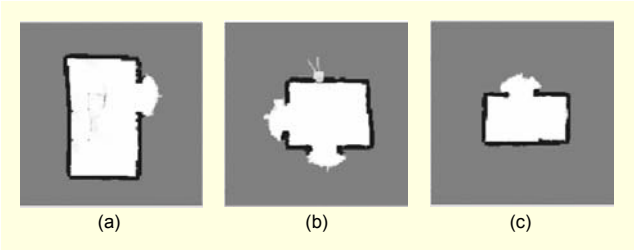

Fig. 12. Local grid map of three-node environment: (a) node 1, (b) node 2, and (c) node 3.

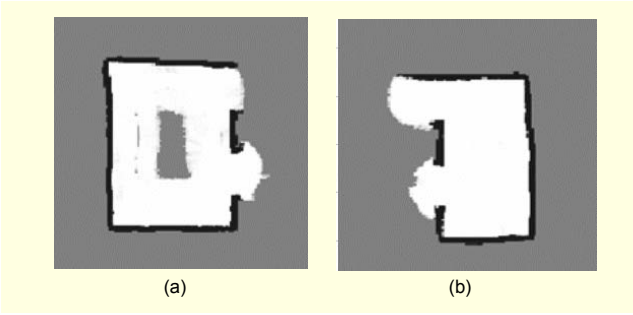

Fig. 13. Local grid map in case 1 environment: (a) node 1 and (b) node 2.

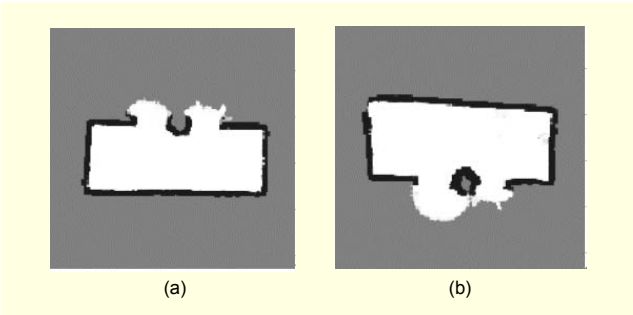

Fig. 14. Local grid map of case 2 environment: (a) node 1 and (b) node 2.

environment. This topological structure is a cyclic loop. If the robot cannot recognize the same node when it comes back to the first node, it falls into an infinite cycle. However, as the region of a given node cannot overlap any other node, when the robot comes back to the first node, it can recognize that the current node is not the third node but first node. This is because the robot has the information of relative position between nodes. When the robot recognizes that the current node is the first node, it does not add a new node and it updates the information of the first node. Figure 13 shows each local grid map after map building.

C. Experiment Result of Case 2 (Fig. 5(b))

In case 2, there are two edges which are formed by an obstacle. The robot can cross to another node using any edge. The topological structure of case 2 is the same as that shown in Fig. 11(b). Figure 14 shows each local grid map after mapping Fig. 10(c).

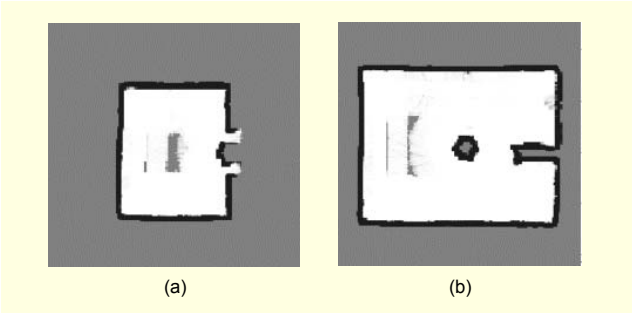

Fig. 15. Local grid map: (a) case 4 and (b) case 5.

D. Experiment Result of Case 3 (Fig. 5(c))

In case 3, there is obstacle in an edge area as in case 2, but the robot detects one edge. There is no difference except a slight change in the starting position of the edge and the robot can move to another node using this edge.

E. Experiment Result of Case 4 (Fig. 5(d))

In Fig. 10(d), the robot is not able to cross to another region because of obstacles. This contradicts assumption 1; therefore, case 4 should not occur during mapping. Figure 15(a) shows the result of mapping the first node.

F. Experiment Result of Case 5 (Fig. 5(e))

Edges are defined as fusion edges when they exist between a wall and an obstacle as shown in Fig. 10(e). After pre-mapping, the robot has two unexplored edges which are fusion edges. However, the robot is aware that the unexplored edges are fusion edges when crossing to a new node. The unexplored edges are eliminated from the topological structure. Figure 15(b) shows the local grid map after mapping.

G. Experiment Result of Case 6 (Fig. 5(f))

In Fig. 10(f), there are two nodes and one dead-end edge.

