Path Planning for Complete and Efficient Coverage Operation of Mobile Robots

Jung Won Kang, Si Jong Kim and Myung Jin Chung
Department of Electrical Engineering
Korea Advanced Institute of Science and Technology
Yuseong-gu, Daejeon, 305-701, Republic of Korea
{kctown, terre}@cheonji.kaist.ac.kr, mjchung@ee.kaist.ac.kr

Hyun Myung, Jun Ho Park and Seok Won Bang
Micro Systems Laboratory
Samsung Advanced Institute of Technology
Giheung, Yongin, 449-712, Republic of Korea
{h.myung, jun45, banggar.bang}@samsung.com

***Abstract* - This paper presents a complete and efficient coverage path planning method for mobile robots. Its applications include robots for de-mining, cleaning, painting and so on. When a mobile robot performs area coverage task, completeness and efficiency of coverage are important factors. To achieve completeness easily, we adopt a divide and conquer strategy. We developed a novel cell decomposition algorithm that divides a given area into several cells. To achieve efficiency, each cell is covered by a robot motion that requires minimum time to cover the cell. The effectiveness of the method is verified using simulations.**

Index Terms - Complete coverage algorithm, Optimal coverage algorithm, Cell decomposition

I. INTRODUCTION

The purpose of coverage path planning for a mobile robot is to make the robot visit all reachable points in a given environment. Area coverage operation of a mobile robot can be used in various applications such as cleaning robots, patrol robots, de-mining robots, and so on.

Various approaches have been proposed to deal with coverage path planning problem. One of the most popular approaches to this problem is the utilization of a divide and conquer strategy [1]. In this approach, a given environment is divided into several regions called cells, and each cell is covered by simple robot motion. Choset [2] introduced a coverage method based on an exact cellular decomposition called boustrophedon decomposition. In this method, a line segment called a slice is swept through a given environment. When there is a change in the connectivity of the slice, a new cell is formed, and the cell is then described by points termed critical points. After cell decomposition, the robot covers each cell using a simple zigzag motion called boustrophedon motion. Acar *et al*. [3, 4] presented a framework for coverage task using Morse decomposition. In this method, the critical points of Morse function are used to form the cells. Using the property that the topology of the slice remains constant between the critical points, this method easily guarantees complete coverage of a cell. Wong and MacDonald [5, 6] proposed a cell decomposition algorithm based on topological structures of the robot's environment. This form of decomposition, termed slice decomposition, uses changes in topology to decompose an environment, where each cell intersects with the sweep line twice as it passes over. In this method, the cell decomposition results are represented as a topological map, and each cell is covered by the robot via simple zigzag motion. These divide and conquer approaches are very powerful in that completeness of coverage can be easily achieved by using simple algorithms to cover each region.

When task efficiency is considered, time efficiency as well as completeness of coverage are important factors. A robot should cover a given area as fast as possible. Some previous researches have dealt with the efficiency of coverage. Huang [7] focused on cell decomposition itself to achieve optimal coverage. In this method, the criterion for optimization is the total number of turns required to cover all sub-regions. For reducing the number of turns, a slice direction that minimizes the sum of the altitudes of all cells is chosen. Huang suggested an algorithm that performs multiple line sweeps to decompose the given environment into cells and applied dynamic programming to combine cells into larger sub regions and to assign a sweep direction for each sub region. Although this approach is limited to polygonal environments and requires a great deal of time for cell decomposition, it considers efficiency of coverage within the divide and conquer approach. Yao [8] attempted to achieve optimal coverage by minimizing extra relocation moves between cells. The main concept of this method is ensuring that entry and exit points for various sub-regions of a given environment are in close proximity.

In this paper, we propose a novel coverage path planning algorithm that achieves completeness and time efficiency. We assume that the robot is operated in an enclosed indoor environment and knows its map, which consists of occupancy grids. Our method basically achieves completeness of coverage by adopting a divide and conquer strategy. We propose cell decomposition for general environments. In this method, each cell is represented as a set of grids. The algorithm decomposes a given area into sub-regions based on changes in relations of sets of free space grids, termed free space segments, which are grids in free space. During the cell decomposition process, small and narrow cells are merged into larger neighbor cells. This merging process reduces the computational burden of path generation as the number of cells to cover is decreased. After cell decomposition, predefined template paths are generated for each cell. For each cell, a path that requires minimum time for coverage is selected among the template paths during path generation for a whole area. The remainder of this paper is organized as follows. Section II describes the cell decomposition algorithm for general environments. Section III presents the path generation method for coverage. A path generation method for each cell and the whole area are presented. Results of simulations conducted to test the algorithm are presented in Section IV. Conclusions and further works are discussed in Section V.

