A Near-Tight Approximation Lower Bound and Algorithm for the Kidnapped Robot Problem *

Sven Koenig† Apurva Mudgal‡ Craig Tovey§

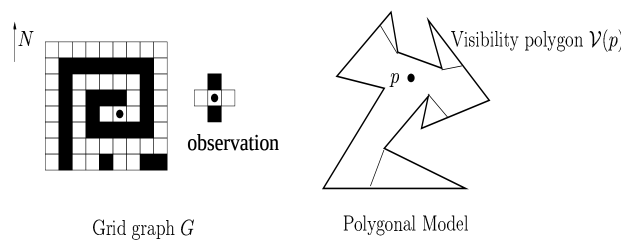
Abstract

Localization is a fundamental problem in robotics. The 'kidnapped robot' possesses a compass and map of its environment; it must determine its location at a minimum cost of travel distance. The problem is NP-hard [6] even to minimize within factor $c \log n$[21], where n is the number of vertices. No approximation algorithm has been known. We give a $O(\log^3 n)$-factor algorithm. The key idea is to plan travel in a 'majority-rule' map, which eliminates uncertainty and permits a link to the $\frac{1}{2}$-Group Steiner (not Group Steiner) problem. The approximation factor is not far from optimal: we prove a $c \log^{2-\epsilon} n$ lower bound, assuming $NP \not\subseteq ZTIME(n^{polylog(n)})$, for the grid graphs commonly used in practice. We also introduce a new hypothesis equivalence decomposition of the plane, built from pairs of aspect graph duals, in order to extend the algorithm to polygonal maps.

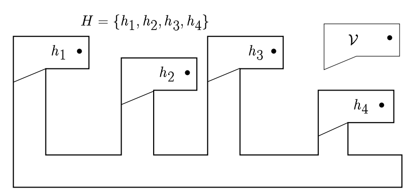
1 Introduction

Consider the following problem: a mobile robot is placed at an unknown position in an environment for which it has a map $\mathcal{E}$. The robot constructs a map $\mathcal{E}'$ of its local environment by going to different places and sensing the environment from there. It rules out positions whose local environment does not agree with map $\mathcal{E}'$, until it infers the unique position where it was located. The objective is to do so by traveling the minimum distance. This is known as the *kidnapped robot* or *localization* problem [4, 15].

Motivation. In general, robots must localize when they are switched on because they may have been moved while switched off. Also, the control systems guiding a robot gradually accumulate error due to mechanical drift and sensor noise [5]. Thus it is necessary to localize from time to time to verify the actual position of the robot in the map, and apply corrections. In this context, localization eliminates the need for complex and expensive position-guidance systems such as radio beacons [3, 4], to be installed in buildings or streets with tall buildings, where three satellites are not in view and so GPS is not effective. For situations in which such systems cannot be built, such as a Mars rover (see [14]), localization is the only possiblity.



(a) Models



(b) Hypothesis generation

Figure 1:

Model. We study localization on two well-known models: grid graphs and polygons. A grid graph G is a finite rectangular area of square cells as shown in Figure 1(a). Each cell can be either blocked or traversable. A robot is always in exactly one traversable cell. It starts in a traversable cell and can then always move to the cell to its north, south, east or west. Tactile sensors allow the robot to find the states (blocked / traversable) of its four neighbouring cells. In the polygonal model [22, 4], the environment is a simple polygon P and the robot occupies exactly one point $p \in P$. The robot is equipped with a *range finder*, a device that emanates a beam (laser or sonic) and determines distance to the first point of contact with P's boundary in that direction.

*supported by NSF awards IIS-99427, IIS-00907, and ITR/AP-01131.

†Computer Science Dept., University of Southern California. skoenig@usc.edu

‡College of Computing, Georgia Tech. apurva@cc.gatech.edu

§College of Computing, Georgia Tech. ctovey@cc.gatech.edu

The robot sends out a series of beams spaced at regular intervals about its position, measuring the distance to the boundary at each of these angles. The "points of contact" are then joined together to obtain a *visibility polygon* $\mathcal{V}$ with m vertices (see Figure 1(a)). We use n to denote the size of the map: for grid graphs n is the number of cells in G, and for the polygonal model it is the number of vertices in map polygon P.

Following Dudek *et al* [6], we view the localization problem as consisting of two phases: *hypothesis generation* and *hypothesis elimination*. The first phase is to determine the set H of hypothetical locations (or *hypotheses*) that are consistent with the sensing data obtained by the robot from its initial location (see Fig. 1(b)). The second phase is to determine, in the case $|H| > 1$, which location $h \in H$ is the true location of the robot. For grid graphs the robot can take H to be set of all traversable cells and proceed to phase 2. For polygons, Guibas *et al* [11] provide an algorithm which generates at most n hypotheses using the visibility polygon observed by the robot. Thus to solve the problem we focus on hypothesis elimination.

By a *strategy* S we mean the hypothesis elimination routine in robot's computer. We measure the effectiveness of a strategy by the *worst-case* criterion. For strategy S, let $W(h, S)$ be the distance traveled to localize if the robot was placed at hypothesis $h \in H$. Then the cost $W(S)$ is taken to be the maximum distance $W(h, S)$ traveled for any starting position h i.e., $W(S) = max_{h \in H} W(h, S)$. The optimal strategy S^* has cost $\min_{S \in \mathcal{S}} W(S)$, where $\mathcal{S}$ denotes the set of all possible localization strategies. OPT(G,H) denotes the cost of the optimal strategy, where G is the map and H the set of hypotheses. We say that a strategy is α-approximate if its cost is at most $\alpha\cdot$ OPT(G,H).

Previous Work. Despite the considerable attention it has received in the robotics literature (e.g. [4, 15, 20, 17, 22]), localization has been the subject of less theoretical work. Dudek *et al* [6] show it is NP-hard to find an optimal strategy; Koenig and Tovey [21] show it is NP-hard to find a $c \log n$-approximate strategy on grid graphs or polygons. Papadimitrou and Yannakakis [16] introduce competitive analysis (Sleator and Tarjan [18]) into the realm of navigation problems with incomplete information. Kleinberg [13] gives an $O(n^{\frac{2}{3}})$-approximate competitive algorithm for localization on geometric trees, which asymptotically improves the "spiral search" technique of Baeza-Yates *et al.* [1]. Dudek *et al.* [6] give a $2(k-1)$-competitive algorithm in the polygonal model, where $k = |H|$.

We briefly discuss the advantages of worst-case cost over the competitive ratio introduced in [16]. Competitive algorithms compare the distance traveled by a robot to that by an *omniscient verifier* i.e, a robot which has *a priori* knowledge of its position $h \in H$ and probes the environment just to verify this information. In other words, the distance traveled by an omniscient verifier is exactly $min_{S \in \mathcal{S}} W(h, S)$ and a α-competitive strategy makes the robot travel at most $\alpha \cdot min_{S \in \mathcal{S}} W(h, S)$. From a practical standpoint, worst-case cost better matches the roboticist's concerns with guaranteed rapid localization, rather than with comparisons against an omniscient verifier. From a theoretical standpoint, it admits a $O(\log^3 n)$ approximation algorithm, whereas it is NP-hard achieve a strategy with competitive ratio $o(\sqrt{n})$ on polygons [6].

