Guaranteed Coverage with a Blind Unreliable Robot

Jeremy S. Lewis,[1] Daniel A. Feshbach,[2] and Jason M. O'Kane[1]

***Abstract*— We consider the problem of coverage planning for a particular type of very simple mobile robot. The robot must be able to translate in a commanded direction (specified in a global reference frame), with bounded error on the motion direction, until reaching the environment boundary. The objective, for a given environment map, is to generate a sequence of motions that is *guaranteed* to cover as large a portion of that environment as possible, in spite of the severe limits on the robot's sensing and actuation abilities.**

We show how to model the knowledge available to this kind of robot about its own position within the environment, show how to compute the region whose coverage can be guaranteed for a given plan, and characterize regions whose coverage cannot be guaranteed by any plan. We also describe a heuristic algorithm that generates coverage plans for this robot, based on a search across a specially-constructed graph. Simulation results demonstrate the effectiveness of the approach.

I. Introduction

The problem of robotic coverage planning —that is, of designing strategies for robots that ensure that they pass near to every point in the environment— has generated sustained interest from the research community [15], [26]. Solutions to such problems have applications in environmental monitoring [47], cleaning and lawncare [52], humanitarian demining [48], and painting [6]. Most of the existing coverage techniques rely on precise control of the robot's motion. For example, techniques based on the boustrophedon decomposition [16], [50] require the robot to be able to travel accurately in straight lines along the coverage passes, and also to be able to transit precisely between the passes. The primary alternative, realized with great success in the original Roomba [56], is to move with some degree of randomness. In that case, one expects the probability of complete coverage to increase as the robot continues its movement, though any guarantees are only probabilistic.

In contrast, this paper considers a coverage problem in which a robot that is very simple —with no feedback sensing, and with highly error-prone actuation— can nonetheless *guarantee* to cover a certain portion of its environment. Specifically, we consider a robot model with only two movement primitives. First, the robot can rotate in place to face a given direction, though this rotation is subject to some unknown bounded disturbance. Second, the robot can move forward from its current position until reaching the environment boundary. The robot cannot measure the distance traveled (it has no odometer nor clock) nor does it have any other sensors to provide feedback about its motion through the world. Our interest in such simple robot models derives both from a practical desire to limit the complexity and expense of robots deployed for such tasks, but also from a desire to understand the underlying information requirements of important robotic tasks.

Fig. 1. A maze-like environment. Our algorithm generates plans that are guaranteed to cover particular regions of the environment. Results from several runs of the algorithm, showing the region guaranteed to be covered by the plans for error bounds ranging from 0.5 degrees to 4.5 degrees of error on each motion, are shown in corresponding colors: 0.5°, 1°, 1.5°, 2°, 2.5°, 3°, 3.5°, 4°, and 4.5°. Layers are stacked by increasing error because the lower (more precise) layers cover everything the layers above them cover. For example, very little red is visible because very little area covered at 0.5 degrees is not also covered at 1 degree.

We describe an algorithm that computes a sequence of motions for this robot to attempt to cover as much of the environment as possible before returning to its start state. The algorithm must confront the dual challenges of navigation and coverage. Navigation with this robot model can be challenging because the available sensor data is so limited, the robot may easily lose track of its own position; coverage with this robot model can be challenging because if the robot does not know its own position with relatively high accuracy, it cannot be certain of which parts of the environment are being covered.

Figure 1 shows an example of our algorithm's output, in which the differently-colored shaded regions illustrate regions that can be covered by this approach for varying bounds on the amount of rotational error. The idea of the algorithm is to construct a directed graph. Vertices of the graph represent contiguous sets of possible states, represented as

[1]J. S. Lewis and J. M. O'Kane are with the Department of Computer Science and Engineering, University of South Carolina, Columbia, South Carolina, USA.

[2]D. A. Feshbach is with the Department of Computer Science at Haverford College, Haverford, Pennsylvania, USA.

line segments along the boundary, in which the robot might know its true state lies. Edges of the graph correspond to achievable transitions between these segments, labeled with the region that is guaranteed to be covered by that transition. After constructing this graph, the planning algorithm is then a process of identifying edges that (a) would be beneficial to cross because they would cover some new portion of the environment, (b) can be reached from the starting position, and (c) can be returned from.

After a brief review of related work in Section II, this paper makes several new contributions. (i) We introduce, in Section III, a new coverage planning problem, suitable for a robot equipped with only an error-prone compass, a contact sensor, and a map of the environment. (ii) We show, in Section IV, how to model the incomplete knowledge that this kind of robot has about its position within the environment. (iii) We characterize, in Section V, the regions whose coverage can be guaranteed for a given plan. We also characterize regions that cannot be covered by any plan. (iv) We describe, in Section VI, an algorithm that generates state space graphs and searches them to find coverage plans. (v) We present, in Section VII, simulation results demonstrating the effectiveness of the approach. Concluding discussion appears in Section VIII.

II. Related work

A. Coverage

The various flavors of coverage problems have been studied so extensively that a full review is impossible here. Recent research has studied the role of environment decomposition [1], [25], [30], [59], particularly on grids [2], [23], [29], [51]; coordination of multiple robots [7], [33], [34], [36], [49], [50], [61]; and different path types such as spirals [12], [28] or Dubins curves [34], [39], [53], [58], [60]. Alam, Bobadilla, and Shell [5] consider probabilistic coverage of grid cells using weighted random movement with a very similar minimal robot. Our work is unique in guaranteeing coverage of particular regions using a robot with such limited sensing and actuation capacities.

We refer the reader to the surveys by Choset [15] and by Galceran and Carreras [26] for a more complete picture.

B. Minimalism in planning

This work draws inspiration from the significant body of prior work on minimalism in robotics. This work, which builds from pioneering work by Erdmann, Mason, and Goldberg, and others [19], [20], [27], [43], has been applied to problems in manipulation [3], [4], [8], [21], [41], [44], [57], navigation [9], [18], [31], [32], [37], [42], and mapping [13], [14], [35], [45], [55]. The robust coverage work of Bretl and Hutchinson [10] might also be viewed as minimalist, as it forms plans guaranteed to succeed despite motion accuracy limitations. Das, Becker, and Bretl likewise considered coverage problems for robots with uncertainty [17].

