

Optimal Coverage of a Known Arbitrary Environment

Raphael Mannadiar and Ioannis Rekleitis

Abstract—The problem of coverage of known space by a mobile robot has many applications. Of particular interest is providing a solution that guarantees the complete coverage of the free space by traversing an optimal path, in terms of the distance travelled. In this paper we introduce a new algorithm based on the Boustrophedon cellular decomposition. The presented algorithm encodes the areas (cells) to be covered as edges of the Reeb graph. The optimal solution to the Chinese Postman Problem (CPP) is used to calculate an Euler tour, which guarantees complete coverage of the available free space while minimizing the path of the robot. In addition, we extend the classical solution of the CPP to account for the entry point of the robot for cell coverage by changing the weights of the Reeb graph edges. Proof of correctness is provided together with experimental results in different environments.

I. INTRODUCTION

The task of covering a bounded region of space is common to many problems such as de-mining, vacuum cleaning, lawn mowing and automated painting. One coverage application that has proven to be extremely successful is the task of robotic vacuum cleaning. The iRobot¹ Roomba vacuum cleaning robot uses a variety of different strategies, such as random walk, wall-following, and the seed-spreader algorithm [1], to achieve, probabilistically, coverage of the whole floor space. The sale of more than two million robots highlights the importance and impact of this application. In the above described applications the problem of robotic coverage of free space is defined as follows: the robot has to pass an end effector, or a sensor, which in most cases is the body of the mobile robot, over all available free space. For example, during mine detection, the robot has to ensure that every location that is not covered by an obstacle is inspected and the position of the discovered mines recorded. In such an application it is of paramount importance to ensure completeness; no accessible area should be left uncovered.

Depending on the target application, the proposed approaches can be characterized according to different requirements. One important division is between algorithms that require a map of the environment, which describes the occupied and free space, and the algorithms that are capable of covering an unknown environment. For unknown environments it is impossible to provide a criterion of optimality as for any design choice a counter-example environment could be constructed. Nevertheless, during coverage of unknown environments it is important in terms of efficiency to avoid

repeated coverage [2] as much as possible. For known environments it is important to accomplish the coverage task in minimum time. A subset of all coverage planning schemas are said to be complete, meaning they guarantee the entire free space will be covered by the provided plan. Such schemas are often based on rigorous representations of the free space such as exact cellular decompositions or spanning trees on which coverage planning is simplified. In this paper we present a new algorithm for the complete coverage of an arbitrary known environment. Our algorithm is using the Boustrophedon cellular decomposition to divide the free space in cells; then, the solution to the Chinese Postman Problem is used to calculate the order in which the cells are going to be covered. We extend the classical Boustrophedon decomposition by splitting some cells in half along the direction of coverage. As a result the robot is capable of covering each cell, or sub-cell, in such order that no cell is traversed twice, and at the end the robot has returned near the starting position. Further improvement in performance is achieved by calculating the width of each cell and estimating the position of the robot at the end of the coverage cycle. As in most previous treatments of coverage, the robot operates under the assumption of accurate localization. Experimental results in different environments illustrate the efficiency of our approach.

Related work would be discussed in the next section, including a brief overview of the Boustrophedon cellular decomposition that provides the basic component to our approach, as well as related concepts from graph theory. In Section III a detailed description of our algorithm is presented, together with a sketch of the optimality proof. Section IV demonstrates the experimental verification of our approach in different classes of simulated environments. Section V contains conclusions and directions of future work.

II. BACKGROUND

Choset and Pignon first introduced a rigorous extension to the Seed Spreader algorithm [1] under the name of Boustrophedon Decomposition [3]. The work was further developed by Acar et al. [4] with experimental verification and for a variety of control Morse functions. The Boustrophedon family of algorithms guarantees the complete coverage of an unknown environment with no claims on the distance travelled. Section II-A provides an outline of these algorithms as they form one of the building blocks of our approach.

Butler *et al.* [5] achieved complete coverage of unknown rectilinear environments using a square robot with contact sensing. They performed an on-line decomposition where each cell, in the shape of a rectangle, was formed such that

Raphael Mannadiar and Ioannis Rekleitis are with the School of Computer Science, McGill University, yiannis@cim.mcgill.ca, raphael.mannadiar@mail.mcgill.ca

¹<http://www.irobot.com/>

it could be covered completely by back-and-forth motions performed parallel to one of the walls of the environment.