Results. The main contributions of this paper are (section 2) a polynomial time $O(\log^3 n)$-approximate strategy, and (section 3) a $\log^{2-\epsilon} n$ approximation lower bound, assuming $NP \not\subseteq ZTIME(n^{polylog(n)})$, for the grid graphs commonly used in practice. The key algorithmic idea is to plan travel in a 'majority-rule' map, which eliminates uncertainty and permits a link to the $\frac{1}{2}$-Group Steiner (not Group Steiner) problem (defined in section 2.3). Section 4 extends the algorithm to robots with line-of-sight (i.e., range finder) sensors in polygons, by means of a new *hypothesis equivalence* decomposition of the plane, built from pairs of aspect graph duals. Section 5 sketches extensions to polygons with holes and robots without compasses.

The basic framework of the strategy is to break the task into a series of 'half-localize' steps:

HALF-LOCALIZE(G,H): Devise a strategy by which the robot can correctly eliminate at least half of the hypotheses in H. The robot should travel (worst-case) distance as small as possible to achieve this. HALF-OPT(G,H) denotes the cost of the optimal strategy.

Intuitively it might appear that an $O(\log^2 n)$ algorithm for half-localization should be a by-product of our $O(\log^3 n)$ localization strategy and not vice-versa. As an example of this, consider the $\frac{1}{2}$-Set Cover problem where the objective is to cover half the elements at minimum cost. There is a constant factor approximation for this; and it is obtained by stopping the $O(\log n)$ greedy algorithm for Set Cover as soon as we cover half the elements. (Another example is the algorithm for $\frac{1}{2}$-Group Steiner [7], which is obtained by stopping the rounding scheme of [9] as soon as the tree covers half the groups.)

However, half-localize seems to play a more fundamental role in our context. We discuss only the simpler grid graph case here. Construct a 'majority-rule' map, in which each cell is blocked or unblocked depending on what the majority of the current hypotheses in H assert. This majority-rule map permits three inter-related simplifications. If the robot tries to follow a route on the majority-rule map, but makes a minority observation (one inconsistent with at least half the hypotheses), the robot has by definition half-localized. This permits a plan to be a path rather than a decision tree. Distances in the real environment are uncertain, but distances on the majority-rule map are fixed. This permits us to model half-localize as a Steiner type problem on a graph, although we are not able to model localization as such. Finally, there is an essential equivalence between optimally half-localizing and halving paths (section 2.2) on the majority-rule map.

2 Strategy for grid graphs

2.1 Preliminaries A hypothesis h is *active* at a point of time if the robot has not yet eliminated it as a candidate starting location. By successive readings on the compass, the robot can keep track of its coordinates with respect to the starting location. We assume that robot's starting location is at the origin $(0,0)$ of the cartesian plane and it moves along the edges of the integer lattice $\mathcal{L} = \{(a,b)|a,b \in \mathbb{Z}\}$. A move to the north, south, east and west on grid graph G amounts to adding $(0,1),(0,-1),(1,0)$ and $(-1,0)$ to its current coordinate $(x,y) \in \mathcal{L}$. In other words, assume the robot started from hypothesis $h \in H$. Then when it is at coordinate $(x,y) \in \mathcal{L}$, it will be occupying the cell x units east and y units north (negative values of x,y denote west and south respectively) of h in grid G. We say that the robot is *at coordinate* (x,y) relative to h.

Suppose the robot makes an observation when at coordinate $p \in \mathcal{L}$. The outcome depends on its starting location $h \in H$. If the robot started from hypothesis h, the observation will be the same as that by a robot located at coordinate p relative to h. We denote this observation by $\mathcal{O}(h,p)$ and call it the *opinion* of h about coordinate p. If the cell at coordinate p to h is blocked, we set $\mathcal{O}(h,p) = \phi$. The *hypothesis partition* $\mathcal{H}(p)$ is a partition of the set of hypotheses according to the following equivalence relation: $h_1 \sim h_2$ iff $\mathcal{O}(h_1,p) = \mathcal{O}(h_2,p)$. $Maj(p)$ denotes the largest size class of $\mathcal{H}(p)$. The 'majority opinion' at p is the opinion common to the plurality of hypotheses $h \in Maj(p)$. Note it may occur that $|Maj(p)| < \frac{1}{2}|H|$. The lemmas that follow are valid in this case because the robot immediately half-localizes. For an observation $o \in \{b,t\}^4 \bigcup\{\phi\}$ (as there are two choices, *b*locked or *t*raversable, for each of the four neighbouring cells), $G(p,o)$ denotes the class of $\mathcal{H}(p)$ with opinion o at p.

2.2 Halving Paths Let $\mathcal{L}_H$, the majority-rule map, denote the subgraph of the integer lattice formed by coordinates p for which $|G(p,\phi)| \leq \frac{1}{2}|H|$. A *halving path* is a (possibly self-intersecting) path $(p_0,p_1,p_2,\ldots,p_m) \in \mathcal{L}_H$ such that (i) the starting coordinate p_0 is the origin, and (ii) $|\bigcap_{i=0}^{m} Maj(p_i)| \leq \frac{1}{2}|H|$.

LEMMA 2.1. *Let $\mathcal{C}$ be a halving path. There exists a strategy $S(\mathcal{C})$ for half-localizing the robot with travel cost at most $|\mathcal{C}|$.*

Proof. Let $\mathcal{C} = (p_0 = \mathbf{0}, p_1, p_2, \ldots, p_m)$ where p_{i+1} is a neighbour of p_i in lattice $\mathcal{L}_H$. We show that strategy $S(\mathcal{C})$ (see Algorithm 1) half-localizes the robot. After observation o_i, the robot keeps only those hypotheses whose opinion at p_i is o_i. Thus, it updates H' (the set of active hypotheses) correctly. We show that $S(\mathcal{C})$ reduces the set of hypotheses by half. If the robot finds that the cell at coordinate p_i is blocked, it localizes to a set of size at most $|G(p_i,\phi)| \leq \frac{1}{2}|H|$ (since $p_i \in \mathcal{L}_H$). If observation o_i is different from the majority opinion at p_i, $H' \subset G(p_i,o_i)$, which has size at most $\frac{1}{2}|H|$. Thus the robot reaches p_m iff for each $p_i, 0 \leq i \leq m-1$, o_i is the majority opinion at p_i. In this case, $H' = \bigcap_{i=0}^{m} Maj(p_i)$ which is again less that $\frac{1}{2}|H|$ (since $\mathcal{C}$ is a halving path).

Algorithm 1: Strategy $S(\mathcal{C})$

Data : Grid graph G, set of hypotheses H and a halving path $(p_0 = \mathbf{0}, p_1, \ldots, p_m) \in \mathcal{L}_H$.
Result : The robot half-localizes in at most m steps.
Initialize $H' = H$;
for $i = 0$ *to* $m-1$ **do**
  begin
    Make observation o_i at coordinate p_i;
    Update $H' = H' \bigcap G(p_i,o_i)$. Stop if $|H'| \leq \frac{1}{2}|H|$;
    Move to coordinate p_{i+1};
  end
end
Make observation o_m at p_m; Update $H' = H' \bigcap G(p_m,o_m)$. Stop.;

LEMMA 2.2. *Let S be a strategy for half-localization. There exists a halving path $\mathcal{C}(S)$ of length at most $W(S)$.*

Proof. Imagine a robot guided by $\mathcal{S}$ which stops as soon as it half-localizes. Let $\mathcal{C}(S) = (p_0 = \mathbf{0}, p_1, p_2, \ldots, p_m)$ be the maximum length path traced by the robot in lattice $\mathcal{L}$ for any (initial) position in H. Let H_i denote the set of active hypotheses just after the robot makes an observation at coordinate p_i. For $0 \leq i < m$, $|H_i| > \frac{1}{2}|H|$ since otherwise the robot would have stopped at p_i itself. Each coordinate p_{i+1} is unblocked for at least $|H_i| \geq \frac{1}{2}|H|$ hypotheses and hence $\mathcal{C} \in \mathcal{L}_H$.

