DOI: XXX-XXX-XXXX

Scan Matching Online Cell Decomposition for Coverage Path Planning in an Unknown Environment

Batsaikhan Dugarjav[1], Soon-Geul Lee[1,#], Donghan Kim[2], Jong Hyeong Kim[3] and Nak Young Chong[4]

1 School of Mechanical Engineering, Kyung Hee University, Seocheon-dong, Giheung-gu, Yongin, 446-701, South Korea
2 School of Electronic Engineering, Kyung Hee University, Seocheon-dong, Giheung-gu, Yongin, 446-701, South Korea
3 School of Mechanical System Design Engineering, Seoul National University, Sillim-dong, Gwanak-gu, Seoul, 151-742, South Korea
4 School of Information Science, Japan Advanced Institute of Science and Technology, Asahidai, Nomi, Ishikawa 923-1292, Japan

Corresponding Author / E-mail: sglee@khu.ac.kr, TEL: +82-031-201-2506, FAX: +82-031-201-2506

KEYWORDS : scan matching, sensor-based online incremental cell decomposition, oriented rectilinear decomposition complete coverage, path planning

This paper presents a novel sensor-based online coverage path-planning algorithm that guarantees the complete coverage of an unknown rectilinear workspace for the task of a mobile robot. The proposed algorithm divides the workspace of the robot into cells at each scan sample. This division can be classified as an exact cell decomposition method, which incrementally constructs cell decomposition while the robot covers an unknown workspace. To guarantee complete coverage, a closed map representation based on a feature extraction that consists of a set of line segments called critical edges is proposed. In this algorithm, cell boundaries are formed by extended critical edges, which are the sensed partial contours of walls and objects in the workspace. The robot uses a laser scanner to sense the critical edges. Sensor measurement is sampled twice in each cell. Scan matching is performed to merge map information between the reference scan and the current scan. At each scan sample, a two-direction oriented rectilinear decomposition is achieved in the workspace and presented by a closed map representation. The construction order of the cells is very important in this incremental cell decomposition algorithm. To choose the next target cell from candidate cells, the robot checks for redundancy in the planned path and for possible positions of the ending points of the current cell. The key point of the algorithm is memorizing the covered space to define the next target cell from possible cells. The path generation within the defined cell is determined to minimize the number of turns, which is the main factor in saving time during the coverage. Therefore, the cell's long boundary should be chosen as the main path of the robot. This algorithm is verified by an experiment under the LABVIEW environment.

Manuscript received: August XX, 201X / Accepted: August XX, 201X

1. Introduction

The task of covering a bound region of space is common to numerous applications. For example, cleaning robots are designed to automatically clean indoor workspaces. Cell decomposition is often employed in solving coverage problems. In this method, a target position must be reached without colliding with obstacles while the robot is moving. The main idea of cell decomposition is to decompose a given bound workspace into a set of non-overlapping regions. Each region is termed cell. The combination of these regions covers or approximates a subset of interest from the workspace, namely, the regions not occupied by obstacles.

In the early research of cellular decomposition, a popular technique that yields a complete coverage path solution is the trapezoidal decomposition.[1] In this technique, the robot's free space is decomposed into trapezoidal cells. The coverage for each cell can be easily achieved with simple back-and-forth motions because each cell is a trapezoid. Unfortunately, the trapezoidal approach has too much redundancy to guarantee complete coverage. To overcome this problem, there was introduced the boustrophedon decomposition, which was an enhancement of the trapezoidal decomposition.[2] This method was designed to minimize the number of excess lengthwise motions by merging narrow cells into one cell. Another development of the cell decomposition technique is the Morse decomposition algorithm presented by Acar,[3] where the location of cell boundaries is indicated by using Morse functions. Instead of analyzing the vertices of the given workspace, Acar's algorithm looks for connectivity changes of the slice in the free space to locate the cell boundaries. These decomposition methods are formed by sweeping a line over the known workspace. A new cell boundary is created whenever the sweep line encounters a vertex. The performance analysis of the coverage path planning based on cell decomposition is investigated,[4] where the algorithm converts the coverage path planning problem into a flow network problem by exact cellular decomposition. The key point of this algorithm was to determine the minimum cost path from the start node to the final node by visiting every node exactly once in the given flow network for cell decomposition.

The robot can decompose the cells in an initially unknown workspace by using incoming sensor data to plans its path and executes coverage. This task is called sensor-based coverage. It directs a robot operating in such workspaces to explore every point of the workspace. The method that presented by Choset incrementally generates an explicit representation of the workspace in a graph as it performs coverage, and uses the graph to direct coverage and to decide when a task is completed.[5] This method creates a set of cells that collectively covers the environment while simultaneously exploring each cell. Additional aspects on the general idea of incremental cell decomposition can be found in other publications.[6-8] These algorithms were based on terms of critical point of Morse function. To achieve coverage in an unknown workspace, the robot simultaneously covered the workspace and incrementally constructed the Reeb graph representation that represents the critical points as nodes and cells as edges. Another algorithm, the hierarchical decomposition for coverage with an extended range detector, was introduced.[8,9] It combines Morse decompositions and the Voronoi diagram. Also, the critical-point sensing method was developed.

