Multi-Robot Exploration of an Unknown Environment, Efficiently Reducing the Odometry Error

Ioannis M. Rekleitis[1] and **Gregory Dudek**[1] and **Evangelos E. Milios**[2]

[1]Centre for Intelligent Machines, McGill University,
3480 University St., Montreal, Québec, Canada H3A 2A7
[2]Department of Computer Science, York University,
North York, Ontario, Canada M3J 1P3

Abstract

This paper deals with the intelligent exploration of an unknown environment by autonomous robots. In particular, we present an algorithm and associated analysis for collaborative exploration using two mobile robots. Our approach is based on robots with range sensors limited by distance. By appropriate behavioural strategies, we show that odometry (motion) errors that would normally present problems for mapping can be severely reduced. Our analysis includes polynomial complexity bounds and a discussion of possible heuristics.[1]

1 Introduction

The problem of exploring an unknown environment and constructing a map is central to mobile robotics. The ability to build an internal representation of the environment is also critical to most intelligent organisms. Existing approaches that have been proposed for this problem range from idealized solutions involving perfect virtual robots to practical solutions of indeterminate complexity with real robots. The range of environment and terrain types that have been considered has been similarly varied. These range from sets of polygons on the plane and abstract 3D shapes used in geometric explorations, up to real world environments, such as office buildings, lunar surfaces, underground mines, and underwater terrains. This work deals with questions of efficiency and feasibility from a theoretical standpoint. We model the world as a collection of closed 2-dimensional curves. For simplicity, we approximate free space as a polygon with holes, and then we extend the work to general shapes.

The simplest robot that can perform exploration is one equiped only with a contact sensor and internal odometry sensors. In this case, an environment can be explored (assuming free space is entirely reachable) by having the robot traverse a space filling-curve, covering the whole free space while avoiding obstacles on the way. The obvious disadvantages of this approach are:

- the long path length that must be traversed (of infinite length for a point robot, of finite length for a robot of finite size or sensing range);
- the inaccuracy of the map due to accumulated position errors (dead reckoning error).

In the case of an ideal robot with no odometry error and an ideal range scanning sensor, Lumelsky [Lumelsky *et al.*, 1990] was one of the first to develop provably correct exploration strategies, which fully map every object in the environment by circumnavigating it. Other techniques [Rao, 1995; Oommen *et al.*, 1987], representative of existing approaches, assume a polygonal world, which the robot maps by traversing the visibility graph ensuring every part is visited. Other idealized models deal with the world at a purely topological level [Deng and Mirzaian, 1996; Kuipers and Levitt, 1988]. Experimental approaches to environment exploration have also been developed [Balch and Arkin, 1994; Walker *et al.*, 1993; Bulata and M.Devy, April 1996; Elfes, 1987], demonstrating satisfactory performance in limited environments but without a performance guarantee. In contrast to these approaches, we present theoretical results but deal explicitly with the need to compensate both for odometric error and for sensing the accuracy of which deteriorates with increasing distance. We compensate for these problems by using multiple cooperating robots to explore the environment.

The organisation of this paper is as follows. In Section 2 we present the description of the world and the robot model. In Section 2.1 we analyze the advantages of cooperative robots versus a single one. In Section 3 an algorithm for exploring large areas (compared to the sensing range of the two robots) is presented. In Section 4, a performance analysis is presented, and in Section 5 the triangulation algorithm for exploring small areas is analysed. Section 6 has the conclusions and suggests

[1]Appeared in "International Joint Conference in Artificial Intelligence", vol. 2 pages 1340-1346, Nagoya, Japan, Aug. 1997.

possibilities for future work.

2 Model description and terminology

A fundamental model for the world is a simple polygon in 2D with holes. A polygon is *simple* if there is no pair of non-consecutive edges sharing a point [Preparata and Shamos, 1985]. The model of the world is essentially a set of simple polygonal obstacles contained within a larger polygonal boundary.