Finally we show that $I = \bigcap_{i=0}^{m} Maj(p_i)$ is of size at most $\frac{1}{2}|H|$. For this assume a robot initially located at some $h \in I$. Guided by S, the robot will follow path $\mathcal{C}(S)$ and observe the majority opinion o_i at all p_i (since $I \subset Maj(p_i)$). But then $|I| = |\bigcap_{i=0}^{m} G(p_i, o_i)| = |H_m| \le \frac{1}{2}|H|$ and hence $\mathcal{C}(S)$ satisfies the lemma.

2.3 Computing halving paths

Let $\mathcal{C}_H^*$ denote the optimal halving path in $\mathcal{L}_H$. We solve the problem of approximating the optimal halving path by reducing it to an instance $\mathcal{I}_H$ of the $\frac{1}{2}$-Group Steiner problem.

$\frac{1}{2}$-Group Steiner Problem. In the Group Steiner problem, the input consists of a graph $G = (V, E)$ with edge weights $w : E \to \mathbb{R}^+$ and k subsets of vertices (called *groups*) $g_1, g_2, \ldots, g_k \subset V$. A tree T in G covers group g_i if $T \bigcap g_i$ is not empty. The goal is to find a minimum weight tree that covers all the groups. In the $\frac{1}{2}$-Group Steiner problem, the goal is to find a min-cost tree that covers at least half the groups. In the *rooted* versions of these problems, tree T is also required to contain a root vertex $r \in V$.

The reduction is essentially a restatement of the problem in terms of groups:

INSTANCE $\mathcal{I}_H$: Take G as lattice $\mathcal{L}_H$. All edges have unit weight. Take origin as the root vertex. Let $H = \{h_1, h_2, \ldots, h_k\}$ be the set of hypotheses. Make k groups, one for each hypothesis $h_i \in H$. Group g_i is the set of all lattice points $p \in \mathcal{L}_H$ such that h_i does not share the majority opinion at p i.e., $h_i \notin Maj(p)$.

Thus tree $T \subset \mathcal{L}_H$ covers k' groups iff $\bigcap_{x \in T} Maj(x)$ has size $k - k'$. In particular, T covers at least half the groups iff $|\bigcap_{x \in T} Maj(x)| \le \frac{1}{2}|H|$.

LEMMA 2.3. *There exists a $O(\log^2 n)$ factor approximation algorithm for the optimal halving path.*

Proof. Assemble a $O(\log^2 n)$-approximate algorithm for $\frac{1}{2}$-Group Steiner from the following components. An $O(\log^2 n)$ algorithm for Group Steiner was given by Garg *et al* [9] for the case when G is a tree. They first solve a linear programming relaxation to get a fractional solution and then use an innovative randomized rounding scheme. A modification of their algorithm can be used to solve $\frac{1}{2}$-Group Steiner on trees within $O(\log n)$ factor [7]. For general graphs, first the graph is approximated by a tree using the tree-metric technique of [8] (which is a recent improvement to Bartal [2]) and then an $O(\log^2 n)$ approximation is found using the algorithm of [9]. The approximation by tree costs $O(\log n)$ factor in approximation, thus giving a $O(\log^2 n)$ algorithm for $\frac{1}{2}$-Group Steiner.

Let T be the tree output by the algorithm on $\mathcal{I}_H$. Then weight of T is at most $O(\log^2 n) \cdot w(T^*)$, where T^* is the optimal $\frac{1}{2}$-Group Steiner tree. Let $\mathcal{C}$ be the path of length $2 \cdot w(T)$ traced by a depth-first search on T starting from the origin. $\mathcal{C}$ is a halving path since $|\bigcap_{x \in \mathcal{C}} Maj(x)| = |\bigcap_{x \in T} Maj(x)| \le \frac{1}{2}|H|$. Since the optimal halving path $\mathcal{C}_H^*$ covers half the groups, $w(T^*) \le |\mathcal{C}_H^*|$. Therefore $|\mathcal{C}| \le O(\log^2 n) \cdot |\mathcal{C}_H^*|$.

Algorithm 2: Strategy **RHL** - **R**epeated **H**alf **L**ocalization

```
Data   : Grid graph G, the set of hypotheses H
Result : The robot localizes to its initial position
         h ∈ H
while |H| > 1 do
    begin
        Compute a halving path C (lemma 2.3);
        Half-localize by strategy S(C) (lemma 2.2);
        Move back to the starting location;
    end
end
```

2.4 Putting everything together

THEOREM 2.1. *A robot guided by strategy* **RHL***(Algorithm 2) correctly determines its initial position $h \in H$ by traveling at most $O(\log^2 n \log k) OPT(G, H)$ distance where $k = |H|$ and n is the size of G. Further, the computation time of the robot is polynomial in n.*

Proof. Since the number of hypotheses reduces by at least half after each phase, the robot localizes in $m \le \lceil \log |H| \rceil = \lceil \log k \rceil$ phases. Let H_i denote the set of active hypotheses at the start of the ith phase. By lemma 2.3, the distance traveled by the robot in ith phase is at most $O(\log^2 n) \cdot |\mathcal{C}_{H_i}^*|$. By lemma 2.2, $|\mathcal{C}_{H_i}^*| \le$ HALF-OPT(G,H_i) $\le$ OPT(G,H), where the last inequality follows from the fact that any localization plan also reduces the set of hypothesis by half. Therefore, the distance traveled by the robot in each phase is at most $O(\log^2 n) \cdot$ OPT(G,H). Since there are $O(\log k)$ phases, the total worst-case travel distance is $O(\log^2 n \log k) \cdot$OPT(G,H). Since instance $\mathcal{I}_H$ can be constructed in $O(nk)$ time, the computation time is at most $O(T(nk) \cdot \log n)$ where $T()$ (a polynomial) is the time taken by the approximation algorithm for $\frac{1}{2}$-Group Steiner.

3 Lower Bound

Preliminaries. A tree is said to be of *arity d* if every non-leaf vertex has d children. A rooted tree has *height* H if all its leaves are at distance H from the root. As usual, the *level of a vertex* is its distance from the root; the root itself is at level 0 and there are $H + 1$ levels.

Definition: [2] A *hierarchially well-separated tree* (HST) is defined as a rooted, weighted tree in which (i) all leaves are at the same distance from the root; and (ii) the weight of every edge is exactly $\frac{1}{\tau}$ times the weight of its parent edge, where $\tau \ge 1$ is any desired constant.