II. Cell Decomposition

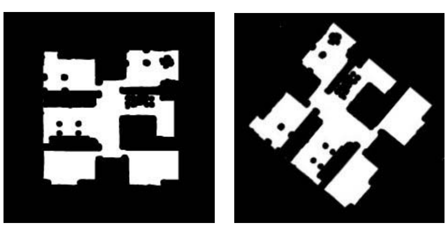
A. Preprocessing

The occupancy grid map, where the value for each grid represents the probability of an obstacle occupancy, is used to represent a given environment. We assume that the robot knows the map of the environment before coverage.

Maps of the same environment can vary due to the orientation of each map. To obtain consistent results of cell decomposition regardless of the orientation of the map, an angle, termed the *orientation invariant angle*, is calculated using the Hough transform (1) from the binary boundary map. The binary boundary map is created by applying a threshold to a map and extracting the boundary.

$$\rho = x\cos\theta + y\sin\theta \tag{1}$$

Among the line orientation values θ, the most dominant value is chosen as an orientation invariant angle. A given map is rotated by the orientation invariant angle. Through this process, consistent cell decomposition results can be obtained regardless of the orientation of the map. Fig. 1 shows the effect of orientation invariant angle. The two maps in Fig. 1 (a) are the same except the orientation. As the two maps are rotated by each orientation invariant angle, we can get the identical maps.



(a)

(b)

Fig. 1 Effect of orientation invariant angle.
(a) Given binary maps. (b) Maps rotated by each orientation invariant angle.

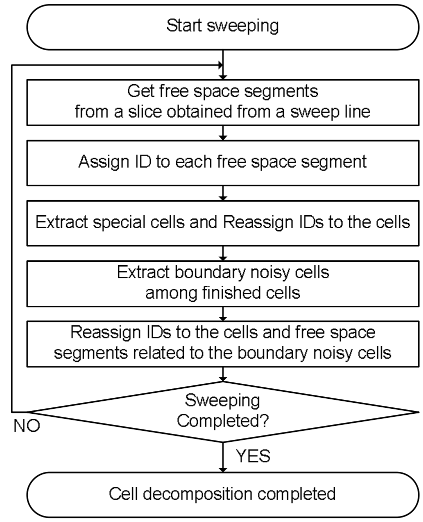


Fig. 2 Cell decomposition algorithm.

B. Cell Decomposition

We propose a novel cell decomposition algorithm for general environments. A given area is decomposed into cells based on changes in relations of segments in free space, which are free space grids. In our algorithm, each cell is described by a set of grids.

A flow chart of the cell decomposition algorithm is shown in Fig. 2. As shown in Fig. 3, a sweep line sweeps from the left of the environment to the right. At each sweep, each free space segment, which is a set of connected free space grids, is extracted from a slice. An identifier is then assigned to the free space grids according to the relation between the current free space segment and previous free space segments. There are only 4 cases, as shown in Fig. 4. In this figure, each pattern in a grid designates each identifier, and a cell denotes a set of free space grids that have the same identifier. If a cell splits into multiple cells ('cell splitting' case) or multiple previous cells are merged into one cell ('cell merging' case) or a cell is started ('cell starting' case), then a new identifier is assigned to each current free space segment. However, if a cell is continued ('cell continuing' case) without the occurrence of any of the above three cases, then the existing identifier of the cell is assigned to the current free space segment. However, in this case, if the shape of the cell is abruptly changed, a new identifier is assigned to the current free space segment, i.e. a new cell is created.