To guarantee coverage in grid map representation, approximate cellular decomposition methods, including spiral-STC,[10] backtracking spiral algorithm[11] and linked spiral path,[12] are all based on low-resolution grid map representation, where the cell size is set to the size of the sweeping tool. Also, high-resolution grid map representation based approach is presented,[13] where the cell size is smaller than the tool size. Other article is proposed sector-based online coverage algorithm.[14] In this method, Target workspace is decomposed by sector that is a flexible region. Sectors are created incrementally as the robot systematically explores the unknown workspace, primarily by wall following. Hyun focused on extracting virtual door to achieve cell decomposition.[15] In this paper, the virtual doors are extracted by combining a Generalized Voronoi Diagram (GVD) and a configuration space eroded by the half of the door size, and the region to region cleaning algorithm is also proposed based on the closing and opening operations of virtual doors. These approaches have considerable advantages, but also significant disadvantages, where sector is extracted after wall access then new sector is decomposed in to cells, and the secter can improve cleaning effieciency but it tends to increase redundancy for coverage path depending on overlapped areas of the sectors.

In this paper, a sensor-based a robot coverage algorithm that combines scan matching and oriented rectilinear decomposition (ORD) is presented and is used to guarantee complete coverage. Scan matching is an efficient tool used to incrementally explore unknown workspace for problems of simultaneous localization and mapping (SLAM).[16-19] First, we assumed the major structures of the indoor workspace is sets of lines that are parallel or perpendicular to each other as in the literature.[20] Second, the robot can create cell while it covering the workspace. To incrementally construct cell decomposition, scan matching and cell decomposition method are combined to create an online cell decomposition for the coverage task. When collecting data from the workspace, the robot merges the pervious map and current data by scan matching. Scan matching is done by Hough scan matching, and then the robot extracts the line feature from the updated map. The extracted line features are then align position and orientation and used to compose cells. After creating cells, the robot covers it by employing template based path planning.

This paper is organized as follows: Section 2 introduces the map representation, feature extraction and incremental map building based on scan matching. Section 3 presents the complete coverage path planning based on scan matching and ORD. Section 4 and 5 demonstrate the simulation and experimental implementation, and we conclude in Section 6.

2. Closed map representation

In the proposed method, the robot builds a feature map using a laser-range finder (LRF) sensor in real time. In the beginning of the mapping process, the current position of the robot is initialized with (x, y, rotation) = (0, 0, 0°). The robot localization in the cell is obtained by particle filter algorithm. A laser scanned data is gathered and used to build the map. To obtain feature map information M_i, preprocessing such as filtering, feature extraction, scan matching, and map merging should be used. LRF's scan is sampled twice in the cell as same as assumed in the previous work[21] where the RUN1 event is defined as the longitudinal one while the RUN2 event is perpendicular to RUN1 for robot motion as depicted in Fig. 1. When the RUN1 or RUN2 event occurs, new slice information of the workspace is obtained as Fig. 2. The map representation of the workspace is updated between these two events and it can be expressed by Eq. (1).

$$M_i = M_{i-1} \cup \{ d'_{i,j} \mid \nexists m_{i-1,k} \in M_{i-1} : \| d'_{i,j} - m_{i-1,k} \| < d_{th} \}, \quad (1)$$

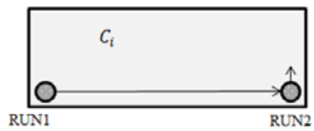

Fig. 1. RUN1 and RUN2 events for the robot motion in the cell.

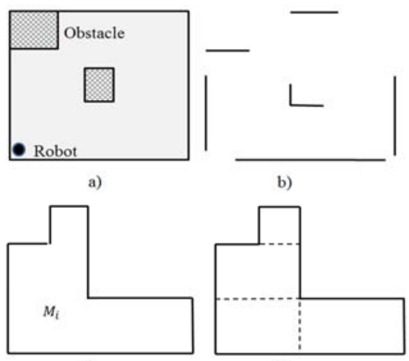

Fig. 2. Map representation at the i-th scan sample: a) Initial state of the workspace, b) Unclosed map representation, c) Closed map representation, d) Sweep invariant decomposition.

DOI: XXX-XXX-XXXX

where $d'_{i,j}$ is the new data points in the i- th collected data D_i and d_{th} is the threshold value for map merging. The scan matching technique effectively combines the current data D_i and the map M_{i-1} to get new map representation. Scan matching procedure is consecutive correction of the scanned recording position, i.e. the current position of the robot. In this way, the current estimation of the scanned recording position is updated and an incremental map is built.

2.1 Scan matching

Scan matching technique is used in various robotic applications for localization and for SLAM. The scan matching is a popular way of recovering the mobile robot's motion, and it can be defined as finding the translation $\mathbf{T}$ and rotation $\mathbf{R}_\theta$ between two consecutive scans. Hough scan matching (HSM) method whose main concept was introduced by Cenci[22] is used to merge maps. This method is novel in many aspects, and has various advantages. It produces a set of ranked hypothesis rather than a single one and is capable of merging maps quite less time than iterative approaches. The Hough transform (HT) exploits the fact that lines can be represented through the polar equation:

$$p = x\cos\theta + y\sin\theta\,, \tag{2}$$

where p is the distance of the line to the reference frame origin, and θ is the angle between the x axis and normal from origin to the line. Generally, HSM is performed through the following steps:

1. Discrete HT (DHT) and Hough spectrum (HS) are extracted from a reference scan and a current scan, respectively.
2. Local maxima of the spectra cross correlation are used to generate hypotheses on ϕ.
3. For each hypothese ϕ, linear constraints for $|\mathbf{T}|$ are produced by correlating columns of the HT.

