Localization: Approximation and Performance Bounds to Minimize Travel Distance

Craig Tovey and Sven Koenig

***Abstract*—Localization, which is the determination of one's location in a known terrain, is a fundamental task for autonomous robots. This paper presents several new basic theoretical results about localization. We show that, even under the idealized assumptions of accurate sensing and perfect actuation, it is intrinsically difficult to localize a robot with a travel distance that is close to minimal. Our result helps to theoretically justify the common use of fast localization heuristics, such as greedy localization, which always moves the robot to a closest informative location (where the robot makes an observation that decreases the number of its possible locations). We show that the travel distance of greedy localization is much larger than minimal in some terrains because the closest informative location can distract greedy localization from a slightly farther, but much more informative, location. However, we also show that the travel distance of greedy localization can be larger, but not much larger, than the terrain size n. Thus, the travel distance of greedy localization scales well with the terrain size and is much larger than minimal in some terrains, not because it is large with respect to the terrain size, but because the minimal travel distance is exceptionally small in these terrains. As a corollary to our analysis, we show that the travel distance of greedy mapping (which always moves the robot to a closest location, where it makes an observation that increases its knowledge of the terrain) cannot be much larger than the terrain size. In theoretical terms, we prove the NP-hardness of minimization of travel distance for localization to within a logarithmic factor of the terrain size. We prove that the travel distance of greedy localization is at least order $n/\log^2 n$ larger than minimal in some terrains and that it is at least order $n \log n/\log\log n$ in the worst case. Finally, we prove that the travel distance of both greedy localization and greedy mapping is at most order $n \log n$. Previously, it was only known that it is NP-hard to localize with minimal travel distance and that the travel distances of greedy localization and greedy mapping are at most order $n^{3/2}$.**

***Index Terms*—Greedy localization, greedy mapping, heuristic algorithms, NP-hardness, worst-case analysis.**

I. Introduction

THIS PAPER presents several new basic theoretical results about localization. The localization problem (sometimes also called the kidnapped-robot problem) is used to determine the unknown location of a robot when a map of the terrain is given. We show that, even under the idealized assumptions of accurate sensing and perfect actuation, it is intrinsically difficult (NP-hard) to localize a robot with an execution time (i.e., worst-case travel distance with respect to all possible starting locations) that is close to minimal. We prove our result for both tactile (short-distance) and long-distance terrain sensors in both polygonal regions and grid graphs. It, thus, applies to localization in any terrain that is more complex or general than polygonal regions and grid graphs and, thus, to virtually all terrains of interest in robotics. To gain an intuitive understanding of our result, note that a terrain, such as an office building, can have many similar areas. Hence, the robot might have to move to several locations to gather sufficient information to be able to localize. Choosing a small set of locations that, taken together, allow the robot to localize is a form of the set-cover problem, which is known to be NP-hard to approximate within a logarithmic factor.

Our result helps to justify theoretically the common use of fast localization heuristics. Therefore, it is important to understand the behavior of these heuristics. Greedy localization always moves the robot (with minimal travel distance) to a closest informative location (where the robot makes an observation that decreases the number of its possible locations). It has several attractive features. For example, it is simple to integrate into complete robot architectures because it is conceptually simple and does not need to have control of the robot at all times. This is important because planning methods should only provide advice on how to act, and work robustly even if this advice is ignored from time to time [1]. We show that the execution time of greedy localization can be much larger than minimal in some terrains because the closest informative location can distract greedy localization from a slightly farther, but much more informative, location, thus suggesting that improvements need to take into account the clustering of informative locations. However, we also show that the execution time of greedy localization can be larger, but not much larger, than the terrain size. Thus, the execution time of greedy localization scales well with the terrain size and can be much larger than minimal in some terrains, not because it is large with respect to the terrain size, but because the minimal execution time is exceptionally small in these terrains. In theoretical terms, we prove NP-hardness of minimization of execution time for localization to within a logarithmic factor of the terrain size. We also prove that the worst-case execution time of greedy localization (with respect to all possible terrains of a given terrain size n) is $\Omega(n/\log^2 n)$ larger than minimal and that it is $\Omega(n \log n/\log\log n)$ and $O(n \log n)$ in grid graphs, which

Manuscript received August 24, 2009; revised October 23, 2009. This paper was recommended for publication by Associate Editor C. Stachniss and Editor L. Parker upon evaluation of the reviewers' comments. This work was supported in part by the National Science Foundation under Contract IIS-0098807 and in part by the U.S. Army Research Laboratory and the U.S. Army Research Office under Contract W911NF-08-1-0468. This paper was presented in part at the International Symposium on Artificial Intelligence and Mathematics, Fort Lauderdale, FL, January 4–6, 2004 and the IEEE International Conference on Intelligent Robots and Systems, Las Vegas, NV, October 27–31, 2003.

C. Tovey is with the Department of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: cat@gatech.edu).

S. Koenig is with the Department of Computer Science, University of Southern California, Los Angeles, CA 90089 USA (e-mail: skoenig@usc.edu).

Digital Object Identifier 10.1109/TRO.2009.2039035

implies nearly sharp bounds. Previously, it was only known that it is NP-hard to localize with minimal execution time and that the worst-case execution time of greedy localization is $O(n^{3/2})$ in grid graphs.

Our analysis of the upper bound on the execution time of greedy localization is quite general, since it applies to any greedy graph traversal algorithm of the form "always move the robot to a closest informative location" provided that, when a location is visited, it, and possibly other locations, become uninformative, but uninformative locations cannot become informative. This generality allows us to apply our result to mapping. The mapping problem is to determine a map of the terrain. Greedy mapping always moves the robot (with minimal travel distance) to a closest location where it makes an observation that increases its knowledge of the terrain. Our result implies that the execution time of greedy mapping cannot be much larger than the terrain size n. In theoretical terms, we prove that the worst-case execution time of greedy mapping is $O(n \log n)$ in grid graphs. This result, together with our previous result that it is $\Omega(n \log n / \log \log n)$ in grid graphs, implies nearly sharp bounds. Previously, it was only known that the worst-case execution time is $O(n^{3/2})$ in grid graphs.

II. Assumptions

The localization problem is to determine the unknown location of a robot when a map of the terrain is given. We assume a point robot with accurate sensing, perfect actuation, and knowledge of its orientation from an onboard compass. Our assumptions are realistic in some cases. Greedy localization, for example, has been used on Nomad 150 mobile robots. The success rate of moving was at least 99.57%, and the success rate of sensing was at least 99.38% [2]. These large success rates enable one to ignore actuator and sensor noise, especially since the rare failures are usually quickly noticed when the number of possible locations drops to zero, in which case the robot simply reinitializes its belief state to all possible locations and then continues to use the localization algorithm unchanged. However, our assumptions are idealized in many cases. In general, we expect that that all of our lower bound results should continue to hold under less ideal conditions and our upper bound results can only get worse. Our results, therefore, remain informative under more real-world conditions.

We consider both (continuous) polygonal regions and (discrete) grid graphs as models of 2-D terrain. A *polygonal region* is a polygon that can contain obstacle polygons. The polygon and obstacle polygons consist of finite sets of line segments [3]. The robot moves continuously in the polygonal region, but cannot pass through line segments. The terrain size of polygonal regions is the length of the encoding of all line segments, where each line segment is encoded by two pairs of integer coordinates, one for each of its two endpoints. A *grid graph* corresponds to a gridworld, which discretizes terrain into unit square cells that are either traversable or untraversable. The cells beyond the perimeter of the gridworld are considered untraversable. The robot is in exactly one traversable cell and can move in one of the four main compass directions to an adjacent traversable cell

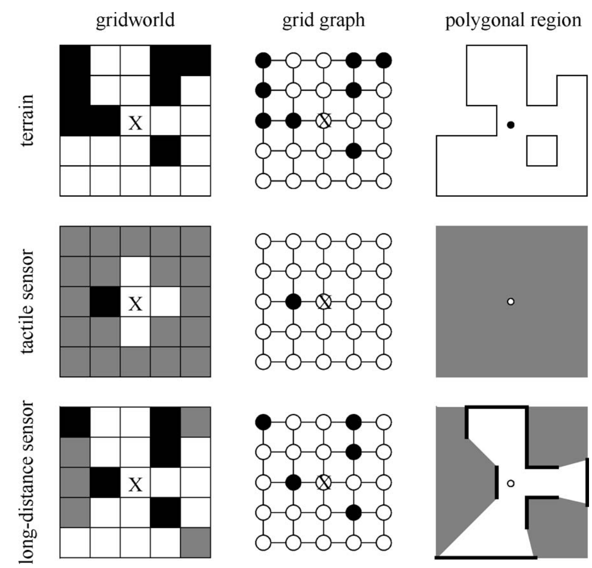

Fig. 1. Terrain models.

with travel distance 1. (A gridworld can be modeled as a polygonal region, but the robot moves differently.) A grid graph has a vertex for every cell in the gridworld. The traversability of the vertex is the same as the traversability of the cell. Two vertices are connected by an edge iff they correspond to adjacent cells. The terrain size of a grid graph is the number of traversable vertices of its largest connected component.