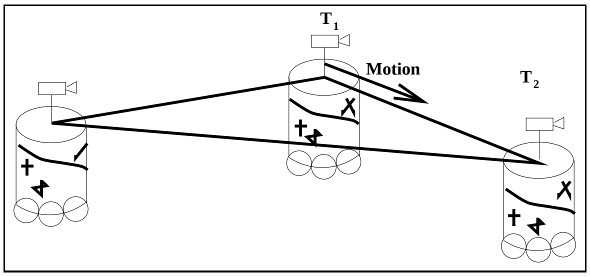
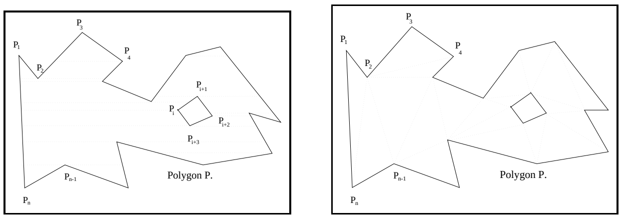


Figure 1: Accurately tracking the position of a moving robot.

Our model for robots is minimal but easily generalized: robots are points that can move in any direction, and they are equipped with two sensors. The first sensor is an *object detector*, able to detect any object in the immediate vicinity of the robot. The object detector allows wall following and object avoidance, and, in practice, the detector would be implemented by a sonar ring, an infra-red device, or even a tactile sensor. The range of the object detector is limited. The second sensor is a *robot tracker*, with the ability to locate another robot when there is a free line of sight between them, and to report accurately the distance to the second robot and its orientation. Examples of this type of sensor are a vision system that could locate a pattern on the other robot [Dudek *et al.*, 1995] (see Figure 1) or a laser range finder with a retroreflective target on the other robot. We assume that the range of the robot tracker is much larger than that of the object detector (i.e., we can see further than we can reach).

The robots explore the unknown environment by progressively covering free space in the polygonal world. Several planar decompositions have been proposed in the computational geometry literature [O'Rourke, 1987; Preparata and Shamos, 1985]. Although they apply to worlds that are completely known, they can be used as a starting point to develop "on-line" versions that construct the decomposition as part of the exploration process. The advantage offered by this approach is guarantee of full coverage without duplication, and a standard description for use in higher level reasoning.

One such systematic method is to cover free space with trapezoids. see Figure 2a. [2] An alternative decomposition of a simple polygon is by triangulation. The interior of the polygon is decomposed into triangles without adding vertices by using non-intersecting diagonals (see Figure 2b). [3]



(a) (b)

Figure 2: (A) Trapezoidation of a simple polygon with holes. (B) Triangulation of the same polygon.

2.1 Why Use Multiple Robots (Aren't Single Robots Trouble Enough)?

The use of multiple agents provides distinct advantages over single-agent systems in several contexts (this lesson has not been ignored by many insect species). The design of a robust error-free general-purpose range sensor has remained a difficult challenge. In general, high accuracy entails a limited range of operation for most devices. This in turn imposes serious constraints on the classes of mapping algorithm a robot can execute. It is possible in many applications to consider the robot and its sensor range as a *point* or a small disk that covers the space by moving through it. In that case the overall path necessary to be travelled before the whole map is constructed defines an area-filling curve swept by the robot/sensor system.

On the other hand, having one robot of a two-robot team observe and track another cooperating robot is a comparatively simple task (since there is no need to measure reflected energy from unpredictable materials in the environment, as is the case with a range sensor). If we use a pair of robots with the above described tracking sensors, then by moving one of them across the base of a triangle (for example AB) with the other at the opposite corner (for example C), they would map as free the area inside the triangle ABC ($\frac{1}{2}|AB|\alpha$ where α the distance of C to AB) by travelling only the distance $d = |AB|$. This can constitute an arbitrarily large im-

[2] For a simple polygon known *a priori* there are algorithms that construct the trapezoid decomposition in worst case $O(n\log(n))$ time.