Perhaps most closely related is prior work that considers localization [22], [46] and navigation [40] problems for robot models very similar to the one we use here. We build upon those results to show how coverage planning problems can also be solved under this robot model.

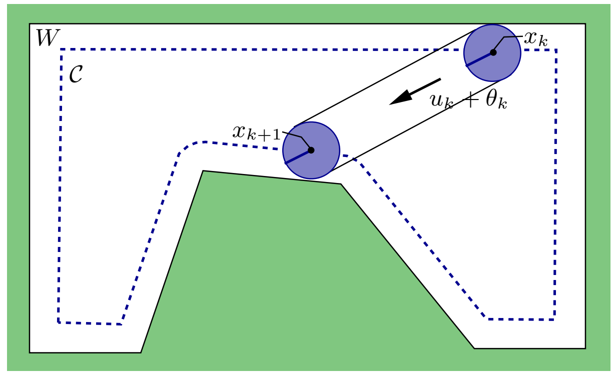

Fig. 2. An illustration of the basic notation. At stage k, the robot moves in direction $u_k + \theta_k$, from x_k to x_{k+1}, covering a portion of the environment W along the way. Both x_k and x_{k-1} are points along the boundary of $\mathcal{C}$. However, the robot does not necessarily know x_k, and certainly does not know θ_k.

III. Problem statement

This section formalizes our robot model and the coverage problem we address in the paper.

A. Robot model

A disk-shaped robot with radius ρ moves through a known, bounded, planar, polygonal environment $W \subseteq \mathbb{R}^2$. Using the center of the robot as its reference point, the configuration space $\mathcal{C}$ is the set of positions within W with distance at least ρ from the boundary of the environment:

$$\mathcal{C} = \{x \in W \mid B(x, \rho) \subset W\}.$$

We follow the usual convention by writing $B(p, r)$ to denote the open ball in $\mathbb{R}^2$ with radius r, centered at p. Note that, though W has a polygonal boundary, the boundary of $\mathcal{C}$ may include both line segments and circular arcs. See Figure 2. Informally, the robot's goal is 'drive over' —that is, to move within distance at most ρ of— as much of W as possible.

We model time as a series of discrete stages $k = 1, 2, \ldots, K$. The robot's state at stage k is denoted $x_k \in \mathcal{C}$. In each stage, the robot selects a movement direction $u_k \in [0, 2\pi]/\!\sim$, in which $\sim$ is an equivalence relation that identifies 0 with 2π. This motion is perturbed by an unknown error $\theta_k \in [-\theta_{\max}, \theta_{\max}]$, in which $\theta_{\max}$ is a known bound on the accuracy of the robot's angular orientation. Because we are interested in guarantees of coverage, we do not assume that any probability model applies to the selection of each θ_k; the disturbances may be selected at random, or adversarially, or through any other mechanism.

From a given state x_k, the robot moves in direction $u_k + \theta_k$. The motion continues until the edge of the robot's body reaches the boundary of W (or, equivalently, until the center of the robot reaches the boundary of $\mathcal{C}$.) The state resulting from from this motion is denoted x_{k+1}, and we denote this state transition function by f, so that

$$x_{k+1} = f(x_k, u_k, \theta_k).$$

The starting state x_1 is assumed to be known.

This robot model could be implemented, for example, with a robot equipped with a noisy compass and a contact sensor, but no way of measuring the distances it travels. An unusual feature of the model is that, because there is no meaningful feedback from any sensors, the robot's strategy can be fully described as a sequence of motion directions. There is no need to consider any branching or looping in plans executed by this robot.

B. Minimalist coverage

We can now consider the coverage problem for this type of robot.

Definition 1: A point $p \in W$ is *covered* by a given sequence of actions $u_1, \ldots, u_K$ and disturbances $\theta_1, \ldots, \theta_K$ if there exist $k \in \{1, \ldots, K\}$ and $\alpha \in [0, 1]$ such that

$$||p - (\alpha x_k + (1 - \alpha) x_{k+1})|| \leq \rho.$$

Note that Definition 1 refers to a specific sequence of disturbances, and recall that the specific disturbance values are unknown to the robot. Thus, we are interested, as the next definition clarifies, in points that we can guarantee are covered, regardless of the specific disturbances in any particular execution.

Definition 2: A point $p \in W$ is *certainly covered* by a given sequence of actions $u_1, \ldots, u_K$ if p is covered by that action sequence under *any* disturbance sequence $\theta_1, \ldots, \theta_K$.

Definition 3: The *certainly covered region*, denoted $\mathrm{CCR}(u_1, \ldots, u_K)$, is the set of points in W that are certainly covered by $u_1, \ldots, u_K$.

The goal is to select actions that certainly cover some desired fraction of the environment. Specifically, the problem is:

> **Given** an environment W, a start state x_1, a robot radius ρ, and the error bound $\theta_{\max}$, **select** a sequence of actions $u_1, \ldots, u_K$ **to maximize** $\mathrm{Area}(\mathrm{CCR}(u_1, \ldots, u_K)) / \mathrm{Area}(W)$.

In the remainder of the paper, we describe a specific heuristic approach to this algorithmic problem.

IV. Safe actions and possible states

Because of the unknown disturbances, as the robot moves through W, it will in general be uncertain of its position. In our approach, we reason about this uncertainty using a worst-case model. That is, we keep track of which states are possible, based on the history, and which are not.

Specifically, we say that a state $x \in \mathcal{C}$ is a *consistent with* a series of actions $u_1, \ldots, u_k$ if there exists some sequence of disturbances $\theta_1, \ldots, \theta_k$, under which the robot's final position x_k is equal to x. In our approach, we follow our own precedent [40] by considering only plans for which the set of states consistent with the action history is a line segment along the boundary of $\mathcal{C}$. We write $\overline{p_k q_k}$ to denote this segment of possible states at stage k. For consistency, we use the naming convention that a positive rotation of the vector $q_k - p_k$ about p_k is into W. When the robot's position happens to be known with certainty (as happens, for example, before the first action is executed) then $p_k = q_k$ and the segment is a single point.