We consider both tactile (short-distance) and long-distance sensors. *Tactile sensors* model bump sensors. In polygonal regions, they detect whether the current location of the robot is on a line segment [4]. In grid graphs, they detect the traversability of all vertices adjacent to the current vertex. *Long-distance sensors* model laser range finders. In polygonal regions, they detect the distances to all visible line segments [5]. They operate in all 360° directions for unlimited distances but cannot see through line segments. In grid graphs, they operate in the same way if the robot is considered positioned in the center of its current cell in the polygonal region that represents the corresponding gridworld. Assume, for example, that a robot in cell $[0, 0]$ is directly below an untraversable cell $[0, 1]$ and that all other cells are traversable. Then, the robot observes the traversability of cells $[1, 1]$, $[2, 2]$, etc., but not of cells $[1, 2]$, $[2, 3]$, etc., because it cannot see through the lower line segment of the untraversable cell.

Fig. 1 illustrates the terrain models and sensors. The left column shows a gridworld (illustrating the assumptions of grid graphs), the center column shows the corresponding grid graph, and the right column shows the corresponding polygonal region. Different from the example, polygonal regions can consist of arbitrary polygons. The top row shows the terrain, the center row shows in black and white what a robot with tactile sensors observes in the given starting location (marked with an X for gridworlds and grid graphs and a dot for polygonal regions), and the bottom row shows what a robot with long-distance sensors observes. The gray shadow shows the unobservable part

TABLE I
SUMMARY OF LOWER BOUNDS ON WORST-CASE PERFORMANCE RATIOS

	tactile (short-distance) sensor	long-distance sensor
Greedy Localization in polygonal regions	$\Omega(n/\log^2 n)$	$\Omega(n/\log n)$
Greedy Localization in grid graphs	$\Omega(n/\log n)$	$\Omega(n)$
Depth-First Search in grid graphs	$\Omega(n/\log n)$	$\Omega(n)$

of the terrain. The traversability of a partially observed cell is completely known for long-distance sensors in grid graphs.

III. DEFINITIONS

Solving a localization problem requires a reconnaissance plan in which the robot moves to gather sufficient information to be able to localize. A (valid) localization plan P for terrain M is a deterministic reconnaissance plan that always specifies the next move of the robot, when its sensor and movement history are given, and terminates with either the correct current location of the robot or the correct conclusion that the location of the robot cannot be uniquely determined. Also, the algorithm is not required to localize correctly if its map is not correct. The location cannot be determined if there are two connected components of the terrain that are identical and the robot is located in one of them. The execution time $v(P, M)$ of localization plan P for terrain M is the resulting worst-case travel distance with respect to all possible starting locations. (Technically speaking, the execution time for polygonal regions is defined as the supremum rather than the maximum, because the maximum might not exist.) The objective is to find a localization plan with a small execution time for a given terrain M. Localization plan P^* is optimal for terrain M iff $v(P^*, M) \leq v(P, M)$ for all localization plans P, i.e., if its execution time in terrain M is minimal. We refer to this execution time as the minimal execution time $v^*(M) = v(P^*, M)$ for map M. The performance ratio of localization plan P for terrain M is $v(P, M)/v^*(M)$. The worst-case travel distance is with respect to all possible starting locations, while the worst-case execution time and performance ratio are with respect to all possible terrains of a given size.

IV. OVERVIEW OF RESULTS

Dudek *et al.* [5] proved, for long-distance sensors in polygonal regions, that it is NP-hard to find localization plans that are optimal, i.e., whose execution time is minimal. In other words, given a polygonal region M and value T, it is NP-hard to determine if $v^*(M) \leq T$. Tovey and Koenig [6] proved the same result for tactile sensors in grid graphs. These results extend readily to the cases of tactile sensors in polygonal regions and long-distance sensors in grid graphs. In this paper, we prove a stronger result. In Section VI, we show that there exists a constant $c > 0$, such that it is NP-hard to find localization plans with performance ratio at most $c \log n$ in terrains with terrain size n for both tactile and long-distance sensors in both polygonal regions and grid graphs. Thus, under the assumption that $P \neq$ NP, one cannot find a localization plan in polynomial time whose execution time is at most a factor of $c \log n$ larger than minimal. Koenig *et al.* [7] used a modification of our construction for this proof, which is detailed in this paper, to prove, under the stronger assumption that NP $\not\subseteq$ ZTIME$(n^{\mathrm{polylog}(n)})$, that one cannot find a localization plan in polynomial time whose execution time is at most a factor of $c \log^{2-\epsilon}(n)$ larger than minimal for both tactile and long-distance sensors in grid graphs. We extend this result to both tactile and long-distance sensors in polygonal regions as a corollary.

In Section VII, we assess the performance ratios of two fast localization heuristics, namely, a simple depth-first search (DFS) algorithm and greedy localization, and we find that they are, in a word, terrible. Table I summarizes our results. We find that greedy localization has a poor worst-case performance ratio of $\Omega(n/\log^2 n)$ for both tactile and long-distance sensors in both polygonal regions and grid graphs, i.e., its worst-case execution time is much larger than minimal. The results in polygonal regions differ by a logarithmic factor from the results in grid graphs because the terrain size n must be calculated differently in both kinds of terrains. The DFS algorithm has a similarly poor worst-case performance ratio for both tactile and long-distance sensors in grid graphs.

Finally, in Sections VIII and IX, we assess the execution time of greedy localization for both tactile and long-distance sensors in grid graphs. We find that the worst-case execution time is $\Omega(n \log n/ \log \log n)$ and $O(n \log n)$ for both tactile and long-distance sensors in grid graphs. Our analysis is quite general, which allows us to find that the execution time of greedy mapping is $O(n \log n)$ for both tactile and long-distance sensors in grid graphs as well. We have not been able to develop satisfactory analogous results for greedy localization in polygonal regions. In Section X, we discuss the scaling problems encountered in our attempts to find a suitable definition of the terrain size for this purpose.

V. RELATED WORK

Localization algorithms have been studied for some time. Passive-localization algorithms track the set of possible locations of the robot as it moves. We are interested in active localization algorithms, which move the robot so that it reduces the set of possible locations to one with small execution time. A variety of geometric algorithms can form the basis of passive and active localization algorithms. Guibas *et al.* [3] gave a polynomial-time algorithm for long-distance sensors in polygonal regions to determine the set H of locations that are consistent with the current sensor scan. Brown and Donald [8] gave an alternative algorithm in the case where the long-distance sensors are point-and-shoot sensors. Aronov *et al.* [9], Zarei and Ghodsi [10], and Bose *et al.* [11] have performed related work on computing visibility polygons.

