2013 13th International Conference on Control, Automation and Systems (ICCAS 2013)
Oct. 20-23, 2013 in Kimdaejung Convention Center, Gwangju, Korea

An online complete coverage approach for a team of robots in unknown environments

Hoang Huu Viet*, SeungYoon Choi, and TaeChoong Chung

Department of Computer Engineering, Kyung Hee University, Gyeonggi, 446-701, Korea
(Tel : +82-031-201-2952; E-mail:{viethh,sychoi84,tcchung}@khu.ac.kr) *Corresponding author

Abstract: This paper presents a novel approach to deal with the online complete coverage problem for a team of robots in unknown environments. In our approach each robot covers an unvisited region using a single boustrophedon motion until a robot reaches an ending point, which is surrounded by covered positions or obstacles. At the ending point the robot detects backtracking points based on the accumulated knowledge, plans the shortest backtracking path to the next starting point based on the proposed Theta* with multi-goals. Then, it follows the planed path to the next starting point to cover the next unvisited region. The robot team finishes the coverage task when no backtracking point is detected. Computer simulations show that our proposed approach is efficient for the complete coverage task of a robot team in terms of the coverage rate and the coverage path length.

Keywords: Cleaning robots, robot team, path planning, Theta* algorithm.

1. INTRODUCTION

The complete coverage is an important task of service robots due to its wide range of applications such as floor cleaning, lawn mowing, mine hunting, harvesting, and so forth. For the cleaning task in small domestic environments such as homes, workplaces, restaurants, etc., it is appropriate to use a single robot. However, for large environments such as supermarkets, train stations, or airports, a team of robots can accomplish the coverage task more quickly than a single robot can by dividing the task into sub-tasks and executing them concurrently in application domains where the tasks can be decomposed. In addition, a multiple robot system may succeed in face of failures, since even if some robots fail, the remaining robots can cover the entire workspace.

Multiple robots for cleaning task have attracted a lot of attention from researchers in recent years. Several researchers partition the accessible region in the workspace and allocate each robot to cover a sub-region. Min et al. [1] partition the whole accessible area of the workspace into a number of sub-regions with the same area. Afterward, each robot is assigned with a task to sweep autonomously an uncovered sub-region. The coverage mission of robots is finished if all the sub-regions are swept. Guruprasad et al.[2] propose a complete coverage of an initially unknown environment by multiple robots using Voronoi partition. The robots decompose the environment into Voronoi cells based on their initial positions. Each robot is responsible for covering its Voronoi cell using any existing algorithm for single robot coverage such as spanning tree coverage or boustrophedon coverage. To achieve provable completeness, approximate cellular decomposition of the environment has also been used for solving the coverage problem of multiple robots. Rekleitis et al. [3], [4] propose a coverage algorithm that uses the same boustrophedon decomposition as the single robot approach [12], but is extended to handle robot teams. Hazon et al. [5] present an online coverage algorithm for a team of robots based on an approximate cell decomposition. Several researchers have also extended coverage algorithms for a single robot to multiple robots. Hazon et al. [6], [7] extend the spanning tree-based coverage (STC) algorithm [8] to multi-robot spanning tree coverage (MSTC) algorithms, called non-backtracking MSTC and backtracking MSTC algorithm. Agmon et al. [9] present a coverage spanning tree for both online and offline coverage algorithms. Gerlein et al.[10] propose a multi-robot cooperative model extended from the backtracking spiral algorithm (BSA)[11].

In this paper we propose a computationally low-cost but efficient approach to deal with the online complete coverage task for a team of robots in completely unknown environments. The robots perform the coverage task based on the boustrophedon motions and Theta* algorithm [14], [15]. The boustrophedon motion mimics the simple back-and forth motion, in which the path is similar to an ox-plowing path in a field [12], [13], as illustrated in Fig. 1. This approach requires four regular conditions. First, the workspace must be closed. Second, the accessible regions are connected and can be reached by the robots from any initial positions. Third, the robot is capable of correctly detecting obstacles by using necessary sensors during the navigation time. Finally, the communication between the robots must guarantee that each robot can send its positions to the other robots. Our approach decomposes the coverage task of the robots into four main sub-tasks. First, each robot sweeps an uncovered region using a single boustrophedon motion until a robot reaches an ending point, which is surrounded by covered positions or obstacles. Second, the robot at the ending point detects backtracking points based on the knowledge accumulated from its coverage process and the other robots. Third, it plans the backtracking path