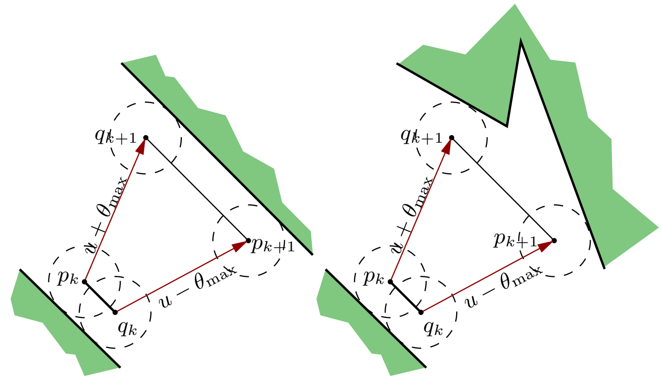

Fig. 3. [left] An example of a safe action. [right] This action is unsafe, because p_{k+1} and q_{k+1} lie on different edges of the boundary.

We say that an action u_k is *safe* from a segment $\overline{p_k q_k}$ along the boundary of $\mathcal{C}$ if the resulting set $\overline{p_{k+1} q_{k+1}}$ of possible states for stage $k+1$ is likewise a segment along the boundary of $\mathcal{C}$. See Figure 3.

Given a segment of possible states $\overline{p_k q_k}$ and the next action u_k, we can use the following procedure to simultaneously test whether u_k is safe from $\overline{p_k q_k}$ and, if so, to compute $\overline{p_{k+1} q_{k+1}}$. First, we define a function $\mathrm{ShootRay}(x, u)$ which returns the first point of intersection with δC from a ray emanating from the point x in the direction u. This is a standard operation from computational geometry [11], [54]. To account for all possible disturbances, $\overline{p_{k+1} q_{k+1}}$ is calculated from $\overline{p_k q_k}$ as follows:

$$\begin{aligned} p_{k+1} &= \mathrm{ShootRay}(q_k, u - \theta_{\max}) \\ q_{k+1} &= \mathrm{ShootRay}(p_k, u + \theta_{\max}) \end{aligned}$$

Next, we test to ensure that the area through which a translating robot may attempt to pass between $\overline{p_k q_k}$ and $\overline{p_{k+1} q_{k+1}}$ is fully within $\mathcal{C}$. A quadrilateral is formed by $p_k q_{k+1} p_{k+1} q_k$ and each edge is checked against $\delta\mathcal{C}$ to ensure no intersections exist. It is also necessary to ensure the quadrilateral contains no vertices of $\mathcal{C}$ to ensure no holes are fully contained within. If the quadrilateral is indeed empty, and if p_{k+1} and q_{k+1} lie on the same segment of the boundary of $\mathcal{C}$, then u_k is safe, and we return $\overline{p_{k+1} q_{k+1}}$. Otherwise, we declare u_k unsafe. (A similar algorithm originally appeared in the context of the navigation problem for a similar robot model [40].)

V. Characterizing the certainly covered region

Before considering the broader question of choosing sequences of actions to cover the environment, we must first characterize how the CCR changes as the robot moves. Specifically, in this section, we present two results, one positive and one negative. First, in Section V-A, we show how to compute the set of states that are certainly covered a given motion of the robot. Then, in Section V-B, we state a condition under which certain points can never be certainly covered by any action sequence.

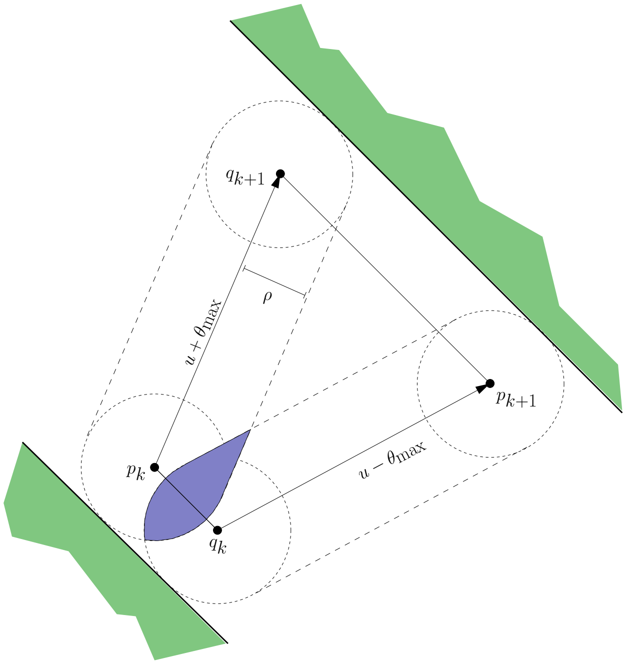

Fig. 4. Computing the CCR for a single safe action, as described in Theorem 1.

A. The region covered by a single movement

Suppose that, at stage k, we know that the robot's state x_k lies within some segment $\overline{p_k q_k}$ along the boundary of $\mathcal{C}$. From there the robot executes action u_k. What can we say about the points, if any, that are certainly covered by this motion?

Definition 2 would appear to require us to reason about each of the infinitely many possible disturbances θ_k to establish that a point is certainly covered. Fortunately, we can show that it is sufficient to consider only the extremal disturbances $-\theta_{\max}$ and $+\theta_{\max}$ instead.

Before stating the result, we need the following preliminary definition.

Definition 4: Given two points p and p' and a radius r the *stadium* between p and p' with radius r, denoted $\mathrm{Stad}(p, q, r)$ is the locus of points within distance r of any point along the segment $\overline{pp'}$.

Visually, the stadium between p and q is a rectangle bisected by the segment $\overline{pq}$, capped by two semicircles of radius r centered at p and q (⬬).

Now we can describe the region covered by a single motion.