For efficient implementation, line detection is performed using DHT, which can be expressed as:

$$DHT_{\theta,\rho} = \{(x_i, y_i);\quad \rho \le x\cos\theta + y\sin\theta < \rho + \Delta\rho\}\,, \tag{3}$$

where $\Delta\rho, \Delta\theta$ are discrete and finite.

Once DHT is computed, the HS is defined as the vector with n_θ elements obtained by summing column wise the squared values of

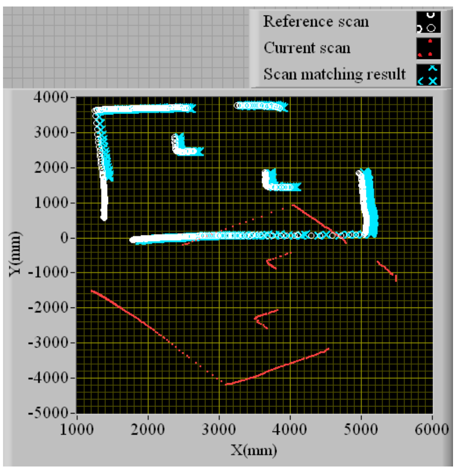

Fig. 3. Hough scan matching: Reference scan and current scan are matched.

DHT. If the DHT result is stored in a matrix $\mathbf{H}$ with columns n_θ and row n_p, then HS is obtained by the relationship

$$HS(k) = \sum_{i=1}^{n_\rho} H^2(i,k)\,,\ \ 1 \le k \le n_\theta\,. \tag{4}$$

We can estimate ϕ by correlation of HS of the current scan and reference scan. The correlation can be computed as follows:

$$CC(\phi) = \sum_k HS_C(\theta) * HS_R(\theta - \phi)\,, \tag{5}$$

where $1^\circ \le \theta \le 360^\circ,\ 1^\circ \le \varphi \le 360^\circ$. The corresponding rotating angle φ_0 for the translation is computed by taking the angle that shows the maximum cross correlation between the HS of the reference scan and the current scan as Eq. (6).

$$\phi_0 = \arg\max_\phi CC(\phi)\,, \tag{6}$$

To find the magnitude of the translation $|\mathbf{T}|$, the cross correlation between the distance spectra of the reference scan and the current scan should be considered as given by Eq. (7):

$$DC(k) = \sum_k DS_C(d) * DS_R(d-k)\,, \tag{7}$$

where $0 \le d \le D, 0 \le k \le D$, and DS_C, DS_R are the distance spectra of the reference scan and the current scan, respectively. Finally, translation $|\mathbf{T}|$ is derived as follows:

$$|\mathbf{T}| = \arg\max_k DC(k)\,. \tag{8}$$

Then, the translations in the x and y axes are calculated as relationship Eq. (9):

$$\begin{aligned} \mathbf{T}_x &= |\mathbf{T}| * \cos\phi_0 \\ \mathbf{T}_y &= |\mathbf{T}| * \sin\phi_0 \end{aligned}\,. \tag{9}$$

Finally, the scanned recording position is updated by following relationships:

$$\begin{bmatrix} X_{k+1} \\ Y_{k+1} \\ \phi_{k+1} \end{bmatrix} = \begin{bmatrix} X_k \\ Y_k \\ \phi_k \end{bmatrix} + \begin{bmatrix} \mathbf{T}_x\cos(\phi+\phi_0) - \mathbf{T}_y\sin(\phi+\phi_0) \\ \mathbf{T}_x\sin(\phi+\phi_0) + \mathbf{T}_y\cos(\phi+\phi_0) \\ \phi_0 \end{bmatrix}. \tag{10}$$

Fig. 3 shows that HSM result where LRF scanned twice at different positions and current scan matched to the reference scan.

The comparison among scan matching techniques, which are iterative closest points (ICP), HSM, and Histogram, was performed with datasets of scan pairs obtained from real environments using a laser scan sensor. The computation time of the comparison for each dataset are shown in Table 1. Obviously, Histogram method is much faster than ICP and HSM. Fig. 4 shows the error components of each scan matching in x, y direction and in rotation angle. Histogram method shows the smallest error for x-direction and angle. Similarly, HSM shows almost same errors for x and y direction. ICP shows the largest errors for x-direction and angle.

Table 1. Comparison of the computational time.

Type	Computation time (sec)
ICP	0.56
HSM	1.2
Histogram method	0.02

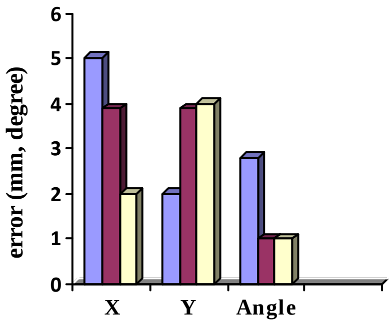

Fig. 4. Error comparison among scan matching methods.