[3] The worst case time complexity of triangulating a known polygon is $O(n)$ [Chazelle, 1990].

provement over a space-filling sweep algorithm[4].

Another major problem that arises in practice is odometry error. Due to imperfections in the construction of a real robot and the properties of the environment, mobile robots cannot avoid building up small errors in their position and orientation estimates when they move. After several steps, the robot's estimate of its position can be very different from the actual position. The traditional self-contained solution for the localisation problem is to correct the robot's position estimate by making reference to external landmarks observed using the robot's sensors. Detecting and recognizing landmarks is a difficult task in general, especially when the environment is much larger than the sensing range, and therefore the landmarks are far apart from each other.

In our work, two or more robots are used in conjunction to limit the size of odometry errors. This is accomplished by having only one robot move at any time, while the other robot(s) observes it. This allows the stationary robot to track the moving one and measure its position with higher accuracy than using simple dead reckoning. Later on, the roles are reversed: the robot that had been moving becomes the observer while another robot can move. This approach reduces the odometry error and guarantees better performance than a single robot. [5] For now, full communication is assumed, as the moving robot can obtain its current position from the observer's at any time [Dudek *et al.*, 1995]. This allows positioning to be accomplished based on the observing robots positions and independent of any environmental characteristics.

3 Exploring a large environment with two robots

When the size of the free space is much larger than the sensing range of the robots, then the trapezoid decomposition guides the exploration strategy. At any time only one robot moves, and it maps a part of the free space; then the two robots exchange roles; then the other robot moves with another part of the free space being mapped. The exploration algorithm consists of two logical parts: the *local* exploration, which sweeps a horizontal stripe of free space inside one trapezoid, and the *global* one, which connects the stripes together and decides which part to explore next.

3.1 Local exploration

When one robot moves in a straight line, then the area mapped has the shape of a triangle as in Figure 3. If the only parameter to be optimized was the total path traveled, then each robot would move by a δl at distance $d = R$, and then the two robots would exchange roles and the other one would move by a δl, continuing in this way up until they reach a wall. In terms of path travelled, the algorithm would be optimal because each time they cover the maximum area possible, and no area is covered twice. Unfortunately, the exchange of roles has some cost, associated with acceleration/deceleration and location update. Therefore, the number of exchanges should be taken into account. We model the cost as a function of the angle θ as in Figure 3. The simplest way to explore a stripe is by moving the robots into two parallel lines, at a distance appropriate for the number of exchanges that are considered acceptable (the closer they come the smaller the number of exchanges). In this model, each time one robot moves a triangular area is covered. Table 1 presents the number of exchanges, the total path travelled, and the number of rotations for the exploration of a rectangular area XY. While that algorithm has good performance, it is not optimal in terms of path length or the number of exchanges. The optimal length path [Rekleitis *et al.*, 1997] occurs when the two robots explore the maximum area possible at any time without overlaps. An example of the optimal path can be seen in Figure 4, where the area explored each time is forming a diamond shape (e.g. $T_2T_0T_3$). Table 1 presents the number of exchanges, the total path travelled, and the number of rotations for the exploration of a rectangle XY, when each robot move covers a diamond shaped area.

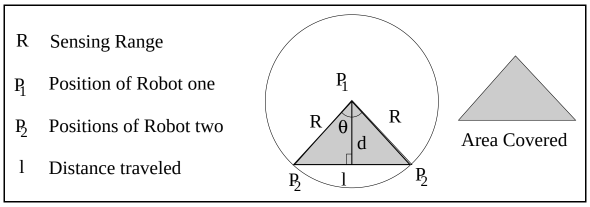


Figure 3: Area covered when one robot moves in a straight line.

