Generating Topological Map from Occupancy Grid-map using Virtual Door Detection

Kwangro Joo, Tae-Kyeong Lee, Sanghoon Baek, and Se-Young Oh, *Senior Member, IEEE*

Abstract—**This paper proposes a method for a cleaning robot to generate a topological map from an occupancy grid-map. Virtual door is defined as the candidates of real door, and the virtual doors are detected as edges of the topological map by extracting corner features from the occupancy grid-map; using this method, an initial topological map is generated, which consists of nodes and edges. The final topological map is generated using a genetic algorithm to merge the nodes and reduce the edges. As a result, the generated topological map consists of nodes divided by virtual doors and edges located in real doors. The proposed methods provide a topological map for the user interaction and the cleaning robot, and the topological map can be used to plan more efficient motion including room-to-room path planning and covering each room.**

I. Introduction

In mobile robotics, mapping the environment is widely used for localization and navigation [1]. A mobile robot uses actuators such as electric motors to explore the environment, and uses sensors such as range sensors [2] or vision sensors [3] to obtain information about the surroundings. The information includes many range readings of obstacle detections or many images of the surroundings; however, storing or using this information needs large memory space and increases computational complexity for the execution of localization and navigation. Consequently, the information needs to be simplified by extracting some features from the information and by mapping them for the mobile robot.

Topological mapping and metric mapping are the main approaches to obtain a map. The topological map [4] consists of nodes that represent the significant spaces in the environment, and edges that represent the connection between the spaces. For this reason, the topological map is compact and suitable for global path planning. In contrast, the metric map, for example, an occupancy grid-map [5], represents the environment using grid cells that model the occupied and empty space, so the occupancy grid-map is suitable for elaborate tasks such as coverage path planning. Which approach is used for a mobile robot is determined by the sensor's specification of the mobile robot.

Most cleaning robots use infrared (IR) or sonar sensors rather than laser sensors as range sensors because of cost limitations. Laser sensors provide dense and accurate range readings, but they are expensive. In contrast, IR and sonar sensors provide relatively accurate range readings including uncertainty, but they are cheap. In this case, the uncertainty can be reduced by dividing the environment into grids and by representing the environment using range readings in each region of a grid cell. Therefore, the occupancy grid-map is a useful representation of the environment for the cleaning robot. The occupancy grid-map also can be used for the cleaning robot to perform dense path planning and to check the coverage of a cleaning robot by giving a property to each grid cell. However, the occupancy grid-map represents the environment as one large space that has no division, and it causes the cleaning robot to plan an inefficient coverage in a domestic environment. The domestic environments consist of sequentially connected small spaces, so room-to-room coverage path planning is more efficient than whole space coverage. However, global path planning for this task is complex in the occupancy grid-map.

The objective of this study is to generate a topological map from an occupancy grid-map for a cleaning robot. The topological map can be used to perform a global path planning in the occupancy grid-map, and the topological map makes the cleaning robot simply perform room-to-room path planning by dividing the environment. For example, divided space becomes the node of topological map and a waypoint for the efficient path planning. To extract the topological information from an occupancy grid-map, several researchers have tried to generate a topological map from an occupancy grid-map [6]-[9]. Although their methods work well, the method is only applied in the environments that have narrow passages that are uncommon in domestic; or the methods just divide the indoor environment into several sub-spaces without considering an interface between users and a cleaning robot. In domestic, user sometimes selects one room and requires the cleaning robot to clean up only the room. Therefore, the topological map needs to be matched real environment, in other words, the indoor environment needs to be divided by the location of real doors.

In this paper, virtual door is defined as all possible locations of real doors in an occupancy grid-map, and the occupancy grid-map is obtained in orthogonal environment [10]. Virtual doors are detected by extracting corner features from occupancy grid-map, and an initial topological map is generated which has many nodes (rooms) and edges (virtual doors). The final topological map is generated by using a

Manuscript received January 31, 2010. This work was supported (in part) by Ministry of Knowledge Economy under Human Resources Development Program for Convergence Robot Specialists.

K. R. Joo, T. K. Lee, S. H. Baek, and S. Y. Oh are with Pohang University of Science and Technology (POSTECH), Pohang, Gyeongbuk, 790-784, South Korea (phone: 82-54-279-2880, 2904, 2214; fax: 82-54-279-5594; e-mail: {troll, devilee, tkeb100, syoh}@postech.ac.kr).