A different approach was followed by Huang [6]: the environment was subdivided in different regions aiming to minimize the amount of rotations the robot has to perform. The total path travelled was ignored. Yao proposed an improved algorithm [7] that reduces the path travelled compared to [6]. Illustrative examples were shown, but no proof of optimality. More recently, Kang et al. [8] proposed a scheme where a set of precalculated motion strategies are selected in order to minimize repeat coverage. In all of the above the claims are only supported by examples in simulation.

Gabriely and Rimon [9], [10] used a grid based approach for planning a complete coverage path. The main requirement is for the environment to be decomposable to a grid. For known terrain, algorithms proposed by Zheng et al. guarantee a performance of at most eight times the optimal cost [11]. This is achieved by superimposing a grid with each grid cell having the size of four footprints of the robot. Our approach eliminates the grid restrictions on the environment and guarantees an optimal path.

Several researchers have worked on the problem of distributing a group of nodes (mobile sensors) such that they achieve the maximum coverage of an area of interest. The robots, after reaching their position, do not move unless there are dynamic changes in the environment. In particular, Cortés et al. proposed an approach which utilizes the centers of the Voronoi cells [12], [13]. Using artificial potential fields, Howard et al. control the robots to move away from each other, thus increasing the covered area [14]. More recently, Schwager et al. proposed a unifying scheme for multi-robot coverage which combines the previous methods [15]. All of the above are concerned with the sensor coverage at the final position of the robots e.g. in surveillance operations as opposed to the path-planning problem more common to applications such as de-mining, vacuum cleaning, etc.

Other approaches used genetic algorithms [16] and landmarks [17] to improve the speed of coverage. Easton and Burdick [18] used a variant of the Chinese Postman Problem to solve the problem of boundary coverage. For information on several more algorithms on coverage see [19]. In addition, many authors have worked on the area of multi-robot coverage; for an extensive survey please refer to [20].

A. Boustrophedon Cellular Decomposition

Our approach utilizes the concepts of Boustrophedon Cellular Decomposition of unknown environments to optimally schedule the order of coverage. To better describe our algorithm, the following terms from single robot coverage are used: *slice*, *cell*, *sweep direction*, and *critical point* [21], [4]; see Fig. 1a. The Boustrophedon decomposition [3] is a type of Morse decomposition where the slice is a line. The robot follows the intersection of the slice and the area to be covered, thus covering the area with vertical back and forth motions. A *cell* is a region defined by the Boustrophedon decomposition where slice connectivity does not change. In other words, no obstacle breaks the connectivity of the slice

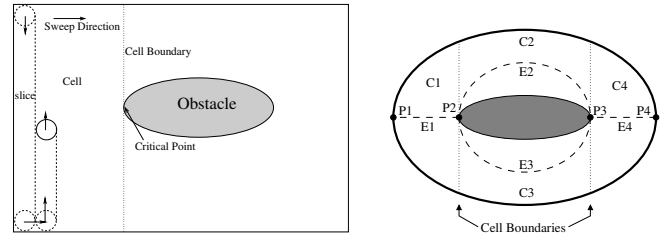


Fig. 1. (a) Illustration of the terms borrowed from single robot coverage with a single robot and one obstacle in the target environment. The robot is performing coverage with simple back-and-forth motions. (b) Here a simple Reeb graph is overlaid on top of a simple elliptical world with one obstacle. P1-P4 are critical points which represent graph nodes. E1-E4 represent edges which directly map to cells C1-C4.

inside each cell. *Sweep direction* refers to the direction in which the slice is swept. Lastly, a *critical point* represents a point on an obstacle which causes a change in the slice connectivity. Thus, the free space is divided into regions (cells) of constant slice connectivity, each of which can be covered with a vertical back and forth motion.

Another concept used here is the Reeb graph [21], [4]. A Reeb graph is a graph representation of the target environment where the nodes represent the critical points and the edges represent the cells; see Fig. 1b. Due to the nature of the Boustrophedon decomposition, all concave critical points are connected to exactly one cell, i.e., a node of degree one in the Reeb graph. Similarly, all convex critical points are connected to exactly three cells, i.e. a node of degree three in the Reeb graph.

B. Graph Theory

Different algorithms from graph theory have been used in robotics to guide exploration [22], mapping [23], [24], and coverage [25] in the past. Edmonds and Johnson [26] present an overview of different graph algorithms that are directly applicable to the problem of optimal coverage of a known environment.

