Distributed Coverage with Multi-Robot System

Chan Sze Kong
DSO National Laboratories
cszekong@dso.org.sg

New Ai Peng
DSO National Laboratories
naipeng@dso.org.sg

Ioannis Rekleitis
Canadian Space Agency
yiannis@cim.mcgill.ca

Abstract – In this paper, we proposed an improved algorithm for the multi-robot complete coverage problem. Real world applications such as lawn mowing, chemical spill clean-up, and humanitarian de-mining can be automated by the employment of a team of autonomous mobile robots. Our approach builds on a single robot coverage algorithm, Boustrophedon decomposition. The robots are initially distributed through space and each robot is allocated a virtually bounded area to cover. The area is decomposed into cells where each cell width is fixed. The decomposed area is represented using an adjacency graph, which is incrementally constructed and shared among all the robots. Communication between the robots is available without any restrictions. Experiments on both simulated and physical hardware demonstrated the viability of employing the algorithm to perform distributed coverage of a given unknown area with multiple robots.

Index Terms – Multi-Robot System, Coverage, Mine Field Clearance, Cooperative Robotics.

I. INTRODUCTION

The task of covering an unknown environment is of high interest in a number of industries. Among them are manufacturers of automated vacuum/carpet cleaning machines and lawn mowers, emergency response teams such as chemical or radioactive spill detection and clean-up, and humanitarian de-mining. In addition, interesting theoretical problems have emerged especially in the areas of path planning, task (re)allocation and multi-robot cooperation. In particular, this paper presents a multi-robot distributed coverage algorithm to solve the humanitarian de-mining problem. The robots' operating environment is unstructured and unknown (i.e. no detailed map representation available for the robots). The algorithm has to ensure complete coverage of the minefield less those areas that are not reachable by the robots.

To achieve complete coverage, the robots need to locate the obstacles and to represent the environment in a way that enables them to navigate to any uncovered areas. In order for multiple robots to cooperate effectively, the robots need to exchange their knowledge of the environment and have a mechanism to allocate the coverage tasks among themselves. A common task selection protocol is used for the task allocation. For hardware implementation, the localization problem is abstracted by allowing the robots to localize themselves using a laser range finder and retro reflective objects placed in the environment.

In the next section we present relevant related work on the coverage task. Section III provides an overview of our algorithm and Section IV presents our experimental results in simulated and real world environments. Finally, Section V provides the conclusion and future work.

II. RELATED WORK

Due to space limitations we will briefly outline the major approaches in multi-robot coverage (for a more detailed survey please refers to [7]). This work takes root in the Boustrophedon decomposition [6], which is an exact cellular decomposition, where each cell can be covered with simple back-and-forth motions.

Deterministic approaches have been used to cover specialized environments [16] sometimes resulting in repeat coverage [3, 10, 14]. Moreover, based on the communication capabilities, two different approaches have been recently proposed that guarantee complete coverage [7, 8]. Non-deterministic approaches include the use of neural networks [1], chemical traces [9], and swarm intelligence [2, 11, and 13]. The non-deterministic approaches can not guarantee complete coverage. Finally, when the terrain is known, existing algorithms guarantee a performance of at most eight times the optimal cost [15].

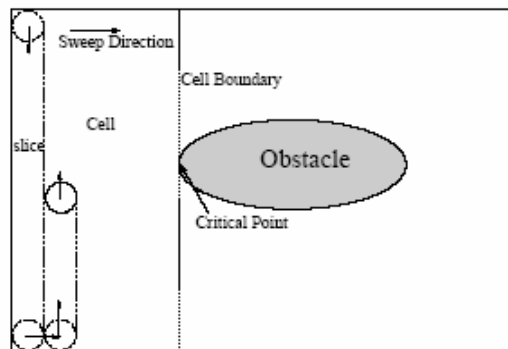### A. Boustrophedon/Morse Decomposition

Figure 1. Illustrates the terms used in a single robot coverage

To better describe the multi-robot coverage algorithm, we borrow the following terms from single robot coverage: *slice, cell, sweep direction,* and *critical point* (see Figure 1). The critical points have been described in length in [4] (see Figure 2a for an overview). For the single robot sensor-based coverage of unknown environments, Acar et al [4] uses Morse

Decompositions. The robot performs coverage and during the process, the robot detects critical point. When a critical point is detected, the robot would insert the critical point to the model, and insert new cell(s) to the Reeb graph, if any. Once the cell is completely covered, the robot will obtained the next uncovered cell from the model and proceed to cover this new cell.