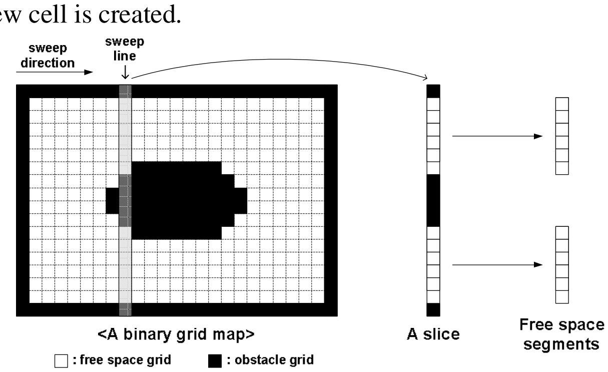


Fig. 3 Free space segments extraction.

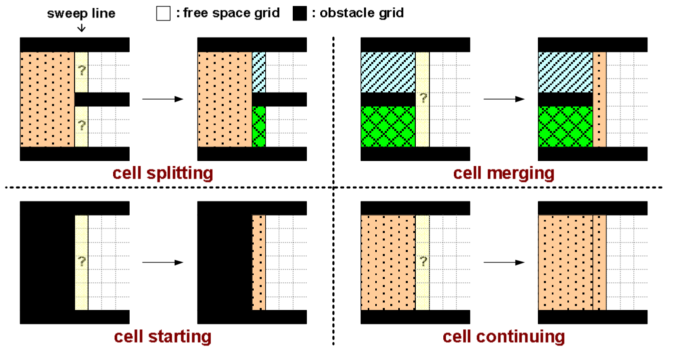


Fig. 4 Assigning an identifier to a free space segment.

In noisy environments due to complex structures and sensor noise during the map building process, numerous small and narrow cells termed *noisy cells* are created. Such noisy cells increase the complexity of the path generation, and finally prevent the realization of time efficiency for coverage. Therefore, noisy cells should be eliminated if time efficiency is to be considered. In our method, noisy cells are merged into larger neighbor cells. Through this process, the number of small cells that do not need to be covered is reduced.

A method for merging noisy cells is shown in Fig. 5. Let us define a parent segment as a previous segment, and a child segment as the next segment. At each sweep, after assigning an identifier to free space segments, noisy cells that are located beside obstacles are extracted among the finished cells. When a cell splitting or cell merging event occurs, if there is only one cell to cause the event except a noisy cell, the event is considered invalid, and is cancelled and then the noisy cell is merged, as shown in 'only 1 other parent segment', 'only 1 child segment' case of Fig. 5. On the other hand, if there are more than one cells to cause the event except a noisy cell, just the noisy cell is merged without canceling the event, as shown in 'more than 1 other parent segments', 'more than 1 other child segments' case. By this method, numerous noisy cells are removed. Finally, as a result of cell decomposition, regions of each cell and an adjacency graph representing the connection relations between each cell are obtained.

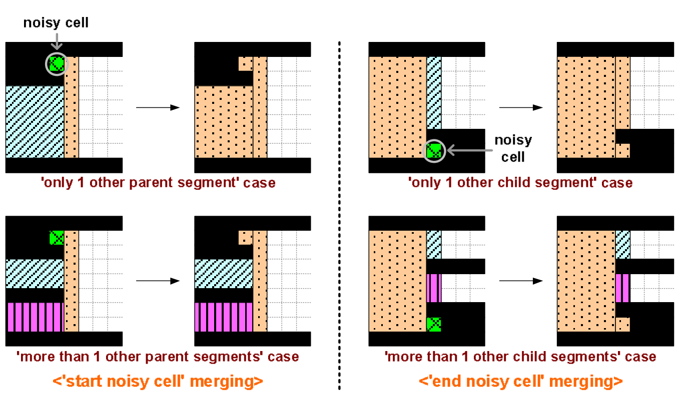


Fig. 5 Noisy cell merging method.