After mapping the first node, the robot has left two edges unexplored, one of which is a dead-end edge. When the robot crosses the dead-end edge, the robot is aware that it is not connected to another node. It contradicts the properties of an edge, so the dead-end edge is eliminated. Figure 16 shows the local grid map after mapping.

The results of the presented experiments are the initial results of our algorithm, but the basic structure of the algorithm need not be changed even if it is performed in a more realistic environment. Our algorithm is suitable for cluttered indoor environments such as homes and offices. In a cluttered environment, it is difficult for a robot to make a path using previous methods of map composition;

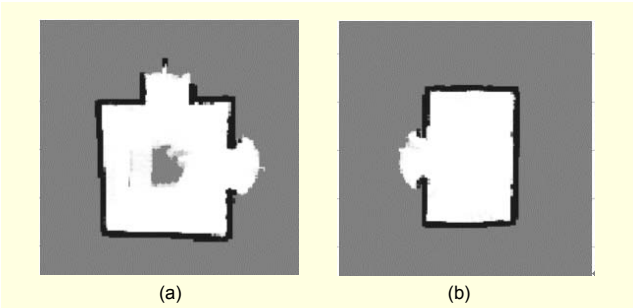

Fig. 16. Local grid map of two nodes including one dead-end edge environment: (a) node 1 and (b) node 2.

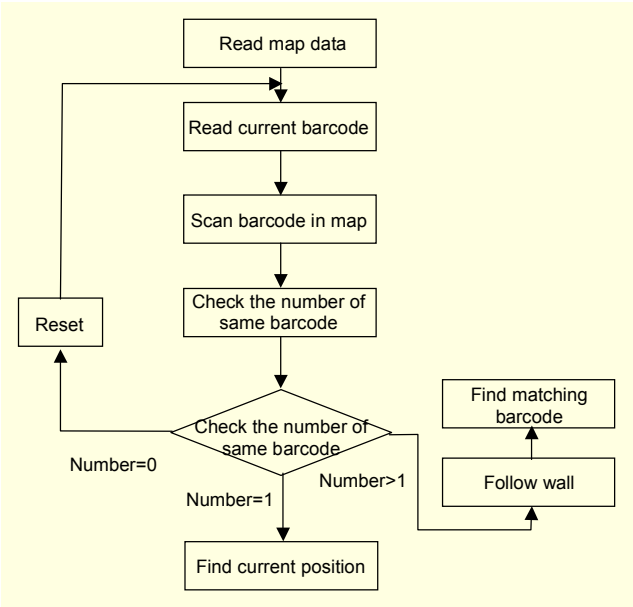

Fig. 17. Overall procedure of kidnap recovery.

however, a robot can make a path much more easily using our algorithm. When making a path to a certain position, instead of scanning the whole map, the robot can search indexes connecting the start node with the target node and paths to the starting positions of edges in each node. This is possible because the robot has the local grid map of every node. For this reason, paths using our algorithm are more optimized than other paths.

4. Kidnap Recovery

To verify that a robot can find its absolute location using our system, we carried out an experiment to solve the kidnapped robot problem [17]. After mapping, a robot has the barcode information of the whole region, thus the robot should be able to recover from a kidnap situation in which it would be moved to a random position without being told. Using our system, the robot can represent the indoor environment using the topological structure. When the robot reads a barcode, it compares the barcode to all of the barcodes in the local maps of every node. Because there is no rule for the placement of each piece of flooring except that the pieces are laid with a y-directional interval of three barcodes, it is possible for identical barcodes to exist near each other. Therefore, the robot cannot find its location after reading only one barcode. However, after the robot reads several barcodes, it can find its location by comparing an array of barcodes in the local maps. The overall procedure of kidnap recovery is shown Fig. 17. This recovery was experimentally verified. In the experiment of kidnap recovery, we commanded the kidnapped robot to move to a specific position after mapping. We validated that the robot could find its absolute location after finding the matching barcode.

VI. Conclusion

This paper proposed a localization and navigation algorithm to provide topological information for the movement of robots in indoor environments using invisibly barcoded flooring. The proposed algorithm has several advantages.

First, we solved the localization problem using invisible barcodes. This solution shows excellent localization performance with only a few centimeters of error for the whole region of a homelike environment. After mapping, it is also possible for a robot to recover from being kidnapped. We also proposed a compensation algorithm of prediction and correction to cover the case when a barcode is damaged or when a sensor signal is noisy. Moreover, we suggested new definitions of node and edge and proposed a navigation algorithm which is suitable for indoor navigation, especially for large area having multiple rooms, many walls, or many static obstacles. Because the navigation algorithm uses topological information, a robot can move into other zones after exploring one region. This algorithm also has the advantage that the errors occurring in each node are mutually independent. It also has computational efficiency to make paths. Simulation and experimental results showed that the proposed algorithm is suitable solution for commercial robot navigation in indoor environments, although the system requires new flooring.