An Euler tour is a circuit that covers every edge in a graph exactly once. Euler demonstrated that a necessary and sufficient condition for the existence of such a tour in a graph is that all nodes of the graph have even degree – such graphs are called *Eulerian Graphs*. A similar problem is the *Chinese Postman Problem* (CPP): find a shortest tour that traverses every edge at least once. If a graph is Eulerian then all of its Euler tours are solutions to the CPP. For non-Eulerian graphs, a standard approach to solving the CPP is to *double* selected edges in the graph – i.e. given an edge e that connects nodes V_i and V_j , add an edge f that also connects nodes V_i and V_j – to make the resulting graph Eulerian and to then choose one of the Euler tours as the solution. Different strategies can be applied to determine which edges to double. All of the new edges will by definition be part of the Euler tour; hence, the challenge is to choose edges such that the total cost – the sum of the individual costs of all the edges – of this Euler tour be minimized. The constraints and objective function of a linear programming system that can be solved to choose the edges to double is described in [26]. In addition, these edges can be chosen

by utilizing algorithms from *matching theory*. The simplest of these algorithms essentially determines the shortest paths between every *odd-node* (i.e. nodes of odd degree) in the original graph, the total costs of which are used to determine which edges should be doubled. Consequently, an Euler tour can be extracted from an Eulerian graph. A simple and wildly inefficient algorithm was proposed [26]: let G be the original graph. Repeatedly choose and remove edges whose deletion would not disconnect G unless there is no other choice until G is empty. The sequence of chosen edges is the Euler tour. More efficient, but also more complex, algorithms iteratively build disjoint non-Euler tours, which they connect in a certain fashion. When there are no more unvisited edges, the constructed sequence is an Euler tour. For more information please refer to [26].

Next, the optimal coverage algorithm of an arbitrary known environment is presented.

III. OPTIMAL COVERAGE ALGORITHM

The optimal coverage algorithm is divided in two parts: first, an off-line analysis of the environment, construction of the Boustrophedon Cellular Decomposition (BCD) and the Reeb Graph (RB), formulating and solving of a linear programming problem for the construction of an Euler tour; and second, on-line coverage that uses the sequence of edges in the Euler tour to guide the coverage from one cell to the next. Each component will be discussed next

A. Construction of the Reeb Graph

The input to our algorithm is a bitmap representation of the environment. This choice comes naturally as the testing environment that we employ, Player/Stage, uses the same representation. In addition, a bitmap representation provides maximum flexibility in the modeling of the free/occupied space, without any restrictions on the shape of the obstacles. As can be seen in the experimental results section, different environments, such as indoor office space, outdoor, sparsely populated areas, as well as completely unstructured areas were used to verify the performance of our approach.

A common assumption in most cellular decomposition algorithms is that no two critical points change the slice connectivity at the same time. This assumption is trivial to enforce by a small rotation of the sweep direction. It is worth noting that in this work we extend the notion of a critical point to include critical regions; for example a whole wall that changes the slice connectivity. See for example the left and right walls in Fig. 3a. In a structured environment, such as an office space, it is beneficial to perform coverage parallel to the walls. The above mentioned extension allows it.

The first task is to sweep the bitmap. Without loss of generality we assume a sweep direction along the x-axis and record all the critical points/regions. During the sweep, the location of the cells is also recorded. Finally, the critical points and the cells are encoded in the Reeb Graph. The resulting Reeb Graph $G = \langle V, E \rangle$ is used as input to the next step of the algorithm that calculates an Euler tour.

B. Construction of the Euler Tour

The primary contribution of our algorithm is using the solution to the CPP in order to find the optimal order, in terms of distance travelled, in which the cells are covered. Given the Reeb graph, the next task would be to calculate an Euler tour. As mentioned in Sect. II-B, this can be achieved by doubling selected edges of the Reeb Graph. Consequently, when the robot covers one of these doubled edges, it will split the cell into top and bottom sub-cells and assign each sub-cell to one of the two doubled edges. From [26], no edge would be duplicated more than once. In addition, we bias the edge duplication in cells that are wider in order to facilitate easier coverage². An instance of a linear programming problem described in Eq. 1 is created. The solution to the LP represents the Euler tour that is the routing of the robot.