III. Path Generation For Efficient Area Coverage

A. *Path Generation for Each Cell*

After a given environment is divided into cells, predefined template paths are generated for each cell. These template paths are used to find an optimal path to cover a cell. Using this template based method, the computational burden for finding an efficient path is reduced. Paths for two kinds of back & forth motion and spiral motion are predefined as shown in Fig. 6. There are four types of a path for each motion, depending on the start point of each motion. As a first stage of path generation for each cell, the orientation invariant angle is calculated for each cell, and each cell is then rotated by each angle so that the directions of the template paths are aligned along the boundary of the cell. Here, the role of the orientation invariant angle is different from that of cell decomposition. As the orientation invariant angle is derived from Hough transform, the angle implies a direction of the most dominant line component of the map. Therefore, the orientation invariant angle can be used to align the boundary of the cell with directions of the template paths. When a robot covers a cell, the number of turns of the robot required to cover the cell is main factor for time cost. The alignment of a cell using an orientation invariant angle reduces the number of turns by minimizing the sum of altitudes of the cell [7].

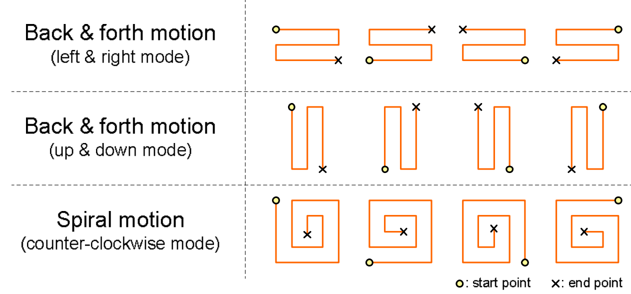


Fig. 6 12 Predefined template paths.

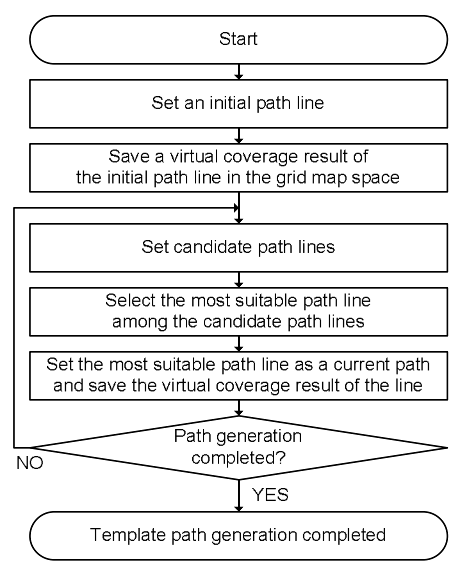


Fig. 7 Template path generation algorithm.

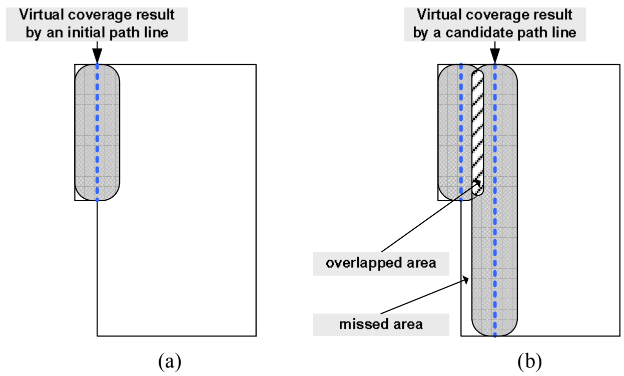


Fig. 8 Using a virtual coverage result.
(a) An initial path line. (b) A candidate path line

An algorithm for generating 12 template paths for each cell is presented in Fig. 7. In this algorithm, the virtual coverage result is used in order to achieve completeness of coverage. The most outer boundary line of the cell is set to an initial path line, as in Fig. 8(a). Based on this line, several candidate paths are set, as in Fig. 8(b), and the cost of each path is evaluated by J in (2), the sum of the overlapped area with the coverage result of the previous path line and the missed coverage area. In (2), α and β are weighting factors of each element. A relatively high α produces a path without an overlapped area. In contrast, a relatively high β produces a path without a missed area. Among the candidate path lines, the path having the lowest cost is selected. The above process is continued until path line generation for a cell is finished.

$$J = (\alpha \times \text{overlapped area}) + (\beta \times \text{missed area}) \tag{2}$$

B. *Path Generation for the Whole Area*

After template paths for each cell are generated, a path for the overall area is formed by selecting the path that requires minimum time for each cell. This is a kind of graph search problem. The algorithm is shown in Fig. 9. The criterion for selecting a cell to cover is the connection relation with neighbor cells and the time to cover it.