B. Multi-Robot Coverage

For multi-robot system [7, 8], the approach was to introduce an exploration task. In [7], each robot is allocated a stripe, and the robot moves around the stripe to build a Reeb graph representation of the stripe. These Reeb graphs are shared among the robots and hence each robot has a global Reeb graph of the environment (less the obstacles located fully within a stripe). The introduction of the exploration task allows some preliminary information to be gathered and helps to optimize allocation of the coverage task. However, the exploration task channels resources away from doing the actual work. In [8], the exploration of each cell is performed by two explorer robots using line-of-sight (LOS) communication. Any lost of LOS communications implies the existence of a critical point. This approach is not preferred as a minimum of two robots equipped with LOS sensors are required.

C. Detecting Critical Points

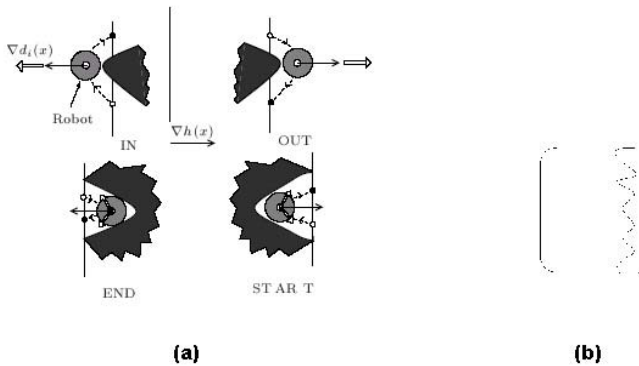(a)   (b)

Figure 2-a Types of Critical Point
2-b Obstacles with critical points that are difficult to detect and represent with Reeb graph

A common feature in Morse Decomposition is the need to detect critical points, created by the presence of obstacles in the environment, to perform decomposition of space. Figure 2a shows the 4 types of critical point: IN, OUT, START and END. However, it is very difficult to detect critical points created by obstacles with many critical points near each other (see Figure 2b). For this project, we have decided to simplify the task decomposition by not decomposing the space with critical points.

D. Task Decomposition

In the coverage problem, the task of the robot team is for the robots to move over the entire free space (assuming the detector or clearance device is the same as the footprint of the robot). Hence, task decomposition can be seen as the decomposition of the free space. In a multi-robot paradigm, the way the task is decomposes has an impact on the efficiency of the algorithm. The task has to be "big" enough for a robot to perform uninterrupted for a period of time, yet "small" enough so as not to create situations where some robots are busy with coverage while others are idling, waiting for the coarsely decomposed tasks to be completed.

In the work of Acar and Choset ([4] and [5]), Morse Decomposition was employed to decompose the free space. Figure 3 illustrates Morse Decomposition using the Boustrophedon decomposition function to decompose the space. Reeb graph is use to represent the decomposed space, where the edge represents a cell and a node represents a critical point.

However, it is noted that tasks obtained from Morse Decomposition can be quite coarse (depending on the environment). As an example, consider three robots assigned to cover the area in Figure 3 starting at cells A, B and C. The robot covering cell C (which is the smallest of all tasks) would complete the task first and move on to cover cell D (the biggest task). The other two robots that were covering cell A and B would be idle after they completed their task, while waiting for cell D to be completed. The performance of the multi-robot system will be improved if the idling period can be reduced or eliminated.