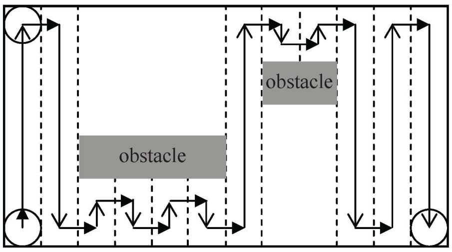

Fig. 1 The boustrophedon path found by a boustrophedon motion

to the next starting points based on our proposed Theta* with multi-goals. Finally, it follows the planed path to the next starting point of an unvisited region. This process is repeated until no backtracking point is detected. Computer simulations show that our approach is efficient for the complete coverage task of a team of robots in terms of the coverage rate and the coverage path length in unknown workspaces.

The rest of this paper is organized as follows: Section 2 describes a short review of the Theta* algorithm. Section 3 presents our approach. Section 4 discusses the simulation results. Finally, Section 5 concludes our work.

2. THETA* ALGORITHM

Theta* [14], [15] is a variant of the A* search [16] for any-angle path planning. Theta* takes a starting vertex s_s and a goal vertex s_g as inputs to find the shortest path connecting them through a sequence of consecutive accessible vertices. Theta* defines the cost of a vertex s by $f(s) = g(s) + h(s)$, where $g(s)$ is the length of the shortest path from vertex s_s to vertex s that has been found so far, and $h(s)$ is an estimated distance from vertex s to vertex s_g. It can be seen that $h(s)$ plays the role of guiding the direction of expanding the search space.

Unlike the A* search, in which the parent of a vertex must be a successor vertex, Theta* allows the parent of a vertex to be any vertex, thereby improving the solution path thanks to a smoothing process. Theta* inserts the smoothing process by checking the line-of-sight paths into the iterations of the searching process. To examine all of the candidate vertices that might be able to construct the shortest path requires that Theta* uses three lists of vertices: *open*, *closed*, and *parent.* The *open* list memorizes all of the neighboring vertices of the vertices that are already expanded. Any vertex in the *open* list is eliminated from the list if the vertex is expanded. The *closed* list stores all expanded points in order to make sure that no point is checked more than once. Meanwhile, the *parent* list stores the parent-child relations between the vertices for extracting the solution path when Theta* ends. Theta* is described in detail in Algorithm 1 [14], [15], where $c(s, s')$ is the Euclidean distance between two vertices s and s'. If steps 6.3 and 6.4 are ignored, Theta* becomes the A* search.

Algorithm 1: Theta* algorithm

Inputs : The starting vertex s_s and the goal vertex s_g
Outputs: The s_s-s_g path

1. Assign $open = \emptyset$, $closed = \emptyset$, $g(s_s) = 0$, $f(s_s) = g(s_s) + h(s_s)$, and $parent(s_s) = s_s$.

2. Add to $open$ the vertex s_s and $f(s_s)$.

3. If $open = \emptyset$, return "*no solution*".

4. Find a vertex s with the minimum $f(s)$ in the $open$ list. If $s = s_g$, return "*the s_s-s_g path*"; otherwise, take the following steps:

5. Remove vertex s from the *open* list and add it to the *closed* list.

6. Expand vertex s by checking each neighboring vertex s' of vertex s:

6.1. If $s' \in closed$, ignore it and goto step 3.

6.2. If $s' \notin open$, assign $g(s') = \infty$, and $parent(s') = s'$.

6.3. If there exists a line of sight from the $parent(s)$ to s', goto step 6.4; otherwise, goto step 6.5.

6.4. If $g(parent(s)) + c(parent(s), s') < g(s')$,

i) Assign $g(s') = g(parent(s)) + c(parent(s), s')$, $f(s') = g(s') + h(s')$, and $parent(s') = parent(s)$.

ii) If $s' \in open$, update $f(s')$ in the $open$ list; otherwise, add s' and $f(s')$ to the $open$ list.