$$\begin{aligned} x_e \text{ is integer } e \in E; w_n \text{ is integer } n \in V; \\ x_e \geq 0, \quad e \in E; w_n \geq 0, \quad n \in V; \\ \sum_{e \in E} a_{ne} x_e - 2w_n = b_n, \quad n \in V; \\ z = \sum_{e \in E} c_e x_e \text{ is minimized.} \end{aligned} \quad (1)$$

where: x_e is the number of added copies of edge e in the solution. $\sum_{e \in E} a_{ne} x_e$ represents the number of added edges to node $n \in V$ in the solution. Note that for the solution to be an Eulerian graph, an odd number of edges has to be added to nodes with odd degree and an even number of edges has to be added to nodes with even degree. b_n is 1 for nodes with odd degree and 0 for nodes with even degree. w_n is a variable that will force $\sum_{e \in E} a_{ne} x_e$ to be odd for odd nodes and even for even nodes. a_{ne} is 1 if node n meets edge e , and 0 otherwise; c_e is the cost of edge e .

The GLPK³ library was used to solve the above linear programming problem. The solution is used to guide the robot from cell to cell, and each cell is covered using a modified version of the Boustrophedon Coverage.

C. Boustrophedon Coverage

The simple back-and-forth motion used for covering the interior of a cell is well documented in the literature [21], [4], [3], [19], and covering it is beyond the scope of this paper. The novel contribution of our algorithm is the treatment of the cells, which had their corresponding Reeb Graph edge doubled in order to generate an Euler tour. These cells are necessary to be traversed twice. The coverage algorithm is modified to cover during the first traversal the top (or bottom) half of the cell, and in the second traversal to cover the bottom (or top) half of the cell correspondingly, see Fig. 2. The choice of which half to cover is dictated by the position of the robot at the end of the previous coverage

²In the degenerate case of a narrow corridor, with width equal to the size of the robot, if that is the only route, then the robot would have to double the coverage of that area.

³<http://www.gnu.org/software/glpk/>

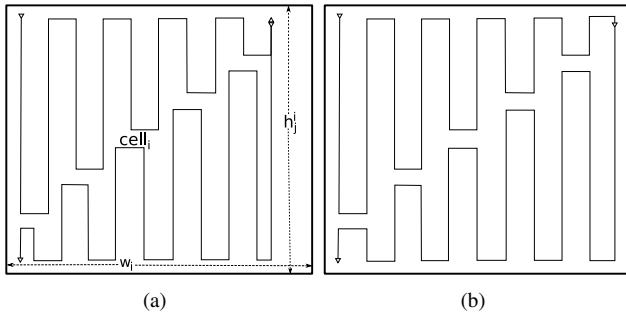


Fig. 2. Coverage of a single cell, depending on the number of up-down motions: (a) odd number, (b) even number.

task. Therefore, at the end of the coverage, the robot will have covered the complete cell without duplicating any work.

It is worth noting that the robot is operating in a known environment. As such, when the robot enters a cell to cover it, it has the information on the length and the height of that cell. Consequently, an informed decision can be made, if the robot covers only the top half of the cell, where to stop each downward motion; see Fig. 2. More formally, for a cell c_i with width w_i and a slice width of w_s there are going to be $k = w_i/w_s$ vertical motions to cover the c_i . When the robot begins a downward motion, it will cover only h_j distance, see Eq. 2.

$$h_j = h_j^i * \frac{k-j-1}{k}, \quad j = 0 \dots k-1 \quad (2)$$

where, h_j^i , is the length of the slice at vertical motion j , which is calculated by the height of cell c_i at that point.

By varying the height of the coverage progressively, the robot is capable of controlling the exit point from each cell, thus minimizing the distance to the entry point to the next cell.

D. Proof of Correctness

The correctness of the algorithm and the optimality follow directly from the properties of the cellular decomposition and the Euler tour used. By definition, the Reeb graph provides a complete model of the environment. By ensuring that each edge of the graph is traversed (covered) we guarantee that all available free space has been covered. The Euler tour resulting from the doubling of selected edges provides an order in which the cell of the Reeb graph should be visited. By definition, there is no edge of the Euler tour that is traversed twice, which means that no area is covered twice. Therefore, the proposed algorithm is optimal, as all free space is covered exactly once.