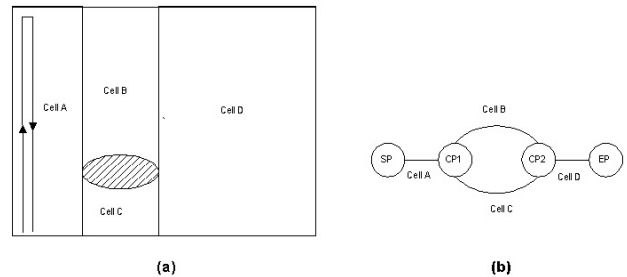(a)   (b)

Figure 3a Morse Decomposition using Boustrophedon approach
3b Reeb Graph to represent the Morse Decomposition

At the other extreme, one can decompose the area using a grid-based decomposition and represent the area using a fine resolution occupancy grid. In this case, the task would be for the robots to move over all the unoccupied grids. One logical grid resolution will be size of the robot, as reaching the centroid of the grid cell implies the coverage of that cell. Besides the problem of partially occupied grid cells, the coverage paths of multiple robots are difficult to coordinate and optimize. We believe that this is not an efficient way to decompose the coverage task for multi-robot systems.

II. ALGORITHM DESCRIPTION

A. Task Decomposition and Representation

In this paper, we use the Boustrophedon approach and decompose the task as fixed width cells (the height of the cell being determined by boundaries provided by operator or physically bounded by obstacles). The width of each cell will be twice the lap width. In this case, each cell can be covered by an atomic "cycle algorithm" behavior. This is illustrated in Figure 4 using the same environment as before. The Adjacency graph is used to represent the decomposition of this space. The nodes in the adjacency graph represent the cells and the edges represent the connectivity.
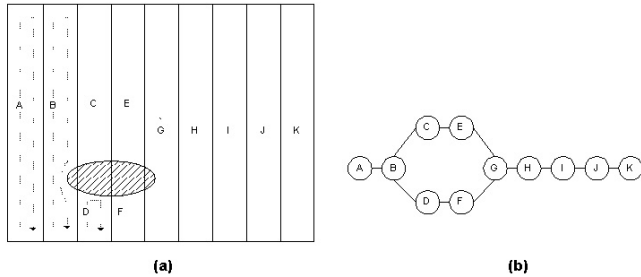

Figure 4a Cellular Decomposition with fixed size cell width.
The arrowed line shows the path that the robot will take to cover the cell
4b Adjacency Graph to represent the decomposition of (a)

B. Cycle Algorithm - Coverage and Detecting Change in Connectivity

The cycle algorithm was proposed by Acar and Choset in [4]. It has 3 phases: forward phase, reverse phase and closing phase. The difference in our algorithm is that the robot will not look for Critical Points. Instead, the robot will determine by the presence of obstacles if the cell being covered is divided into disconnected parts. There are 3 possible scenarios that can occur while the robot is executing the cycle algorithm (refer to Figure 5):

- Cell with no obstacles (e.g. Cell A) – the robot performs the forward and reverse phases without meeting any obstacle. The robot can therefore conclude that the current cell coordinates are correct and update the graph to indicate that the cell is covered. In addition, it can also infer that the cells beside it will have the same size as the current cell. As such, 1 or 2 new cells can be added to the Adjacency Graph, if the new cells are within the area to be covered and the cell is not already in the graph. In the case as shown in Figure 5, a cell D is added to the right of cell A. The current cell (A) and the new cell (D) are connected on the graph. The updated graph is shared with the other robots in the team.

- Cell with obstacles blocking part of the cell's width (e.g. Cell B) – while performing forward or reverse phase, the robot encounters an obstacle partially obstructing its lapping path. At the obstacle location, the robot uses its sensor to determine that the obstacle is blocking part of

the cell's width (cell B of Figure 5). After passing the obstacle, it will take note of its location. This location will be used as the starting point of the adjacent cell to be added later. The robot continues the coverage of the cell. At the boundary of the bounded area, the robot updates the state of cell B and adds cell E, F and G to the graph.