2.2 Feature extraction

Feature extraction involves simplifying the amount of resources required to accurately describe a large set of data. By definition, features are the recognizable structures of the geometric primitives in a given workspace. They can usually be extracted from sensor measurement. Line extraction is used to explore a closed map that contains a set of horizontal and vertical lines. In this step, an initial estimate of the line parameters in Eq. (12) is calculated for each data segment from the segmentation process. This step aims to identify and separate sets of segments related to interesting targets such as walls, a desk, or a person. Line extraction can be performed after segmentation, and the split and merge algorithm is used to obtain critical edges because it is probably the most popular and the fastest among other algorithms.[23] The main idea of the split and merge algorithm is shown in Fig. 5 and the flowchart is described in Fig. 6. Segments can be expressed by the following representation:

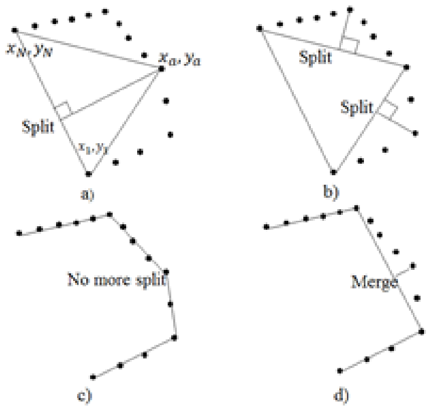

Fig. 5. Split and merge algorithm.

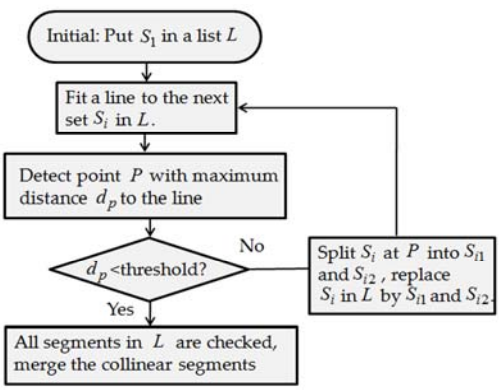

Fig. 6. Flowchart of the split and merge algorithm.

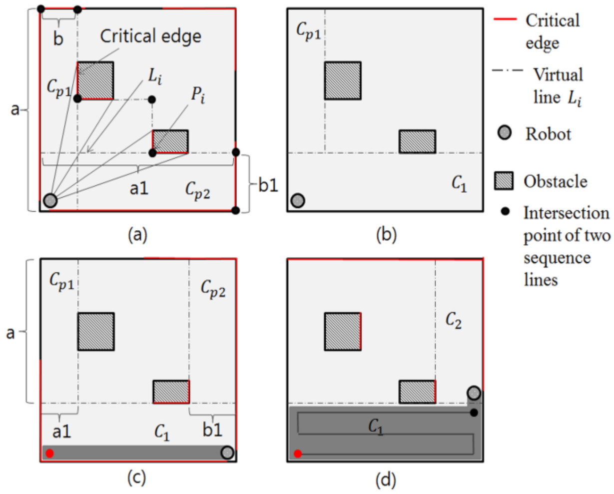

Fig. 7. Sensor-based incremental cell decomposition algorithm.

$$S_i = \{(x_i, y_i), i = k:n\}, 1 \le k < n < N, \tag{11}$$

where (x_i, y_i) is 2-D Cartesian coordinate, k and n denote the start index and the end index of the segment, respectively. The least square line fitting is given by Eq. (12) and Eq. (13). The general form of the line is:

$$ax + by + c = 0. \tag{12}$$

The best estimated line coefficients are calculated by follows:

$$\begin{aligned}\hat{a} &= \sum_{i=1}^{N} x_i \sum_{i=1}^{N} y_i^2 - \sum_{i=1}^{N} y_i \sum_{i=1}^{N} x_i y_i \\ \hat{b} &= \sum_{i=1}^{N} y_i \sum_{i=1}^{N} x_i^2 - \sum_{i=1}^{N} x_i \sum_{i=1}^{N} x_i y_i \\ \hat{c} &= \left(\sum_{i=1}^{N} x_i y_i\right)^2 - \sum_{i=1}^{N} x_i^2 \sum_{i=1}^{N} y_i^2\end{aligned}. \tag{13}$$

3. Online cell decomposition for coverage path planning

3.1 Overview of the previous work

This section introduces an online cell decomposition method that combines a scan matching and the cell decomposition based on the two directions ORD to generalize the online algorithm. This online cell decomposition was introduced in our previous work.[21] This method can be summarized as follows: if a robot is equipped with an LRF, it can sense the contours of walls and/or objects in the workspace, which are termed critical edges in Fig. 7. These edges are composed of points measured by the LRF and are represented as lines. To create new cell boundaries, a line extraction is used to identify the critical edges. After defining the critical edges and each intersection point of two adjacent critical edges the virtual line L_i corresponding to the critical edge of an obstacle is extracted to determine cell boundary as shown in Fig. 7. By using alternatives to create cell boundaries, the proposed algorithm then decomposes the workspace into cells.

3.2 Two-direction oriented rectilinear decomposition based on scan matching

DOI: XXX-XXX-XXXX

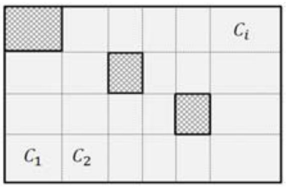

Fig. 8. Example of the sweep invariant cell decomposition.

