Proceedings of the 2001 IEEE
International Conference on Robotics & Automation
Seoul, Korea • May 21-26, 2001

Optimal Line-sweep-based Decompositions for Coverage Algorithms

Wesley H. Huang
Department of Computer Science
Rensselaer Polytechnic Institute
Troy, New York 12180

Abstract

Robotic coverage is the problem of moving a sensor or actuator over all points in a given region. Ultimately, we want a coverage path that minimizes some cost such as time. We take the approach of decomposing the coverage region into subregions, selecting a sequence of those subregions, and then generating a path that covers each subregion in turn. In this paper, we focus on generating decompositions based upon the planar line sweep.

After a general overview of the coverage problem, we describe how our assumptions lead to the optimality criterion of minimizing the sum of subregion altitudes (which are measured relative to the sweep direction assigned to that subregion). For a line-sweep decomposition, the sweep direction is the same for all subregions. We describe how to find the optimal sweep direction for convex polygonal worlds.

We then introduce the minimal sum of altitudes (MSA) decomposition in which we may assign a different sweep direction to each subregion. This decomposition is better for generating an optimal coverage path. We describe a method based on multiple line sweeps and dynamic programming to generate the MSA decomposition.

1 Introduction

Coverage algorithms have received much attention in the past several years because of demining operations in several parts of the world. Landmines must be detected and removed in order to make these areas safe for public activity, and we would like to use robots to carry out the demining. Such a robot must pass a specialized sensor over all points in an area. These robots should be guaranteed to cover the entire area and should perform this task efficiently.

Coverage algorithms also have much broader applicability. Like demining, some applications require that a sensor be passed over all points in a given area. These applications include mapping (creating image mosaics), inspection, and search and rescue operations. Other applications require some sort of actuator to be passed over a given area: spraying coatings (such as paint), vacuum cleaning, lawnmowing, and various agricultural tasks. An application such as snow removal requires not only coverage but consideration of all the snow that is collected!

There have been several coverage algorithms published in the robotics literature, including online and offline algorithms, and single and multiple robot algorithms; however, none address the cost of the path generated to cover the given area.

Efficiency is important in many coverage applications. Time is critical in search and rescue operations, and in industrial applications, even a small improvement in efficiency can result in large cost savings through improved cycle times or reduced material use.

A coverage algorithm must generate what we will call a *coverage path*, i.e. a detailed sequence of motion commands for a robot that sweep the given sensor or actuator over a specified region. An optimal coverage algorithm would return a coverage path that minimized, for example, the time required to execute that path.

Several existing algorithms take the following basic approach to generating a coverage path: the region to be covered is decomposed into subregions, a traveling-salesman algorithm is applied to generate a sequence of subregions to visit, and a coverage path is generated from this sequence that covers each subregion in turn. These algorithms all use a single line sweep in order to decompose the coverage region into subregions, and these subregions are individually covered using a back and forth motion in rows perpendicular to the sweep direction. All subregions use the same sweep direction.

After finishing one row, the robot must turn around to start the next row, and we claim that minimizing the number of these turns is the most important factor in an efficient solution. The number of turns is directly related to the altitude of the subregion (measured along the sweep direction), so our optimality criterion is to minimize the sum of subregion altitudes.

In Section 3, we show that the optimal line sweep decomposition for convex polygonal worlds is generated by sweeping a line parallel to one of the boundary edges. We have also shown this result holds for nonconvex polygonal worlds.

0-7803-6475-9/01/$10.00© 2001 IEEE

In Section 4, we show that by allowing different sweep directions to be assigned to each subregion of a decomposition, we can produce a lower sum of subregion altitudes and thus a cheaper coverage path. We propose an algorithm to solve this minimal sum of altitudes (MSA) decomposition problem which uses multiple line sweeps and dynamic programming.

1.1 Previous work

Two similar and relatively recent algorithms for coverage are by Choset and Pignon [2] and Hert *et al.* [6].

Choset and Pignon describe an offline planning algorithm for polygonal worlds which explicitly performs a line sweep decomposition (the "Boustrophedon" decomposition) and creates a sequence of subregions (cells) using an heuristic traveling-salesman algorithm. This work includes experiments on a synchro-drive mobile robot. Hert *et al.* describe an online algorithm for nonpolygonal worlds which implicitly uses a line sweep decomposition and an heuristic Traveling Salesman algorithm. This work is described in the context of an autonomous underwater vehicle that creates an image mosaic of the ocean floor.

Schmidt and Hofner [9] describe a floor cleaning robot which has nonholonomic constraints. They use an offline planning algorithm to generate a coverage path based on a line sweep decomposition. A vocabulary of "basic motion macros" are used to maneuver the robot (i.e. for the portions of the coverage path that are not straight lines).

Kurabayashi *et al.* [8] describe an offline algorithm for planning coverage paths for multiple robots. It appears to generate a single coverage path, based on both "direction parallel" and "contour-parallel" motion. Zelinsky *et al.* [10] describes a grid-based coverage algorithm.

More recent results include: Gabriely and Rimon [5], who formulated a coverage algorithm based on traveling about the perimeter of a minimum spanning tree that fills the coverage region; Butler *et al.* [1], who created a distributed algorithm for multiple robots to cover an unknown rectilinear environment; and Choset *et al.* [3] who have extended the Boustrophedon decomposition to higher dimension Euclidean spaces.