- Cell divided by obstacle (e.g. Cell C) – during the forward phase, the robot will encounter the obstacle. By following the obstacle, the robot will move laterally one cell width. This means that there is an obstacle dividing the cell into 2 disconnected parts. At the end of the cycle algorithm, the robot updates the state and the size of the current cell. The new cells (cell H and I) added are assumed to have the same height as the current cell (cell C).

After covering the current cell, the robot will select the next uncovered cell with the highest utility function from the Adjacency graph. For simplicity, we set the utility function as the inverse of the distance between the robot and the cell. Addition components can be added to the utility function to further enhance the performance.
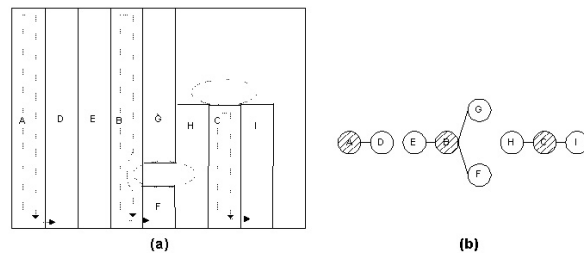

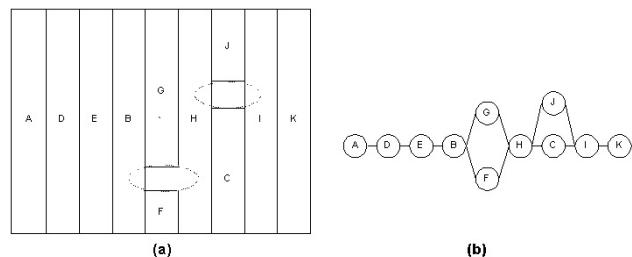Figure 5 Adding of new cells to the model after completing the coverage of cell A, B and C.

Figure 6 The model of the entire area when it completes

When a new cell is added, the dimension of the new cell is assumed to be similar to the current cell if no obstacles are detected. If obstacles are present, the dimension of the new cell is considered to be limited by the robot's perception of the obstacle's location and size. As in the case of cell H and I in Figure 6, the height of these cells actually stretches the entire area. Therefore, the robot must be able to rectify any wrong assumptions that were made. The Cycle Algorithm is able to handle this situation. This can be illustrated with an example with reference to Figure 6. With the complete coverage of cell C, cell I is added with a smaller length. Assume that the robot moved on to cover cell I. When it reaches the obstacle at cell

I, the robot will be able to detect that the obstacle is only covering part of the cell width. The Cycle Algorithm will bring the robot to the upper boundary of the region and when the coverage is completed, the robot will be able to correct the height of cell I and update the Adjacency graph.

When adding a new cell, the robot may find that the cell has already been added. The position and dimension of the current new cell may be different than the one already in the model. The merging of the cells is handled as follows:

1) Information of a COVERED cell will be considered as accurate and final. Therefore, information on COVERED cell will take precedence.
2) If both the cells are in an UNKNOWN state, the 2 cells will be combined and the final cell in the model will cover the size of the 2 cells. An example can be seen in cell H of Figure 6. The robot at cell C will insert a cell that is of the same size as cell C and the robot at cell G will insert a cell that is of the same size as cell G. The final combined cell in the model will be the size of C plus G.
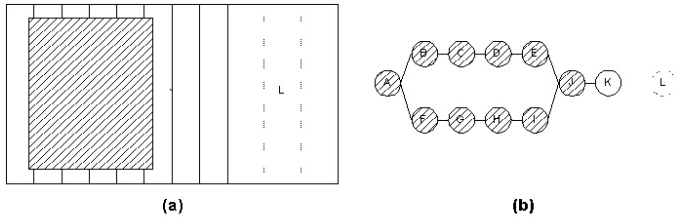
C. Introducing New Tasks

(a)   (b)

Figure 7 Insertion of a new task L when no new cell can be found.