In this paper, another method is used to completely decompose the workspace of the robot. One technique used to uniquely decompose a known rectilinear workspace is the sweep invariant decomposition (SID).[24, 25] The general idea of the SID is that the workspace or the boundary (critical edge as termed[21]) is extended until a perpendicular wall is reached. Therefore, cell boundaries can be defined by extended critical edges. As an example, cell decomposition that uses SID is illustrated in Fig. 8 where a rectilinear workspace is separated into number of cells $C_1, C_2, ..., C_i$. Although the SID is successfully achieved, the next process, such as coverage path planning, cannot be processed automatically because a boundary of the two adjacent cells can be inefficiently revisited by the robot. This means that time required to complete the coverage task would be taken more than expected. To improve the coverage efficiency, a smaller cell should be merged into an adjacent cell.

In order to employ SID in real time applications, our algorithm consolidates SID with scan matching and it allows the robot to perform coverage and celling decomposition simultaneously. Exact cell decomposition algorithm used in this research generates the cell directly after several pre-processes while the robot is covering because pre-processes are needed to reduce the noise and uncertainty of map observation. As shown in Fig. 1, scanning, that is the process of getting workspace information, is carried out when RUN1 and RUN2 events occur once in the cell. Thus, sensor measurement is only sampled twice in each cell. The scan matching is conducted to merge map information between RUN1 and RUN2 events. On the other hand, the closed map representation is updated twice in each cell by Eq. (1). After the first scan is taken at the starting point, the parts of the critical edge of the workspace are

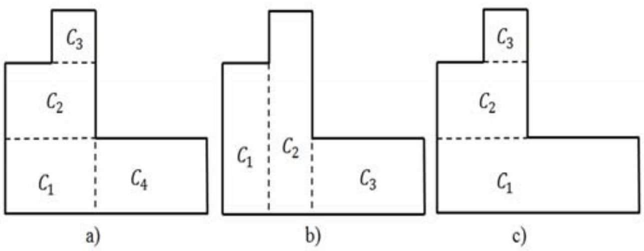

Fig. 9. SID versus vertical-direction ORD and horizontal-direction ORD.

a) SID algorithm, where cells C_1, C_2, C_3 or C_1, C_4 should be merged into one cell referred to as the current cell.

b) Vertical direction ORD, where the current cell is selected by C_1. The longitudinal direction is the vertical axis.

c) Horizontal direction ORD, where the current cell is chosen by C_1 with a horizontal longitudinal direction.

determined by line extraction. Each intersection point of two adjacent critical edges is obtained to generate a closed map representation M_i. Then, the robot can have a map of the workspace from the closed map representation and cell decomposition is easily to be done.

The SID is modified by a two-direction ORD which is achieved by extending both horizontal and vertical edges to indicate the cell boundaries. Two-direction ORD will also yield the same results as SID for the cell decompose and ORD has advantage that cell merging process is not necessary to ORD. Fig. 9 shows process of the SID and the ORD. After all the vertical critical edges and all the horizontal critical edges are extended, the robot chooses an efficient direction for the extending critical edge to compose the current cell. For instance, the robot is assumed to start from any corner of the workspace. The laser scanner senses the critical edges of the workspace. The robot explores the closed map with cells C_1, C_2, C_3 using two-direction ORD as shown in Fig. 10(a, b). To determine the first cell to be covered, the robot compares the lengths and widths of the candidate cells in each direction. The robot is assumed to be aware of the information for all cells in order to compare the dimensions. The longer and wider cell is chosen first to be covered as shown in Fig. 10(b) and, in this case, the longitudinal direction is the horizontal axis. The robot plans a coverage path for the current cell C_1 with back-and-forth motions that cause the ending point of the cell. The cell C_2 at horizontal direction is going to be the next possible cell.

After the robot travels along the longer boundary of the cell and reaches one of its corners, or until the RUN2 event occurs, the robot can gather information about the next possible cells. The robot plans a coverage path again and that path eventually comes to the ending point. Then, the robot decides the path which includes few or no redundant coverage depending on an ending point. The next target cell C_2 is chosen from possible candidates by the ending point of the current cell as shown in Fig. 10(c). When the robot finishes covering the current cell, it goes to the next target cell.

Then, the RUN1 event occurs and the laser scanner gathers information about the workspace. The closed map representation is updated by scan matching as depicted in Fig. 10(e). The algorithm maintains three lists: D (discovered cells), C (the current cell)

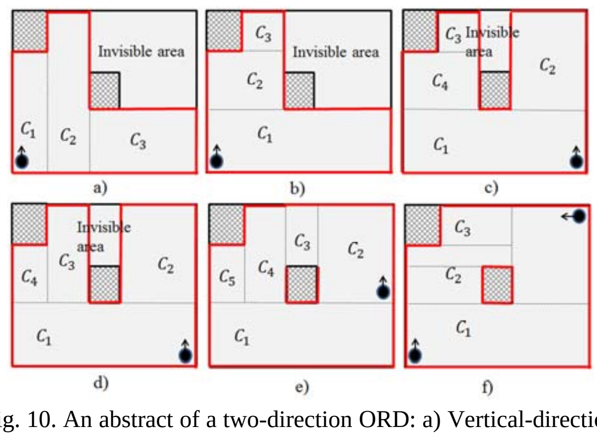

Fig. 10. An abstract of a two-direction ORD: a) Vertical-direction ORD. b) Horizontal-direction ORD. c), d), e), f) Closed map was updated during RUN1 and RUN2 events.

and C_{pi} (next candidate cells). The list D stores all discovered cells that were visited and the list C contains the cell that the robot is currently covering. Once the robot finishes covering the current cell, the current cell is removed from list C and added to the list D, and the list C_{pi} stores the next candidate cells for the one-scan sample. The robot needs to memorize which cell is currently covering and which ones have already covered. The algorithm is shown in Fig. 11.