To prove the lower bound, we use the recent result of Halperin *et al* [12] which establishes $\Omega(\log^{2-\epsilon} n)$ hardness for Group Steiner problem on HSTs. The next theorem, extracted from their proof, states their result in a detailed form suited to our purpose:

THEOREM 3.1. *[12] Let L be any NP-complete language. Then there exist constant c_0 and an algorithm $\mathcal{A}$ that, given an instance $\mathcal{I}$ and a sufficiently large constant α, produces in expected running time $O(|\mathcal{I}|^{polylog(|\mathcal{I}|)})$ an instance $\mathcal{I}' = (T, r, \mathcal{G})$ (r is also the root of T) of Group Steiner problem such that:*

1. *For some $m \leq |\mathcal{I}|^{c_0}$, T is a HST with height $H = (\log m)^\alpha$, arity $d = m^{O(\log m)}$ and $\tau = m^{\log m}$. Further, each group $g \in \mathcal{G}$ is a subset of the leaves of T and there are $k = m^{O((\log m)^{\alpha+1})}$ groups.*
2. *If $\mathcal{I} \in L$, then there is a (rooted) tree $T' \subseteq T$ of weight $(\log m)^\alpha$ covering all the groups.*
3. *If $\mathcal{I} \notin L$, then every tree $T' \subseteq T$ covering all the groups has weight $\Omega((\log m)^{3\alpha+2})$.*

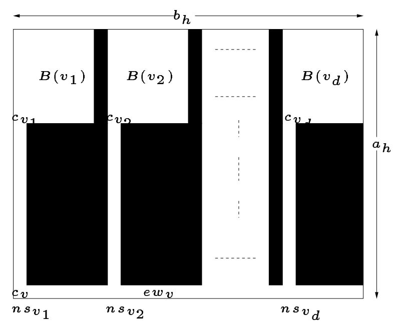
THEOREM 3.2. *Let L be any NP-complete language. Then there exist constant c_0 and an algorithm $\mathcal{A}'$ that, given an instance $\mathcal{I}$ and a sufficiently large constant α, produces in expected running time $O(|\mathcal{I}|^{polylog(|\mathcal{I}|)})$ an instance $\mathcal{I}'' = (G, H)$ of the robot localization problem on gridgraphs such that*

1. *For some $m \leq |\mathcal{I}|^{c_0}$, G has $N = m^{O((\log m)^{\alpha+1})}$ cells and H has $m^{O((\log m)^{\alpha+1})}$ hypotheses.*
2. *For some $\beta = m^{O((\log m)^{\alpha+1})}$:*
   - *(a) If $\mathcal{I} \in L$, then there exists a localization plan with worst-case cost $O(\beta \cdot (\log m)^\alpha)$.*
   - *(b) If $\mathcal{I} \notin L$, every localization plan has cost $\Omega(\beta \cdot (\log m)^{3\alpha+2})$.*

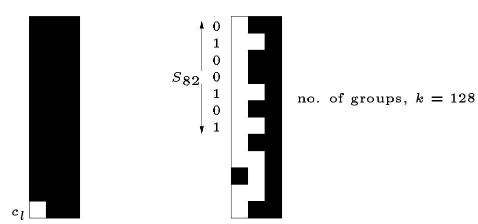
Proof. Let $\mathcal{I}' = (T(V, E), r, \mathcal{G})$ be the instance of Group Steiner on HSTs obtained by running algorithm $\mathcal{A}$ on $\mathcal{I}$ (see Theorem 3.1 above). Let d, H and τ denote the arity, height and weight factor of HST T, and k denote the number of groups in $\mathcal{G}$. G consists of $k+1$ (disjoint) copies $B_0, B_1, \ldots, B_k$ of grid graph B, where B is an 'embedding' of HST T respecting the weights on its edges.

The embedding B is best described inductively. Let $B(v)$ denote the embedding of the subtree rooted at vertex $v \in T$. Cell c_v at the southwest corner of each $B(v)$ corresponds to vertex v. For a leaf l, $B(l)$ is a $3 \times (\lceil \log k \rceil + 5)$ rectangle with a single traversable cell c_l at its southwest corner (Figure 2(b)). The reason for adding blocked space to c_l will be clear later, when we use it to add "signatures" to leaf l. For a non-leaf vertex v, $B(v)$ is formed by combining the embeddings of the subtrees rooted at its d children $v_1, v_2, \ldots, v_d$ (see Figure 2(a)). $B(v_1), B(v_2), \ldots, B(v_d)$ are positioned along the top edge of $B(v)$ separated by north-south walls of width 1. There is an east-west corridor ew_v running along the bottom edge of $B(v)$. Cell c_{v_i} is connected to this corridor by a north-south corridor ns_{v_i} which corresponds to edge $vv_i \in T$. We make the length of ns_{v_i} proportional to the weight of vv_i: if v is at level h, $|ns_{v_i}| = \beta \cdot \frac{1}{\tau^h}$, where β is a *scaling factor* to be chosen later. Finally $B = B(r)$ where r is the root of T.

Let a_h, b_h be the length and breadth of the grid required to embed the subtree rooted at a level h vertex $v \in T$. To see that the tree 'fits', observe that $B(v)$ fits in an $a_h \times b_h$ rectangle where $a_h = d \cdot a_{h+1} + (d-1)$ and $b_h = b_{h+1} + \frac{\beta}{\tau^h}$. Hence $b_h = (\lceil \log k \rceil + 5) + \beta \cdot \sum_{\alpha=h}^{H-1} \frac{1}{\tau^\alpha}$, and by induction one can show that $a_h = 4 \cdot d^{H-h} - 1$.



(a) Block $B(v)$



(a) Leaf block $B(l)$ (b) Adding signature S_{82} to block $B(l)$

(b) A leaf block with signature

Figure 2:

Let w_{xy} denote the weight of path connecting $x, y \in T$. Let P_{uv} be the unique path connecting cells c_u and c_v in B. We show that choosing $\beta = 5d^H \cdot \tau^H$ makes B an embedding of T in the following sense: for all vertices $x, y \in T$, $\beta \cdot w_{xy} \leq |P_{xy}| \leq 2\beta \cdot w_{xy}$. First observe that the length of any north-south corridor ns_v is now at least $5d^H$ while any east-west corridor is less than $4d^H$. Therefore $|ew_x| \leq |ns_v|$ for all $u, v \in T$. We charge the distances traveled along east-west corridors

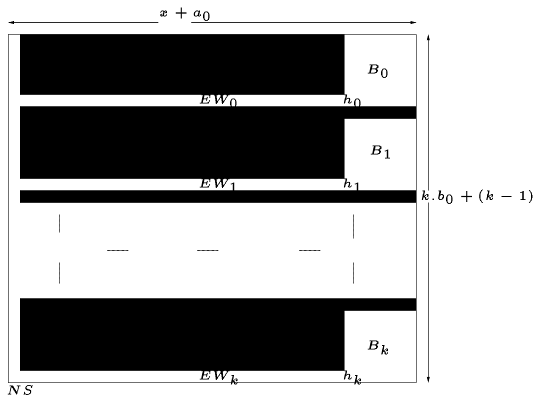


Figure 3: Gridworld G

to the north-south corridors immediately preceding it. First assume x is the parent of y. Then P_{xy} consists of the north-south hallway ns_y along with portion of ew_x connecting c_x to ns_y. Clearly, $\beta \cdot w_{xy} = |ns_y| \leq |P_{xy}| \leq |ns_y| + |ew_x| \leq 2\beta \cdot w_{xy}$. Next consider the case when x, y are siblings with common parent z. P_{xy} consists of north-south corridors ns_x, ns_y along with the portion of ew_z connecting them. Hence, $\beta \cdot w_{xy} = \beta \cdot (w_{xz} + w_{zy}) = |ns_x| + |ns_y| \leq |P_{xy}| \leq |ns_x| + |ns_y| + |ew_z| \leq 2\beta \cdot w_{xz} + \beta \cdot w_{zy} \leq 2\beta \cdot w_{xy}$. For general x, y, let $c_{z_0=x}, c_{z_1}, \ldots, c_{z_m=y}$ be be the cells corresponding to vertices of T, in the order they occur along path P_{uv}. By the construction of B, we know that for each i either (i) z_{i+1} is a parent of z_i or vice-versa, or (ii) z_i, z_{i+1} are siblings. Therefore $\beta \cdot w_{z_i z_{i+1}} \leq |P_{z_i z_{i+1}}| \leq 2\beta \cdot w_{z_i z_{i+1}}$. Since $|P_{xy}| = \sum |P_{z_i z_{i+1}}|$, the length of P_{uv} is within factor 2 of $\beta \cdot \sum w_{z_i z_{i+1}} = \beta \cdot w_{xy}$.