978-1-4244-8126-2/10/$26.00 ©2010 IEEE

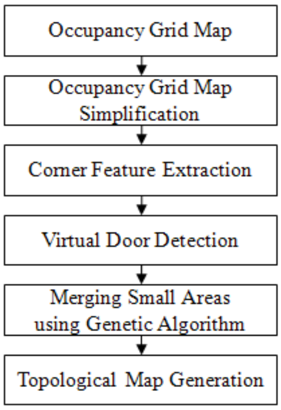


Fig. 1. Flowchart for generating topological map from occupancy grid-map.

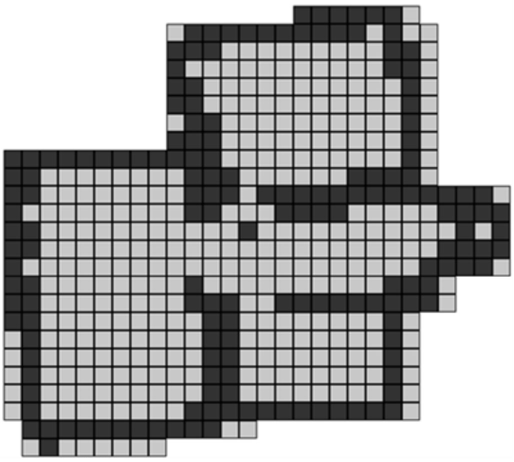


Fig. 2. Occupancy grid-map. Black rectangles: occupied space; gray rectangle: empty space; white space: unexplored space.

genetic algorithm (GA) to merge the nodes (Fig. 1). The topological map consists of real doors that accurately detected.

The detail descriptions of the proposed methods are described in Section II and III, the experimental results and discussions are given in Section IV, and a conclusion is given in Section V.

II. Detection of Virtual Door

We use an occupancy grid-map (Fig. 2) to extract corner features and to generate a topological map for a cleaning robot. Occupancy grid-maps are obtained by a vacuum cleaning robot (RoboKing LG Electronics Inc.) and each grid cell represents 20 cm by 20 cm space. The cleaning robot has a camera on its top for vision based simultaneous localization and mapping (SLAM), and IR and sonar sensors in its front for detecting obstacles. The occupancy grid-map is presented by range sensory data, and is simplified for corner feature extraction. Some obstacle grid cells are extracted as corner features, and the corner features are used to detect virtual doors and to set the boundary of rooms. The occupancy grid-map is divided by the virtual doors, and then an initial topological map is generated.

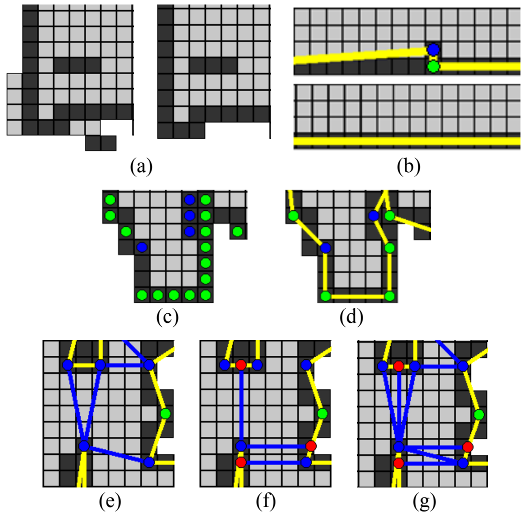


Fig. 3. Example of the process for detection of virtual doors. Green circles: concave features; blue circles: convex features; red circles: virtual feature for virtual door detection; yellow line: boundary of indoor environment; blue lines: virtual doors. (a),(b) occupancy grid-map simplification; (c),(d) corner feature extraction; (e),(f),(g) virtual door detection.

A. *Occupancy Grid-map Simplification*

An occupancy grid-map is simplified by removing the grid cells that are inaccessible to the robot (Fig. 3a). Although the occupancy grid-map is obtained by SLAM, the map still includes errors because IR and sonar sensors are not completely accurate. Some grid cells are not covered or reached by a cleaning robot that moves only in an enclosed indoor environment. The information about these grid cells is unnecessary; therefore the grid cells should be removed for the effective generation of a topological map.