During coverage the border areas adjacent to obstacles could be wider than a single pass but narrower than two. In such cases the robot sensor would cover these areas twice. In general, these areas, with the exception of “fractal-like” counterexamples, are an order of magnitude smaller compared to the interior area to be covered. Therefore, repeat coverage of the boundaries does not affect the optimality.

It is worth noting that the structure of the Reeb graph is exploited in order to provide a polynomial time solution

to the complete coverage, minimum distance, path planning problem.

IV. EXPERIMENTAL RESULTS

Numerous experiments were conducted for a variety of environments using the robotic simulation package Player/Stage⁴. A simulated Pioneer robot was used to perform coverage of all the available free space. In particular, three different classes of environments were used as test cases. First, office like environments such as the one in Fig. 3a were used, inspired by the service robotics applications. Second, open fields with sparse tree and rock like obstacles were used, the target application being humanitarian demining; see Fig. 3d. Finally, arbitrary environments with convex and concave objects were used to ensure the performance of the algorithm under arbitrary conditions; see Fig. 3g.

Figure 3a illustrates the application of our algorithm in an indoor environment, such as an office building. The solution presented in Fig. 3b guides the robot to cover each one of the thirty eight cells and sub-cells, and return at the starting position. Figure 3c shows the path of the robot as it covers sub-cell 1 and cell 2. A similar structure is followed in the rest of Fig. 3. Figure 3d shows a sparsely populated area. The free space is subdivided in fifty four different cells; Figure 3f shows the path of the robot while covering areas 1 to 5. Finally, Fig. 3g-i demonstrates the application of our algorithm in an environment populated by arbitrary concave and convex obstacles, where the resulting Euler tour results in twenty one cells and sub-cells.

Figure 4 presents another example of an arbitrary environment. Figure 4a shows the Boustrophedon Cellular Decomposition as dashed lines, together with the resulting Euler tour. Figure 4(b-f) presents a sequence of screen-shots of the Stage simulation environment with the robot together with a limited trace. In Fig. 4(b) the robot is covering the cell 11. In Fig. 4(c) the top sub-cell (13) is being covered. The robot continues to the top sub-cell (16), see Fig. 4(d). The cell 18 and then the cell 20 are covered in Fig. 4(e) and (f) accordingly.

V. CONCLUSIONS

In this paper we presented a new algorithm for the complete coverage of a known arbitrary environment. The algorithm guides a mobile robot through a sequence of areas to be covered without wasting energy and time by moving through already covered areas. The solution to the Chinese Postman Problem from graph theory is adapted for the calculation of the cell ordering. The single cell coverage used in the Boustrophedon Cellular Decomposition algorithm is modified in order to eliminate repeat coverage by splitting selected cells into two components.

Experiments using real hardware are scheduled for the near future in order to further validate our approach. A iRobot Create mobile robot will be used in combination with an overhead camera for accurate localization. Furthermore,