Let $g_1, g_2, \ldots, g_k$ be the k groups in $\mathcal{G}$. We make $k+1$ copies $B_0, B_1, \ldots, B_k$ of embedding B. B_i's are the same except for distinguishing 'signatures' at some leaf blocks. $B_0 = B$ is the *dummy* copy and contains no signatures. For $i > 0$, B_i is formed by adding signature s_i (a binary encoding of i) to every leaf block $B(l)$ of B such that $l \in g_i$ (Figure 2(b)). To add s_i, first cell c_l is extended to a north-south corridor along the left edge of $B(l)$. Then a set of $\log k$ eastern "alcoves" encoding i in binary are placed along the eastern edge: the jth alcove from the top is blocked iff the jth bit in the binary form is 0. A robot located at c_l can read the value of i by going north and sensing the alcoves to its right for blockage.

Let $x = 2 \cdot a_0 \cdot b_0$. Grid graph G is a $(x + a_0) \times ((k+1) \cdot b_0 + k - 1)$ rectangle formed by connecting group blocks $\{B_i\}_i$ as shown in Figure 3. $B_0, B_1, B_2, \ldots B_k$ are placed along the right edge of G separated by east-west walls of width 1. A north-south corridor NS of width 1 runs alongside the left edge of G. The south-west cell of each block B_i is connected to this corridor by an east-west corridor EW_i of length x. The set of hypotheses H equals $\{h_0, h_1, \ldots, h_k\}$ where h_i denotes the cell at the south-west corner of block B_i. Substituting values of k, H, d, τ as given by Theorem 3.1, we get $\beta = 5d^H \cdot \tau^H = m^{O((\log m)^{\alpha+1})}$, $|G| = O(a_0 b_0^2 k) = m^{O((\log m)^{\alpha+1})}$ and $|H| = k = m^{O((\log m)^{\alpha+1})}$, where $m \leq |\mathcal{I}|^{c_0}$. We complete the proof by showing that the optimal localization plans for $\mathcal{I}'' = (G, H)$ in the 'yes' ($\mathcal{I} \in L$) and 'no' ($\mathcal{I} \notin L$) cases differ by a factor of $\Omega((\log m)^{2\alpha+2})$.

'Yes' case: Suppose $\mathcal{I} \in L$. By theorem 3.1, there exists a tree $T' \subseteq T$ of weight $(\log m)^\alpha$, which covers all groups in $\mathcal{G}$. As all groups $g \in G$ consist of leaves of T, w.l.o.g. every root to leaf path in T' ends at a leaf of T. Let $l_0, l_1, \ldots, l_{t-1}$ be the leaves of T' in the order they are visited by a depth-first search from the root. Consider the following plan: read the signatures at leaf blocks $B(l_0), B(l_1), \ldots, B(l_{t-1})$ in that order. As soon as a non-zero signature $s_{i_0}, i_0 > 0$ is read, localize to h_{i_0}. Otherwise, localize to h_0.

To prove correctness, assume the robot was placed (without its knowledge) at hypothesis h_{i_0}. If $i_0 = 0$, the robot will read zero signatures at all leaf blocks and correctly localize to h_0. Suppose $i_0 > 0$. Since T' covers all groups, group g_{i_0} contains at least one leaf vertex from T'. The robot will read signature s_{i_0} at the first such vertex in the sequence $l_0, l_1, \ldots, l_{t-1}$ and localize to h_{i_0}.

The total travel cost of the robot is $|P_{rl_0}| + \sum_{i=0}^{t-2} |P_{l_i l_{i+1}}| \leq 2\beta \cdot (w_{rl_0} + \sum_{i=0}^{t-2} w_{l_i l_{i+1}}) \leq 2\beta \cdot w(T') = O(\beta \cdot (\log m)^\alpha)$. We neglect the cost of reading signatures at l_i, as it is $O(t \cdot \log k) = O(d^H \log k) \leq \beta$.

'No' case: Suppose $\mathcal{I} \notin L$. Assume that we have found a localization plan with cost $o(C \cdot (\log m)^{3\alpha+2})$. The number of movements for the plan is no larger than the length of an east-west hallway EW_i. Now assume the robot starts at cell h_0. Thus, it cannot visit a different east-west hallway and, as part of the localization, must determine that no leaf block in its group block has a non-zero signature. Let $B(l_0), B(l_1), \ldots, B(l_{t-1})$ be all the leaf blocks, in the order they are visited by the robot. The collection of groups that these leaves cover must equal $\mathcal{G}$, for otherwise the robot could not distinguish between hypotheses h_0 and h_i for the groups g_i not covered by them.

Let T' be the Group Steiner tree formed by taking the union of paths connecting r to l_0 and l_i to l_{i+1} for $0 \leq i \leq t-2$. By Theorem 3.1, weight of T is $\Omega((\log m)^{3\alpha+2})$. Therefore, the cost of the localization plan is at least $|P_{rl_0}| + \sum_0^{t-2} |P_{l_i l_{i+1}}| \geq \beta \cdot (w_{rl_0} + \sum_{i=0}^{t-2} w_{l_i l_{i+1}}) \geq \beta \cdot w(T') = \Omega(\beta \cdot (\log m)^{3\alpha+2})$.

COROLLARY 3.1. *For every fixed $\epsilon > 0$, the robot localization problem cannot be approximated within ratio $\log^{2-\epsilon} N$ on gridworlds of size N, unless $NP \subseteq ZTIME(n^{polylog(n)})$.*

Proof. Apply the algorithm in Theorem 3.2 with $\alpha = 2 \cdot (\frac{1}{\epsilon} - 1)$. The logarithm of the size of grid graph G is $\log N = O((\log m)^{\alpha+2})$, where $m \leq n^{c_0}$. The optimum localization plans in the 'yes' and 'no' case differ by a factor of $\Omega((\log m)^{2\alpha+2}) = \Omega((\log N)^{2-\epsilon})$.

COROLLARY 3.2. *For every fixed $\epsilon > 0$, the robot localization problem cannot be approximated within ratio $\log^{2-\epsilon} N$ on polygons with N vertices, unless $NP \subseteq ZTIME(n^{polylog(n)})$.*

Proof. The grid graph G in Theorem 3.2 above can be viewed as a polygon P with at most N vertices. Let h'_i denote the center of the cell h_i in G. Consider the localization problem on P with hypotheses set $H' = \{h'_0, h'_1, \ldots, h'_k\}$. The optimal localization plan in the 'yes' case has cost $O(\beta \cdot (\log m)^{\alpha})$, as a robot with a range finder can only do better. However when $\mathcal{I} \notin L$, a robot with a range finder may read the signatures from a distance, and localize at lesser cost. To rule this out, put small 'twists' in polygon P just before every signature. Thus the robot cannot read the signatures at a distance, and therefore will travel at least $\Omega(\beta \cdot (\log m)^{3\alpha+2})$ distance as in Theorem 3.1 above. The 'yes' and 'no' cases differ by $\Omega((\log m)^{\alpha+2})$ and the bound follows by choosing $\alpha = 2 \cdot (1 - \frac{1}{\epsilon})$.