After the unnecessary grid cells are removed, obstacle grid cells are flattened to remove noise that interferes in corner feature extraction. For example, although an obstacle grid cell that juts into inner empty space is actually not a corner; the grid cell can be extracted as a corner feature (Fig. 3b). The reason for this error is that the curvature between the jut and its neighboring obstacle grid cells is high as if the jut is a corner.

B. *Corner Feature Extraction*

Corner features are extracted using adaptive curvature estimation that is a modified version of [11]. Curvature at each obstacle grid cell is the correlation of sequential grid cells in the neighbors of the grid cell. For adaptive curvature estimation, obstacle grid cells are sequentially sampled and numbered according to Left-Hand Path. The curvature at *i-th* sampled grid cell is estimated by the following steps:

1) Calculation of the number of neighbors at *i-th* sample. The number of frontward neighbors $K_f[i]$ and the number of backward neighbors $K_b\,[i]$ are the largest value that satisfies the following equations respectively:

$$d(i,\ i\ +\ K_f[i])\ >\ l(i,\ i\ +\ K_f[i])\ \times\ U_k \quad (1)$$

$$d(i,\ i\ -\ K_b[i])\ >\ l(i,\ i\ -\ K_b[i])\ \times\ U_k \quad (2)$$

where $d(x, y)$ is the Euclidean distance from *x-th* sample to *y-th* sample; $l(x, y)$ is a real length between *x-th* sample and *y-th* sample; U_k determines the nosie ratio that can be tolerated. In our case, U_k = 0.9 works well in all occuapancy grid maps.

2) Calculation of the local vectors between *i-th* sample and its neighbors. The forward vector f_i is the vector between *i-th* and $(i+K_f[i])$*-th* sample, and the backward vector b_i is the vector between *i-th* and $(i-K_b\ [i])$*-th* sample. The vectors are defined as the following equations:

$$\overrightarrow{f_i} = (x_{i+K_f[i]} - x_i,\ y_{i+K_f[i]} - y_i) \quad (3)$$

$$\overrightarrow{b_i} = (x_{i-K_b[i]} - x_i,\ y_{i-K_b[i]} - y_i) \quad (4)$$

where (x_i, y_i) is the coordinates of *i-th* sampled grid cell.

3) Calculation of the curvature at *i-th* sample. The curvature ρ_i estimated by the following equation:

$$\rho_i = \left(arccos\left(\frac{\overrightarrow{f_i}\cdot\overrightarrow{b_i}}{\left|\overrightarrow{f_i}\right|\cdot\left|\overrightarrow{b_i}\right|} \right) \right)^{-1} \quad (5)$$

The obstacle grid cell that has $\rho_i > 1/155$ extracted as a candidate corner feature and each candidate is defined as convex feature and concave feature according to the following equation:

$$\alpha = arctan\left(\frac{f_{iy}}{f_{ix}}\right) - arctan\left(\frac{b_{iy}}{b_{ix}}\right) \quad (6)$$

The candidates that have $\alpha < -180^\circ$ or $0 \le \alpha < 180^\circ$ are defined as concave features and the candidates that have $\alpha \ge 180^\circ$ or $-180^\circ \le \alpha < 0$ are defined as convex features (Fig. 3c). After the candidates are extracted, the candidates with the local maximum ρ are identified as the corner features (Fig. 3d).

C. Virtual Door Detection

In domestic space, doors are located in narrow pathways around convex corners. Therefore, the location of virtual doors can be defined in narrow pathways around the corner feature, especially convex feature. When a couple of convex corner features is not more than 1.2 m away (Fig. 3e), the location of virtual door is defined between the two features. When a single convex corner feature and the closest boundary from the feature are not more than 1.2 m away (Fig. 3f), the location of virtual door is defined between the feature and the boundary.

The virtual doors are used to generate the initial topological map. The virtual doors are defined as edges of the topological map and the indoor environment is divided into several nodes by the edges. Each node includes the information about which features form a boundary of the room, whether each feature is convex or concave and where the room's center is. Each edge represents how neighboring nodes are connected. At the initial topological map, there are too many nodes and edges. In other words, the indoor space has too many rooms and doors because all virtual doors are used. To increase the robot's coverage and time efficiency, the initial topological map is optimized by deciding which virtual doors should be used. The detail descriptions of this are described in next section.