Several results subsequently appeared that study active localization algorithms from the standpoint of competitive analysis

[12], which were applied to navigation problems first by Baeza-Yates *et al.* [13], [14]. Dudek *et al.* [5] gave a best possible competitive ratio of $|H| - 1$ for long-distance sensors in polygonal regions, where H is computed from the starting location. Both Schuierer [15] and Karch *et al.* [16] provided a further analysis of the time and space complexity. Kleinberg [17], who was followed by Fleischer *et al.* [18], analyzed the competitive ratio in specific graphs, namely, geometric trees. An online competitive analysis compares the travel distance of a localization algorithm to the travel distance of an omniscient robot that knows its location at the outset and seeks only to verify that location. Minimizing the ratio of these quantities minimizes regret in the sense that it minimizes the value of k such that the robot could have localized k times faster. We, on the other hand, analyze two other measures that are of at least equal concern in practice. The first measure is the performance ratio, which is the classical measure of algorithm analysis applied to localization problems. It compares the execution time of a localization algorithm, namely, the worst-case travel distance with respect to all starting locations, to the execution time of an optimal localization algorithm that does not know the location of the robot at the outset. A small measure guarantees that the execution time is close to minimal. The second measure is the worst-case execution time. A small measure guarantees that the execution time scales well with the terrain size.

Localization algorithms can also be based on next-best-view algorithms, which determine the location where the robot makes an observation that adds the most information to the one that has already been gathered. The resulting localization algorithms ignore the travel distance and attempt to minimize the number of observations. We, on the other hand, analyze greedy localization, which ignores the cost of the observations and attempts to minimize the travel distance. Gonzalez-Banos and Latombe [19] and Murta *et al.* [20] proposed hybrid-localization algorithms that always move the robot to a most informative location within a region around the current location that is guaranteed to be traversable. Rao *et al.* [21] proposed a similar localization algorithm, except that it samples locations with the traversable region randomly and moves the robot to a location that trades off between its informativeness and the travel distance needed to reach it. Koenig *et al.* [7] proposed a localization algorithm that identifies a "majority-rule" region around the starting location that, if untraversable when expected to be traversable or *vice versa*, at least halves the number of possible locations, and then attempts to visit informative locations within that region. It finds a localization plan in polynomial time whose execution time is at most a factor of $O(\log^3(n))$ larger than minimal. However, its runtime is $\Omega(n^{12})$, which is not close to practical.

VI. Complexity of Approximately Minimizing Execution Time

We now use a connected-grid-graph construction, which is a simplification of the one presented in [6], to prove NP-hardness of the minimization of execution time for localization to within a logarithmic factor of the terrain size for both tactile and long-distance sensors in both polygonal regions and grid graphs.

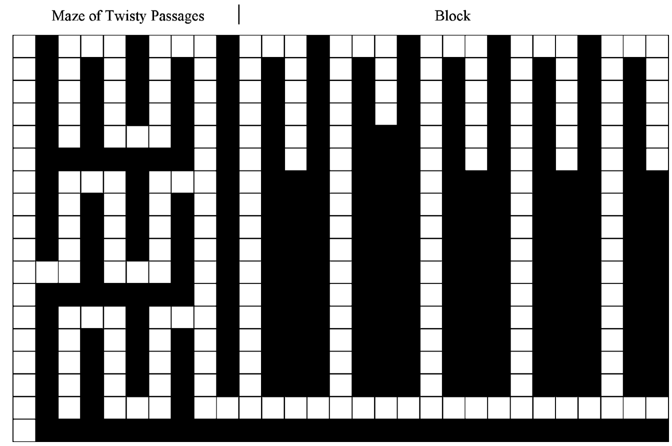

Fig. 2. Conceptual block for Theorem 6.1

Theorem 6.1: There exists a constant $c > 0$, such that it is NP-hard to find a localization plan with performance ratio at most $c \log n$ for both tactile and long-distance sensors in connected grid graphs of terrain size n.

Proof: We reduce from set cover. An instance of set cover consists of the base set $S = \{1, \ldots, t\}$ and a collection of sets $S_1, \ldots, S_y \subseteq S$. A set cover is a subcollection of these sets, whose union is S. Let y^* denote the minimum cardinality of a set cover of the given instance. For $c' < 1$, to find a set cover of cardinality at most $c' y^* \log t$ is NP-hard [22]. Moreover, y can be used to satisfy $\log y \leq c'' \log t$ for some $c'' > 0$ (equivalent to y being polynomially bounded in t) [22]. We reduce this problem to that of finding a localization plan in a grid graph M that corresponds to a connected gridworld of size $n_1 \times n_2$ with $n_1 = 4y + 2t + O(1)$ and $n_2 = t^2 y + O(t)$, whose execution time is within factor $c \log(n_1 n_2)$ of minimal. The traversable cells in the gridworld are not sparse, i.e., $n = \Theta(n_1 n_2)$, which proves the result.

We construct the gridworld for a given instance of set cover as follows. It consists of a column of t blocks of size $(4y + O(1)) \times (ty + O(1))$ each, one for each base element $i = 1, \ldots, t$. The right side of Fig. 2 shows a conceptual example of a block. The second to lowest row of each block contains a left-right corridor. There are y "hooking" corridors connected to the left-right corridor, each running up for $ty - 2$ cells, right for two cells, and then down for t cells. The hooking corridors are spaced four cells apart. The jth hooking corridor represents the set S_j. If $i \in S_j$, then the last i cells of hooking corridor j are untraversable, and we say that the corridor is revelatory. The key idea is that the length of a revelatory hooking corridor identifies the block, but the robot needs to move almost to its end to explore it for both tactile and long-distance sensors. The robot then basically needs to explore a group of hooking corridors that correspond to a set cover to guarantee that it explores a revelatory hooking corridor, which localizes the robot. The blocks would suffice for the construction, except that the grid graph would not be connected. Therefore, to the left of each block, append an up-down corridor connected to the left-right corridor and attach to it a maze of t twisty left-right passages. Each twisty passage fits in a region of size $(2t + O(1)) \times (y + O(1))$ and has length $ty + O(1)$. The ith twisty passage of block $i = 1, \dots, t$, has a left exit. All other twisty passages of block i are dead ends. Connect the exit twisty passages with an up-down hallway. The left side of Fig. 2 shows a conceptual example of a maze with twisty passages. The grid graph can be constructed in polynomial time in the size of the set cover. The hooking corridors and twisty passages ensure that at least 25% of the cells are traversable. Thus, we have $n = \Theta(n_1 n_2)$, as required.

We now calculate an upper bound z on the minimal execution time $v^*(M)$. Consider the following localization plan: If the robot starts in the hallway, it localizes within travel distance $4ty + O(1)$ by moving up and identifying either the end of the hallway or an exit twisty passage, which it can then follow. Otherwise, the robot moves within travel distance $4ty + O(1)$ to the lower left corner of a block. The robot then moves right and explores, in turn, each of the y^* hooking corridors that correspond to a minimal set cover. The robot must eventually explore a revelatory hooking corridor, which localizes the robot. The execution time of this (valid) localization plan is at most $4ty + 2(ty + t)y^* + O(1) \leq 2.01tyy^* = z$.

Finally, we show that a localization plan with a small execution time corresponds to a small set cover. Suppose localization plan P is a (valid) localization plan with execution time at most $0.99 \log t$ times the minimal execution time. Its execution time can thus be at most $0.99z \log t < 1.99tyy^* \log t$. We analyze how it localizes the robot from the lower left corner of a block: It obtains new information only at the end of a hooking corridor or twisty passage, since the blocks and mazes differ only in those places. It does not move to any other block because it localizes immediately when it explores an exit twisty passage. It also localizes immediately when it explores a revelatory hooking corridor. The twisty passages provide the same information as would hooking corridors that represented singleton sets $\{i\}$, for $i = 1, \dots, t$. Therefore, we may think of localization plan P as exploring hooking corridors until it finds a revelatory one or has eliminated all but one block. Choose a block i for which it explores the most hooking corridors. The hooking corridors that it explores must correspond to a set cover with the possible exception of i, since it would otherwise need to explore additional hooking corridors. Therefore, it finds a subcollection of hooking corridors that is either a set cover or one set short of a set cover. Any singleton set $\{i\}$ that corresponds to a twisty passage may trivially be replaced by any other set containing i. It takes travel distance at least $2ty$ to explore a nonrevelatory hooking corridor. Consequently, localization plan P can sample at most $(1.99tyy^* \log t)/(2ty)$ hooking corridors and provides a set cover of cardinality at most $1 + 0.995y^* \log t < y^* \log t$. ■

Next, we show that any hardness result for grid graphs applies to polygonal regions, within constant factors, which enables us to prove hardness of localization in polygonal regions without having to construct a lengthy proof similar to that of Theorem 6.1.