6.5. If $g(s) + c(s, s') < g(s')$,

i) Assign $g(s') = g(s) + c(s, s')$, $f(s') = g(s') + h(s')$, and $parent(s') = s$.

ii) If $s' \in open$, update $f(s')$ in the $open$ list; otherwise, add s' and $f(s')$ to the $open$ list.

7. Goto step 3.

3. PROPOSED APPROACH

This section presents our solution for the online complete coverage problem using a team of robots in unknown environments.

3.1 Cooperation in the robots

Cooperation is one of the fundamental requirements in multiple robot systems. It refers to an information exchange between robots in order to achieve the goal of a team of robots. The robots in our approach, as mentioned earlier, perform two main sub-tasks: cover unvisited regions and backtrack to unvisited regions. Therefore, cooperation must be designed for each sub-task.

In covering task the cooperation enables robots to cover unvisited regions more efficiently. When each robot performs a boustrophedon motion to cover an unvisited region, it broadcasts its position to the other robots. By such way, each robot shares its accumulated knowledge to each other, i.e. each robot knows the model of the workspace, which is built by all of the robots. Let $\mathcal{M}$ be the model build incrementally when all of the robots perform the coverage task. When each robot moves to a uncovered cell, $\mathcal{M}$ is build as

$$\mathcal{M} = \mathcal{M} \cup \{s_k\}, k = 1, 2, \cdots, n, \tag{1}$$

where n is the number of the robots in the coverage system and s_k is the position of the robot k.

A conflict can appear when some robots perform the backtracking task while the other robots are performing boustrophedon motions. The cooperation enables the robots to avoid this problem. To avoid the conflict when robots choose the same backtracking point, before following the backtracking path, a robot broadcasts a message about its backtracking point to the other robots. To avoid the collision between robots when a robot is following the backtracking path, before moving to the next position, it broadcasts a message about the next position in the model $\mathcal{M}$ to the other robots. If there exist some robots that tend to move to that position, the robot with smallest identifier among the robots will get the highest priority and the other robots have to wait to get their priority. With these two cooperation mechanisms, conflict in the backtracking task is solved.

3.2 Construction of the boustrophedon motion

In this work each robot covers an unvisited region using a boustrophedon motion constructed based on the its sensors and the accumulated knowledge, i.e. the model $\mathcal{M}$. At each time step each robot moves to one of its four adjacent positions based on the four directions: north, south, east, and west, provided that the directions do not lead the robot into a *blocked position*, which is either an obstacle or a covered position. As a result, each robot covers an unvisited region while simultaneously constructing the model $\mathcal{M}$. Figure 2 shows an illustrative example, where three robots start at positions S1, S2, and S3. When robot 1 reaches the ending point E1, its knowledge $\mathcal{M}$ is region covered by itself and by the other two robots. Obviously, the knowledge $\mathcal{M}$ of robot 2 and robot 3 is the same as that of robot 1.

Algorithm 2 known as the BM algorithm is used for each robot to sweep an uncovered region using a single boustrophedon motion. In order to detect blocked position, each robot uses its sensors to identify obstacles, and uses the model $\mathcal{M}$ to identify the covered cells. The algorithm ends when the robot reaches an ending point.

3.3 Backtracking mechanism of robots

The workspace cannot be covered completely with only a single boustrophedon motion. When a robot reaches an ending point, it has to find and move to the closest unvisited region for the next coverage. The backtracking mechanism involves the determination of the

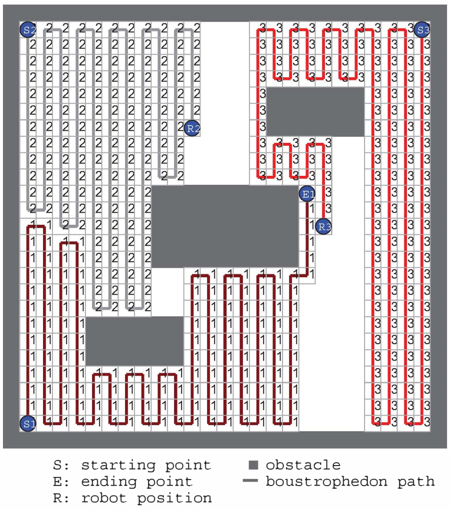

Fig. 2 The boustrophedon path found by a boustrophedon motion

---

Algorithm 2: BM algorithm for a single robot

Inputs : The robot position and the model $\mathcal{M}$
Outputs: The robot position and the model $\mathcal{M}$