III. Topological Map Optimization using GA

To increase the robot's coverage and time efficiency, small areas of the initial topological map are merged to make larger areas. Each node has the information of related features, and the locations of the corner features can be used to calculate the area of each room. In the initial topological map, the virtual doors' locations are concentrated in several areas because each convex feature generates several virtual doors (Fig. 3g). This means that many nodes are generated, which have small area. Each node shares the information about the features with neighboring nodes, and each node becomes a base waypoint for room-to-room path planning; therefore the presence of many small areas implies that the cleaning robot needs a large memory and a long time to complete the task. Merging small areas reduces the amount of memory and makes cleaning more efficient.

Genetic algorithm (GA) [12] is a global search approach used to find an optimized solution. GA is inspired by the mechanics of natural genetics and natural selection. GA evolves solutions to obtain optimized solution about a problem, and the evolution is modeled on natural evolution process. Solutions are created and evaluated, and the favored solutions are adaptively preserved with every generation. Therefore, GA is suitable for complex problems or for poorly understood problems that have large search space.

The generation of an optimized topological map is the problem that has large search space. For example, if twenty virtual doors are detected, the problem has 2^{20} solutions in the search space because each virtual door has two options: removed or not. In the case of exhaust search, all solutions need to be evaluated, this task has large computational load, and takes long time. GA can be used to decrease the computational load and time for the optimization of the problem. In the front example, the maximum search space is reduced to (population size) × (maximum generations) = 50×30 by the following steps:

1) Initialization phase - The initial population size is fifty and each string size r is the number of the edges of the initial topological map. A string S_i is represented by a bit as $S_i = e_{i1}e_{i2}e_{i3}...e_{ir}$ where e_{ij} = 1 means that j-th edge is removed to merge the neighboring nodes, and e_{ij} = 0 means that j-th edge remains in the topological map.