4 Extension to polygons

In this section, we extend our algorithm to polygons. The outline of the algorithm is the same: the robot works in phases; in each phase reducing the set of hypotheses by half. However since the robot moves continuously, the set of possible coordinates $\mathcal{M}$ is the whole of euclidean plane $\mathbb{R}^2$. As before, let opinion $\mathcal{O}(h,p)$ be the visibility polygon observed by a robot at coordinate p if it started from hypothesis h. If the point at coordinate p lies outside $\mathcal{P}$, take $\mathcal{O}(h,p) = \phi$. The hypothesis partition $\mathcal{H}(p)$ partitions hypotheses in H according to their opinions at p. Let $\mathcal{M}_H$ denote the set of coordinates with $|G(p,\phi)| \leq \frac{1}{2}|H|$.

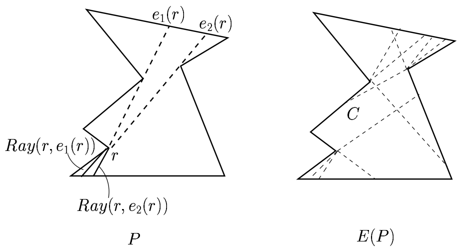
Redefine a *halving path* $\mathcal{C}$ as a 'curve' in the plane with one endpoint at the origin such that $|\bigcap_{x \in \mathcal{C}} Maj(x)| \leq \frac{1}{2}|H|$. Here x goes over all the (infinite) coordinates in $\mathcal{C}$. It is straightforward to extend lemmas 2.1 and 2.2 to the case of polygons with this new definition. However since $\mathcal{M}_H$ contains an infinite number of points, one cannot compute an approximation to the optimal halving path $\mathcal{C}^*_H$ by reducing it to $\frac{1}{2}$-Group Steiner on a finite number of coordinates (as in lemma 2.3). Instead we 'discretize' the problem to a finite (and polynomial size) set of coordinates Q_H such that the optimal halving path on Q_H has length within constant factor of the optimal halving path in $\mathcal{M}_H$.

To do so, we first compute what we call the *hypothesis equivalence decomposition* $D(P,H)$. $D(P,H)$ is a partition of coordinate plane $\mathcal{M}$ into polygonal cells such that the hypothesis partitions are the same for any two points in the same cell. This new construct is more complex than what was needed previously (the *overlay arrangement* of Dudek *et al* [6]). Thus we can define $\mathcal{H}(C)$ and $Maj(C)$, where C is a cell of $D(P,H)$. $\mathcal{M}_H$ is the union of all cells $C \in D(P,H)$ such that $|G(C,\phi)| \leq \frac{1}{2}|H|$. This partitions $\mathcal{M}_H$ into finite number of cells. Further the robot needs to make an observation only when it enters an unvisited cell of $D(P,H)$. In particular, it is enough to consider piecewise linear paths with break points on segments in $D(P,H)$.

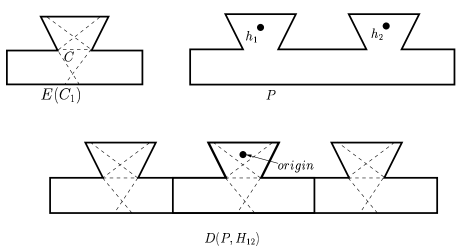
Let L denote the set of line segments comprising $D(P,H)$, and V denote the end points of segments in L. For point $v \in V$ (we call them vertices) and line segment $l \in L$, let $\pi(v,l)$ denote the point closest to v on l. The set Q_H is define as $\{\pi(v,l) | v \in V, l \in L\}$. In section 4.2, we show that shortest halving path having breakpoints in $Q_H \subset L$ has length at most $5|\mathcal{C}^*_H|$. Finally, an $O(\log^2 n)$-approximate halving path is obtained by solving an instance of $\frac{1}{2}$-Group Steiner problem on Q_H (as in lemma 2.3).

4.1 Hypothesis Equivalence Decomposition Let P be the map polygon and $H = \{h_1, h_2, \ldots, h_k\}$ the set of hypotheses. First we briefly review the *aspect graph dual*, which will be used as a building block in the hypothesis equivalence decomposition. The idea is to partition the plane into regions from which the set of visible edges is the same. An edge e is *visible* from a point $p \in P$ if at least one interior point (i.e., a point except the end points) of e is visible from p. For a point $p \in P$, the *edge skeleton* $E^*(p)$ is the subset of edges of P visible from p. The aspect graph dual $E(P)$ is a partition of P into polygonal cells such that the edge skeletons are the same for any two points in the same cell. $E(P)$ is a standard construct in computational geometry (see [10] and references therein), and can be constructed in polynomial time. For sake of completeness, we describe its construction.

See Figure 4(a). Let $e(p)$ denote the line segment containing the subset of points on e visible from p. Denote the (possibly) two end points of $e(p)$ by $e_1(p)$ and $e_2(p)$ respectively. For points $p, q \in P$, $Ray(p,q)$ denotes a ray starting from p which is collinear with line segment pq and is directed away from p. For each reflex vertex r and each edge e visible from r, introduce two rays $Ray(r, e_1(r))$ and $Ray(r, e_2(r))$ in the interior of P. Assume that these rays are line segments by restricting them to the interior of P. Write $E(P)$ for the partition of P formed by these line segments. By a *cell* $C \in E(P)$ we mean a maximally connected region containing no line segments. We state without proof that $E^*(p) = E^*(q)$ for every two points p and q contained in a cell $C \in E(P)$. Clearly, $E(P)$ can be computed in time polynomial in n and has cardinality $O(n^4)$.



(a) Dual of Aspect Graph



(b) The subpartition $D(P, H_{12})$

Figure 4:

Let P_i denote a copy of the map polygon P in the coordinate plane such that hypothesis h_i coincides with the origin $\mathbf{0}$. For each pair of distinct hypotheses $H_{ij} = \{h_i, h_j\}$ construct the hypothesis equivalence decomposition $D(P, H_{ij})$ (see Fig 4(b)) as follows: first superimpose P_i and P_j to get a preliminary partition $P_i \bigcup P_j$ of the coordinate plane. $D(P, H_{ij})$ is then formed by taking the duals of aspect graphs $E(C)$ for each cell (a maximally connected region containing no line segments) $C \in P_i \bigcup P_j$. (Note that we do not partition the region formed by the intersection of the exteriors of P_i and P_j.) Finally, the hypothesis equivalence decomposition $D(P, H)$ for H is constructed by taking the union of all line segments in the partitions $\{D(P, H_{ij}) | 1 \leq i < j \leq k\}$.