As the model is incrementally constructed, it may happen that a robot cannot find any new cells even though the area is not completely covered. Figure 7 illustrates an example where 2 robots are covering an area with a big obstacle in it. Robot 1 starts with cell A and Robot 2 starts with cell J. Robot 2 completes the coverage of cell J and moves on to cell K. Robot 1 finishes all the cells left of cell J. Robot 1 cannot find a new cell from the graph. Unless Robot 2 detects an obstacle subsequently (which will introduce 2 new tasks simultaneously), Robot 1 will remain idle for the rest of the time. To handle such situation, each robot is allowed to insert a new cell at other regions of the free space that has no information (provided the unknown area is of a certain pre-specified size). Cell L is an illustration of Robot 1 inserting a new cell and generated a new task for itself.

D. The Cooperation Mechanism of the Robots

When each robot completes the coverage of a cell, the robots will broadcast the new graph information to the team. Upon receiving the new graph information, the robot will synchronize its own graph with the new information. In this way, all members of the team have a common global picture of the state of the world. In this manner, they are able to coordinate their effort to complete the coverage in the shortest possible time.

In the event of attrition, the remaining team is able to continue with the coverage without been affected. The lost of communication however will result in repeat coverage.

E. Algorithm Summary

The flowchart in Figure 8 summarizes the algorithm. After the coverage command has been issued, the algorithm creates 2 threads, one to listen for and synchronize the new graph information and the other to perform the cycle algorithm.
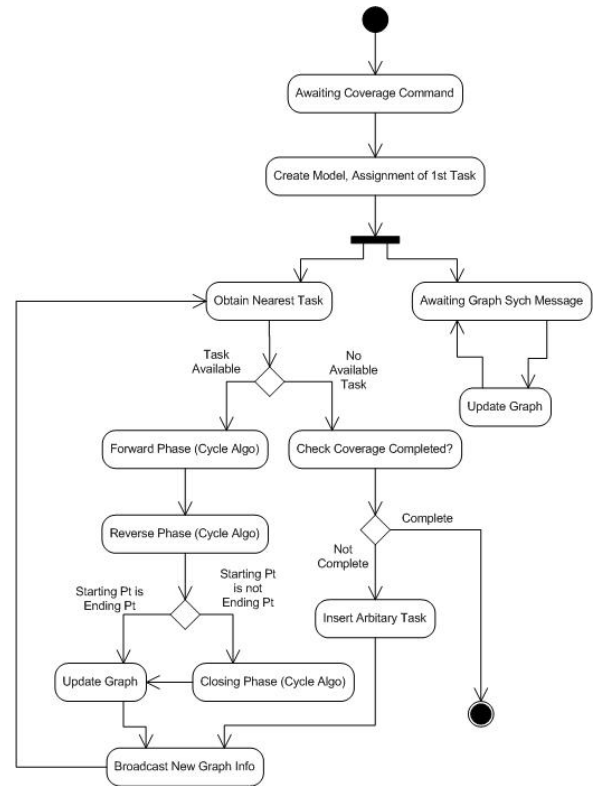

Figure 8 Flowchart of Algorithm

III. EXPERIMENTAL RESULTS

Our algorithm has been validated via a variety of experiments in simulation and with real robots. In the simulated environments ground truth information verified the correctness of the algorithm. In hardware implementation, real robots demonstrated the feasibility of our approach and highlighted hardware limitations.

A. Single Robot Coverage in Simulation

The Player/Stage [12] multi-robot simulator was utilized with the Pioneer 3AT model to develop and test the distributed coverage algorithm. The footprint for the coverage is set at 0.5 m, and therefore the cell width is 1.0 m. In Figure 9a, a

robot is deployed to cover an open area of 5 m X 5 m. The robot started at the arbitrarily set coordinates of (-7,-8) and is tasked to cover the area defined by (-7,-7) to (-2,-2). After obtaining the command from the Operator, the robot inserts a cell into the model (Figure 9a). The robot moves to the start point at (-6.75, -7), and performs the cycle algorithm to cover the first cell. At the end of the cycle algorithm, the robot will insert a new cell to the right of the covered cell and proceed to cover the next cell. This sequence is repeated until the entire area is covered.

In an open outdoor area there are often small obstacles (e.g. trees) that are smaller than the width of the cell (1 meter). Such obstacles do not divide the cell into 2 disconnected parts. Figure 9b shows the result of the robot performing coverage in such an environment.