⁴<http://playerstage.sourceforge.net/>

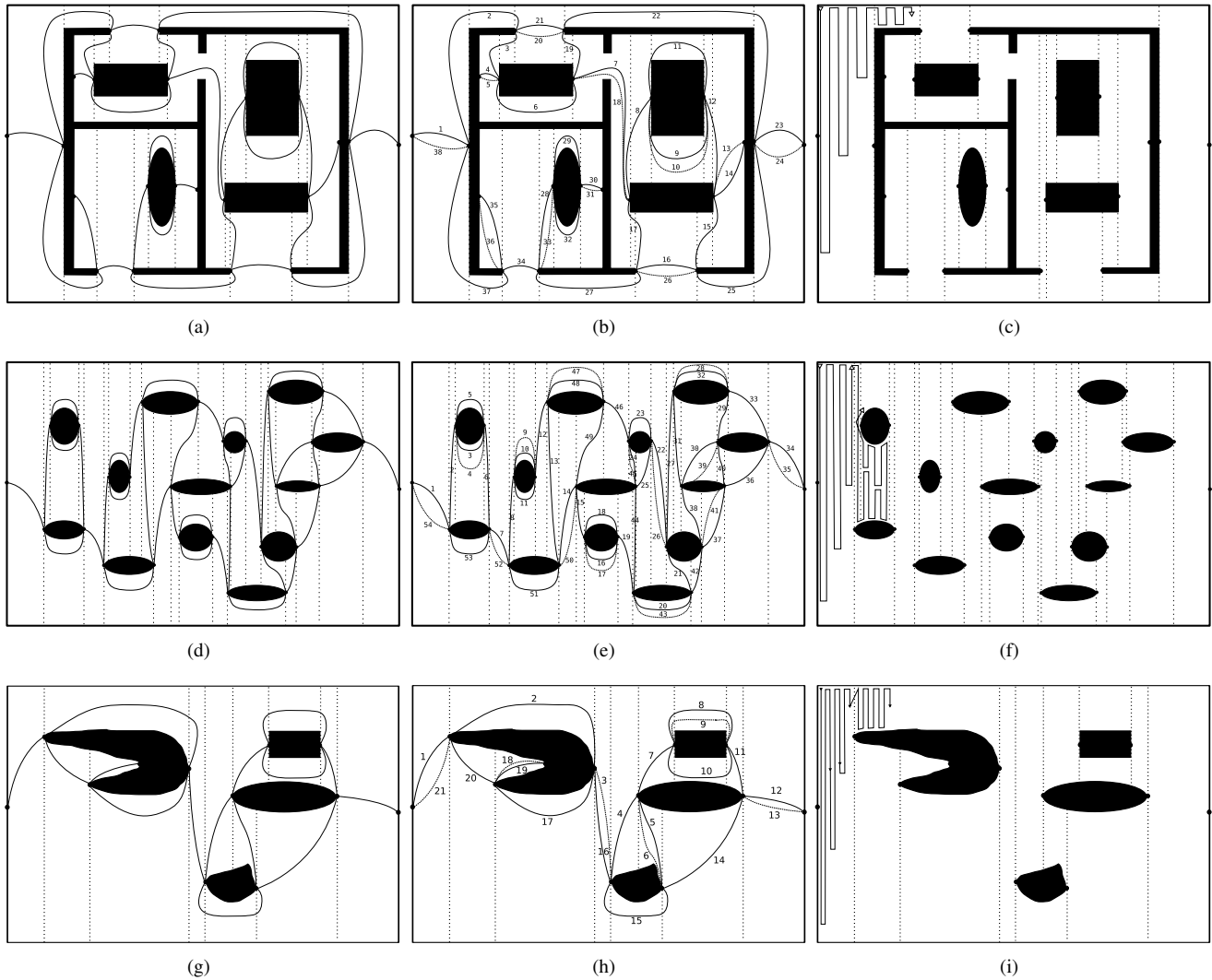


Fig. 3. (a) An environment representative of an office space, with the Reeb graph (RG, solid lines) and the Boustrophedon Cellular Decomposition (BCD, dashed lines). (b) the Euler tour for (a) with the order in which the cells are going to be visited marked. Doubled RG edges marked as dense dashed lines. (c) A snapshot of coverage. (d) An open space with sparse obstacles in line, representing an open minefield with RG and BCD. (e) The Euler tour for (d). (f) Coverage of (d). (g) An arbitrary environment populated with convex and concave obstacles with RG and BCD. (h) The Euler tour for (g). (i) Coverage of (g).

we are planning to investigate the effect of rotations versus the distance travelled in order to propose time optimal strategies extending the current approach.

Automated coverage by mobile robots is an area with many applications. With the expected increase in service robots in private households as well as in public spaces, improving coverage performance is essential. The presented algorithm is a significant step in this direction.

ACKNOWLEDGMENT

We would like to thank the generous support of Microsoft Research and of the National Research Council of Canada (NSERC) that partially funded this work. In addition we would also like to thank Prof. D. Avis for the useful discussions on Graph theory and the Chinese Postman Problem.

REFERENCES

- [1] V. J. Lumelsky, S. Mukhopadhyay, and K. Sun, "Dynamic path planning in sensor-based terrain acquisition," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 4, pp. 462–472, Aug. 1990.
- [2] I. Rekleitis, V. Lee-Shue, A. P. New, and H. Choset, "Limited communication, multi-robot team based coverage," in *Proc. of the 2004 IEEE Int. Conf. on Robotics and Automation*, 2004.
- [3] H. Choset and P. Pignon, "Coverage path planning: The boustrophedon cellular decomposition," in *Proc. of the Int. Conf. on Field and Service Robotics*, Canberra, Australia, 1997.
- [4] E. U. Acar and H. Choset, "Sensor-based coverage of unknown environments: Incremental construction of morse decompositions," *The Int. Journal of Robotics Research*, vol. 21, no. 4, pp. 345–366, April 2002.
- [5] Z. Butler, "CC R : A complete algorithm for contact-sensor based coverage of rectilinear environments," The Robotics Institute, Carnegie Mellon University, Tech. Rep. CMU-RI-TR-98-27, 1998.
- [6] W. Huang, "Optimal line-sweep-based decompositions for coverage algorithms," in *Proc. the IEEE Int. Conf. on Robotics and Automation*, vol. 1, 2001, pp. 27 – 32.
- [7] Z. Yao, "Finding efficient robot path for the complete coverage of a known space," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Beijing, China, 2006, pp. 3369–3374.