Proposition 6.2: For any grid graph G, one can construct a polygonal region H in linear time, such that any localization plan for either tactile or long-range sensors on H can be transformed in linear time to a localization plan for tactile sensors in G so that the execution time of the latter is at most $\sqrt{2}$ times the execution time of the former. Moreover, the optimal localization execution time and size of H are at most a constant multiple of the optimal localization execution time and size, respectively, of G.

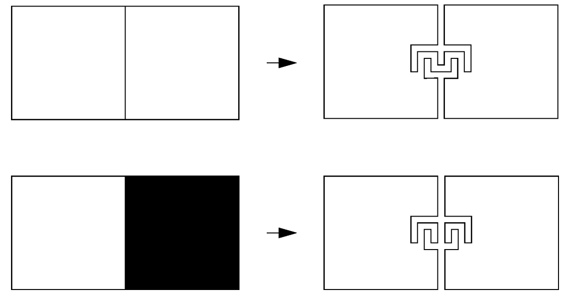

Fig. 3. Converting gridworlds to polygonal regions.

Proof: Given a grid graph G, construct the polygonal region H by replacing each boundary between a traversable cell in the grid graph and any of its adjacent cells with a wall through which a twisty tunnel passes. The tunnel is blocked at the crossing point iff the boundary is untraversable. Fig. 3 shows an example. This construction requires linear time because each twisty tunnel is $\theta(1)$ in size and precision. It follows that the size of H is not more than a constant multiple of the size of G.

Consider a robot that localizes in H. Each time the robot in H moves in order to gain information, it travels at least $1/\sqrt{2}$. The corresponding robot in G either already has this information or, if the robot in H had moved to an adjacent cell, it (the robot in G) can travel a distance 1 to the corresponding cell, from which it acquires this information. Each of these movements can be computed in constant time. The robot in the grid graph always has at least as much information as the robot in the polygonal terrain. Therefore, we have a localization plan for G, which costs at most $\sqrt{2}$ times the cost of the plan in H.

Now consider a robot that localizes optimally in G. When it moves from a cell to an adjacent cell in G (with travel distance 1), the corresponding robot in the polygonal region then moves to the corresponding cell and then immediately moves into each tunnel of its new cell to check the crossing point, thus incurring travel distance at most $1 + 4/\sqrt{2}$. Both robots always have identical information about the traversability of cells, since the twisty tunnels render long-distance sensors no more useful than tactile sensors. Therefore, there is a localization plan for H that costs at most $1 + 2\sqrt{2}$ times the optimal cost of localization in G. ■

Proposition 6.2 allows us to extend Theorem 6.1 to polygonal regions.

Corollary 6.3: There exists a constant $c > 0$, such that it is NP-hard to find a localization plan with performance ratio at most $c \log n$ for both tactile and long-distance sensors in polygonal regions of size n.

Thus, under the assumption that $P \neq \text{NP}$, one cannot find a localization plan in polynomial time whose execution time is at most a factor of $c \log n$ larger than minimal for both tactile and long-distance sensors in both polygonal regions and grid graphs. Under the stronger assumption that $\text{NP} \not\subseteq Z\text{TIME}(n^{\text{polylog}(n)})$, one cannot find a localization plan in polynomial time whose execution time is at most a factor of $c \log^{2-\epsilon}(n)$ larger than minimal for tactile sensors in grid graphs [7]. Proposition 6.2 allows us to extend this theorem to polygonal regions.

Corollary 6.4: There exists $c > 0$, such that, if $\text{NP} \not\subseteq Z\text{TIME}(n^{\text{polylog}(n)})$, one cannot find a localization plan in polynomial time whose execution time is at most a factor of $c \log^{2-\epsilon}(n)$ larger than minimal, where n is the size of the polygon. The result holds for both tactile and long-range sensors.

As noted earlier, it was already known that it is NP-hard to find localization plans that are optimal for both tactile and long-distance sensors in both polygonal regions and grid graphs, i.e., whose execution time is minimal. Our results provide stronger guarantees for both tactile and long-distance sensors in both polygonal regions and grid graphs, namely, that it is NP-hard (and, thus, likely impossible with polynomial planning time) to find localization plans that are close to optimal, i.e., whose execution time is close to minimal. They, as with all lower bound results in this paper, can be interpreted pessimistically. Since they hold under our idealized assumptions, they should continue to hold under less ideal conditions and in virtually all terrains of interest in robotics, thus confirming earlier empirical results that performing complete AND–OR searches in belief space to determine optimal-localization plans is often infeasible [23].

VII. Performance Ratios

We now describe two fast-localization heuristics, namely, a simple DFS algorithm and greedy localization. DFS is a strawman localization algorithm. Greedy localization has been used by the delayed planning architecture with the viable-plan heuristic [24]. Nourbakhsh pioneered the delayed planning architecture in robot programming classes, where Nomad 150 mobile robots had to navigate in gridworlds that were built with 3-ft-high and 40-in-long cardboard walls [2]. Subsequently, it was generalized in [25] and [26].

1) *DFS* operates in grid graphs. First, it determines the connected components M_i of the given map of the grid graph. Second, it acquires a map M' of the component that the robot is in by moving the robot in a DFS manner. Third, it determines which of the components M_i are identical to map M' using a DFS for every component, starting from the leftmost vertex of all uppermost traversable vertices. If exactly one component M_i matches map M', it has localized the robot. Otherwise, it can conclude that the robot cannot localize. It is unclear how to generalize DFS to operate in polygonal regions.
2) *Greedy localization*, on the other hand, operates in both polygonal regions and grid graphs. It always moves the robot (with minimal travel distance) to a closest informative location (where the robot makes an observation that decreases the number of its possible locations) [15]. The algorithm by Guibas *et al.* [3] can be used to determine the locations where the robot makes an observation that decreases the number of its possible locations.

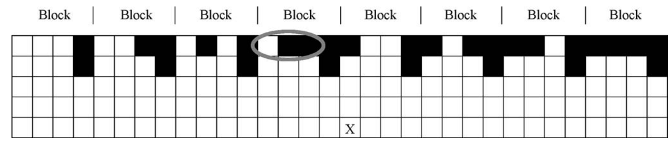

Fig. 4. Gridworld for Theorem 7.2.

The planning and execution times of DFS and greedy localization are polynomial, and thus imply two properties, since Theorem 6.1 and Corollary 6.3 suggest that it is impossible with polynomial planning time to find localization plans that are close to optimal: First, DFS and greedy localization substantially reduce the sum of planning and execution time compared with localization algorithms that find localization plans that are optimal or close to optimal. Second, DFS and greedy localization can find localization plans that are not close to optimal. Their worst-case performance ratios are $\Omega(\log n)$ according to Theorem 6.1 and Corollary 6.3. In the following, we prove much larger lower bounds on their worst-case performance ratios.

The worst-case performance ratios of DFS and greedy localization could be argued to be arbitrarily poor, for a trivial reason. Consider, for example, the performance ratios of DFS and greedy localization in a polygonal region or grid graph that consists of two identical copies of the polygonal region or grid graph from Theorem 6.1 or Corollary 6.3. The minimal execution time is zero, since the robot cannot localize. However, DFS and greedy localization move the robot for both tactile and long-distance sensors, thus resulting in an infinite performance ratio. We, therefore, assume in the following that the robot can localize from all starting locations.

Theorem 7.1: The worst-case performance ratio of greedy localization is $\Omega(n/\log^2 n)$ for tactile sensors in connected polygonal regions of terrain size n.