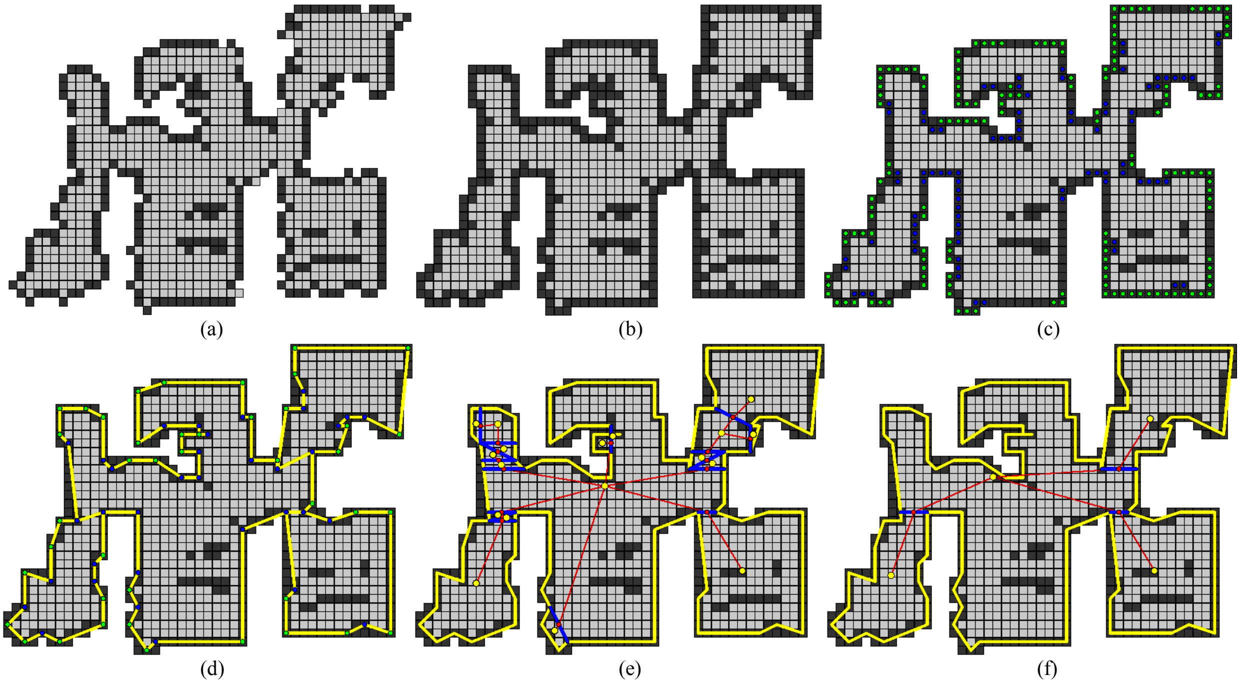


Fig. 5 Topological map generation from occupancy grid-map from [8]. Green circles: concave features; blue circles: convex features; yellow circles: nodes located in centers of rooms; red circles: edges located in centers of virtual doors; yellow line: boundary of indoor environment; blue lines: virtual doors; red lines: connections between each edge and neighboring nodes. (a) occupancy grid-map, (b) simplified occupancy grid-map, (c) sampling of obstacle grids, (d) corner feature extraction, (e) initial topological map, (f) merged topological map.

Parent $S_1 = \boldsymbol{e_{11}}\ \boldsymbol{e_{12}}\ \boldsymbol{e_{13}}\ \boldsymbol{e_{14}}\ \boldsymbol{e_{15}}\ \ldots\ \boldsymbol{e_{1r}}$
Parent $S_2 = e_{21}\ e_{22}\ e_{23}\ e_{24}\ e_{25}\ \ldots\ e_{2r}$
Crossover: | | ... |
Offspring $S_1 = e_{21}\ e_{22}\ \boldsymbol{e_{13}}\ \boldsymbol{e_{14}}\ \boldsymbol{e_{15}}\ \ldots\ e_{2r}$
Offspring $S_2 = \boldsymbol{e_{11}}\ \boldsymbol{e_{12}}\ e_{23}\ e_{24}\ e_{25}\ \ldots\ \boldsymbol{e_{1r}}$
(a)

String = 0 0 0 1 0 1 1 0 0 ... 0
Mutation: | |
Result = 1 0 0 1 0 1 0 0 0 ... 0
(b)

Fig. 4. Genetic operations. (a) crossover, (b) mutation.