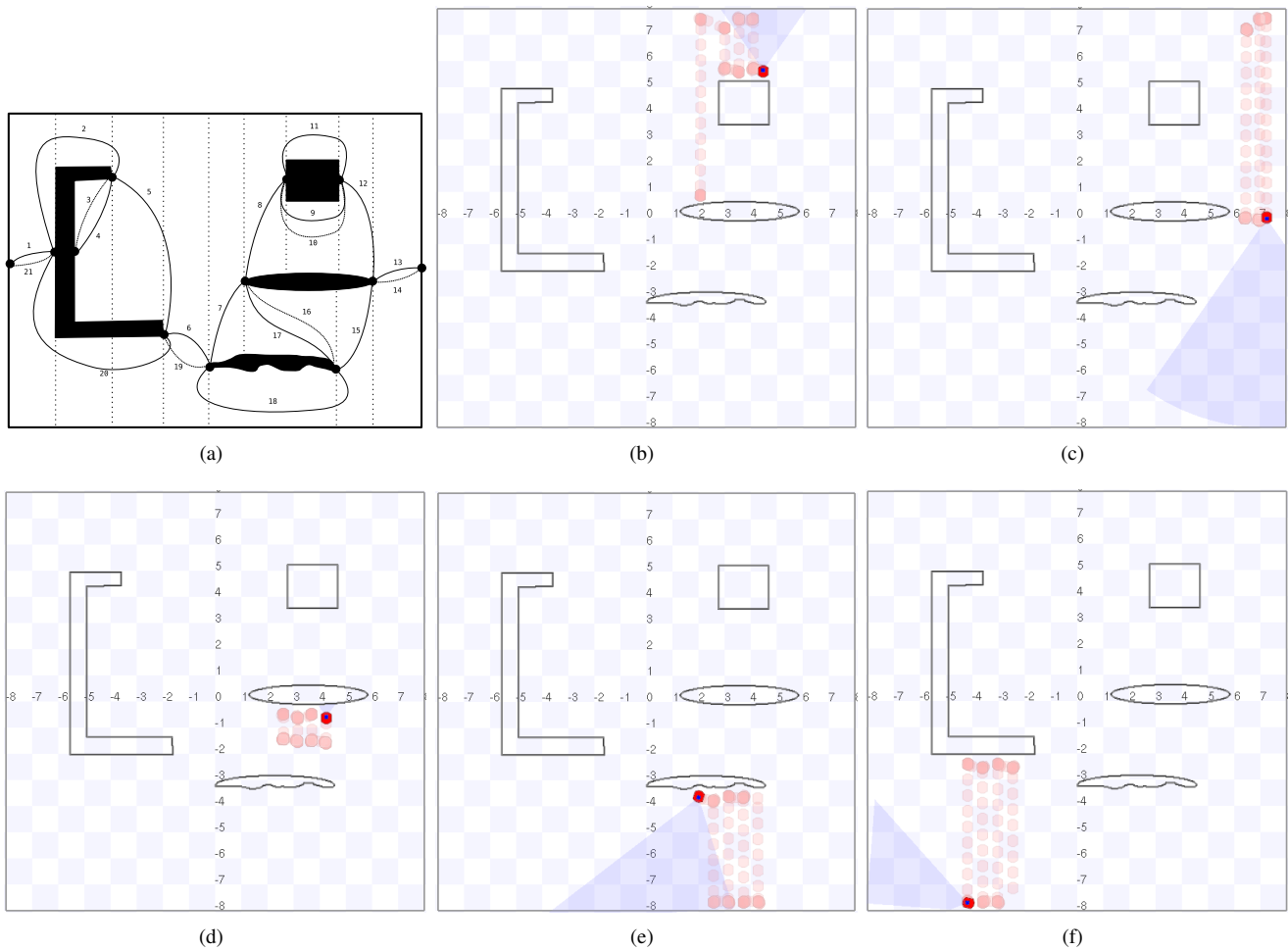


Fig. 4. (a) An arbitrary environment with the Boustrophedon Cellular Decomposition (BCD, dashed lines) and the Euler tour. (b)-(f) Screenshots from the Stage simulation of the coverage.