***Theorem** 1:* Suppose the robot has executed a sequence of safe actions $u_1, \ldots, u_{k-1}$. Let segment $\overline{p_k q_k} \subset \mathcal{C}$ denote the segment of possible states at stage k. Consider a safe action u_k, and let $\overline{p_{k+1} q_{k+1}}$ denote the segment of possible states resulting from this motion. Then

$$CCR(u_1, \ldots, u_k) = CCR(u_1, \ldots, u_{k-1}) \cup \left(\mathrm{Stad}(p_k, p'_k, \rho) \cap \mathrm{Stad}(q_k, q'_k, \rho) \right). \quad (1)$$

Proof: First, note that for any p, if $p \in CCR(u_1, \ldots, u_{k-1})$, then $p \in CCR(u_1, \ldots, u_k)$. Thus, we need only to consider the points certainly covered by the motion from x_k to x_{k+1}. Let R denote this set. We must show that $R = \mathrm{Stad}(p_k, p'_k, \rho) \cap \mathrm{Stad}(q_k, q'_k, \rho)$.

($\subseteq$) Let $p \in R$. Note that, since p is certainly covered by this motion, it must be specifically covered in the case where $x_k = p_k$ and $\theta_k = \theta_{\max}$. Thus, $p \in \mathrm{Stad}(p_k, p'_k, \rho)$. A similar argument shows that $p \in \mathrm{Stad}(q_k, q'_k, \rho)$.

($\supseteq$) Let $p \in \mathrm{Stad}(p_k, p'_k, \rho) \cap \mathrm{Stad}(q_k, q'_k, \rho)$. We need to show that $p \in R$, which means that for every possible starting point $x_k \in \overline{p_k q_k}$ for the motion, and every possible disturbance $\theta_k \in [-\theta_{\max}, +\theta_{\max}]$, the robot passes within distance ρ of p. The set of locations from which this occurs, for a particular x_k and θ_k, is $\mathrm{Stad}(x_k, f(x_k, u_k, \theta_k), \rho)$. Because this must hold for all x_k and θ_k, we know that if

$$p \in \bigcap_{x_k} \bigcap_{\theta_k} \mathrm{Stad}(x_k, f(x_k, u_k, \theta_k), \rho),$$

then $p \in R$. However, this intersection is fully determined by the two extremal stadia $\mathrm{Stad}(p_k, p'_k, \rho)$ and $\mathrm{Stad}(q_k, q'_k, \rho)$, which are known by construction to contain p. Thus p is also in R. ■

Theorem 1 leads directly to an algorithm for computing the CCR achieved by any motion sequence: Start from the empty set, and iterate over the actions. At each step, compute the union of the previously covered region with the intersection of stadia described in Equation 1.

B. Regions that cannot be covered

We can use a similar idea to the proof of Theorem 1 to rule out certain states from being certainly covered by *any* sequence of motions.

***Theorem** 2:* Given a point $p \in W$, an error bound $\theta_{\max}$, and a robot radius ρ, let q denote the nearest point on the boundary of W to p. If

$$||p - q|| > \rho \frac{\tan \theta_{\max} + 1}{\tan \theta_{\max}}, \quad (2)$$

then p cannot be certainly covered by any motion sequence.

Proof: Theorem 1 characterizes the region certainly covered at each step as the intersection of two stadia. This intersection is largest when the robot begins at a known position (that is, when $p_k = q_k$) and extends the furthest into the interior of W when the motion direction u_k is perpendicular to the environment boundary. Thus, if p can be certainly covered at all, it can be certainly covered starting at x_k and moving directly toward p. It is a simple matter of trigonometry to determine that the most distant point this region has distance $\rho \frac{\tan \theta_{\max}+1}{\tan \theta_{\max}}$ from q. See Figure 5. ■

The intuition is that by imagining the robot at the point nearest to p, with no position uncertainty, we construct the best-case opportunity to include p in the CCR. If p cannot be certainly covered under those ideal conditions, then there is no hope to certainly cover p.

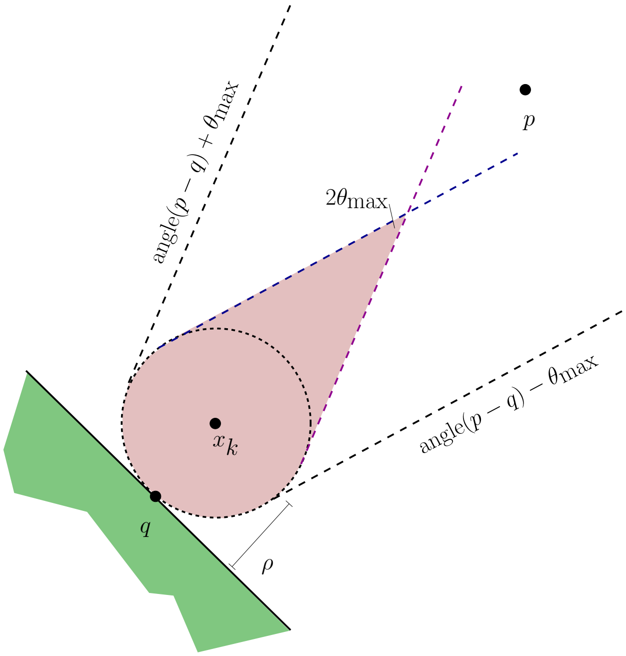

Fig. 5. Point p is too far from the boundary to be certainly covered by any plan under our robot model. See Theorem 2.

VI. Algorithm description

In this section, we describe a method to maximize $\mathrm{Area}(\mathrm{CCR}(u_1,\ldots,u_K))$. The method takes into account the uncertain nature of the robot model's motions and constructs a plan which covers the environment while maintaining a set of states known to contain the robot's true state.

The approach is divided into two parts. The first generates the graph, generating parameter-described layers of line segments on the boundary of W as nodes, and then adding edges where there are safe actions between segments. The second generates the actual action sequence, by determining which edges in this graph may be traversed in a cycle (Algorithm 1).