References

[1] iRobot. website. [Online]. Available: http://www. roombavac.com

[2] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte Carlo Localization for Mobile Robots," *Int'l Conf. Robotics and Automation*, 1999, pp. 1322-1328.

[3] H.H. Gonzalez-Banos and J.C. Latombe, "Navigation Strategies for Exploring Indoor Environments," *Int'l J. Robotics Research*, vol. 21, no. 10, 2002, pp. 829-848.

[4] Javier Minguez, "Integration of Planning and Reactive Obstacle Avoidance in Autonomous Sensor-Based Navigation," *Int'l Conf. Intelligent Robots and Systems*, 2005, pp. 270-276.

[5] C. Madsen and C. Andersen, "Optimal Landmark Selection for Triangulation of Robot Position," *Int'l J. Robotics and Autonomous Systems*, vol. 23, no. 4, 1998, pp. 277-292.

[6] S. Hernaóndez, J.M. Torres, C.A. Morales, and L. Acosta, "A New Low Cost System for Autonomous Robot Heading and Position Localization in a Closed Area," *Autonomous Robots*, vol. 15, no. 2, 2003, pp. 99-110.

[7] Evolution Robotics. website. [Online]. Available: http://www.evolution.com/products/northstar/

[8] E. Gonzaólez, A. Suaórez, C. Moreno, and F. Artigue "Complementary Regions: a Surface Filling Algorithm," *Int'l Conf. Robotics and Automation*, 1996, pp. 909-914.

[9] W.H. Huang, "Optimal Line-Sweep-Based Decompositions for Coverage Algorithms," *Int'l Conf. Robotics and Automation*, 2001, pp. 27-32.

[10] P. Althaus and H.I. Christensen, "Automatic Map Acquisition for Navigation in Domestic Environments," *Int'l Conf. Robotics and Automation*, 2003, pp. 1551-1556.

[11] N. Tomatis, I. Nourbakhsh, and R. Siegwart, "Simultaneous Localization and Map Building: A Global Topological Model with Local Metric Maps," *Proc. IEEE Int'l Conf. Intelligent Robots and Systems*, Oct. 2001, pp. 421-426.

[12] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller, "An Atlas Framework for Scalable Mapping," *Proc. IEEE Int'l Conf. Robotics and Automation*, Sep. 2003, pp. 1899-1906.

[13] A.C. Victorino and P. Rives, "Global Consistency Mapping with a Hybrid Representation," *Proc. IEEE Int'l Conf. Intelligent Robots and Systems*, Aug. 2005, pp. 2403-2408.

[14] E. Fabrizi and A. Saffiotti, "Extracting Topology-Based Maps from Gridmaps," *Proc. IEEE Int'l Conf. Robotics and Automation*, Sep. 2000, pp. 2972-2978.

[15] P. Buschka and A. Saffiotti, "A Virual Sensor for Room Detection," *Proc. IEEE Int'l Conf. Intelligent Robots and Systems*, Oct. 2002, pp. 637-642.

[16] S. Thrun, W. Burgard and D. Fox, *Probabilistic Robotics*, MIT Press, 2005.

[17] D. Fox, S. Thrun, W. Burgard, and F. Dellaert, "Particle Filters for Mobile Robot Localization," A. Doucet, N. de Freitas, and N. Gordon (eds.), *Sequential Monte Carlo Methods in Practice*, Springer Verlag, 2001, pp. 499-516.

Jinwook Huh received his BS degree in mechanical engineering from POSTECH in 2004 and his MS degree in mechanical engineering from POSTECH in 2006. His research interests include navigation, mapping, and localization for mobile robots.

Woong Sik Chung received the BS degree in instrumentation engineering from Seoul National University of Technology in 1999. He is a Chief Researcher in Microrobot Co., Ltd. His research interests include path planning and navigation for service robots.

Sang Yep Nam received the BS, MS, and PhD degrees in electronics engineering from Dankook Unirersity, 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 BS degree in mechanical design from Seoul National University in 1981, his MS degree in mechanical engineering from KAIST in 1983, and his PhD in production engineering from KAIST in 1987. He is a Professor in the School of Mechanical Engineering, POSTECH (he joined the faculty in 1987). His research interests include localization and navigation for mobile robots, underwater robots, and the development of a robust controller for precision motion control. He is a Director of the National Research Laboratory for Intelligent Mobile Robot Navigation. He is a member of IEEE.