Proof: Consider the following gridworld that consists of a row of x blocks of size $(\log x + 1) \times 5$ each, where $x \geq 8$ is a power of two. The uppermost row of each block contains a "signature." For block $k = 1, \ldots, x$, this signature encodes $k - 1$ in binary form, which needs $\log x$ bits. The signature is in the form of a pattern of traversable and untraversable cells, which is followed by a separator that consists of a column of two untraversable cells. The remainder of the block consists of traversable cells. Fig. 4 shows an example with the signature 0 1 1 circled. The gridworld has $\Theta(x \log x)$ traversable cells and is connected. A robot with tactile sensors can localize from anywhere by moving up until it reaches a signature, and then along the signature to read it. Now consider the polygonal region that corresponds to the gridworld. Its terrain size is $n = \Theta(x \log^2 x)$, since the representation of the integer coordinates needs, on average, $\Theta(\log x)$ bits. The minimal execution time is $O(\log x)$, since the robot can localize from anywhere within this time. Assume that the robot starts approximately in the center cell of the lowest row of the gridworld. DFS and greedy localization can then move the robot right. The robot does not localize

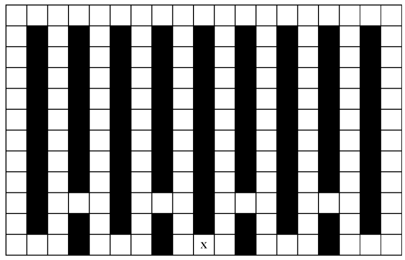

Fig. 5. Gridworld for theorem 7.3.

before it reaches the rightmost wall. Consequently, the execution time is $\Omega(x \log x)$ and thus results in a performance ratio of $\Omega(x \log x / \log x) = \Omega(x) = \Omega(n / \log^2 n)$. ■

Corollary 7.2: The worst-case performance ratio of DFS and greedy localization is $\Omega(n/\log n)$ for tactile sensors in connected grid graphs of terrain size n.

Proof: Consider the grid graph that corresponds to the gridworld from the proof of Theorem 7.1. Its terrain size is $n = \Theta(x \log x)$ (due to the different calculation of the terrain size), and the minimal execution time is $O(\log x)$. DFS and greedy localization can exhibit the behavior outlined in that proof (except that the robot does not localize before it reaches the second to rightmost cell), thus resulting in a performance ratio of $\Omega(x \log x / \log x) = \Omega(x) = \Omega(n/\log n)$. ■

The lower bound of Corollary 7.2 cannot be improved for DFS if the robot can localize from all starting cells, for the following reason: If the robot can localize with execution time x, then it can experience at most 16×8^x different sequences of observations and, thus, can distinguish at most among these many cells. (Initially, it observes the traversability of the four cells adjacent to its starting cell, thus resulting in one of 16 possible observations. After each move to an adjacent cell, it observes one of at most eight possible observations, since the cell it came from is traversable.) Thus, it must be the case that $n \leq 16 \times 8^x$, and thus, $x = \Omega(\log n)$ in grid graphs of terrain size n. The minimal execution time is thus $\Omega(\log n)$. The execution time of DFS is $O(n)$ in connected grid graphs, thus resulting in a worst-case performance ratio of $O(n/\log n)$ and, thus, $\Theta(n/\log n)$.

Theorem 7.3: The worst-case performance ratio of greedy localization is $\Omega(n/\log n)$ for long-distance sensors in connected polygonal regions of terrain size n.

Proof: Consider the following gridworld of size $(8x + 3) \times 12$, where $x > 0$. It contains $4x + 1$ up-down walls that alternate with $4x + 2$ up-down corridors. The uppermost cell of every wall is traversable, thus forming a left-right corridor in the uppermost row. One cell near the bottom of every wall is also traversable. It is called the chink cell. The chink cells alternate between the lowest cell (type-1 walls) and the cell 2 above the lowest cell (type-2 walls), and thus form a zigzag corridor. The leftmost, center, and rightmost walls are type-1 walls. Fig. 5 shows an example. The gridworld has $\Theta(x)$ traversable cells and is connected. A robot with long-distance sensors can localize from anywhere by moving up until it reaches the left-right corridor. Now consider the polygonal region that corresponds to the gridworld. Its terrain size is $n = \Theta(x \log x)$, since the representation of the integer coordinates needs on average order $\log x$ bits. The minimal execution time is $O(1)$, since the distance to the left-right corridor is $O(1)$. Assume that the robot starts in the center of the chink cell of the center wall. Greedy localization observes the chink cells of the adjacent type-2 walls, and therefore, it knows that it is not in the rightmost or leftmost type-1 wall but cannot distinguish among the other possible type-1 walls. It then moves diagonally to the center of the chink cell of the adjacent wall to its left or right because this travel direction is perpendicular to the line of sight to the next chink and, thus, eliminates one wall with minimal travel distance. If the robot moves to the right, then it does not localize before it reaches the chink in the rightmost wall. Consequently, the execution time is $\Omega(x)$ and, thus, results in a performance ratio of $\Omega(x/1) = \Omega(n/\log n)$. ■

Corollary 7.4: The worst-case performance ratio of DFS and greedy localization is $\Omega(n)$ for long-distance sensors in connected grid graphs of terrain size n.

Proof: Consider the grid graph that corresponds to the gridworld from the proof of Theorem 7.3. Its terrain size is $n = \Theta(x)$, and the minimal execution time is $O(1)$. DFS and greedy localization can exhibit a behavior similar to the one outlined in that proof (except that they move in a step pattern rather than diagonally), thus resulting in a performance ratio of $\Omega(x/1) = \Omega(n)$. ■

To summarize, our results show that DFS has a poor worst-case performance ratio for both tactile and long-distance sensors in grid graphs, even if the robot can localize from all starting locations. Similarly, greedy localization has a poor worst-case performance ratio for both tactile and long-distance sensors in both polygonal regions and grid graphs under the same assumption. Thus, their worst-case execution times are much larger than minimal. In particular, the closest informative location can distract greedy localization from a slightly farther, but much more informative, location, such as a unique signature of length $\log n$. For tactile sensors, the robot needs to move along the signature to read it for a travel distance of $O(\log n)$. For long-distance sensors, the robot could observe it from one well-chosen location for a travel distance of $O(1)$, which explains, in essence, why the lower bounds for tactile sensors differ from the lower bounds for long-distance sensors by a factor of $\log n$. The terrain sizes of polygonal regions and grid graphs are calculated differently, which explains, in essence, why the lower bounds (for greedy localization) in polygonal regions differ from the lower bounds in grid graphs by a factor of $\log n$.

VIII. Lower Bound on Execution Time

We now prove a lower bound on the execution time of greedy localization for both tactile and long-distance sensors in grid graphs.

Theorem 8.1: The worst-case execution time of greedy localization is $\Omega((n \log n/(\log \log n))$ for both tactile and long-distance sensors in grid graphs of terrain size n.

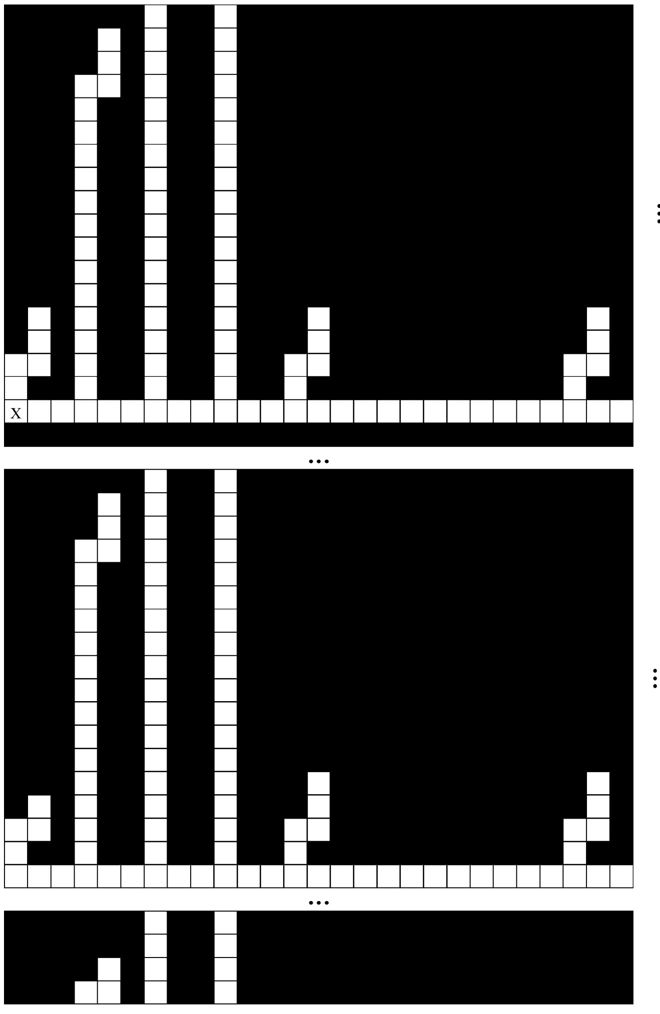

Fig. 6. Gridworld for Theorem 8.1.