1. Observe the first available direction in the priority of north-south-east-west.

2. If all directions are blocked then break the loop.

3. Move one step along the available direction.

4. Broadcast the robot's position to the other robots.

5. Receive the positions of the other robots.

6. Update the robots' position to the mode $\mathcal{M}$. Goto step 1.

---

backtracking points as the next starting points of uncovered regions and the shortest collision-free path as the backtracking path to the best starting point.

Considering that the workspace is unknown in advance to the robots, the starting point of the next boustrophedon motion must be detected from the positions that have been covered so far, i.e. from the model $\mathcal{M}$. In principle, a next starting point can be any element of the model $\mathcal{M}$ for which at least one cell in its four neighboring cells of directions, east, north, west, or south, is uncovered. For example shown in Fig. 2, when robot 1 arrives at the ending point E1, it can choose a next starting point on the edges of the covered regions. If so, the starting point on the edges of the covered regions can result in many boustrophedon motions, and thus lengthen the coverage path. Therefore, we propose that the next starting points should be chosen at the corners of the covered regions because a starting point at the corner of a boustrophedon region creates less fragments than a starting point on the edge of a boustrophedon region and therefore, the coverage path is shortened. More specifically, the starting points at the corners of the covered regions is proposed

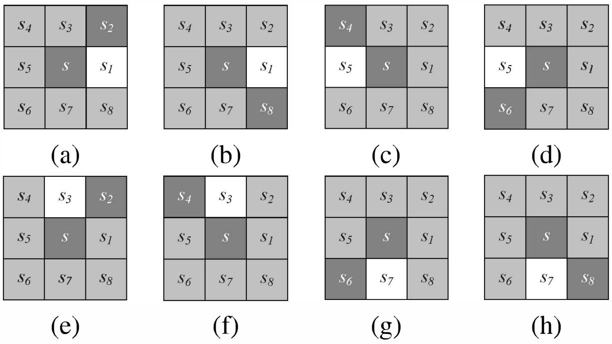

Fig. 3 Starting points at the corner of the covered regions

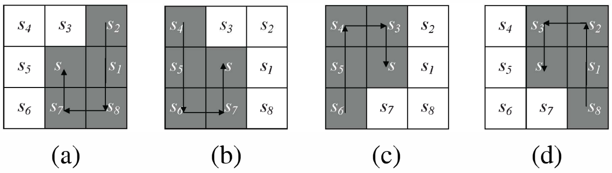

Fig. 4 Starting points around the robot

as in Fig. 3, where each black cell denotes a *blocked position*, each white cell denotes an *uncovered position*, and the other cells denotes a either blocked or uncovered position. For example, Fig. 3(a) shows that s_1 is a candidate of the next starting point. If we define a function $v(s_i, s_j)(i, j = 1, 2, \cdots, 8)$ that returns 1 if cell s_i is uncovered and cell s_j is blocked, and 0 otherwise, a starting point is taken under condition

$$N(s) = \begin{pmatrix} v(s_1, s_2) + v(s_1, s_8) + \\ v(s_5, s_4) + v(s_5, s_6) + \\ v(s_3, s_2) + v(s_3, s_4) + \\ v(s_7, s_6) + v(s_7, s_8) \end{pmatrix} > 1. \tag{2}$$

The set of the next starting points at the corners of the covered regions is determined as

$$S = \{s | s \in \mathcal{M} \text{ and } N(s) > 1\}. \tag{3}$$

It should be noted that while the robot at the ending point is determining the next starting points, some robots can be performing boustrophedon motions, the other robots can be backtracking to the next starting point. For the former case, the position of each robot is a blocked position and therefore, the set S contains it and the starting points surrounding it, as illustrated in Fig. 4. These positions must be eliminated from the set S. Let s be the position of the k^{th} robot, the position of the k^{th} robot and the starting points surrounding the k^{th} robot must be eliminated from the set S including $S_k = \{s, s_2, s_4, s_6, s_8\}$. For the latter case, the set S can contain the next starting point chosen by another robot. To avoid conflict when robots choose the same next starting point, it must be eliminated from the set S. Let P_l is backtracking path found by Theta* with multi-goals of the l^{th} robot, it means that the last element s_l of P_l, i.e. the next starting point of the l^{th} robot, must be eliminated from the set S.