- [8] J. W. Kang, S. J. Kim, M. J. Chung, H. Myung, J. H. Park, and S. W. Bang, "Path planning for complete and efficient coverage operation of mobile robots," in *Proc. of the IEEE Int. Conf. on Mechatronics and Automation*, 5-8 Aug. 2007, pp. 2126 – 2131.
- [9] Y. Gabriely and E. Rimon, "Spanning-tree based coverage of continuous areas by a mobile robot," in *Proc. of the 2001 IEEE Int. Conf. on Robotics and Automation*, 2001.
- [10] —, "Spiral-stc: an on-line coverage algorithm of grid environments by a mobile robot," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, vol. 1, 11-15 May 2002, pp. 954 – 960.
- [11] X. Zheng, S. Jain, S. Koenig, and D. Kempe, "Multi-robot forest coverage," in *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, Edmonton Alberta, Canada, Aug. 2005, pp. 3852 – 3857.
- [12] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics*, vol. 20, no. 2, pp. 243–255, 2004.
- [13] S. Martinez, J. Cortes, and F. Bullo, "Motion coordination with distributed information," *IEEE Control Systems Magazine*, vol. 27, no. 4, pp. 75–88, 2007.
- [14] A. Howard, M. J. Matarić, and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem," in *Proc. of the Int. Symposium on Distributed Autonomous Robotic Systems*, 2002, pp. 299–308.
- [15] M. Schwager, J. J. Slotine, and D. Rus, "Unifying geometric, probabilistic, and potential field approaches to multi-robot coverage control," in *Proc. of the Int. Symposium on Robotics Research (ISRR 09)*, Lucerne, Switzerland, August 2009.
- [16] P. Jimenez, B. Shirinzadeh, A. Nicholson, and G. Alici, "Optimal area covering using genetic algorithms," in *Proc. of the IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, 2007, pp. 1 – 5.
- [17] M. Weiss-Cohen, I. Sirotn, and E. Rave, "Lawn mowing system for known areas," in *Proc. of the Int. Conf. on Computational Intelligence for Modelling Control and Automation*, 2008, pp. 539 – 544.
- [18] K. Easton and J. Burdick, "A coverage algorithm for multi-robot boundary inspection," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 18-22 April 2005, pp. 727 – 734.
- [19] H. Choset, "Coverage for robotics - a survey of recent results," *Annals of Mathematics and Artificial Intelligence*, vol. 31, pp. 113–126, 2001.
- [20] I. M. Rekleitis, A. P. New, E. S. Rankin, and H. Choset, "Efficient multi-robot coverage: An algorithmic approach," *Annals of Mathematics and Artificial Intelligence*, vol. 52, no. 2-4, pp. 109–142, Apr. 2008.
- [21] E. U. Acar, H. Choset, A. A. Rizzi, P. N. Atkar, and D. Hull, "Morse decompositions for coverage tasks," *The Int. Journal of Robotics Research*, vol. 21, no. 4, pp. 331–344, April 2002.
- [22] D. Meger, I. Rekleitis, and G. Dudek, "Heuristic search planning to reduce exploration uncertainty," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2008, pp. 3382 – 3399.
- [23] I. M. Rekleitis, G. Dudek, and E. Miliotis, "Multi-robot collaboration for robust exploration," *Annals of Mathematics and Artificial Intelligence*, vol. 31, no. 1-4, pp. 7–40, 2001.
- [24] H. Choset and J. Burdick, "Sensor based planning, part ii: Incremental construction of the generalized voronoi graph," in *Proc. of the IEEE Conf. on Robotics and Automation*. Nagoya, Japan: IEEE Press, May 1995, pp. 1643 – 1648.
- [25] H. Choset, "Coverage of known spaces: The boustrophedon cellular decomposition," *Autonomous Robots*, vol. 9, pp. 247–253, 2000.
- [26] J. Edmonds and E. L. Johnson, "Matching, euler tours and the chinese postman," *Mathematical Programming*, vol. 5, pp. 88–124, 1973.