(a)   **(b)**
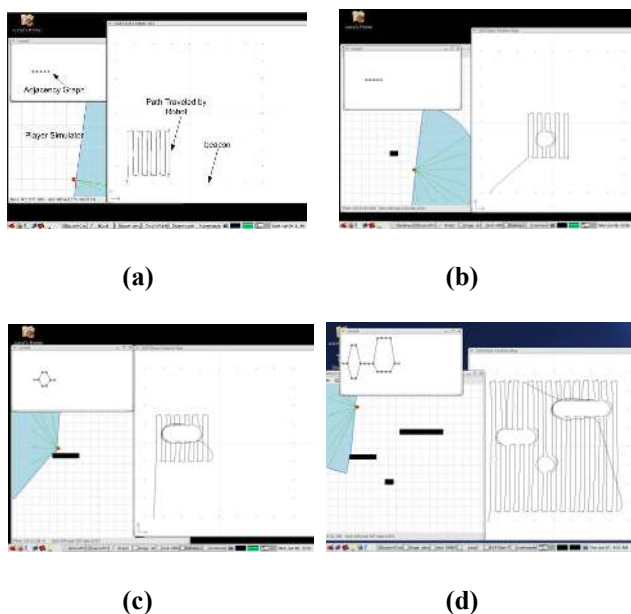

(c)   **(d)**

Figure 9 (a) Single Robot Coverage without obstacles. (b) Single Robot Coverage a small obstacle (e.g. a tree). (c) Single Robot Coverage a big obstacle. (d) Single Robot Coverage with multiple obstacles

Figure 9c illustrates the robot's capability to handle obstacles that divide the cells into 2 disconnected parts. When the robot encounters such an obstacle, it adds 2 cells to the graph. The robot then proceeds to cover the area below the obstacle. When the robot reaches the right boundary of the area to be covered, it goes back to the uncovered cells.

Figure 9d illustrates the capability of the algorithm to perform complete coverage of a more complex area. The time taken to complete the 34 m X 34 m area with 3 unknown obstacles is 48 minutes.

B.   Multi-Robot Coverage in Simulation

Two robots were deployed to cover the same area (34 m X 34 m). The time taken by the Robot 1 was 26 minutes, and the time taken by Robot 2 was 27 minutes. The total time saved by deploying 2 robots is 21 minutes. Robot 1 finished the task earlier and therefore was left idling for approximately 1 min (see Figure 10 for robots' coverage trace and Adjacency Graph).
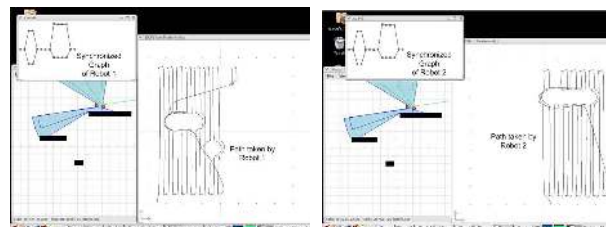

Figure 10 Two Robots Coverage

One important advantage of multi-robot systems is robustness. Figure 11 demonstrates how the algorithm handles attrition. The scenario is that the motor of Robot 2 was disabled while covering the first cell, but the robots are not aware of the problem. Robot 1 continues to perform coverage in its allocated area. When Robot 1 finishes its area, it inserts an arbitrary new cell inside the area allocated to Robot 2. Eventually, Robot 1 covers the rest of the area completely by itself.
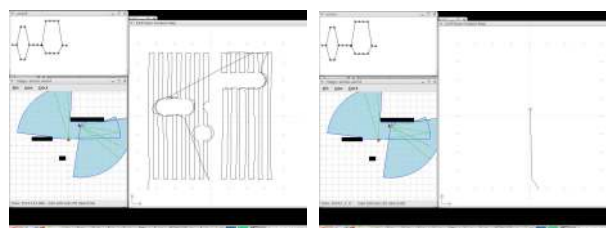

Figure 11 Two Robots Coverage with Attrition

C.   Single Robot Coverage using a Pioneer 3AT Robot

The robot used in our experiments is the Pioneer 3AT robot equipped with a Sick Laser range finder. The laser range finder performed obstacle detection and localization via retro-reflective objects placed in the operating environment.