4. Simulation and Experiment

4.1 Simulation

The proposed algorithm for coverage path planning is verified by simulation with two different configurations whose workspace area is 5m × 5m and has two rectangular obstacles. Fig. 12 shows each step of complete coverage path planning simulation after Fig. 7, while the robot performs incremental cell decomposition. It shows that the next target cell is dynamically selected according to the ending point of the current cell, and the planned path should include few numbers of turns for each cell. Therefore, the longer side of the cell should be chosen as the main path of the robot as C_2 of Fig. 12(a). The number of turns required to cover the cell is the main factor to save time. For the coverage algorithm, time efficiency is one of the key factors which should be considered to improve the performance of the algorithm. Simultaneously, when RUN1 and RUN2 events occur, the robot scans around and repeats the same procedures; so incremental construction of the boustrophedon decomposition can be achieved. If there is a no remaining space, the algorithm will be terminated. This means that the algorithm can guarantee complete coverage so that the robot passes over all free space in the given unknown workspace. Fig. 12(d) shows completed simulation result for the given unknown workspace. You can see sequence of the cells, $C_1 \rightarrow C_2 \rightarrow C_3 \rightarrow C_4 \rightarrow C_5 \rightarrow C_6 \rightarrow C_7$, that have been covered by the robot.

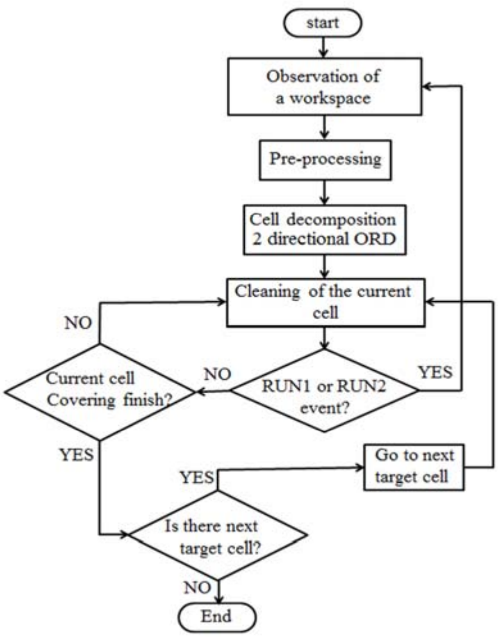

Fig. 11. Complete coverage path planning algorithm based on online two-direction ORD

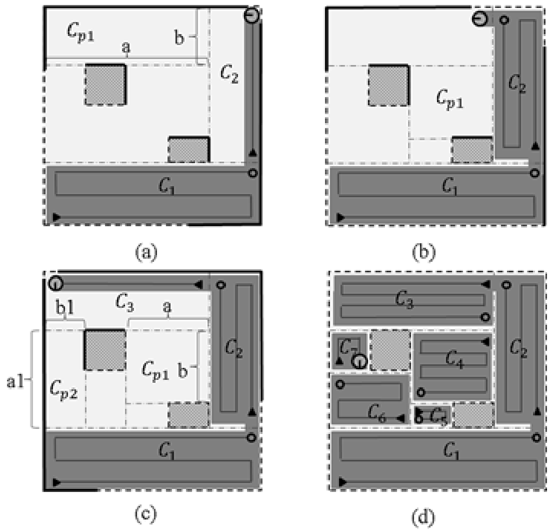

Fig. 12. Implementation sketch of the algorithm.

The intermediate results of the algorithm are shown in Fig. 13, the robot starts from the origin of the workspace's reference frame that is denoted as a red solid circle and completely covers the whole workspace with apparently time efficient paths for both simulations. In the figure, the black solid rectangles represent obstacles, the red dotted rectangles denote decomposed cells and the black solid line is the path of the robot.

4.2 Experiment

To evaluate proposed online algorithm, we loaded the algorithm onto differential-driven mobile robot (X-bot from Yujin Robot in Fig. 14) and conducted several experiments in real workspace of the robot. The robot equipped with the LRF sensor in order to recognizing indoor environmental components. Sensors and robot controllers are interfaced to the remote supervisory control computer with a wireless USB hub and the main control code is written in VI of LabView. Maximum speed of the robot is 30 cm/sec in the experiment. For the first experiment, one obstacle placed center of the workspace. And then the robot easily performed online cell decomposition and covered the each cell.

Fig. 15 shows the result of first experiment. In the second experiment, we added some obstacles in the workspace. In Fig. 16, the red solid rectangles are representing obstacles, the blue solid lines are indicating boundaries of the decomposed cells, and the black

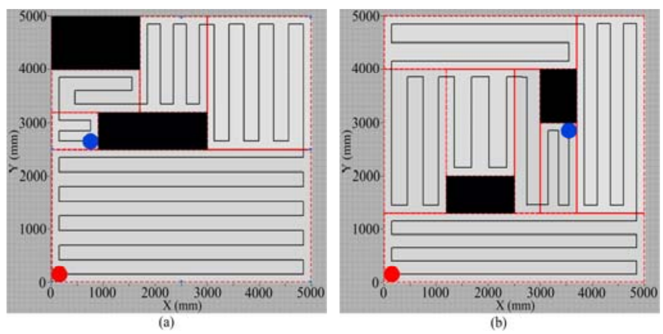

Fig. 13. Simulation results.