Three lists, termed *OPEN*, *CLOSED*, and *PRIORITY* list, are used in this algorithm. The OPEN list contains candidate cells to cover. The PRIORITY list has candidate cells that have high priority for coverage. Finally, the CLOSED list contains cells where coverage is completed. At each step, the OPEN and PRIORITY lists are checked. If the PRIORITY list has any cells, the cell nearest to the robot position is selected. The path that requires minimum time to cover the cell is then selected. If only the OPEN list has a cell, a cell and a path covering the cell are selected, as in the PRIORITY list case. The selected cell is inserted into the CLOSED list, and neighbor cells of the selected cell are inserted into the OPEN list. Among the cells in the OPEN list, cells that have no neighbor cells in the OPEN list are moved to the PRIORITY list. Through the above process, a path for the overall area is generated.

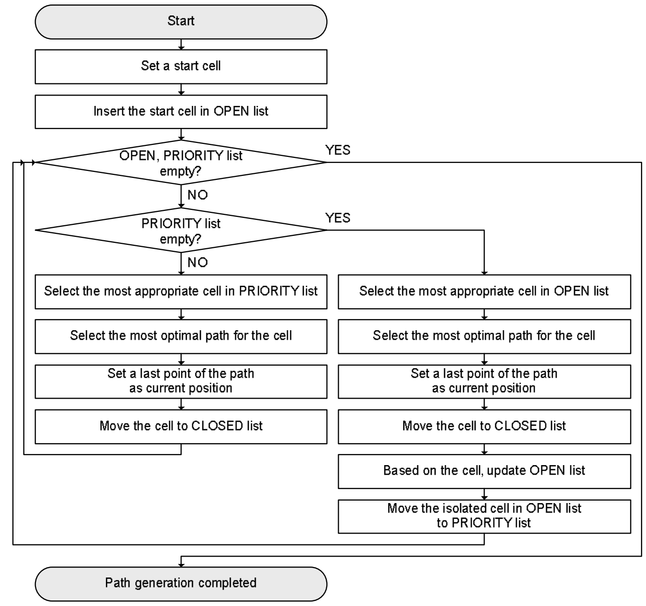


Fig. 9 Path generation algorithm for the whole area.

When a path is generated, the coordinate for the path is translated into the original coordinate i.e. the coordinate before being rotated by the orientation invariant angle.

IV. Results

We verify the effectiveness of our algorithm through simulations. First, we discuss the effect of merging noisy cells and then we present and analyze the results of coverage path generation. We assume that the robot is circular-shaped. The used test map is actually a configuration space where a circular-shaped robot is regarded as a point robot. The configuration space of a grid map can be easily obtained using morphological operation of image processing.

A. *Effect of merging noisy cells*

Fig. 10 shows an example of the effect of merging noisy cells in cell decomposition and path generation. The figures on the left show the cell decomposition results of each case,

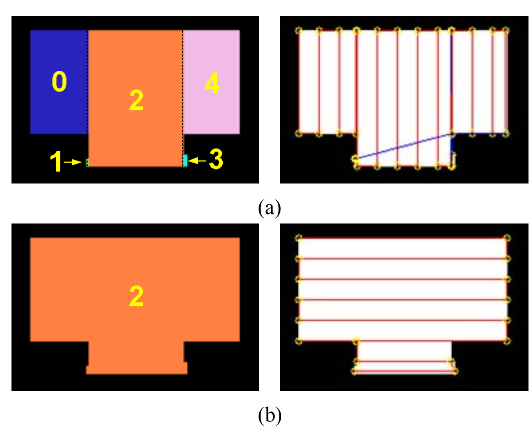
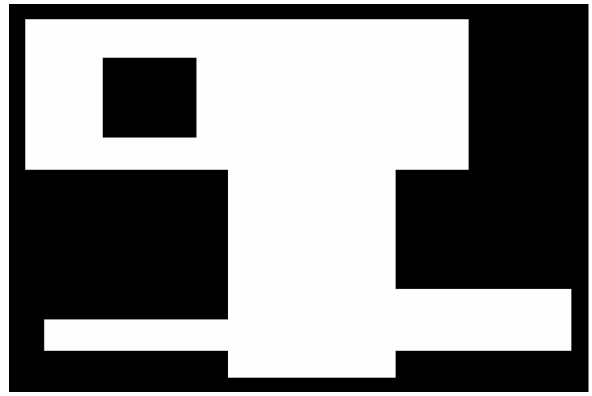


Fig. 10 Effect of merging noisy cells
(a) A result without merging noisy cells. (b) A result with merging noisy cells.