A. Generating the Graph

Our method begins by generating line segments to be graph nodes, by repeatedly calling a procedure AddLayer$(\mathcal{C}, G, l, o_{\max})$. Each call creates a 'layer' (set) of segments all of a given length l to add as nodes of G. The segments may overlap, and are placed evenly along each sufficiently long (at least as long as l) face of $\mathcal{C}$, with the offset between segment starts (and thus amount of overlap) based on parameter $o_{\max}$. The specific choices of l and $o_{\max}$ are tunable parameters. The face is filled from one end to the other with segments of length l, start points spaced $o \leq o_{\max}$ apart, until the final segment ends at the endpoint of the face. The idea of how o is calculated is to fill the face with segments $o_{\max}$ apart until one includes the end of the face, then move the segments closer together (preserving uniform spacing) until the final segment ends exactly on the end of the face. Specifically, if l_f is the length of the face, then $o = \frac{l_f - l}{\lceil (l_f - l)/o_{\max} \rceil}$. See Figure 6. AddLayer$(\mathcal{C}, G, l, o_{\max})$ is called several times with different values for l and $o_{\max}$ to build up several distinct layers of segments.

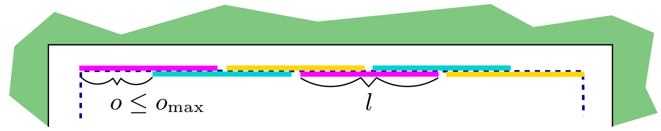

Fig. 6. Placing a layer of segments, along one face of $\mathcal{C}$. The segments are shown vertically offset from the face, and drawn in different colors, to distinguish them visually and show their overlap. Each vertex in the graph corresponds to one such segment.

After the segments are generated and added as nodes to the graph, edges are found by looping through ordered pairs of nodes and adding them where appropriate. A directed edge exists between nodes n_i and n_j if there exists an action u_{ij} under which the robot can be guaranteed a safe translation from n_i to n_j. Each edge is labeled with this u_{ij}, and also with the region the action certainly covers as calculated with Theorem 1.

B. Building cycles in the graph

After the graph is generated, we have a collection of edges, each labeled with a region of the environment that would be covered if the robot were to cross that edge. It might tempting to simply find edges that are reachable from the start position and greedily attempt to cross them. That approach is problematic because the graph is unlikely to be strongly connected; selecting a path that crosses one edge, without regard for the forward connectivity of the resulting node to other locations, may leave the robot stuck in a portion of the graph from which it cannot escape to cover elsewhere.

As a result, our approach to generating the coverage plans is based on generating a series of cyclical 'forays' from a node containing the start position, out through the environment to cover some new territory, and then back to the start node. To begin, we first calculate the shortest paths between all pairs of nodes, using the Floyd-Warshall algorithm [24]. The resulting shortest path matrix has enough information to efficiently determine, for any ordered pair of nodes the graph, whether a directed path exists from the first node to the second node.

We then iterate over the edges of the graph, maintaining a sequence of actions $u_1, \ldots, u_k$ planned to execute, along with $\mathrm{CCR}(u_1, \ldots, u_k)$. For each each e, we check three properties:

1) Is the source node of e reachable from the start node?
2) Is e labeled with a non-empty certainly covered region, which is not already contained in the current CCR?
3) Is the start node reachable from the end node of e?

If all three properties hold, then e represents an opportunity to cover some new portion of the environment. In that case, we generate (using the Floyd-Warshall matrix to determine which states to visit) actions that transit from the start node, across e, and back to the start. For each of the edges crossed by these actions, we include the corresponding certainly covered region in the overall CCR, and remove them from

Algorithm 1 COMPUTECOVERAGEEDGES($\overline{pq}, G$)

$P \leftarrow$ ALLPAIRSSHORTESTPATH(G)
for all edges $e_i \in G$ **do**
  if $P(\overline{pq}, \text{source}[e_i]) \neq \emptyset$
  and $P(\text{target}[e], \overline{pq}) \neq \emptyset$
  and $\text{CCR}[e_i] - \text{CCR}(u_1, \ldots, u_k) \neq \emptyset$ **then**
    Generate actions that travel to e, cross it, and return to the start. Update $CCR(u_1, \ldots, u_k)$ for each of these actions.
  end if
end for

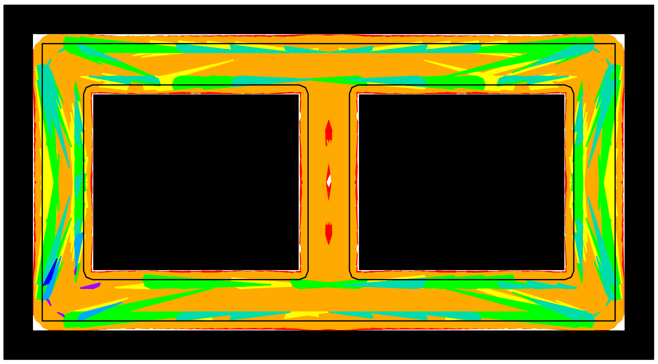

Fig. 7. A simple environment with two large holes. The robot began in the lower left corner and became stuck there as $\theta_{\max}$ reached 3 degrees (■).

consideration in the outer loop. (Note that some of these edges may be labeled with empty coverage regions, for example because they correspond to segments of uncertainty that are too large. This phenomenon explains why the final CCR produced by the algorithm need not be a connected set.)

After each edge has been considered, the planning process terminates. The results is a sequence of actions —the coverage plan itself— that crosses every edge that can be crossed without becoming trapped away from the start vertex, along with the CCR corresponding to that coverage plan.

VII. SIMULATION RESULTS

We implemented our algorithm using C++. We used robot with $\rho = 0.3$, and the layers of segments specified in Table I as our graph nodes. We selected four environments. Figure 1 is a maze-like environment to represent a building or office space. Figure 7 is a simple environment with two large holes separating the convex vertices of the environment. Figure 8 is a large, mostly empty environment to illustrate points far from any edge which the robot cannot be guaranteed to cover, unless $\theta_{\max}$ is very small. Finally, Figure 9 is a more natural cave-like environment.