Proof: Consider a gridworld that consists of a column of unconnected blocks that are variations of those presented in [27]. The second to lowest row of block 0 contains a left-right corridor with cells $i = 0, \ldots, d^d - d$, where $d \geq 12$. There are "towers" connected to the left-right corridor. A tower of height x runs up for x cells, right for one cell (reaching the "zigzag cell"), and then up again for two cells (reaching the "tip cell"). For each $k = 0, \ldots, \lfloor d/4 \rfloor$, there is a tower of height $2 + \sum_{j=1}^{k} d^j$ connected to the left-right corridor at cell $i + 3k$, for $i \equiv 0 \pmod{d^{k+1}}$. Thus, the leftmost tower of any height is to the left of the leftmost tower of any larger height, and the rightmost tower of any height is to the right of the rightmost tower of any larger height. The number of blocks is equal to the number of towers plus one. All blocks are identical to block 0, except that each additional block makes the tip cell of a different tower untraversable. Fig. 6 shows part of such a gridworld. Now consider the grid graph that corresponds to the gridworld. Its terrain size is $n = \Theta(d^d)$, since each block contains one corridor with $\Theta(d^d)$ traversable cells and $\Theta(d^{d-k-1})$ towers with $\Theta(d^k)$ traversable cells each for each of the $\Theta(d)$ different heights. Assume that the robot starts in the leftmost corridor vertex of block 0. The key idea is that greedy localization determines that it is in the leftmost corridor vertex of some block but needs to observe the tip vertices of all towers to determine the block and localize the robot. It needs to move to the zigzag vertex of a tower for long-distance sensors and one cell further for tactile sensors to observe its tip vertex. The travel distance from the attachment point of a tower of height $h = 2 + \sum_{j=1}^{k} d^j$ to the zigzag vertex of the nearest other tower of the same height is $d^{k+1} + h + 1 = 3 + \sum_{j=1}^{k+1} d^j$ and is, thus, smaller than the travel distance from the attachment point of the tower to the zigzag vertex of the nearest tower of any larger height. Therefore, greedy localization traverses the corridor from left to right, visiting the zigzag vertices of all towers of the smallest height, traverses the corridor from right to left, visiting the zigzag vertices of all towers of the next largest height, etc. It traverses the corridor $\Omega(d)$ times at travel distance $\Omega(d^d)$ each. Consequently, its execution time is $\Omega(d^{d+1})$ and, thus, results in a performance ratio of

$$\Omega(d^{d+1}) = \Omega\left(\frac{d^d\, d \log d}{\log d}\right) = \Omega\left(\frac{n \log n}{\log \log n}\right). \quad (1)$$

■

To summarize, our results show that the execution time of greedy localization can be larger than the terrain size for both tactile and long-distance sensors in grid graphs.

IX. Upper Bound on Execution Time in Grid Graphs

We now prove an upper bound on the execution time of any *greedy graph traversal algorithm*, which then also applies to greedy localization and greedy mapping for both tactile and long-distance sensors in arbitrary graphs, including grid graphs. Initially, all vertices are untagged. Let the starting vertex be x_0. The greedy-graph traversal algorithm always tags the current vertex x_{i-1} of the robot and possibly other vertices. (We make no assumptions whether it tags other vertices and, if so, which ones it tags.) It then moves the robot (with minimal travel distance) to a closest untagged vertex x_i. We make no assumptions about how it breaks ties. If all vertices are tagged, the algorithm terminates. Let $d(x, v)$ denote the travel distance from vertex x to vertex v. Let B_i denote the set of vertices that have been tagged when the robot reaches untagged vertex x_i. (B_0 is the empty set.) If $d(x_{i-1}, v) < d(x_{i-1}, x_i)$, then $v \in B_i$, because otherwise, v would be a closer untagged vertex than x_i. This intuitively suggests that the execution time of a greedy-graph traversal algorithm is small, since large values of $d(x_{i-1}, x_i)$ force many vertices to be tagged. We now analyze the execution time of greedy-graph traversal algorithms by defining marking sequences that abstract some of their properties away.

A marking sequence for graph $G = (V, E)$ is a sequence of triples $\{v_i, r_i, M_i\}$, for $i = 0, 1, \ldots, m$, where $r_i \geq 0$ is an integer, $v_i \in V$, and $M_i \subseteq V$ satisfy the following properties.

Property 1: $v_i \notin M_i$.

Property 2: $M_i \subseteq M_{i+1}$.

Property 3: $d(v_i, v) \leq r_i$ implies $v \in M_{i+1}$.

The cost of the marking sequence is $\sum_{i=0}^{m-1}(1 + r_i)$.

Any greedy graph traversal algorithm forms an associated marking sequence on the same graph with $v_i = x_i$, $r_i =$

$d(x_i, x_{i+1}) - 1 < |V| - 1$ (r_m can be set to zero) and $M_i = B_i$. The cost of this marking sequence equals the travel distance of the graph traversal algorithm, since $1 + r_i = d(x_i, x_{i+1})$. In general, marking sequences are less restrictive than greedy-graph traversal algorithms, since v_{i+1} need not be within travel distance $1 + r_i$ of v_i. Marking sequences consist of a sequence of choices of an untagged vertex v_i (i.e., one not in M_i) and a radius r_i. All vertices within travel distance r_i from v_i (and possibly others) are tagged and the sequence continues.

Lemma 9.1: The cost of any marking sequence is at most $|V| + 2|V| \ln |V|$ on connected graphs $G = (V, E)$.

Proof: Let $\{v_i, r_i, M_i\}$, for $i = 1, 2, \ldots, m$, be a marking sequence on connected graph $G = (V, E)$. Define $S_t = \{v_i | r_i \geq t, 0 \leq i < m\}$. We first prove that $|S_t| \leq 2|V|/t$. If $d(v_i, v_j) \leq r_i$, then $j \leq i$ because $v_j \notin M_j$ according to Property 1, but $v_j \in M_{i+1}$ according to Property 3, and thus, $v_j \in M_k$ for all $k > i$ according to Property 2. Therefore, no distinct vertices $v_i, v_j \in S_t$ may have $d(v_i, v_j) \leq t$ because otherwise, $j \leq i$ and $i \leq j$, and the vertices could not be distinct. For each $x \in S_t$, consider the ball $B(x)$ of radius $t/2$ around x, i.e., all vertices within travel distance $t/2$ of x. These balls are pairwise disjoint, because a nonempty intersection of $B(v_i)$ and $B(v_j)$ would imply $d(v_i, v_j) \leq t$ by the triangle inequality. Each ball must contain at least $1 + \lfloor t/2 \rfloor$ vertices, since G is connected. Consequently, there can be at most $\lfloor |V|/(1 + \lfloor t/2 \rfloor) \rfloor \leq |V|/(t/2) = 2|V|/t$ such balls and, thus, at most this many vertices in S_t, which implies that $|S_t| \leq 2|V|/t$. Then, the cost of the marking sequence is given as follows:

$$\sum_{i=0}^{m-1} (1 + r_i) = m + \sum_{i=0}^{m-1} r_i = m + \sum_{t=0}^{|V|-1} t(|S_t| - |S_{t+1}|)$$

$$= m + \sum_{t=1}^{|V|-1} |S_t| \leq m + \sum_{t=1}^{|V|-1} 2|V|/t \leq |V| + 2|V| \ln |V|.$$

Note that this is a natural log. This proves the lemma. ■

Corollary 9.2: The worst-case execution time of any greedy-graph traversal algorithm is $O(|V| \log |V|)$ in connected graphs $G = (V, E)$.