DOI: XXX-XXX-XXXX

Fig. 14. Mobile robot system for the experiment.

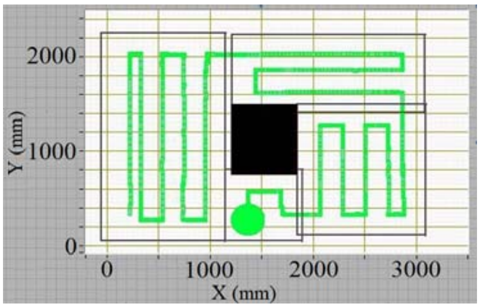

Fig. 15. First experiment result.

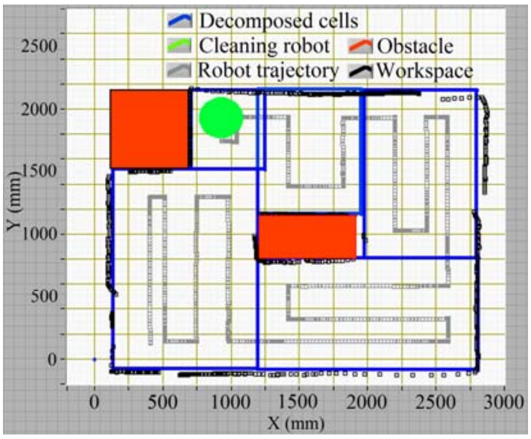

Fig. 16. Experiment result of the proposed online coverage algorithm.

dotted line is showing critical edge of the workspace those are from scan matching. The grey dotted line and the green solid circle are representing coverage path of the robot and robot, respectively. The robot covered about 99.7% of the its workspace within 3.5 min. Finding closed map representation was more difficult than in the first experiment because one more obstacles were placed.

However, it apparently looks good in the Fig. 16. As we already mentioned about two-direction ORD based on scan matching in section 3.2, online cell decomposition was performed incrementally, covering order of the cell is dynamically selected according to the ending point of the current cell, and the planned path included few number of turns for each cell. You can see from Fig. 16. The experimental result of the proposed algorithm is compared in Table 2 with the results of other algorithms used in each of [13], [14] and [24]. The sizes of the workspaces in [13], [14] and [24] are 49.2 m^2, 21.5 m^2 and 21.5 m^2, respectively. The dimension varies considerably that direct comparison of the workspaces provides no relevant result, but the coverage rates can be compared. The higher value of the coverage rate means that it is more efficient. The total time cannot be normalized because the maximum robot velocities in the previous studies are different from each other and the speed of the robot changes during the experiment. But we can say that the total time of the proposed algorithm is efficient than others based on assumption comparing workspace to be covered. For example, workspaces of [14] and [24] are approximately 3 times bigger than our experimental area. If our algorithm is applied to the same workspace area as [14] and [24], the equivalent total time will be 3 times bigger from the current one. However, it is still smaller than the total time of other algorithm.

Table 2. Comparison of the performance of the algorithms.

	Total time	Coverage rate	Ratio of the workspace
Result of the proposed algorithm	3.5 min	99.7%	6.05 m^2
Simulation result of [13] (Sweeping tool size is 5cm.)	15.8 min	98%	49.2 m^2
Experimental result of [14]	16 min	99.5%	21.5 m^2
Experimental result of [24]	20 min	99%	21.5 m^2

5. Conclusions

This paper presented an online two-direction oriented rectilinear decomposition based on scan matching. Most of the previous studies on cell decomposition are conducted with the assumption that information about the workspace is initially given. Although few of those studies develop a cell decomposition algorithm for an unknown workspace, their algorithms are based on ultrasonic sensors or contact sensors. In those studies, the robot can directly recognize the critical points with the use of sensors while traveling without any information about the possible cell. That is, if the robot detects a critical point, new cells are created unconditionally. The proposed algorithm in this research incrementally (one by one) decomposes an unknown workspace into cells based on closed map representation. A new closed map representation is created after identifying the critical edges, and two-direction ORD is achieved to create new cell. Hence, this algorithm is applicable and efficient for the coverage path planning of a mobile robot. Specifically, it can reduce the number of times the robot revisits cells that were already covered. The robot also plans its path inside the cell with a fewer number of turns, which means that coverage time for a single cell is more efficient. Based on the assumptions above, the coverage time for the entire workspace can be optimized.

ACKNOWLEDGEMENT

This research was partially supported by the IT R&D program of MKE/KEIT [KI10040990, A Development of Communication Technology with UTIS & Vehicle Safety Support Service for Urban Area] and by the Implementation of Technologies for Identification, Behavior, and Location of Human based on Sensor Network Fusion Program through the MKE (Grant Number 10041629). It was also supported by the Technology Innovation Program (10040992) of MKE/KEIT.