In summary, when a robot reaches an ending point, the set of the starting points for it is determined as

$$\mathcal{L} = S - \bigcup_{k=1}^{n_1} S_k - \bigcup_{l=1}^{n_2} s_l, \tag{4}$$

where n_1 is the number of robots performing boustrophedon motions and n_2 is the number of the robots backtracking to the next starting points.

The next task of the robot at the ending point is the plan the shortest collision-free path as the backtracking path to a point in list $\mathcal{L}$ that be nearest to the ending point. As mentioned before, Theta* can be a solution for the path-planning task. However, Theta* takes a starting vertex and a goal vertex as inputs and thereby, it has to determine all m backtracking paths from the ending point to all m candidates in list $\mathcal{L}$, and then chooses the shortest one. To decrease m times the time complexity for the determination of the backtracking task, we propose a variant of Theta*, called *Theta* with multi-goals*, in which the heuristic function $h(s)$ is minimum distance from cell s to a bunch of goals,

$$h(s) = \min_{s_i \in \mathcal{L}} (d(s, s_i)), \tag{5}$$

where $d(s, s_i)$ denotes the Euclidean distance between two cells s and $s_i (i = 1, 2, \cdots, m)$. By this proposal, we can see that Theta* with multi-goals always chooses the goal cell s_i that is nearest to cell s when it explores the search space and the path to goal cell s_i provides the shortest path to the ending point. It should be noted that the search space of Theta* with multi-goals is region that has been accessed by the robots, that is, the model $\mathcal{M}$ built so far and thereby, the backtracking path are certainly found by Theta* with multi-goals.

Finally, to sweep the next uncovered region, the robot at the ending point has to follow the backtracking path. This task is implemented by determining the angle and the distance between two points $s_i = (x_i, y_i)$ and $s_{i+1} = (x_{i+1}, y_{i+1}), (i = 1, 2, \cdots, l-1)$ on the backtracking path $P_l = [s_1, s_2, \cdots, s_l]$. Let θ be the heading angle of the robot at the point s_i, the angle required the robot to turn at s_i is

$$\alpha = \beta - \theta, \text{ mapped into } (-\pi, \pi], \tag{6}$$

where β is the direction of vector $\overrightarrow{s_i, s_{i+1}}$, or

$$\beta = \arctan \frac{y_{i+1} - y_i}{x_{i+1} - x_i}, \text{ mapped into } (-\pi, \pi]. \tag{7}$$

The distance required robot to move from s_i to s_{i+1} is

$$d = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}. \tag{8}$$

3.4 Proposed algorithm

This section proposes an algorithm for online complete coverage in unknown environments using a team of robots based on our proposed techniques. Specifically, the steps of the algorithm are described in Algorithm 3.

Algorithm 3: Proposed algorithm

Inputs : A team of robots
Outputs: The workspace is covered

1. Initialize $\mathcal{M} = \emptyset$.

2. Each robot covers the workspace based on Algorithm 2.

3. Check whether a robot reaches an ending point. If yes,

3.1. Detect the backtracking list $\mathcal{L}$ based on Eq. (4)

3.2. Check whether the backtracking list $\mathcal{L}$ is empty. If yes,

i) If there exists a robot covering the workspace, wait until it finishes the boustrophedon motion.

ii) If all of the robots finish their boustrophedon motions, stop the coverage task.

3.3. Plan a backtracking path from the ending point to the next starting points in list $\mathcal{L}$ using Theta* with multi-goal on model $\mathcal{M}$.

3.4. Follow the backtracking path from the ending point based on Eqs. (6) and (8). Goto step 2.

4. SIMULATION

This section presents simulations to evaluate the performance of our approach. Simulations are implemented by object-oriented programming in Matlab, where each robot is modeled by an object. In all of the simulations the workspaces of robots are binary images of 30×30 pixels, in which each pixel is marked to indicate whether it belongs to an obstacle or not based on its value being one or zero, respectively. We also assume that each robot is modeled by a point and the velocity of robots is equal.