More precisely, in the example in Figure 4, the two robots are "awakened" at time T_0 next to each other. After an initial scan of the environment, the robot R_2 moves away from robot R_1, which remains stationary until R_2 reaches a distance $d = R$, distance that gives the maximum covered area while accurately locating the position of R_2, (time T_1). Then the robot R_2 moves to a new position (time T_2), mapping the area $T_0T_1T_2$ as free space. Consequently, the robot R_2 becomes stationary and the robot R_1 maps a new area $T_0T_2T_3$ (time T_3). Then they exchange roles again and continue. When the two robots reach the end of the stripe, (time T_n), they

[4] In practice, even line of sight tracking is range limited and can be described as a sweep, but in this case the sweeping figure can be extremely large.

[5] The corrections from the stationary robot could be combined with the dead reckoning technique, using Kalman filtering to give more accurate estimates.

move to the new positions (time T_{n+1}) and explore the next stripe in the opposite direction. It is worth noting the effect of the reflex vertices in the order of exploration.

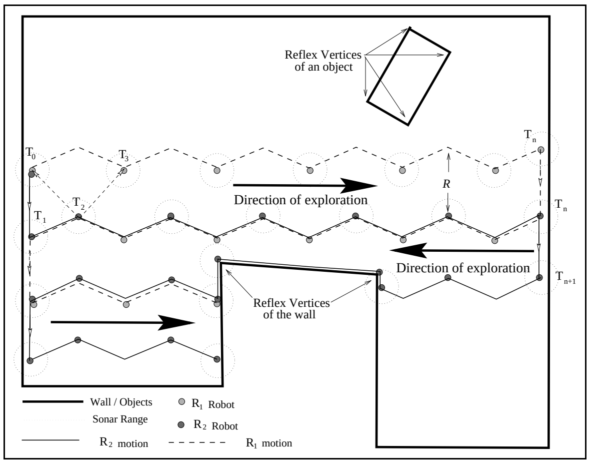


Figure 4: Exploration of a large storage space.

3.2 Global exploration

The order in which the stripes are explored is given by a depth first search algorithm. From the collection of the stripes we construct the dual graph by matching every explored stripe to a vertex of the graph and every pair of adjacent stripes to an edge connecting the two corresponding vertices. At any point, after visiting a node in the graph (in other words, after exploring the corresponding stripe of free space), there are a maximum of two choices (except for the initial step) for which stripe/node is to be explored next. In general every stripe is connected to one above and one below. If, during the exploration, a robot encounters a reflex vertex, [6] then a decision point is introduced and consequently an extra edge is added on that node (see Figure 5). Every time a reflex vertex is encountered a decision is made, and one branch of the graph is followed. In our approach a depth first search strategy is used in order to determine which edge of the dual graph the robots are going to follow in the exploration. It is worth noting that, in order to have optimal results, the deepest branch of the graph should be explored last, but without a-priori knowledge this is impossible to determine in advance. Various heuristics could be applied, such as exploring the narrowest or the widest opening first, depending on previous environmental knowledge. Regarding the complexity of the dual graph, if the environment has obstacles in it (modeled as holes in the simple polygon), then the graph contains cycles; otherwise it is a tree.

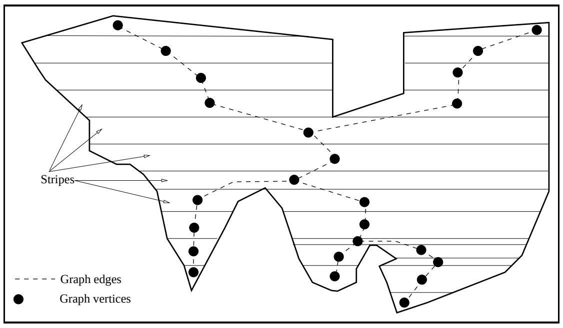


Figure 5: Stripes to Graph modeling

The area exploration problem now is equivalent to a graph exploration, and the complexity (in terms of robot edge transitions) is linear in the number of reflex vertices in the environment.