LEMMA 4.1. *Let C be a cell of $D(P, H_{ij})$. Then exactly one of the following holds: (i) $\mathcal{O}(h_i, p) = \mathcal{O}(h_j, p)$ for all coordinates $p \in C$ or, (ii) $\mathcal{O}(h_i, p) \neq \mathcal{O}(h_j, p)$ for all coordinates $p \in C$.*

Proof. If C is the intersection of exteriors of P_i and P_j, then $\mathcal{O}(h_i, p) = \mathcal{O}(h_j, p) = \phi$ and we are done. If not, let C_1 be the cell of $P_i \bigcup P_j$ containing C (see Fig. 4(b)). Let $S \subset B(C_1)$ be the subset of edges of C_1 visible from every point in C (recall that C is a cell of aspect graph dual $E(C_1)$). If an edge $e \in S$ belongs to just P_i or P_j, then $\mathcal{O}(h_i, p) \neq \mathcal{O}(h_j, p)$ for all coordinates $p \in C$. Otherwise assume that all edges in S belong to both P_i and P_j. Then a robot at coordinate p will see the same visibility polygon regardless of whether it was initially at h_i or h_j, and hence $\mathcal{O}(h_i, p) \neq \mathcal{O}(h_j, p)$ for all coordinates $p \in C$.

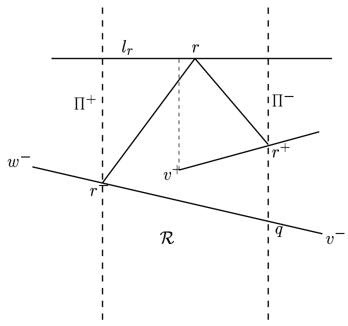
LEMMA 4.2. *$D(P, H)$ satisfies the following:(i) The cells of $D(P, H)$ are polygonal regions, and(ii) $\mathcal{H}(p) = \mathcal{H}(q)$ for every two coordinates p, q contained in a cell C of $D(P, H)$.*

Proof. Every cell $C \in D(P, H)$ is equal to the intersection of cells $C_{ij} \in D(P, H_{ij})$ containing it. Since C_{ij}'s are polygons, C will be a polygon itself. For the second part, assume two points p, q contained in C such that $\mathcal{H}(p) \neq \mathcal{H}(q)$. Choose a pair of hypotheses (h_m, h_n) such that they belong to the same class in $\mathcal{H}(p)$ but not in $\mathcal{H}(q)$. Then $p, q \in C_{mn} \in D(P, H_{ij})$ are two points such that $\mathcal{O}(h_m, p) = \mathcal{O}(h_n, p)$ but $\mathcal{O}(h_m, q) \neq \mathcal{O}(h_n, q)$, thereby contradicting lemma 4.1 above.

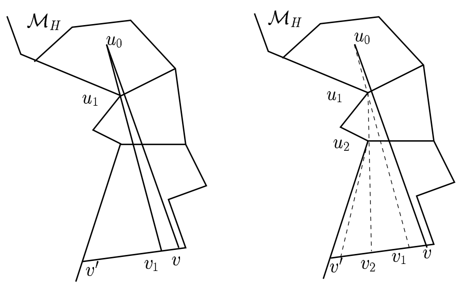
Since $|P_i \bigcup P_j| = O(n^2)$, we have $|D(P, H_{ij})| = O(n^4)$. Hence $D(P, H)$ is formed by at most $O(k^2 n^4)$ lines and is of total cardinality $O(k^4 n^8)$. Thus we can compute $D(P, H)$ in polynomial time by the construction above. Since $Q_H = |V| \cdot |L|$, the set of coordinates Q_H is also of size polynomial in n.

4.2 The Set of Coordinates Q_H For each edge e of $D(P, H)$, the first time the robot hits e (at an endpoint or interior) it can visit the interior of each adjacent cell at arbitrarily small cost. New information is acquired only at these limited times. Between two such points $p_1, p_2 \in L$ the robot can follow the shortest path joining them in $\mathcal{M}_H$, which has break points only at vertices $v \in V$ forming the boundary of $\mathcal{M}_H$.Thus by ignoring an arbitrarily small positive cost, robot movement can be assumed to be piecewise linear with breakpoints at edges of $D(P, H)$. We redefine halving path in the new framework: a curve $\mathcal{C}$ is said to *cover* a cell $C \in D(P, H)$ if $\mathcal{C}$ intersects C or *its boundary.* The cover of $\mathcal{C}$, $Cover(\mathcal{C})$, is the set of all cells $C \in D(P, H)$ covered by $\mathcal{C}$. Thus a *halving path* is a curve $\mathcal{C}$ with $|\bigcap_{C \in Cover(\mathcal{C})} Maj(C)| \leq \frac{1}{2}|H|$.

We next show that there exists a halving path of length $5|\mathcal{C}^*_H|$ with break points in Q_H. Call a break point of $\mathcal{C}^*_H$ a *reflection point* if it lies in the interior of a line segment $l \in L$. Since all other break points are vertices and hence belong to Q_H, our aim will be to 'shift' the reflection points to Q_H. The proof consists of two steps: first we show that for each reflection point r one can find a point $r' \in Q_H$ not far from it (lemma 4.3). Then we describe an operation **Shift**$(\mathcal{C},v,v')$ that shifts the endpoint of a given curve $\mathcal{C}$ from v to v', maintains the set of cells covered by it and introduces no new reflection points; while increasing its length by at most $|vv'|$. The result is proved by 'shifting' each subpath of $\mathcal{C}^*_H$ between two reflection points r_1, r_2 to its new endpoints $r'_1, r'_2 \in Q_H$, which are found by lemma 4.3.



(a) Proof of lemma 4.3



(a) (b)

(b) Operation **Shift**$(\mathcal{C},\ v,\ v')$

Figure 5:

LEMMA 4.3. *Let r be a reflection point of $\mathcal{C}^*_H$ and let r^- and r^+ be break points preceding and succeeding it. Let $l_r \in L$ be the line segment containing it. Then there exists a point $r' \in l_r \bigcap Q_H$ such that $|rr'| \le |r^-r| + |rr^+|$.*

Proof. Let Π^-, Π^+ be lines perpendicular to l_r passing through r^-, r^+ respectively. Let $\mathcal{R}$ be the region of the plane between Π^-, Π^+. Suppose there exists a vertex $v \in V$ in $\mathcal{R}$. Then $r' = \pi(v, l_r) \in Q_H$ satisfies the lemma. Assume, for the sake of contradiction, that there exists no vertex in $\mathcal{R}$. Then r^-, r^+ are reflection points. Let w^+v^+ and w^-v^- be line segments in L containing r^+, r^- respectively. Since $\mathcal{R}$ contains no vertices, both w^-, v^- lie outside $\mathcal{R}$ (see figure 5(a)). W.l.o.g. assume w^- lies to the left and v^- to the right of $\mathcal{R}$. Let q be the point where w^-v^- intersects Π^-. Then at least one endpoint v^+ of the anchor at r^+ must lie inside the quadrilateral rr^-qr^+. We can take r' to be the projection of that endpoint on l_r (a contradiction).

Let $\mathcal{C}$ be a piecewise linear curve with endpoints u, v. Suppose v lies in the interior of segment $l \in L$. Let v' be another point on l. Figure 5(b) illustrates how **Shift**$(\mathcal{C},\ v,\ v')$ works. Let u_0 be the break point preceding v in $\mathcal{C}$. We fix u_0 and slide endpoint v of line segment u_0v on l till it reaches v'. If we hit the vertex u_1 of a cell covered by $\mathcal{C}$, we make u_1 a 'new' break point and repeat sliding on u_1v_1. This is repeated to give new break points $u_1, u_2, \ldots$, till v reaches v'. Since $u_i \in V$, we introduce no new reflection points. As new break points $\{u_i\}_i$ are made every time the curve is about to exit a cell in its cover, $Cover(\mathcal{C}) \subseteq Cover(\mathcal{C}')$. Finally by the triangle inequality, $|\mathcal{C}'| \le |\mathcal{C}| + |vv'|$.

