Exploring Unknown Environments with Obstacles

Susanne Albers* Klaus Kursawe* Sven Schuierer†

1 Overview

We study exploration problems where a robot has to construct a complete map of an unknown environment using a path that is as short as possible. In the first problem setting we consider, a robot has to explore a geometric scene with n rectangles. We show that no deterministic or randomized online algorithm can be better than $\Omega(\sqrt{n})$-competitive, solving an open problem by Deng *et al.* [2]. We also generalize this bound to the problem of exploring three-dimensional rectilinear polyhedra without obstacles. In the second problem setting we study, a robot has to explore a grid graph with obstacles in a piecemeal fashion. The piecemeal constraint was defined by Betke, Rivest and Singh [1] and implies that the robot has to return to a start node every so often. Betke *et al.* give an efficient algorithm for exploring grids with rectangular obstacles. We present an efficient strategy for piecemeal exploration of grids with arbitrary rectilinear obstacles.

2 The New Lower Bound

We consider geometric exploration problems where a robot is placed in a room with obstacles. The exterior wall of the room as well as the obstacles are modeled by simple polygons. The robot has 360° vision. Its task is to move through the scene so that it sees all parts of the room. More precisely, every point in the room must be visible from some point on the path traversed. Deng *et al.* [2] showed that no online exploration algorithm can achieve a constant competitive ratio in scenes with arbitrary polygonal obstacles. The obstacles used are diamonds. We can prove that no online exploration algorithm can be better than $\Omega(\sqrt{n})$-competitive in scenes with n obstacles, even if the obstacles are rectangles. Our proof is based on a new, recursive construction of a scene.

THEOREM 2.1. *Let A be an online algorithm for exploring two-dimensional scenes with n rectangles. If A is c-competitive, then $c = \Omega(\sqrt{n})$.*

We sketch how to prove this theorem. Given an exploration algorithm A, we construct a scene with n rectangles. Let n be a positive integer and $k = \lfloor\sqrt{n/2}\rfloor$. Let $\varepsilon = \frac{1}{2}(2k)^{-k}$. The obstacles used in the construction are *k-combs*, as depicted in Figure 1. A k-comb consists of k *spike rectangles*, or *spikes* for short, that span the whole width of the k-comb, and $k-1$ *base rectangles* that have width of $w_b = 1$. The side of a base rectangle that is aligned with the spikes is called the *outer side*; the opposite side is called the *inner side*. The distance between a spike and a base rectangle is $\varepsilon' = \varepsilon^2/(2k)$.

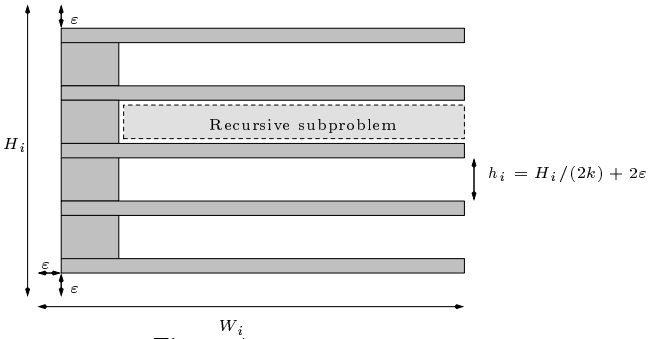

Figure 1: A k-comb in stage i.

The construction of the scene proceeds in k *stages*. In each stage i, $1 \le i \le k$, exactly one k-comb is used. Thus, $k(2k-1) \in \Theta(n)$ rectangles are placed in total. The crucial property of the scene is that the k-comb of stage i, $2 \le i \le k$, is placed between two spikes of stage $i-1$. For $i = 1, \ldots, k$, let W_i and H_i be the lengths available in x- and y-direction to place the k-comb of stage i. We set $H_1 = k$ and $W_1 = 2k$. The construction is such that $W_i = 2k - (w_b + \varepsilon)(i-1) \ge k$ and $H_{i+1} = H_i/(2k)$. We specify the placement of the k-comb for stage $i+1$ in stage i, $1 \le i \le k$. In this abstract we only consider the case that the exploration algorithm A used by the robot is deterministic. Assume that the robot is located at the start point of the scene, which is either the lower left or the upper left corner of the k-comb in stage i. The k-comb of stage i is placed such that the robot faces the outer sides of the base rectangles. Thus, if the robot enters stage i to the right of a spike in stage $i-1$, then the k-comb is a mirror image of the arrangement given in Figure 1. To find the k-comb of stage $i+1$, one option of the robot is to explore the inner sides of the base rectangles, one by one. It will succeed only at the very last base rectangle. Alternatively, the robot can move to the other side of the k-comb. In this case the k-comb of stage $i+1$ may be between two spikes of any unexplored base rectangle. We can show that the path used by A is at least $\Omega(k) = \Omega(\sqrt{n})$ times as long as the path used by an optimal offline algorithm OPT.

In our construction, the rectangles become very thin. We can modify the scene so that a unit diameter circle can

*Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, FRG. Email: {albers,kursawe}@mpi-sb.mpg.de

†Institut für Informatik, Am Flughafen 17, Geb. 051, D-79110 Freiburg, FRG. Email: schuiere@informatik.uni-freiburg.de

be inscribed into any rectangle. Furthermore, we can prove the following result.