Proof: The execution time of any greedy-graph traversal algorithm is at most $|V| + 2|V| \ln |V|$, since this is the cost of the corresponding marking sequence according to Lemma 9.1. ■

Corollary 9.3: The worst-case execution time of greedy localization is $O(n \log n)$ in graphs, including grid graphs, of terrain size n.

Proof: Greedy localization always moves the robot to a closest informative vertex (where the robot makes an observation that decreases the number of its possible vertices). Define a vertex to be tagged iff it is uninformative. Then, greedy localization is a greedy-graph traversal algorithm in the connected component of the graph that contains the robot. Therefore, Corollary 9.2 applies. ■

Previously, it was only known that the execution time of greedy localization is $O(n^{3/2})$ in grid graphs of terrain size n [6]. Our upper bound, on the other hand, is quite close to the lower bound of the previous section. It shows that the execution time of greedy localization cannot be much larger than the terrain size. Thus, the execution time of greedy localization scales well with the terrain size and can be much larger than minimal in some terrains, not because it is large with respect to the terrain size, but because the minimal execution time is exceptionally small in these terrains.

Our analysis is quite general and also applies to mapping. The mapping problem is to determine a map of the terrain. *Greedy mapping* is a mapping algorithm that always moves the robot (with minimal travel distance) to a closest location where it makes an observation that increases its knowledge of the terrain. It has been used on a nomad-class tour-guide robot that offered tours to museum visitors [28] and Super Scouts [29].

Corollary 9.4: The worst-case execution time of greedy mapping is $O(n \log n)$ in graphs, including grid graphs, of terrain size n.

Proof: Greedy mapping always moves the robot to a closest informative vertex (where the robot makes an observation that increases its knowledge of the terrain). Define a vertex to be tagged iff it is uninformative. Then, greedy mapping is a greedy-graph traversal algorithm in the connected component of the graph that contains the robot. Therefore, Corollary 9.2 applies. ■

The same upper bound and proof also hold unchanged for a version of greedy mapping that always moves the robot to a closest unvisited location. Previously, it was only known that the execution time of greedy mapping is $O(n^{3/2})$ in grid graphs of terrain size n [30]. Our upper bound, on the other hand, is quite close to our previous lower bound $\Omega(n \log n / \log \log n)$ [30]. It shows that the execution time of greedy mapping for both tactile and long-distance sensors in grid graphs cannot be much larger than the terrain size and, thus, scales well with the terrain size.

X. Execution Time in Polygonal Regions

So far, we have proved bounds on the execution time of greedy localization in grid graphs as a function of the terrain size. We now explain why we have not been able to prove analogous bounds in polygonal regions. Let P be a polygonal region. We consider possible definitions for its terrain size and explain why each one suffers from unacceptable scaling problems because, in all cases, the ratio of the execution time and the terrain size can be made arbitrarily large. For cases 3) and 4), as shown next, for example, the execution time can be made arbitrarily large for a given terrain size, which means that it is meaningless to express the execution time as a function of the terrain size. Define the terrain size of polygonal region M as follows.

1) By the largest connected area $n_1(M)$, which is similar to the number of traversable vertices of the largest connected component of grid graphs. If a polygonal region is scaled by ϵ, then the execution time changes by factor ϵ, and the terrain size $n_1(M)$ changes by factor ϵ^2.
2) By the length of its encoding $n_2(M)$, where each line segment is encoded by two pairs of integer coordinates. Consider a polygonal region with two unconnected hooking corridors, as used in the proof of Theorem 6.1. The first one runs up for m cells, which is right for two cells and down for two cells. The second one is identical to the

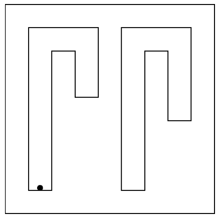

Fig. 7. Gridworld for terrain sizes $n_2(M)$ and $n_3(M)$.

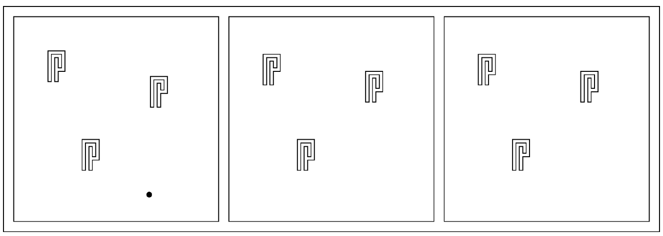

Fig. 8. Gridworld for terrain sizes $n_4(M)$ and $n_5(M)$.

first one, except that it runs down for three cells. Fig. 7 shows an example. If the starting location is far away from the hook in one of the hooking corridors, then the robot needs travel distance $\Theta(m)$ to determine which corridor it is in. Thus, the execution time is $\Theta(m)$, and the terrain size $n_2(M)$ is $\Theta(\log m)$.

3) By the number of line segments of the polygonal region $n_3(M)$. The execution time for the example in the context of $n_2(M)$ is $\Theta(m)$, and the terrain size $n_3(M)$ is $\Theta(1)$.
4) By the length of all line segments in the reachable region $n_4(M)$. There exist instances of the traveling-salesman problem in the unit square with tour length $\Omega(\sqrt{m})$, where m is the number of cities [31]. Pick such an instance and construct a polygonal region by replacing all cities with identical small hooking corridors. Make m copies of the instance, slightly shortening the hook of the hooking corridor of the ith city in the ith copy. Fig. 8 shows an example. Choose the size of the hooking corridors so that the length of all line segments remains constant, as the number m of cities increases, i.e., make each hooking-corridor size $1/m$. Then, a robot needs travel distance $\Omega(\sqrt{m})$ in the worst case to visit all but one hooking corridor and determine which copy it is in. Thus, the execution time is $\Omega(\sqrt{m})$, and the terrain size $n_4(M)$ is $\Theta(1)$.
5) By the square root of the reachable area of its interior $n_5(M)$. The execution time for the example in the context of $n_4(M)$ is $\Omega(\sqrt{m})$, and the terrain size $n_5(M)$ is at most 1.

Thus, several natural definitions of the terrain size of polygonal regions suffer from scaling problems.

XI. Conclusion and Future Work

We showed that it is NP-hard to localize a robot with an execution time that is close to minimal. We proved our result for both tactile (short-distance) and long-distance terrain sensors in both grid graphs and polygonal regions. It, thus, applies to virtually all terrains of interest in robotics. Therefore, it is important to understand the behavior of fast-localization heuristics. We showed that the execution time of greedy localization in grid graphs can be much larger than minimal but cannot be much larger than the terrain size. Thus, the execution time of greedy localization scales well with the terrain size and can be much larger than minimal in some terrains, not because it is large with respect to the terrain size but because the minimal execution time is exceptionally small in these terrains. The upper bound on the worst-case execution time of greedy localization is so general that it applies to greedy mapping as well. Finding results analogous to Theorem 8.1 and Corollary 9.3 for polygonal regions remains as future work, although we have shown in Section X that several natural definitions of the terrain size of polygonal regions suffer from scaling problems. Taking actuator and sensor noise into account, for example, to analyze a generalization of greedy localization that deals with actuator and sensor noise by always moving the robot so that it quickly decreases the entropy of the probability distribution over the possible locations [26], [32] also remains as future work.

Acknowledgment

The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the sponsoring organizations and agencies or the U.S. government.