3.3 More than two robots

An immediate extension of the previous algorithm can be obtained by the addition of more robots. When the two robots sweep one stripe of width d then by adding an extra robot (50% increase) we could double the area swept. In the original algorithm, every robot has only one device to track the other robots; in this case a scheduling algorithm should be applied in the order the robots are moving. If we add a second tracking device, one robot could track robots on both sides, allowing a parallel cover of double the area at the same time.

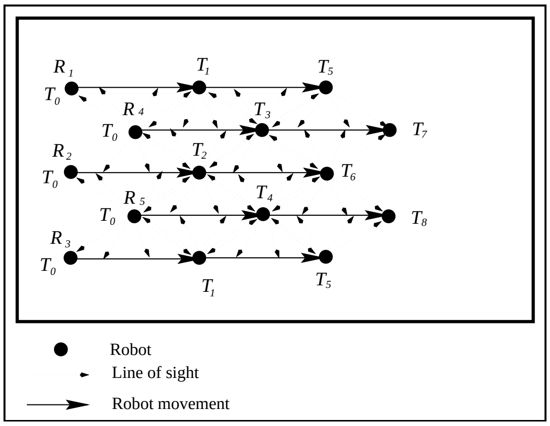


Figure 6: Exploration of a stripe with 5 robots.

In the example in Figure 6 we use five robots ($R_1 \dots R_5$) that are positioned in two lines at time T_0, and we assume that each robot can track only one other robot at a time. First the robots R_1, R_3 move forward,

[6]For a single polygon P a reflex vertices are the concave vertices. For obstacles, inside the polygon, reflex vertices are the convex vertices of the obstacle (see for example Figure 4).

tracked by R_4 and R_5 accordingly, mapping the two triangles as free space, then both R_4, R_5 track R_2, which moves forward at the position (marked as time) T_2. Then it is time for the next column of robots (R_4, R_5) to advance (one at a time due to the tracking devices), marking more area as free space. The tracking is marked with the dotted lines of sight. The same pattern is followed as the two columns alternatively advance, marking a stripe of free space much wider than that possible with only two robots.

The second part of the algorithm concerning the exploration strategy for the whole space and the order in which the trapezoids should be explored is identical to the previous algorithm where only two robots were used.[7]

4 Complexity analysis

The algorithm consists of two states: the local exploration of a stripe, and the global exploration that determines the order in which the adjacent stripes are explored. Consequently the complexity of the algorithm reflects these two states. The exploration is performed in stripes that are covered one after the other, forming trapezoids. The trapezoids then are joined together to form the complete map. The total travel of the two robots while they explore new space is the sum of the perimeters of the different stripes. For a single rectangle, analytical results are given in Table 1. Details of this procedure and associated bounds appear elsewhere [Rekleitis *et al.*, 1997]. The second quantity is the net travel inside known territory to visit the boundaries of the unmapped territory. This is a function of the number of reflex vertices of the free space polygon (given by a depth first search algorithm). The trapezoid decomposition covers all of the free space with a finite number of stripes (trapezoids). The algorithm methodically explores every one of these stripes, and it never repeats the exploration. Therefore, after the two robots explore all the stripes the algorithm is guaranteed to terminate with a complete map.

5 Small Environments - Triangulation

This specialized algorithm operates in an environment where the visual sensing range is at least as large as the diameter of the polygon. The output of this algorithm is a map of the free space decomposed into triangles.

Our proposed exploration algorithm starts from an arbitrary position in the environment and proceeds to map it as a set of convex polygon/shapes of free space connected as a graph, in the case of a simple polygon with holes, or as a tree, in the case of a simple polygon. As an initial step, the two robots sense the closest wall proceed to move to it and position themselves in opposite corners. The triangulation algorithm then moves one robot by following the walls, maintaining line of sight contact with the other robot, which remains stationary at a corner of the polygon. Again the complexity increases linearly with the number of reflex vertices. Every triangle represents a node in the dual graph, and adjacent triangles represent edges among the corresponding nodes. Every time the line of sight is broken by a reflex vertex, a decision is made and another edge is added to the graph. The total path traveled again depends on two measures. The exploration cost is equal to the perimeter of the polygon. The cost of traversing some edges of the path twice, is linear with the number of reflex vertices, and the path length is bounded by the maximum distance between any two points in the polygon for every edge traversed twice.