For the real robot testing, we set up a series of experiments in a basement parking space (see Figure 12a). Localization beacons were placed three meters apart around the perimeter of the area to be covered. The size covered by the robot is 12m x 9 m. There were three obstacles in the environment, of which two are concrete pillars and the other is an artificially added obstacle.
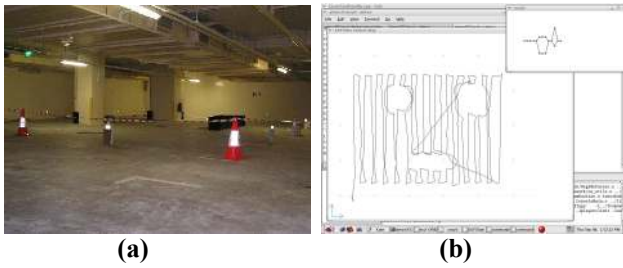
Figure 12 (a) Test Environment. (b) Coverage Trace and Adjacency Graph

The result is shown in Figure 12b. The total time taken by the robot to completely cover the area is about 25 minutes.

D. Multi Robot Coverage using two Pioneer 3AT robots

Multi-robot coverage experiment was conducted using two robots in the same environment as shown in Figure 12(a). Figure 13 (a) to (f) shows the sequence of screenshots during the experiment. In the figures, the graph and coverage trace on the left belong to robot 1 while those on the right belong to robot 2. The robots covered the same area in 13 minutes and 55 seconds, with one robot idling for 20 seconds.

Figure 13(a) shows robot 1 completed its first cell and robot 2 completed the two cells below the large obstacle. While covering the current cell, robot 2 has move passed the obstacle and introduced a new cell to the top of the large obstacle. Figure 13(d) shows that the graph from the 2 robots connected. In addition, robot 2 has reached the right boundary of the area and is moving back to next nearest uncovered cell discovered earlier. Figure 13(f) shows robot 2 completed the last cell, while robot 1 remained idle for a short period of time.

IV. CONCLUSION

In this paper we presented an algorithmic approach to the distributed complete coverage problem. Under the assumption of global communication among the robots, each robot is allocated an area of the unknown environment to cover. Cooperation among the robots is facilitated and performance is improved through regular information sharing and task selection protocol. In our approach, no robot remains idle while there are areas to be covered. Results in simulation and with real robots validate our distributed coverage algorithm. In the hardware experiment, localization is a main challenge to ensure accurate and consistent coverage. We overcome this by planting laser beacons in the environment and using the laser range finder to localise the robots.

Our experiments demonstrated an almost linear improvement with the addition of the second robot. In future work we are planning to examine if this trend continues as more robots are added. Bandwidth limitations and congestion in narrow areas are potential limiting factors.