To characterize the performance of our algorithm as error grows, we conducted several coverage tests in each environment, increasing $\theta_{\max}$ in each iteration. In all executions, the robot's initial position is along the longest possible cycle in the graph. Figures 1, 7, 8, and 9 show the results as color-

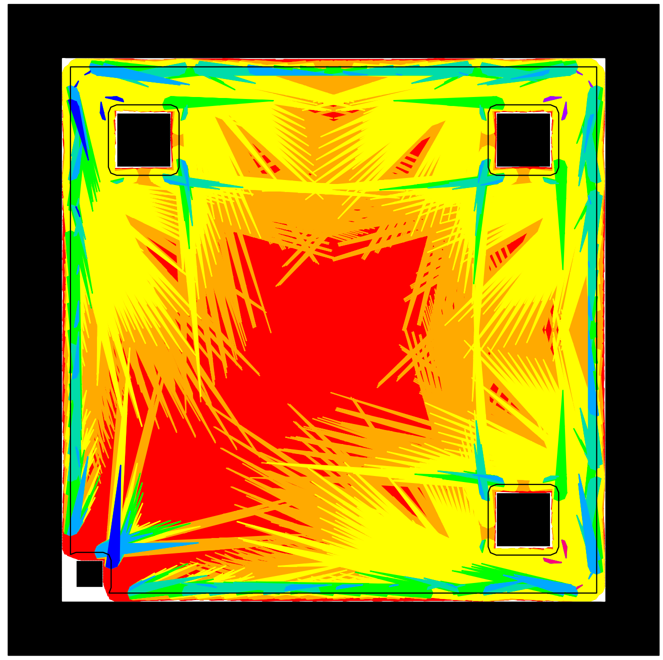

Fig. 8. An environment with a relatively large open middle. As $\theta_{\max}$ increased beyond 0.5 degrees (■), the ability to cover large sections of that area was lost.

TABLE I
LAYERS OF SEGMENTS USED IN SIMULATION

l (length)	$o_{\max}$ (offset)
3	2
2	1
1.5	1
1	0.5
0.5	0.375
0.25	0.1

coded "heat maps" to illustrate the regions that the plans guarantee to cover, using differing colors for each value of $\theta_{\max}$: ■ 0.5°, ■ 1°, ■ 1.5°, ■ 2°, ■ 2.5°, ■ 3°, ■ 3.5°, ■ 4°, and ■ 4.5°. The colors of the heat map are stacked in order of increasing $\theta_{\max}$ because the lower layers cover all the area of all layers above them. Figure 10 shows the area of the certainly covered region, relative to the areas of the environment, for each of these tests. In all cases except the environment in Figure 8, the algorithm achieved close to 100% coverage through $\theta_{\max} = 1$ degrees.

VIII. CONCLUSION

This paper introduced a minimalist coverage problem for an extremely simple class of mobile robots, and showed that even a robot with only an unreliable compass and a contact sensor can still be used to generate plans that are certain to cover significant portions of its environments, in spite of the uncertainty inherent in its motions. However, we have also left a number of stones unturned.

It may be helpful to augment how the graph represents the state space. In particular, it seems likely to be beneficial to include the vertex nodes and the corner-finding subplans

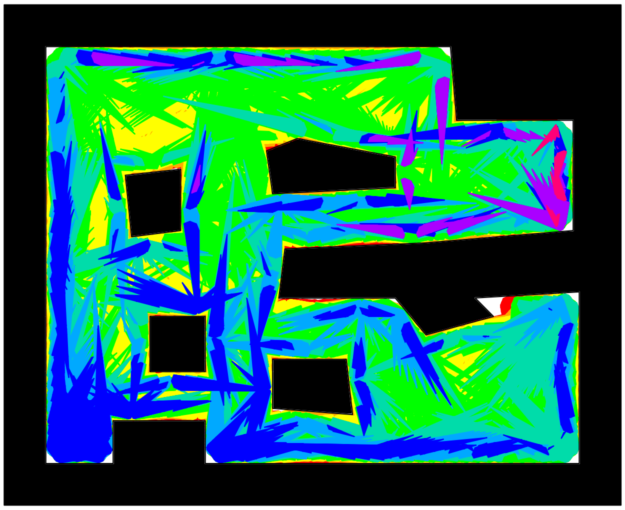

Fig. 9. A more natural cave-like environment. As error grew, the robot retained the ability to navigate around most of the environment until $\theta_{\max}$ reached 4 degrees (■), but lost the ability to cover the more spacious open areas past 2 degrees (■). This environment demonstrates the method's ability to deal with non-uniform features.

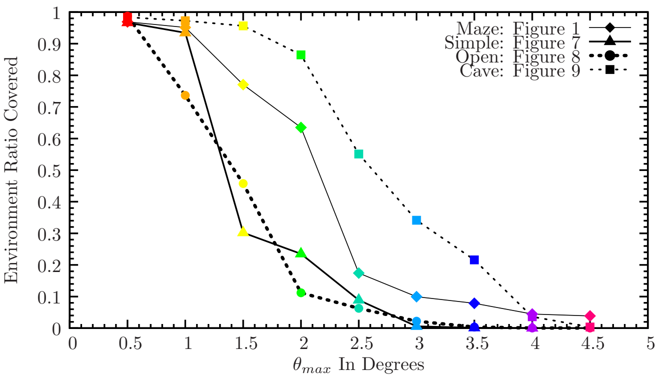

Fig. 10. A plot comparing the ratio of each environment the algorithm can guarantee to cover as accuracy degrades when $\theta_{\max}$ is allowed to grow.

from our earlier work on navigation [40]. Including these elements in the graph will provide the robot with opportunities to re-localize itself at certain points of the environment. This decrease in position uncertainty may generate opportunities to cover some regions that cannot be covered by the current method.

Of perhaps the greatest importance is to improve the mechanism by which the graph is generated. The current approach requires parameter tuning to find segment layers that work well for an environment, and leaves open the possibility that adding more particular segments could increase coverage, either by making edges which cover additional area or by enabling more cyclical navigation. Instead, we may be able to generate the graph in a manner aware of what it can cover and what it can navigate to and from. If we can identify an uncovered but coverable (as per Section V-B) region, generate a pair of segments with a safe action that would certainly cover it, and navigate to and from that edge adding new graph nodes as necessary, then we could generate the entire graph by iterating over the uncovered but coverable regions.

Finally, we focused in this paper solely on the feasibility of coverage plans for our robot model, to the exclusion of optimality concerns. One approach to generating short coverage paths would be to model the problem as a rural Chinese postman problem, a known NP-hard problem, in which the objective is to find the shortest path that crosses each of a selection of edges in a weighted directed graph [38].