[7] There is a possible speedup by splitting up the group in order to explore different parts in critical points, but that would in the end spread the robots too thin.

Covering	Triangle Area	Diamond Area
Total path	$2Y + \frac{XY}{R}\frac{2\sqrt{2}}{\sqrt{1+\cos\theta}}$	$2Y + \frac{XY}{R}\frac{4\sqrt{1-cos(\theta/2)}}{\sqrt{1-cos\theta}}$
# of steps	$\frac{2XY}{R^2}\frac{1}{\sin\theta}$	$\frac{2XY}{R^2}\frac{\sqrt{2}}{2\sqrt{1-\cos\theta}}$
# of turns	$\frac{2\sqrt{2}Y}{R\sqrt{1+\cos\theta}}$	$\frac{2Y}{R} + \frac{4X}{R\sqrt{2-2\cos\theta}}$

Table 1: Analytical complexity of two different path curves.

An example is presented in Figure 7, the two robots start at the two positions marked T_0, and robot R_2 then starts exploring the free space, following the walls of the polygon up until the line of sight is broken by a reflex vertex upon which it switches roles with robot R_1. All of the free space is mapped, except for the areas in which a bifurcation in the sweep was forced due to a reflex vertex. Figure 8 illustrates the final stage of the algorithm which is used to map the remaining areas. The robots plan their path through the mapped area for the fastest route that will take them to the unmapped areas and then proceed to reach these areas and explore them.

In the general case, the two algorithms should be used together: when the robots approach a closed space where they can *"see"* each other from wall to wall, the triangulation algorithm should be used to map it. When they move into an open area, the trapezoid decomposition algorithm should then be used to sweep the area.

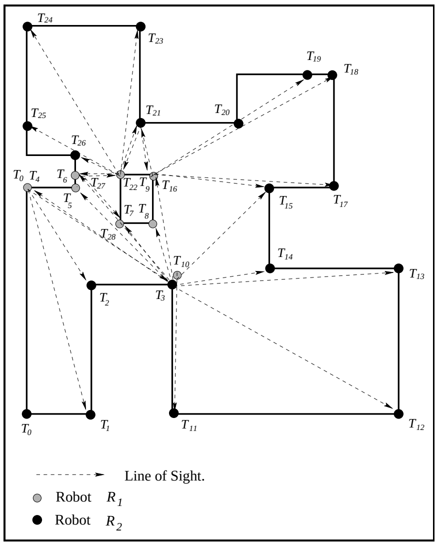


Figure 7: Triangulation like exploration of an unknown environment. The general algorithm.

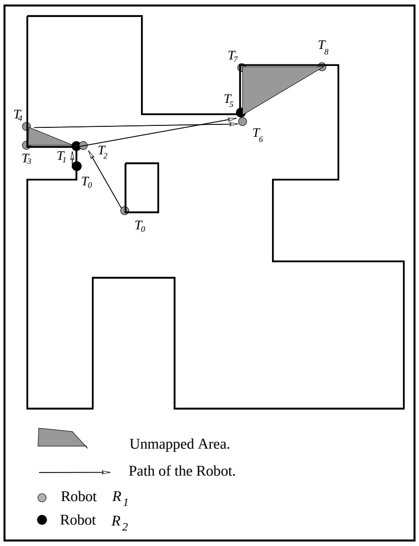


Figure 8: Triangulation-like exploration of an unknown environment. Cleaning-up the leftovers.