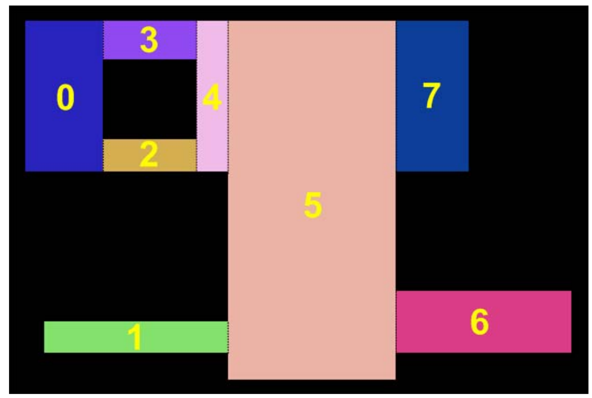
and each number in the figures is the identifier of each cell. The figures on the right show the path generation results when the robot is located at the left-top of the test map, whose size is 177 grid x 112 grid. The width of the free space in the map is longer than the height, and thus left & right back and forth motion is the most efficient motion to cover the space, as shown in Fig. 10(b). In Fig. 10(a), noisy cells such as cells 1 and 3 are not merged, and hence the free space is divided into several cells. Since the height of each cell is longer than the width, up & down back and forth motion is used to cover each cell since this motion is the most efficient for each cell. As a result, case without merging noisy cells requires longer time to cover the whole area. When the velocity of the robot is 0.5m/s, the acceleration is 0.25m/s^2, height & width of each grid is 20mm and the radius of the robot is 0.16m, the required time to cover the whole area in case without merging noisy cells is 128 seconds, and the time for case with merging noisy cells is 84 seconds. By neglecting the effect of noisy cells, we can obtain a more time efficient path. Furthermore, this noisy cell merging process reduces the computational burden of path generation as the number of cells to cover is also reduced.

B. Coverage Path Generation

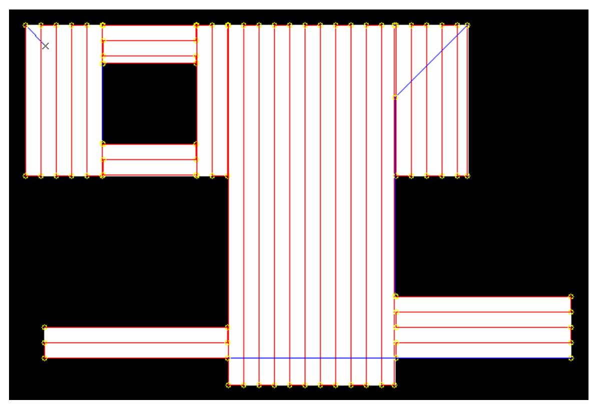
To verify the effectiveness of our method, we compare the result of our method with the method that uses a uniform robot motion to cover all cells. Fig. 11 shows a path generation result for covering a given area. We assume that the robot starts to move at the point marked 'x', i.e. left-top of the test map, as shown in Fig. 11(c). Fig. 11(a) shows the test map. The size of the test map is 433 grid x 643 grid. Fig. 11(b) shows the cell decomposition result. After cell decomposition of a given area, predefined template paths are generated for each cell. Fig. 11(c) shows the path generation result. During the path generation stage for the whole area, a sequence of cells to cover is determined based on an adjacency graph that describes the connection relationships between cells. At this point, time efficiency of coverage is achieved by selecting a path that requires minimum time to cover each cell. In Fig. 11(c), since the robot is initially located in cell 0, cell 0 is selected as the first cell to cover. Among the 12 template paths for cell 0, the path that requires minimum time to cover the cell is selected and generated. After path generation for cell 0, the cell that is the closest to the end point for covering cell 0 is selected as a cell to cover. Here, cell 3 is selected. Among the predefined template paths for cell 3, the most efficient path is selected and generated. This process is continued until all cells are covered. In Fig. 11(c), the sequence of coverage is cell 0, 3, 2, 4, 5, 7, 6, 1. When the velocity, acceleration, size of the robot, and height & width of each grid are the same as those in the previous case, the required time to cover the whole area is 593.92 seconds. On the other hand, when a uniform robot motion is used for coverage, the required time is 690.94 seconds for up & down back and forth motion, 719.98 seconds for left & right back and forth motion.