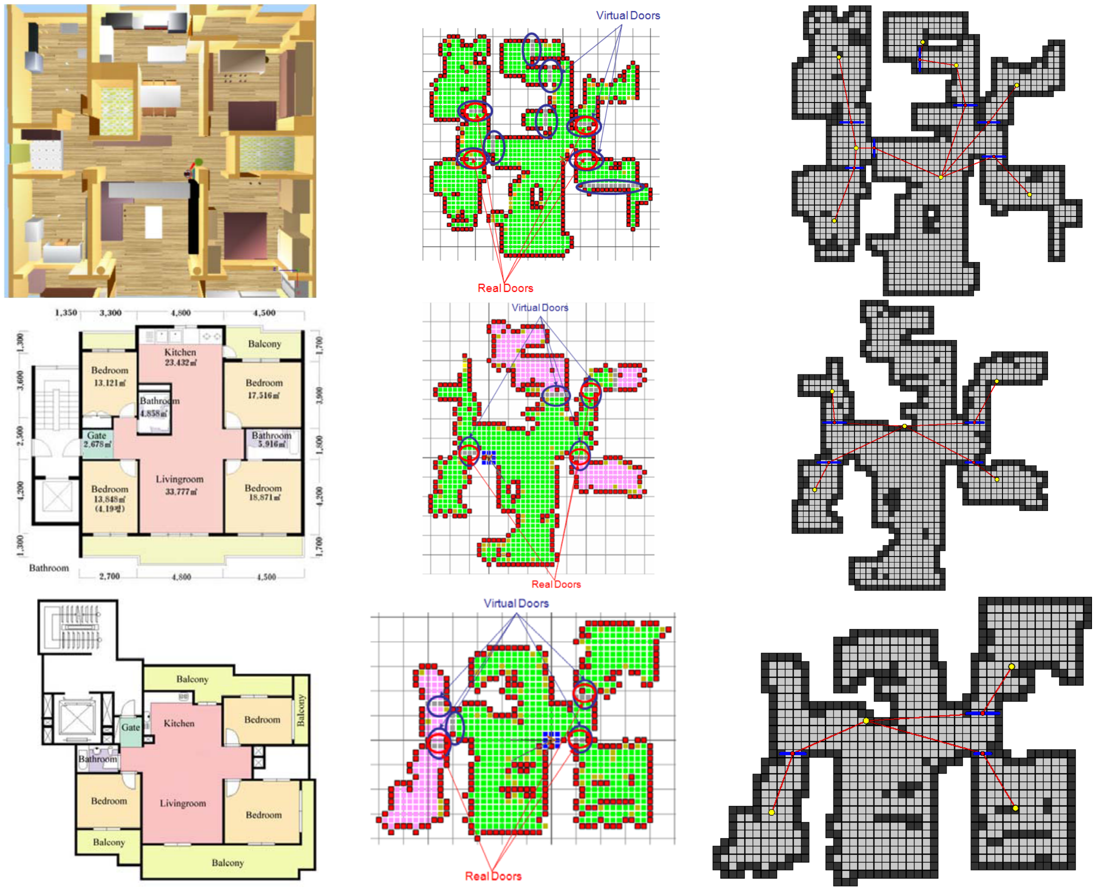
2) Offspring generation phase - Local search is neccessary to find an optimizated solution, and genetic operations (Fig. 4) are used for this tasks. In our case, crossover rate is 0.5, and mutation rate is 0.1 for the mutation for 0 to1 and 0.01 for the mutation for 1 to 0. As a result, fifty offspring strings are generated.
3) Evaluation phase - To evaluate each string, we set three parameters: *NumSamllNode*, *AreaSD* and *DegSD*. *NumSmallNode* is the number of rooms that are samller than room threshold value. In the initial topological map, some nodes have small areas that are hardly considered as room, and *NumSmallNode* performs merging these small nodes. *AreaSD* is the standard deviation of the areas of the rooms from the room threshold value. This parameter is used to prevent generating too big area because the area makes both global path planning and room-to-room coverage path planning complex. *DegSD* is the standard deviation of the angles of the virtual doors. Each angle of the virtual doors is defined as the minimum value of the angle from x-axis or y-axis of the world coordinate. Occupancy grid maps are obtained in the geometric constraint that evrey straight wall is parallel or orthogonal to each other, so the real doors are orthgonal to the axises. Therefore, *DegSD* is used to reduce the virtual doors that have little probabilities of real doors. As a result, the cost of each string is evaluated by the following equation:

$$Cost = w_1 \cdot (NumSmallNode) + w_2 \cdot (AreaSD) + w_3 \cdot (DegSD) \quad (7)$$

where w_1, w_2 and w_3 are normalization factors.
4) Selection phase - The strings of the next population are selected by elitism, and the process is repeated from 2) to 4) until fifty-th genenrations. The cost decrese through these steps. At the end of generations, the best string of final population is seceted and an optimized topological map is generated.

IV. Experimental Results

In this paper, we proposed a method that generates a topological map from an occupancy grid-map. The occupancy grid-map was simplified, and some of the grid cells extracted as the corner features by curvature estimation. The corner features were used to detect virtual doors, the virtual doors removed by GA, and then the topological map was generated (Fig. 5). An occupancy grid-map (Fig. 5a) was simplified by



(a) (b) (c)

Fig. 6 Comparison with existing algorithm [8]. (a) Real environments from [8], (b) Virtual door detections from [8], (c) Topological maps using proposed method.

TABLE I
COMPARISON OF DOOR DETECTION WITH EXISTING ALGORITHM

Environment	Number of doors					Accuracy (B/A)	
	Real	Virtual (A)		Matched (B)			
		Existing	Proposed	Existing	Proposed	Existing	Proposed
case 1	4	9	7	4	4	0.44	0.57
case 2	4	5	4	2	4	0.40	1.00
case 3	3	5	3	3	3	0.60	1.00

eliminating grid cells that are inaccessible to the cleaning robot and that jut into inner empty space. The reduction process flattened the occupancy grid-map and made it easily sampled (Fig. 5b). After curvature estimation, corner feature candidates including 150 concave features and 82 convex features were extracted at some obstacle grid cells that had curvature $\rho_i > 1/155$ (Fig. 5c). Among these corner feature candidates, those having the local maximum curvature were extracted as corner features, and these features defined the boundary of the indoor environment. The extracted features included 43 concave features and 35 convex features (Fig. 5d). Sixteen virtual doors were detected around the convex features, and an initial topological map was generated with virtual doors forming the boundaries of rooms (Fig. 5e). The initial topological map included eighteen nodes and sixteen edges. To make a topological map that the robot can clean efficiently, the initial topological map was merged. In this case, the edges are sixteen, so the search space includes 2^{16} solutions. However, the optimized solution was obtained after 9-th generations by GA, and the computational load was decreased to about $1/2^9$. The final topological map included four nodes and three edges (Fig. 5f).