References

[1] P. Agre and D. Chapman, "Pengi: An implementation of a theory of activity," in *Proc. AAAI Conf. Artif. Intell.*, 1987, pp. 268–271.
[2] I. Nourbakhsh, "Robot information packet," in *Proc. AAAI-96 Spring Symp. Plann. Incomplete Inf. Robot Probl.*, 1996, pp. 86–94.
[3] L. Guibas, R. Motwani, and P. Raghavan, "The robot localization problem in two dimensions," in *Proc. ACM-SIAM Symp. Discr. Algorithms*, 1992, pp. 259–268.
[4] J. O'Kane, "Global localization using odometry," in *Proc. Int. Conf. Robot. Autom.*, 2006, pp. 37–42.
[5] G. Dudek, K. Romanik, and S. Whitesides, "Localizing a robot with minimum travel," in *Proc. ACM-SIAM Symp. Discr. Algorithms*, 1995, pp. 437–446.
[6] C. Tovey and S. Koenig, "Gridworlds as testbeds for planning with incomplete information," in *Proc. AAAI Conf. Artif. Intell.*, 2000, pp. 819–824.
[7] S. Koenig, J. Mitchell, A. Mudgal, and C. Tovey, "A near-tight approximation algorithm for the robot localization problem," *SIAM J. Comput.*, vol. 39, pp. 461–490, 2009.
[8] R. Brown and B. Donald, "Mobile robot self-localization without explicit landmarks," *Algorithmica*, vol. 26, no. 3/4, pp. 515–559, 2000.
[9] M. Aronov, L. Guibas, and L. Zhang, "Visibility queries and maintenance in simple polygons," *Discr. Comput. Geometry*, vol. 27, pp. 461–483, 2002.
[10] A. Zarei and M. Ghodsi, "Query point visibility computation in polygons with holes," *Comput. Geometry*, vol. 39, no. 2, pp. 78–90, 2008.
[11] P. Bose, A. Lubiw, and J. Munro, "Efficient visibility queries in simple polygons," *Comput. Geometry*, vol. 23, no. 3, pp. 313–335, 2002.
[12] S. Gal, *Search Games*. New York: Academic, 1980.
[13] R. Baeza-Yates, J. Culberson, and G. Rawlins, "Searching with uncertainty (extended abstract)," in *Proc. Scand. Workshop Algorithm Theory*, 1998, pp. 176–189.
[14] R. Baeza-Yates, J. Culberson, and G. Rawlins, "Searching in the plane," *Inf. Comput.*, vol. 106, pp. 234–252, 1993.
[15] S. Schuierer, "Greedy robot localization in simple polygons," presented at the Eur. Workshop Comput. Geometry, Würzburg, Germany, 1997.
[16] O. Karch, H. Noltemeier, and T. Wahl, "Robot localization: Theory and implementation," presented at the Eur. Workshop Comput. Geometry, Würzburg, Germany, 1997.

[17] J. Kleinberg, "The localization problem for mobile robots," in *Proc. Symp. Found. Comput. Sci.*, 1994, pp. 521–533.

[18] R. Fleischer, K. Romanik, S. Schuierer, and G. Trippen, "Optimal robot localization in trees," *Inf. Comput.*, vol. 171, no. 2, pp. 224–247, 2002.

[19] H. Gonzalez-Banos and J.-C. Latombe, "Navigation strategies for exploring indoor environments," *Int. J. Robot. Res.*, vol. 21, no. 10/11, pp. 829–848, 2002.

[20] A. C. Murtra, J. M. Tur, and A. Sanfeliu, "Efficient active global localization for mobile robots operating in large and cooperative environments," in *Proc. Int. Conf. Robot. Autom.*, 2008, pp. 2758–2763.

[21] M. Rao, G. Dudek, and S. Whitesides, "Randomized algorithms for minimum distance localization," *Int. J. Robot. Res.*, vol. 26, no. 9, pp. 917–933, 2007.

[22] U. Feige, "A threshold of ln n for approximating set cover," *J. ACM*, vol. 45, no. 4, pp. 634–652, 1998.

[23] I. Nourbakhsh, *Interleaving Planning and Execution for Autonomous Robots*. Norwell, MA: Kluwer, 1997.

[24] M. Genesereth and I. Nourbakhsh, "Time-saving tips for problem solving with incomplete information," in *Proc. AAAI Conf. Artif. Intell.*, 1993, pp. 724–730.

[25] S. Koenig and R. Simmons, "Solving robot navigation problems with initial pose uncertainty using real-time heuristic search," in *Proc. Int. Conf. Artif. Intell. Plann. Syst.*, 1998, pp. 145–153.

[26] D. Fox, W. Burgard, and S. Thrun, "Active Markov localization for mobile robots," *Robot. Auton. Syst.*, vol. 25, pp. 195–207, 1998.

[27] S. Koenig, "Exploring unknown environments with real-time search or reinforcement learning," in *Proc. Neural Inf. Process. Syst.*, 1999, pp. 1003–1009.

[28] S. Thrun, A. Bücken, W. Burgard, D. Fox, T. Fröhlinghaus, D. Hennig, T. Hofmann, M. Krell, and T. Schmidt, "Map learning and high-speed navigation in RHINO," in *Artificial Intelligence Based Mobile Robotics: Case Studies of Successful Robot Systems*, D. Kortenkamp, R. Bonasso, and R. Murphy, Eds. Cambridge, MA: MIT, 1998, pp. 21–52.

[29] L. Romero, E. Morales, and E. Sucar, "An exploration and navigation approach for indoor mobile robots considering sensor's perceptual limitations," in *Proc. Int. Conf. Robot. Autom.*, 2001, pp. 3092–3097.

[30] S. Koenig, C. Tovey, and W. Halliburton, "Greedy mapping of terrain," in *Proc. Int. Conf. Robot. Autom.*, 2001, pp. 3594–3599.

[31] E. Lawler, J. Lenstra, A. R. Kan, and D. Shmoys, *The Traveling Salesman Problem*. New York: Wiley, 1985.

[32] J. Porta, J. Verbeek, and J. Kröse, "Active appearance-based robot localization using stereo vision," *Auton. Robots*, vol. 18, no. 1, pp. 59–80, 2005.

Craig Tovey received the A.B. (*magna cum laude*) degree in applied mathematics from Harvard College, Cambridge, MA, and the M.S. degree in computer science and the Ph.D. degree in operations research from Stanford University, Stanford, CA.

He is currently a Full Professor with the School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, where he is also a Founding Member of the interdisciplinary Algorithms, Combinatorics, and Optimization Ph.D. Program and a Co-Founder and a Co-Director of the Center for Biologically Inspired Design. For more than 20 years, he has been engaged in interdisciplinary applications of operations research methodology. He was a Principal Software Developer with ILOG and was involved in the development cycle for CPLEX 8.0, which is the world's leading optimization engine for linear mixed-integer programming. His algorithms for optimizing electronic circuit board assembly have been used worldwide by Motorola and Ford Electronics. He has authored or coauthored papers in a wide range of high-quality journals and conferences in operations research, computer science, mathematics, economics, biology, political science, and artificial intelligence. He is an Associate Editor of the *Discrete Optimization and Algorithmic Operations Research*.

Prof. Tovey has received the National Science Foundation (NSF) Research Initiation Grant Award, the NSF Presidential Young Investigator Award, the Jacob Wolfowitz Prize, the National Research Council Senior Associateship, and the Georgia Institute of Technology Fellow award.

Sven Koenig received the M.S. degree from the University of California, Berkeley, and the Ph.D. degree in computer science from Carnegie Mellon University, Pittsburgh, PA.

He is currently an Associate Professor of computer science with the University of Southern California, Los Angeles. He was the Editor of several conference proceedings and published papers in various areas of artificial intelligence and robotics. He is an Associate Editor of the *Journal on Advances in Complex Systems* and *Computational Intelligence* and a member of the advisory board and a former Associate Editor of the *Journal of Artificial Intelligence Research*.

Dr. Koenig was a Co-Founder of Robotics: Science and Systems, a highly selective robotics conference, in 2005. He is the Director of the Robotics: Science and Systems Foundation. He was the Conference Co-Chair of the 2002 Symposium on Abstraction, Reformulation, and Approximation (SARA) and the 2004 International Conference on Automated Planning and Scheduling (ICAPS). He was the Program Co-Chair of the 2005 International Joint Conference on Autonomous Agents and Multi-Agent Systems and the 2007 and 2008 Association for the Advancement of Artificial Intelligence Nectar Programs. He is a member of the steering committees of ICAPS and SARA and a former member of the Advisory Committee of the Americas School on Agents and Multiagent Systems. He is a recipient of the Association for Computing Machinery Recognition of Service Award, the National Science Foundation CAREER Award, the IBM Faculty Partnership Award, the Charles Lee Powell Foundation Award, the Raytheon Faculty Fellowship Award, the Mellon Mentoring Award, the Fulbright Fellowship, and the Tong Leong Lim Predoctoral Prize from the University of California.