The first simulation is used for evaluating the coverage rate of robots in a workspace consisting of predefined obstacles. In Fig. 5, two robots coordinate to perform the coverage task. They are placed side by side at S1 and S2. After covering the workspace, the ending positions of the two robots are at E1 and E2. The coverage path length of the robot 1 and 2 is 383.91 and 416.63(diameter), respectively. The simulation result shows that the proposed algorithm succeeds in controlling two robots to cover the entire workspace.

The second simulation is designed to evaluate the performance of our approach in a more complex workspace, as shown in Fig.6. In this simulation, four robots are placed at four corners of the workspace. It can be seen that four robots have completely covered the workspace. The coverage path length of the robot 1, 2, 3, and 4 is 192.59, 177.16, 194.12, and 193.099 (diameter), respectively.

Next, we perform 10 simulations to evaluate the coverage path length of robots. The workspace shown in Fig. 6 is reused for these simulations. The positions of robots are placed randomly in the accessible region of the workspace. The simulation results show that the workspace is covered completely by four robots. Furthermore, if four robots are placed at four corners of the workspace, the total coverage path length of four robots is minimum, as shown in Fig. 6. Figure 7 shows the coverage path length of four robots in 10 simulations. It can be seen that the coverage path lengths of robots are approximate in each simulation, it means that our approach provides a good solution that power consumption of robots are approximate.

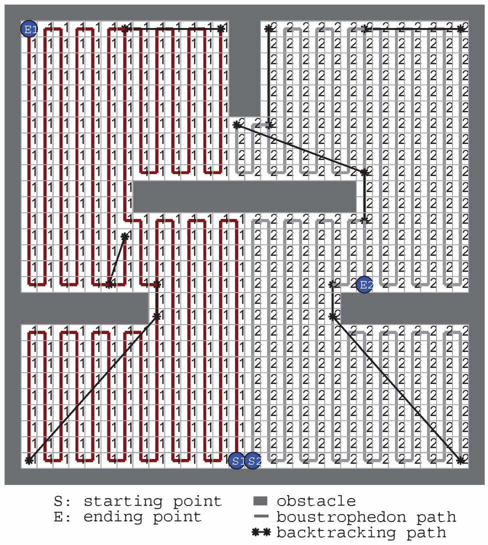

Fig. 5 The coverage regions with two robots

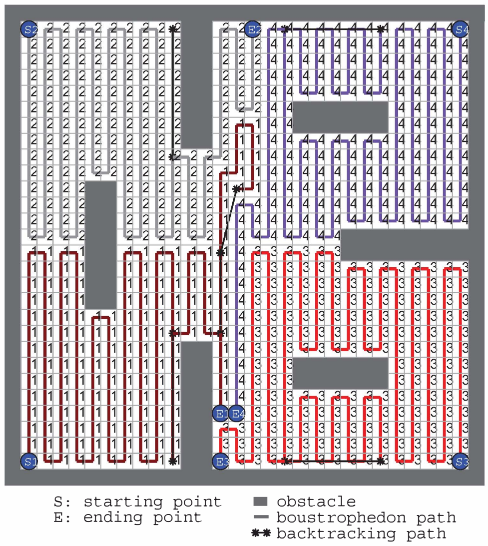

Fig. 6 The coverage regions of four robots

5. CONCLUSIONS

In this paper we have proposed an online complete-coverage approach for a team of robots in unknown environments. In our approach each robot covers an unvis-

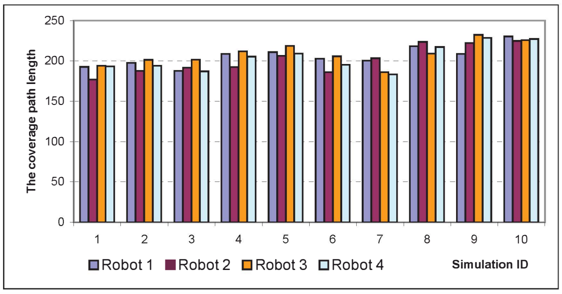

Fig. 7 The coverage path length of four robots

ited region using a single boustrophedon motion until a robot reaches an ending point. At the ending point the robot detects backtracking points based on the model of the workspace built by all robots. Next, it plans the shortest collision-free path as the backtracking path to the next starting point based on the proposed Theta* with multi-goals. Finally, it follows the backtracking path to the next starting point in order to cover the next unvisited region. The coverage process of the robots is implemented until no backtracking point is detected. The simulations show that our approach is efficient for a robot team in unknown environments in terms of the coverage path length and the coverage rate.