THEOREM 2.2. *Let A be an online algorithm for exploring a simple three-dimensional rectilinear polyheron without obstacles. If A is c-competitive, then $c = \Omega(\sqrt{n})$ where n is the number of vertices of the polyhedron.*

3 Piecemeal Exploration

So far we assumed that the robot can see an infinite range as long as no obstacle or exterior wall blocks the view. However, in practice, a robot's sensors are often only able to scan a distance of a few meters. This situation can be modeled by adding a grid to the scene, as shown in Figure 2, and requiring that the robot moves on the nodes and vertices of the grid. A node in the grid models the vicinity that the robot can see at a given point. Now the robot has to explore all nodes and edges of the grid using as few edge traversals as possible. A node is explored when it is *visited* for the first time and an edge is explored when it is *traversed* for the first time. At any node the robot knows its global position and the directions of the incident edges.

Betke, Rivest and Singh [1] introduced an interesting variant of this problem where an additional *piecemeal constraint* has to be satisfied, i.e, the robot has to return to a start node s every so often. These returns might be necessary because the robot has to refuel or drop samples collected on a trip. Betke, Rivest and Singh [1] developed two algorithms for piecemeal exploration of grids with rectangular obstacles. The algorithms, called *Wavefront* and *Ray*, need $O(|E|)$ edge traversals where $|E|$ is the number of edges in the graph.

Consider a grid graph with arbitrary obstacles. Let r be the radius of the graph, i.e., r is the maximum of all shortest path distances between s and any other node in the graph. We assume that the robot can traverse a total of $2R$ edges, with $R = (3 + \alpha)r$ and $\alpha > 0$, between two consecutive visits to s. We have developed an algorithm that explores an unknown grid with arbitrary obstacles using $O(|E|)$ edge traversals.

Our algorithm is a generalization of the *Ray* algorithm [1]. The original *Ray* algorithm only explores grids with rectangular obstacles. It is essential in the algorithm that the robot always knows a path back to s that has a length of at most r. When exploring grids with arbitrary obstacles we cannot always satisfy such a constraint. We solve this problem by developing an efficient strategy for exploring the boundary of arbitrary obstacles.

In this abstract we assume that the exterior boundary of the grid is a rectangle and that the start node s is located in the bottom left corner of the scene. At any time *dist* denotes the number of edges traversed by the robot since the last visit to s. An obstacle is *new* if none of the nodes and edges on the boundary are explored. An obstacle is *mapped fully* if all nodes and edges on the boundary are explored. For an obstacle O, an *open segment* S of O is a maximal sequence consecutive horizontal edges on the boundary of O such that the interior of O is to the south of S.

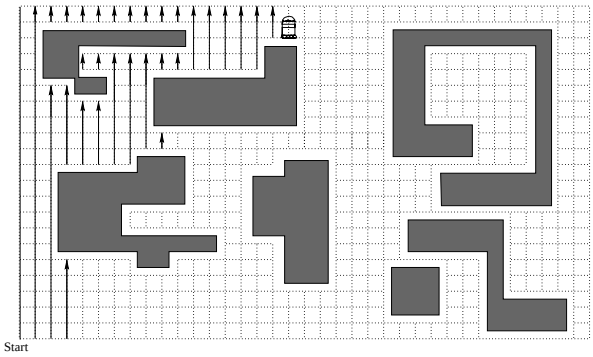

Figure 2: The *Ray* algorithm

The actual exploration of the scene proceeds in rays. Let S be the bottom segment of the exterior boundary and let $s = x_1, \ldots, x_n$ be the vertices of S. For every i, $1 \leq i \leq n$, the robot starts at node x_i and explores a vertical ray of edges in northern direction until an obstacle or the exterior boundary is hit. The *Ray* algorithm applies a depth-first strategy, as illustrated in Figure 2. Whenever the robot hits a new obstacle O at a node y while exploring a ray R_i started at x_i, the robot immediately explores the boundary of O using a procedure *Map-Obstacles* that we will sketch below. The robot then visits the open segments of O in clockwise direction and explores vertical rays started at these segments. The same strategy is applied recursively to new obstacles discovered while the robot explores rays started at O's segments. When all the open segments of O are visited, the robot moves to y and backtracks along R_i to x_i.

The crucial part of our *Ray* algorithm is a procedure *Map-Obstacles* that is called every time the robot hits a new obstacle O at a node x. When *Map-Obstacles* terminates, O and all obstacles hit during the execution of the procedure are mapped fully and the robot is located again on x. Initially, the robot moves in, say, clockwise direction along the boundary of O and tries to reach x again while $dist \leq R$. If it succeeds in reaching x, then O is mapped fully and the call of *Map-Obstacles* terminates. If the robot cannot reach x while $dist \leq R$, then the execution of the procedure is more involved. The robot maintains a set of *partially mapped* obstacles for which only some nodes or edges on the boundary are explored. For any partially mapped obstacle, the robot keeps track of the *explored segments*, which are maximal sequences of explored nodes and edges on the boundary of the obstacle. These explored segments are grown further and further until the obstacle is mapped fully. Details are given in the full paper.

THEOREM 3.1. *The Ray algorithm explores a grid with arbitrary obstacles using $O(|E|)$ edge traversals.*

References

[1] M. Betke, R. Rivest and M. Singh. Piecemeal learning of an unknown environment. *Proc. 5th Conf. on Computational Learning Theory*, 277–286, 1993.

[2] X. Deng, T. Kameda and C. H. Papadimitriou. How to learn an unknown environment. *Journal of the ACM*, 45:215–245, 1998.