LEMMA 4.4. *There exists a piecewise linear halving curve of length at most $5|\mathcal{C}^*_H|$ with break points in Q_H.*

Proof. Let $r_1, r_2, \ldots, r_{k-1}$ be the reflection points of $\mathcal{C}^*_H$. Let $Z_i, 0 < i \le k$ denote the subpath of $\mathcal{C}^*_H$ from r_{i-1} to r_i, where r_0 and r_k are the two endpoints of C^*_H. By lemma 4.3, choose point $r'_i \in Q_H$ on the line segment containing r_i such that $|r_ir'_i| \le |r_i^-r_i| + |r_ir_i^+|$. Let Z'_i =**Shift**$(Z_i,\ r_{i-1},\ r'_{i-1})$ and Z''_i =**Shift**$(Z'_i,\ r_i,\ r'_i)$ be curves obtained by consecutively shifting the two endpoints to r'_{i-1}, r'_i respectively. Let Z'' denote the curve formed by concatenating $Z''_1, Z''_2, \ldots, Z''_k$. Clearly Z'' covers the cells covered by Z, and therefore is a halving curve. Further $|Z''| \le |\mathcal{C}^*_H| + 2 \cdot \sum_{i=1}^{k-1} |r_ir'_i| \le |\mathcal{C}^*_H| + 2 \sum_{i=1}^{k-1} (|r_i^-r_i| + |r_ir_i^+|) \le 5 \cdot |\mathcal{C}^*_H|$ where we use the fact that r'_i were chosen by lemma 4.3.

It is easy to show that an $O(\log^2 n)$ approximation to the optimal halving curve is obtained by solving the following instance of $\frac{1}{2}$-Group Steiner: we take G as the complete graph on Q_H. The weight of edge (q_1, q_2) is taken to be the shortest length path in $\mathcal{M}_H$ joining points q_1, q_2. There are k groups where group g_i consists of all coordinates $q \in Q_H$ such that $h_i \notin Maj(C)$ for at least one of q's neigbouring cells C.

5 Further Extensions

Here we sketch some extensions of our algorithm. If the robot does not possess a compass, but has no actuator uncertainty with respect to changes in orientation, the lower bound remains valid. For the algorithm, redefine a hypothesis to be a (location,orientation) pair. The size of the set H of possible hypotheses remains $O(n)$, and the algorithm extends naturally as the robot operates on the map relative to the original (location,orientation) pair.

When the map polygon contains polygonal holes, the cells of subpartition $D(P, H_{ij})$ will be polygons with holes rather than just simple polygons. A vertex of such a cell may see more than one continuous portion of a particular edge through the windows formed by different obstacles. Therefore the dual of aspect graph needs to be augmented by adding rays $Ray(v, e_1(v)), Ray(v, e_2(v)), \ldots, Ray(v, e_k(v))$ where v is a vertex of the map polygon (including the vertices on holes) and e_i's are the end points of the various portions (each portion will still be a line segment) of e visible from v. This leads to a decomposition of cardinality $O(n^4)$ as before. The hypothesis equivalence decomposition $D(P, H)$ can then be formed by taking intersections of all subpartitions $D(P, H_{ij})$ and hence is of polynomial cardinality. We can verify that the lemmas above hold for polygonal cells with holes, and therefore our algorithm extends with the same approximation guarantee but a concomitant increase in running time.

References

[1] Ricardo A. Baeza-Yates, Joseph C. Culberson and Gregory J. E. Rawlins, *Searching in the Plane*, Information and Computation 2 (1993), pp. 234-252.

[2] Yair Bartal, *Probabilistic Approximations of Metric Spaces and Its Algorithmic Applications*, Foundations of Computer Science (1996), pp. 184-193.

[3] Anit Chakraborty, Hari Balakrishnan and Nissanka B. Priyantha, *The Cricket Location-Support System*, Mobile Computing and Networking (2000), pp. 32-43.

[4] I. J. Cox, *Blanche - an experiment in guidance and navigation of an autonomous robot vehicle*, IEEE Trans. on Robotics and Automation (1991), pp. 193-204.

[5] E. Davis, *Representing and Acquiring Geographic Knowledge*, Pitman and Morgan Kaufmann Publishers, Inc. (1986).

[6] Dudek, Romanik and Whitesides, *Localizing a Robot with Minimum Travel*, SIAM Journal on Computing 27 (1998), pp. 583-604.

[7] G. Even, G. Kortsarz and W. Slany, *On network design: fixed charge flows and the covering Steiner problem*, SWAT 2002, pp. 318-329.

[8] Jittat Fakcharoenphol, Satish Rao and Kunal Talwar, *A tight bound on approximating arbitrary metrics by tree metrics*, STOC 2003.

[9] Garg, Konjevod and Ravi, *A Polylogarithmic Approximation Algorithm for the group Steiner Tree Problem*, Journal of Algorithms 37 (2000), pp. 67-84.

[10] J.E. Goodman and J. O'Rourke, *Handbook of Discrete and Computational Geometry*, Chap. 25, CRC Press 1997.

[11] L. Guibas, R. Motwani and P. Raghavan, *The robot localization problem in two dimensions*, SODA 1992, pp. 259-268.

[12] E. Halperin and R. Krauthgamer, *Polylogarithmic inapproximability*, STOC 2003, pp. 585-594.

[13] Jon M. Kleinberg, *The Localization Problem for Mobile Robots*, FOCS 1994, pp. 521-531.

[14] D. P. Miller, D. J. Atkinson, B. H. Wilcox, and A. H. Mishkin, *Autonomous navigation and control of a Mars rover*, Automatic Control in Aerospace (Selected papers from IFAC Symposium, Tsukuba, Japan, 1989), Pergamon, Oxford, UK, 1989.

[15] C. Ming Wang, *Location estimation and uncertainty analysis for mobile robots*, Autonomous Robot Vehicles (Ed: I. J. Cox and G. T. Wilfong), Springer-Verlag, Berlin (1991).

[16] C. Papadimitriou and M. Yannakakis, *Shortest paths without a map*, Theoretical Computer Science 84 (1991), pp. 127-150.

[17] W.D. Rencken, *Concurrent localisation and map building for mobile robots using ultrasonic sensors*, In Proc. of IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (1993), pp. 2129-2197.

[18] D. Sleator and R. Tarjan, Amortized efficiency of list update and paging rules, Comm. ACM 23 (1985), pp 202-207.

[19] R. Simmons, R. Goodwin, K. Haigh, S. Koenig and J. O'Sullivan, *A layered architecture for office delivery robots*, In Proc. of the First Intl. Conf. on Autonomous Agents (1997).

[20] W. Thompson, H. Pick, B. Bennett, M. Heinrichs, S. Savitt and K. Smith, *Map-based localization: The 'drop-off' problem*, Proc. DARPA Image Understanding Workshop (1990), pp. 706-719.

[21] C. Tovey and S. Koenig, *Gridworlds as Testbeds for Planning with Incomplete Information*, Proc. of the Natl. Conf. on Artificial Intelligence (2000), pp. 819-824.

[22] G. Weib, C. Wetzler and E. Von Puttkamer, *Keeping track of position and orientation of moving indoor systems by correlation of range-finder scans*, Proc. of the Intl. Conf. on Intelligent Robots and Systems (1994), pp. 595-601.