(a)



(b)



(c)

Fig. 11 Coverage path generation result
(a) Given test map (b) Cell decomposition result (c) Path generation result

The above result shows the advantage of our algorithm over other approaches that use a uniform robot motion. If a uniform robot motion is used for coverage, as in other approaches that employ a divide and conquer strategy, completeness for coverage can be achieved, but time efficiency is not guaranteed. Here, we achieve completeness by using a divide and conquer strategy and time efficiency by minimizing the time to cover each cell.

We also tested our algorithm for application to a cleaning robot through simulations, as shown in Fig. 12.

(a) (b) (c)

Fig. 12 Application to a cleaning robot.

(a) Simulator (b) Occupancy grid map (c) Path generation result

A real environment is modeled as a test environment. Through the simulation, we can also verify that our algorithm is available in a real environment.

V. CONCLUSIONS

In this paper, we describe an efficient path generation method for complete coverage operation of a mobile robot. To achieve completeness of coverage, we adopted the well-known divide and conquer strategy. We proposed a novel cell decomposition algorithm. In our algorithm, unnecessary small cells are merged into a larger neighbor cell. By this process, the computational burden of path generation is reduced. As a result of cell decomposition, the region of each cell and an adjacency graph that describes connection relations of cells are obtained. Each cell is represented as a set of grids. After cell decomposition, predefined template paths are generated for each cell. During path generation for the whole area of interest, a path is generated by gathering paths that require minimum time to cover each cell. Time efficiency of coverage is achieved by selecting the most efficient path for each cell. Completeness and efficiency of our algorithm are verified through simulations.

To generate more efficient paths while maintaining completeness, it is necessary to reduce the amount of overlapped paths located between cells. The sequence of cells to cover should also be addressed further. When we consider the real world applications of robots, practicability in dynamic environments where unknown and moving obstacles exist is also a critical issue. We should also address the coverage path planning problem in dynamic environments.

REFERENCES

[1] H. Choset, "Coverage for robotics - A survey of recent results," *Annals of Mathematics and Artificial Intelligence,* vol. 31, no. 1-4, pp.113-126, 2001.

[2] H. Choset, "Coverage of known spaces: the Boustrophedon cellular decomposition," *Autonomous Robots,* vol. 9, no. 3, pp.247-253, 2000.

[3] E. U. Acar, H. Choset, A. A. Rizzi, P. N. Atkar, and D. Hull, "Morse Decompositions for Coverage Tasks," *The International Journal of Robotics Research,* vol. 21, no. 4, pp.331-344, 2002.

[4] H. Choset, E. Acar, A. A. Rizzi, and J. Luntz, "Exact cellular decomposition in terms of critical points of Morse functions," in *Proc. of the IEEE International Conference on Robotics and Automation*, San Francisco, USA, 2000, pp.2270-2277.

[5] S. C. Wong and B. A. MacDonald, "A topological coverage algorithm for mobile robots," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Lag Vegas, USA, 2003, pp.1685-1690.

[6] S. C. Wong and B. A. MacDonald, "Complete coverage by mobile robots using slice decomposition based on natural landmarks," in *Proc. of the 8th Pacific Rim International Conference on Artificial Intelligence*, Auckland, New Zealand, 2004, pp.683-692.

[7] W. H. Huang, "Optimal line-sweep-based decompositions for coverage algorithms," in *Proc. of the IEEE International Conference on Robotics and Automation*, Seoul, Korea, 2001, pp.27-32.

[8] Z. Yao, "Finding efficient robot path for the complete coverage of a known space," in Proc. of the IEEE International Conference on Robotics and Automation, Beijing, China, 2006, pp.3369-3374.