6 Conclusions

Different techniques for mapping the environment have been used since ancient times in Egypt and Rome. Most of them involved collaboration among different observers in order to improve their accuracy. In this paper, a new algorithm for exploring an unknown environment is proposed. Our algorithm uses a well-known planar decomposition form in order to systematically explore the free area of of an unknown environment modelled as a simple polygon with holes. The trapezoid decomposition is used for large areas ensuring an exploration strategy that finishes with the total free space mapped as a set of trapezoids. For small areas, a triangulation of the free space is returned.

Realistic assumptions, such as odometry error, and sensing that deteriorates with distance, are used. Both algorithms return a complete map, while a single robot would encounter great difficulties in such a case. The approach acts to minimize the effects of inherent navigation errors, while providing a performance guarantee (unlike heuristic methods). We are currently involved in experimental evaluation of these algorithms.

References

[Balch and Arkin, 1994] Tucker Balch and Ronald C. Arkin. Communication in reactive multiagent robotic systems. *Autonomous Robots*, 1(1):27–52, 1994.

[Bulata and M.Devy, April 1996] H. Bulata and M.Devy. Incremental construction of a landmark-based and topological model of indoor environments by a mobile robot. *Proc. IEEE Conference on Robotics and Automation*, pages 1054–1060, April 1996.

[Chazelle, 1990] Bernard Chazelle. Triangulating a Simple Polygon in Linear Time. In *Proceedings of the* 31^{st} *Annual Foundations of Computer Science*, pages 220–229. IEEE Computer Society Press, 1990.

[Deng and Mirzaian, 1996] Xiaotie Deng and Andranik Mirzaian. Competitive robot mapping with homogeneous markers. *IEEE Trans. on Robotics and Automation*, 12(4):532–542, August 1996.

[Dudek *et al.*, 1995] Gregory Dudek, Michael Jenkin, Evangelos Milios, and David Wilkes. Experiments in sensing and communication for robot convoy navigation. In *Proceedings IEEE International Conference on Intelligent Robots and Systems (IROS)*, volume 2, pages 268–273, Pittsburgh, PA, August 1995.

[Elfes, 1987] Alberto Elfes. Sonar-based real-world mapping and navigation. *IEEE Transactions on Robotics and Automation*, 3(3):249–265, June 1987.

[Kuipers and Levitt, 1988] B. Kuipers and T. Levitt. Navigation and mapping in large-scale space. *AI Magazine*, pages 25–43, summer 1988.

[Lumelsky *et al.*, 1990] Vladimir J. Lumelsky, Snehasis Mukhopadhyay, and Kang Sun. Dynamic path planning in sensor-based terrain acquisition. *IEEE Transactions on Robotics and Automation*, 6(4):462–472, August 1990.

[Oommen *et al.*, 1987] B. Oommen, S. Iyegar, S. Nageswara, S. Rao, and R. Kashyap. Robot navigation in unknown terrains using learned visibility graphs, part i: The disjoint convex obstacle case. *IEEE J. of Robotics and Automation*, 3(6):672–681, 1987.

[O'Rourke, 1987] J. O'Rourke. *Art Gallery Theorems and Algorithms.* Oxford University Press, New York, NY, 1987.

[Preparata and Shamos, 1985] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction.* Springer-Verlag, New York, NY, 1985.

[Rao, 1995] Nageswara S. V. Rao. Robot navigation in unknown generalized polygonal terrains using vision sensors. *IEEE Transactions on Systems, Man, and Cybernetics*, 25(6):947–962, June 1995.

[Rekleitis *et al.*, 1997] Ioannis Rekleitis, Gregory Dudek, and Evangelos Milios. Optimality proof for the trapezoid decomposition. Technical Report TR-CIM-97-01, Centre for Intelligent Machines, McGill University,, 3480 University St., Montreal, Québec, Canada H3A 2A7, January 1997.

[Walker *et al.*, 1993] Ashely Walker, Hihn Hallman, and David Willshaw. Bee-havior in a mobile robot: The construction of a self-organized cognitive map and its use in robot navigation within a complex, natural environment. In *IEEE Conf. on Neural Networks*, pages 1451–1456, 1993.