Acknowledgements

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science, and Technology (2010-0012609).

REFERENCES

[1] T. W. Min and H. K. Yin, "A Decentralized Approach for Cooperative Sweeping by Multiple Mobile Robots," *Proc. of the International Conference on Intelligent Robots and Systems*, pp. 380-385. 1998.

[2] K.R. Guruprasady, Z. Wilson, and P. Dasgupta, "Complete Coverage of an Initially Unknown Environment by Multiple Robots using Voronoi Partition," *Proc. of the second International Conference on Adcances in Control and Optimization of Dynamical Systems*, Feb. 2012.

[3] I. Rekleitisy, V. L. Shuey, A. P. Newz, and H. Choset, "Limited Communication, Multi-Robot Team Based Coverage," *Proc. of the 2004 IEEE International Conference on Robotics and Automation*, pp. 3462-3468. 2004.

[4] I. Rekleitis, A. P. New, E. S. Rankin, and H. Choset, "Efficient Boustrophedon Multi-Robot Coverage: an algorithmic approach," *Ann Math Artif Intell*, Vol. 52, No. 1, pp. 109142, 2008.

[5] N. Hazon, F. Mieli, and G. A. Kaminka, "Towards Robust On-line Multi-Robot Coverage," *Proc. of the 2006 IEEE International Conference on Robotics and Automation*, pp. 1710 - 1715. 2006.

[6] N. Hazon and G. A. Kaminka, "Redundancy, Efficiency and Robustness in Multi-Robot Coverage," *Proc. of the 2005 IEEE International Conference on Robotics and Automation*, pp. 735-741. 2005.

[7] N. Hazon, and G. A. Kaminka, "On redundancy, efficiency, and robustness in coverage for multiple robots," *Robotics and Autonomous Systems*, Vol. 56, No. 1, pp. 1102-1114, 2008.

[8] Y. Gabriely and E. Rimon, "Spanning-tree based coverage of continuous areas by a mobile robot," *Ann Math Artif Intell*, Vol. 31, No. 1, pp. 77-98, 2001.

[9] N. Agmon, N. Hazon, and G. A. Kaminka, "The giving tree: constructing trees for efficient offline and online multi-robot coverage," *Ann Math Artif Intell*, Vol. 52, No. 1, pp. 143168, 2008.

[10] E. Gerlein and E. Gonzalez, *Multirobot Cooperative Model Applied to Coverage of Unknown Regions, Multi-Robot Systems, Trends and Development*, Dr Toshiyuki Yasuda (Ed.), InTech 2011. DOI: 10.5772/13241.

[11] E. Gonzalez, O. Alvarez, Y. Diaz, C. Parra, and C. Bustacara "BSA: A Complete Coverage Algorithm," *Proc. of the 2005 IEEE International Conference on Robotics and Automation*, pp. 2040-2044. 2005.

[12] H. Choset and P. Pignon, "Coverage path planning: The boustrophedon cellular decomposition," *Proc. of the International Conference on Field and Service Robotics*, Dec. 1997.

[13] H. Choset, "Coverage of known spaces: The boustrophedon cellular decomposition," *Autonomous Robots*, Vol. 9, No. 1, pp. 247-253, 2000.

[14] A. Nash, K. Daniel, S. Koenig, and A. Felner, "Theta*: Any-angle path planning on grids," *Proc. of the AAAI Conference on Artificial Intelligence*, pp. 1177-1183, 2007.

[15] K. Daniel, A. Nash, and S. Koenig, "Theta*: Any-angle path planning on grids," *Journal of Artificial Intelligence Research*, Vol. 39, No. 1, pp. 533-579, 2010.

[16] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, Vol. 4, No. 2, pp. 100-107, 1968.