It is worth noting that in a large variety of experiments, the robots always perform complete coverage in an efficient manner.
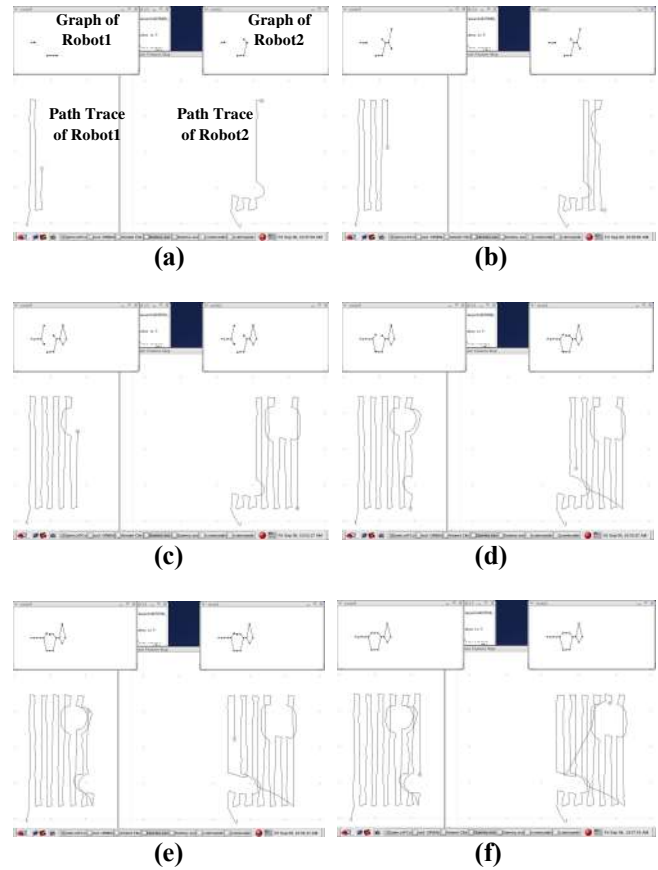

Figure 13 Multi-robot coverage with 2 Pioneer 3AT.

REFERENCES

[1] C. Luo and S. Yang, "A real-time cooperative sweeping strategy for multiple cleaning robots," in *IEEE Internatinal Symposium on Intelligent Control*, Oct. 2002

[2] D. J. Bruemmer, D. D. Dudenhoeffer, M. O. Anderson, and M. D. McKay, "A robotic swarm for spill finding and perimeter formation," *Spectrum*, 2002

[3] D. Kurabayashi, J. Ota, T. Arai, and E. Yoshida, "An algorithm of dividing a work area to multiple mobile robots," in *1995 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1995

[4] E. Acar and H.Choset. Sensor-based Coverage of Unknown Environments: Incremental Construction of Morse Decomposition. The International Journal of Robotics Research, Apr 2002.

[5] E. Acar, H.Choset, A. Rizzi, P. Atkar and D. Hull. Morse Decompositions for Coverage Tasks. The International Journal of Robotics Research, Apr 2002.

[6] H. Choset and P. Pigeon, "Coverage Path Planning: The Boustrophedon Decomposition." in International Conference on Field and Service Robotics 1997

[7] I.Rekleitis, A.P. New and H.Choset. Distributed Coverage of Unknown / Unstructured Environments by Mobile Sensor Networks. 3rd MRS Workshop 2005

[8] I.Rekleitis, A.P. New, H.Choset, and Lee-Shute. Limited Communication, Multi-Robot Team Based Coverage, ICRA 2004

[9] I. Wagner, M. Lindenbaum, and A. Bruckstein, "Distributed covering by ant-robots using evaporating traces," *IEEE Transactions on Robotics and Automation*, 1999

[10] Latimer, Srinivasa, Lee-Shue, Sonne, Choset and Hurst. Towards Sensor Based Coverage with Robot Teams, IROS 2002

[11] M. A. Batalin and G. S. Sukhatme, "Spreading out: A local approach to multi-robot coverage," in *6th International Symposium on DistributedAutonomous Robotics Systems*, 2002.

[12] R. T. Vaughan, B. Gerkey, and A. Howard, "On device abstractions for portable, reusable robot code," in *IEEE/RSJ International Conference on Intelligent Robot Systems*, 2003

[13] S. Ichikawa and F. Hara, "Characteristics of object-searching and object fetching behaviors of multi-robot system using local communication," in *IEEE International Conference on Systems, Man, and Cybernetics, (IEEE SMC '99)* 1999

[14] T. W. Min and H. K. Yin, "A decentralized approach for cooperative sweeping by multiple mobile robots," in *1998 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1998

[15] X. Zheng, S. Jain, S. Koenig and D. Kempe. Multi-Robot Forest Coverage. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS),* pages (in print), 2005.

[16] Z. Butler, A. Rizzi, and R. Hollis, "Distributed coverage of rectilinear environments," in *Proc. of the Workshop on the Algorithmic Foundations of Robotics*, 2001.