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grants No. 1526862 and 1659514.

References

[1] E. U. Acar and H. Choset, "Sensor-based coverage of unknown environments: Incremental construction of morse decompositions," *The International Journal of Robotics Research*, vol. 21, no. 4, pp. 345–366, April 2002.

[2] N. Agmon *et al.*, "The giving tree: constructing trees for efficient offline and online multi-robot coverage," *Annals of Mathematics and AI*, vol. 52(2-4), pp. 143–168, 2008.

[3] S. Akella, W. Huang, K. M. Lynch, and M. T. Mason, "Parts feeding on a conveyor with a one joint robot," *Algorthmica*, vol. 26, no. 3, pp. 313–344, Mar. 2000.

[4] S. Akella and M. Mason, "Posing polygonal objects in the plane by pushing," *International Journal of Robotics Research*, vol. 17, no. 1, pp. 70–88, Jan. 1998.

[5] T. Alam, L. Bobadilla, and D. A. Shell, "Minimalist robot navigation and coverage using a dynamical system approach," in *2017 First IEEE International Conference on Robotic Computing (IRC)*, April 2017, pp. 249–256.

[6] P. N. Atkar, D. C. Conner, A. Greenfield, H. Choset, and A. A. Rizzi, "Uniform coverage of simple surfaces embedded in $\mathbb{R}^3$ for auto-body painting," in *Proc. Workshop on the Algorithmic Foundations of Robotics*, 2004.

[7] G. S. C. Avellar, G. A. S. Pereira, L. C. A. Pimenta, and P. Iscold, "Multi-UAV Routing for Area Coverage and Remote Sensing with Minimum Time," *Sensors*, vol. 15(11), p. 27783, 2015.

[8] R.-P. Berretty, K. Goldberg, M. Overmars, and F. V. der Stappen, "Trap design for vibratory part feeders," *International Journal of Robotics Research*, vol. 20, no. 11, Nov. 2001.

[9] A. Blum, P. Raghavan, and B. Schieber, "Navigating in unfamiliar geometric terrain," *SIAM Journal on Computing*, vol. 26, no. 1, pp. 110–137, 1997.

[10] T. Bretl and S. Hutchinson, "Robust coverage by a mobile robot of a planar workspace," in *Proc. IEEE International Conference on Robotics and Automation*, 2013.

[11] B. Chazelle and L. G. Guibas, "Visibility and intersection problems in plane geometry," *Discrete and Computational Geometry*, vol. 4, pp. 551–589, 1989.

[12] Y.-H. Choi, T.-K. Lee, S.-H. Baek, and S.-Y. Oh, "Online complete coverage path planning for mobile robots based on linked spiral paths using constrained inverse distance transform," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009.

[13] H. Choset and J. Burdick, "Sensor based motion planning: Incremental construction of the hierarchical generalized Voronoi graph," *International Journal of Robotics Research*, vol. 19, no. 2, pp. 126–148, 2000.

[14] ——, "Sensor based motion planning: The hierarchical generalized Voronoi graph," *International Journal of Robotics Research*, vol. 19, no. 2, pp. 96–125, 2000.

[15] H. Choset, "Coverage for robotics - a survey of recent results," *Annals of Mathematics and Artificial Intelligence*, vol. 31, pp. 113–126, 2001.

[16] H. Choset and P. Pignon, "Coverage path planning: The boustrophedon decomposition," in *Proc. International Conference on Field and Service Robotics*, 1997, pp. 3–91.

[17] C. Das, A. Becker, and T. Bretl, "Probably approximately correct coverage for robots with uncertainty," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011.

[18] X. Deng, T. Kameda, and C. H. Papadimitriou, "How to learn an unknown environment I: The rectilinear case," *Journal of the ACM*, vol. 45, no. 2, pp. 215–245, 1998.