To compare the proposed method with the existing algorithm [8], the method was applied to three occupancy grid-maps from [8] (Fig. 6). Compared with the existing algorithm, the proposed method located virtual doors more accurately and detected fewer virtual doors that did not correspond to real doors (Fig. 6). For comparison, the accuracy was defined as the ratio of the extracted virtual doors to the matched real doors (Table I). The existing algorithm extracted more virtual doors than real doors in all cases, and the accuracy of the location of doors was low. In contrast, the proposed algorithm extracted the virtual doors in exact locations. Although, in case 1, the proposed algorithm has low accuracy, the accuracy was higher than the existing method. Therefore, the proposed method is better than the existing algorithm in door detection and topological map generation for a cleaning robot.

V. Conclusion

In this paper, we proposed the method to detect virtual doors and to generate topological map from an occupancy grid-map. The accuracy of door detection was better than the existing algorithm. The door detection is useful for user interaction and topological map generation improve the performance of a cleaning robot by determining the cleaning motion for each room adaptively, so the proposed method will be useful for a cleaning robot.

References

[1] S. Thrun, "Robotic Mapping: A Survey," in *Exploring Artificial Intelligence in the New Millennium*, G. Lakemeyer and B. Nebel, Ed. Morgan Kaufmann, 2002, pp. 1-36.

[2] S. J. Baek, H. W. Park and S. Y. Lee, "Mobile Robot Localization Based on Consecutive Range Sensor Scanning and Optical Flow Measurements," *Advanced Robotics*, 2005, pp. 17-22.

[3] D. H. Won, S. B. Chun, S. K. Sung T. S. Kang and Y. J. Lee, "Improving Mobile Robot Navigation Performance Using Vision Based SLAM and Distributed Filters," *Control, Automation and System*, 2008, pp. 186-191.

[4] E. Fabrizi and A. Saffiotti, "Augmenting topology-based maps with geometric information," *Robotics and Autonomous Systems*, vol. 40, 2002, pp. 91-97.

[5] A. Elfes, "Using Occupancy Grids for Mobile Robot Perecption and Navigation," *IEEE Computer*, vol. 22, 1989, pp. 46-57.

[6] S. Thrun, "Leaning Metric-Topological Maps for Indoor Mobile Robot Navigation," *Artificial Intelligence*, vol. 99, 1998, pp.21-71.

[7] E. Fabrizi and A Saffiotti, "Extracting Topology-Based Maps from Gridmaps," *International Conference on Robotics and Automation*, vol. 3, 2000, pp. 2972-2978.

[8] H. Myung, H. M. Jeon, W. Y. Jeong and S. W. Bang, "Virtual Door-Based Coverage Path Palnning for Mobile Robot," *Federation of International Robot-soccer Association*, 2009, pp. 197-207.

[9] J. W. Choi, M. Y. Choi and W. K. Chung, "Incremental Topological Modeling using Sonar Gridmap in Home Environment," *International Conference on Intelligent Robots and Systems*, 2009, pp. 3582-3587.

[10] Y. H. Choi, T. K. Lee, S. Y. Oh, "A Line Feature Based SLAM with Low Grade Range Sensors Using Geometric Constraints and Active Exploration for Mobile Robot," *Autonomous Robots*, vol.24, 2008, pp. 13-27.

[11] P. Nunez, R. Vazquez-Martin, J. C. del Toro, A. Bandera and F. Sandoval, "Feature Extraction from Laser Scan Data Based on Curvature Estimation for Mobile Robotics," *International Conference on Robotics and Automation*, 2006, pp. 1167-1172.

[12] K. S. Tang, K. F. Man, S. Kwong and Q. He, "Genetic Algorithm and Their Application," *Signal Processing Magazine*, vol. 13, 1996, pp. 22-37.