REFERENCES

1. Latombe, J. C., “Robot motion planning,” Kluwer Academic Publishers, Boston, 1991.
2. Choset, H., “Coverage of known space: the boustrophedon cellular decomposition,” Int. J. of Autonomous robots, Vol. 9, pp. 247-253, 2000.
3. Acar, E., Choset, H., Rizzi, A. A.., Atkar, P. N. and Hull, D., “Morse decompositions for the coverage task,” Int. J. of Robotics Research, Vol. 21, pp. 331–344, 2002.
4. Janchiv, A., Batsaikhan, D., ByungSoo, K., Won Gu, L. and Soon-Geul, L., “Time-efficient and Complete Coverage Path Planning Based on Flow Networks for Multi-Robots,” Int. J. of Control Automation and Systems, 2013.
5. Choset, H., Acar, E., Riazi, A. and Luntz, J., “Exact cellular decompositions in terms of critical points of morse functions,” Int. Conf. of Robot Automation, Vol. 3, pp. 2270-2277, 2000.
6. Acar, E. and Choset, H., “Robust sensor-based coverage of unstructured environment,” Int. Conf. on Intelligent Robots and Systems, Vol. 1, pp. 61-69, 2001.
7. Acar, E. and Choset, H., “Sensor-based coverage of unknown environments: incremental construction of Morse decompositions,” Int. J. Robotics Research, Vol. 21, pp. 345-366, 2002.
8. Choset, H. and Burdick, J., “Sensor-based motion planning: incremental construction of the hierarchical generalized Voronoi graph,” Int. J. of Robotics Research, Vol. 19, No. 2, pp. 126–148, 2000.
9. Acar, E., Choset, H. and Ji Yeong, L., “Sensor-based coverage with extended range detectors,” IEEE Transactions on Robotics, Vol. 22, No. 1, 2006.
10. Gabriely, Y. and Rimon, E., “Spiral-STC: an on-line coverage algorithm of grid environments by a mobile robot,” Int. Conf. on Robotics and Automation, pp. 954–960, 2002.
11. Gonzalez, E., Alvarez, O., Diaz, Y., Parra, C. and Bustacara, C., “BSA: a complete coverage algorithm,” Int. Conf. on Robotics and Automation, pp. 2040–2044, 2005.
12. Young-Ho, C., Tae-Kyeong, L., Sanghoon, B. and Se-Young, O., “Online complete coverage path planning for mobile robots based on linked spiral paths using constrained inverse distance transform,” Int. Conf. on Intelligent Robots and Systems, pp. 5788–5793, 2009.
13. Tae-Kyeong, L., Sanghoon, B., Young-Ho, C. and Se-Young, O., “Smooth coverage path planning and control of mobile robots based on high-resolution grid map representation,” Int. J. Robotics and Autonomous Systems, Vol. 59, pp. 801-812, 2011.
14. Tae-Kyeong, L., Sanghoon, B. and Se-Young, O., “Sector-based maximal online coverage of unknown environments for cleaning robots with limited sensing,” Int. J. of Robotics and Autonomous Systems, Vol. 59, No. 10, pp. 698-710, 2011.
15. Hyun, M., Hae-min, J. and Woo-Yeon, J,. “Virtual door algorithm for coverage path planning of mobile robot,” Int. Symp. On Industrial Electronic, pp. 658-663, 2009.
16. Yoshitaka, H., hirohiko, K., Akihisa, O. and Shinichi, Y., “Map building for mobile robots using a SOKUIKI sensor robust scan matching using laser reflection intensity,” Int Joint Conf. on SICE-ICASE, pp. 5951-5956, 2006.
17. Heon-Cheol, Lee., Seung-Hee, Lee., Seung-Hwan, Lee., Tae-Seok, Lee., Doo-Jin, Kim., Kyung-Sik, Park., Kong-Woo, Lee. and Beom-Hee, Lee., “Comparison and analysis of scan matching techniques for cooperative-SLAM,” Int. Conf. on Ubiquitous Robots and Ambient Intelligence, pp. 165-168, 2011.
18. Diosi, A. and Kleeman, L., “Laser Scan Matching in Polar Coordinates with Application to SLAM,” Int. Conf. on Intelligent Robots and Systems, pp. 3317-3322, 2005.
19. Jacky, Chang, H., Lee, C. S. G., Yung-Hsiang, L. and Hu, Y. C., “P-SLAM : simultaneous localization and mapping with environmental structure prediction,” IEEE Int. Conf. on Robotics and Automation, Vol. 23, pp. 281-293, 2006.
20. Nguyen, V., Harati, A., Martinelli, A., Siegwart, R. and Tomatis, N., “Orthogonal SLAM: a step toward lightweight indoor autonomous navigation,” Int. Conf. on intellegent robots and systems, 2006.
21. Batsaikhan, D., Janchiv, A. and Soon-Geul, Lee., “Sensor-based incremental boustrophedon decomposition for coverage path planning of a mobile robot,” Int. Conf. on Intelligent Autonomous Systems, Vol. 193, pp 621-628, 2012.
22. Censi, A., Iocchi, L. and Grisetti, G., “Scan matching in the Hough domain,” Int. Conf. on Robotics and Automation, pp. 2739-2744, 2005.
23. Nguyen, V., Martinelli, A., Tomatis, N. and Siegwart, R., “A comparison of line extraction algorithms using 2D Laser Range Finder for indoor mobile robotics,” Int. Conf. on Intelligent Robots and Systems, pp. 1929-1934, 2005.
24. Sanghoon, B., Tae-Kyeong, L., Se-Young, O., member., IEEE. And Kwangro, J., “Complete coverage path planning of robotic vacuum cleaners using low cost sensors in an unknown environment,” Int. Conf. on Intelligent Robots and Systems, 2009.
25. Butler, Z. J., Rizzi, A. A. and Hollis, R. L., “Cooperative coverage of rectilinear environments,” Int. Conf. on Robotics and Automations, Vol. 3, pp. 2722-2727, 2001.