[19] M. Erdmann and M. T. Mason, "An exploration of sensorless manipulation," *IEEE Transactions on Robotics and Automation*, vol. 4, no. 4, pp. 369–379, Aug. 1988.
[20] M. A. Erdmann, "Using backprojections for fine motion planning with uncertainty," *International Journal of Robotics Research*, vol. 5, no. 1, pp. 19–45, 1986.
[21] ——, "Understanding action and sensing by designing action-based sensors," *International Journal of Robotics Research*, vol. 14, no. 5, pp. 483–509, 1995.
[22] L. Erickson, J. Knuth, J. M. O'Kane, and S. M. LaValle, "Probabilistic localization with a blind robot," in *Proc. IEEE International Conference on Robotics and Automation*, 2008.
[23] P. Fazli, A. Davoodi, P. Pasquier, and A. Mackworth, "Complete and robust cooperative robot area coverage with limited range," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 5577–5582.
[24] R. W. Floyd, "Algorithm 97: Shortest path," *Commun. ACM*, vol. 5, no. 6, pp. 345–, June 1962. [Online]. Available: http://doi.acm.org/10.1145/367766.368168
[25] Y. Gabriely and E. Rimon, "Spiral-stc: an on-line coverage algorithm of grid environments by a mobile robot," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2002.
[26] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1258–1276, 2013.
[27] K. Y. Goldberg, "Orienting polygonal parts without sensors," *Algorthmica*, vol. 10, pp. 201–225, 1993.
[28] E. Gonzalez, O. Alvarez, Y. Diaz, C. Parra, and C. Bustacara, "BSA: a complete coverage algorithm," in *Proc. IEEE International Conference on Robotics and Automation*, 2005.
[29] N. Hazon and G. Kaminka, "Redundancy, efficiency and robustness in multi-robot coverage," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2005, pp. 735–741.
[30] W. Huang, "Optimal line-sweep-based decompositions for coverage algorithms," in *Proc. the IEEE Int. Conf. on Robotics and Automation*, vol. 1, 2001, pp. 27 – 32.
[31] I. Kamon and E. Rivlin, "Sensory-based motion planning with global proofs," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 6, pp. 814–822, Dec. 1997.
[32] I. Kamon, E. Rivlin, and E. Rimon, "Range-sensor based navigation in three dimensions," in *Proc. IEEE International Conference on Robotics and Automation*, 1999.
[33] N. Karapetyan, K. Benson, C. McKinney, P. Taslakian, and I. Rekleitis, "Efficient multi-robot coverage of a known environment," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vancouver, BC, Canada, Sept. 2017.
[34] N. Karapetyan, J. Moulton, J. S. Lewis, A. Q. Li, J. M. O'Kane, and I. Rekleitis, "Multi-robot dubins coverage with autonomous surface vehicles," in *Proc. IEEE International Conference on Robotics and Automation*, 2018, to appear.
[35] M. Katsev, A. Yershova, B. Tovar, R. Ghrist, and S. M. LaValle, "Mapping and pursuit-evasion strategies for a simple wall-following robot," *IEEE Transactions on Robotics*, vol. 27, no. 1, pp. 113–128, 2011.
[36] C. S. Kong, A. P. New, and I. Rekleitis, "Distributed coverage with multi-robot system," in *Proc. IEEE International Conference on Robotics and Automation*, 2006.
[37] A. Lazanas and J. C. Latombe, "Landmark-based robot navigation," in *Proc. National Conference on Artificial Intelligence (AAAI)*, 1992.
[38] J. K. Lenstra and A. Rinnooy Kan, "Complexity of vehicle routing and scheduling problems," *Networks*, vol. 11, no. 2, pp. 221–227, 1981.
[39] J. S. Lewis, W. Edwards, K. Benson, I. Rekleitis, and J. M. O'Kane, "Semi-boustrophedon coverage with a dubins vehicle," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017.
[40] J. S. Lewis and J. M. O'Kane, "Planning for provably reliable navigation using an unreliable, nearly sensorless robot," *International Journal of Robotics Research*, vol. 32, no. 11, pp. 1339–1354, September 2013.
[41] T. Lozano-Pérez, M. T. Mason, and R. H. Taylor, "Automatic synthesis of fine-motion strategies for robots," *International Journal of Robotics Research*, vol. 3, no. 1, pp. 3–24, 1984.
[42] V. J. Lumelsky and S. Tiwari, "An algorithm for maze searching with azimuth input," in *Proc. IEEE International Conference on Robotics and Automation*, 1994, pp. 111–116.
[43] M. Mason, "Kicking the sensing habit," *AI Magazine*, vol. 14, no. 1, pp. 58–59, 1993.
[44] M. Moll and M. Erdmann, "Manipulation of pose distributions," *International Journal of Robotics Research*, vol. 21, no. 3, pp. 277–292, 2002.
[45] C. Ó. Dúnlaing and C. K. Yap, "A retraction method for planning the motion of a disc," *Journal of Algorithms*, vol. 6, pp. 104–111, 1982.
[46] J. M. O'Kane and S. M. LaValle, "Localization with limited sensing," *IEEE Transactions on Robotics*, vol. 23, pp. 704–716, Aug. 2007.
[47] J. M. Palacios-Gasós, Z. Talebpour, E. Montijano, C. Sagüés, and A. Martinoli, "Optimal path planning and coverage control for multi-robot persistent coverage in environments with obstacles," in *Proc. IEEE International Conference on Robotics and Automation*. IEEE, 2017, pp. 1321–1327.
[48] M. Y. Rachkov, L. Marques, and A. T. de Almeida, "Multisensor demining robot," *Autonomous robots*, vol. 18, no. 3, pp. 275–291, 2005.
[49] I. Rekleitis, V. Lee-Shue, A. P. New, and H. Choset, "Limited communication, multi-robot team based coverage," in *Proc. IEEE International Conference on Robotics and Automation*, 2004.
[50] I. Rekleitis, A. New, E. S. Rankin, and H. Choset, "Efficient boustrophedon multi-robot coverage: an algorithmic approach," *Annals of Mathematics and AI*, vol. 52(2-4), pp. 109–142, 2008.
[51] A. Renzaglia, L. Doitsidis, S. A. Chatzichristofis, A. Martinelli, and E. B. Kosmatopoulos, "Distributed multi-robot coverage using micro aerial vehicles," in *Mediterrean Conference on Control Automation (MED)*, June 2013, pp. 963–968.
[52] H. Sahin and L. Guvenc, "Household robotics: autonomous devices for vacuuming and lawn mowing," *IEEE Control Systems*, vol. 27, no. 2, pp. 20–96, 2007.
[53] K. Savla, F. Bullo, and E. Frazzoli, "The coverage problem for loitering dubins vehicles," in *Decision and Control, 2007 46th IEEE Conference on*, Dec 2007, pp. 1398–1403.
[54] L. Szirmay-Kalos and G. Marton, "Worst-case versus average case complexity of ray-shooting," *Computing*, vol. 61(2), no. 2, pp. 103–131, 1998.
[55] B. Tovar, L. Guilamo, and S. M. LaValle, "Gap Navigation Trees: Minimal representation for visibility-based tasks," in *Proc. Workshop on the Algorithmic Foundations of Robotics*, 2004.
[56] B. Tribelhorn and Z. Dodds, "Evaluating the Roomba: A low-cost, ubiquitous platform for robotics research and education," in *Proc. IEEE International Conference on Robotics and Automation*, 2007.
[57] A. F. van der Stappen, R.-P. Berretty, K. Goldberg, and M. H. Overmars, "Geometry and part feeding," in *Sensor Based Intelligent Robots*, 2000, pp. 259–281.
[58] A. Xu, C. Viriyasuthee, and I. Rekleitis, "Efficient complete coverage of a known arbitrary environment with applications to aerial operations," *Autonomous Robots*, vol. 36, no. 4, pp. 365–381, 2014.
[59] Z. Yao, "Finding efficient robot path for the complete coverage of a known space," in *Proc. IEEE International Conference on Robotics and Automation*, 2006.
[60] X. Yu, T. A. Roppel, and J. Y. Hung, "An optimization approach for planning robotic field coverage," in *Proc. Annual Conference of the IEEE Industrial Electronics Society*, 2015.
[61] X. Zheng, S. Jain, S. Koenig, and D. Kempe, "Multi-robot forest coverage," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005.