2 The Coverage problem

We make the following assumptions about the three elements that describe a coverage problem:

- *coverage region* — the region to be covered is planar (or can be embedded in the plane), is connected, and is defined by an outer perimeter and holes its interior. In this paper we will assume that both the perimeter and holes are polygonal.

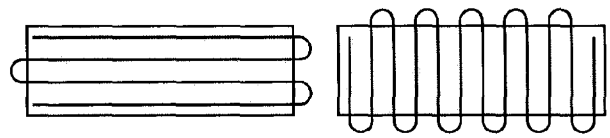

Figure 1: The number of turns is the main factor in the cost difference of covering a region along different sweep directions.

- *robot* — may have nonholonomic constraints, and the shape of the robot can be unrelated (in shape and size) to the sensor/actuator pattern. The starting or ending position of the robot may be specified, or we may insist that the robot end in the same place it starts.

- *sensor/actuator pattern* — the sensor or actuator has a one or two dimensional "coverage" pattern which sweeps out a two dimensional area as the robot moves. Common sensor/actuator patterns include a circle (for radially limited sensors), a rectangle (e.g. a video camera), a line (snowplow). We assume the sensor/actuator pattern does not move relative to the robot.

A coverage algorithm must return:

- *coverage path* — a detailed sequence of motion commands for the robot.

In the rest of this section, we discuss several issues regarding the general coverage problem and describe the assumptions upon which the remainder of the paper is based.

2.1 Optimal coverage

Our approach to the coverage problem is to decompose the coverage region, determine a sequence of subregions, and generate a coverage path that covers each region and then moves on to the next. We seek an optimal solution in this class of solutions.

The planar line sweep, upon which our decompositions are based, divides the coverage region into monotone subregions. These subregions can be easily and efficiently covered by back and forth motion along rows perpendicular to the sweep direction.

We reason that the time to cover a subregion in this manner consists of the time to travel along the rows plus the time to turn around at the end of the rows. Covering a subregion for a different sweep direction results in rows of approximately the same total length; however, there can be a large difference in the number of turns required as illustrated in Figure 1. Furthermore, turns take a significant amount of time: the robot must slow down, make the turn, and then accelerate.

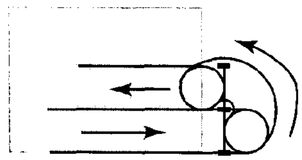

Figure 2: A lawnmower-like robot must drive outside the boundary in order to turn around efficiently.

We therefore wish to minimize the number of turns, and this is proportional to the altitude of the subregion measured along the sweep direction.

An additional cost we have not yet addressed is that of traveling from one subregion to another. In selecting a sequence of subregions, we can take into account the cost of traveling from one subregion to another, but in general, the decomposition of the coverage region cannot be independent of choosing a sequence of subregions to visit and generating a coverage path from that sequence.

We shall put these concerns aside for now by assuming that any gain from choosing a good decomposition is much larger than the variation in the total cost of traveling from subregion to subregion.

Under these assumptions, a decomposition that minimizes the sum of the subregion altitudes (as measured along the sweep direction) will produce an optimal coverage path. This is the problem we address in this paper, first for when the sweep direction must be the same for all subregions and then for when the sweep direction may be different in each subregion.

2.2 Boundaries & obstacles

We differentiate between the boundaries of the coverage region and obstacles within or outside the region (i.e. areas in which the robot cannot travel). Sometimes the two may be coincident, and sometimes they may be independent. For example, when spray painting, a "mask" may define the boundary of the coverage region, but it is permissible to spray beyond this boundary. Alternatively, when mowing a lawn, it is not acceptable to mow over any flowers surrounding the lawn.

A combination of the boundaries, obstacles, sensor/actuator pattern, and the robot may make it technically impossible for a robot to cover a region. For example, a circular robot cannot reach all the way into a convex corner when the boundary and "boundary obstacles" are coincident.

These observations lead us to the following assumption: the coverage region given as input to our algorithm has sufficient room between the boundary and any obstacle (beyond the perimeter or inside a hole) to turn around. This can be accomplished by assuming that the first time the robot encounters a hole or the perimeter, it makes one or more complete circuits around the boundary to create this "buffer zone."

This assumption also helps us deal with nonholonomic constraints of a robot platform. Figure 2 illustrates a simple 180 degree turn for a lawnmower-like robot (differential drive with a circular sensor/actuator pattern in front of the wheels).

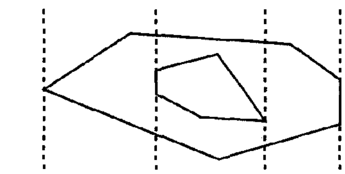

Figure 3: Illustration of the four main types of events in a line sweep from left to right.

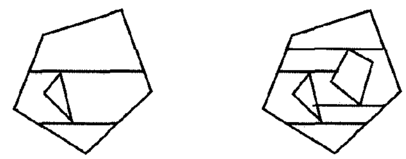

Figure 4: Introducing convex holes increases the subregion altitude sum by the altitude of the holes.

3 Optimal line sweep decompositions

Current coverage algorithms can be improved by determining the optimal sweep direction for a planar line sweep decomposition. We address this problem for convex worlds and then briefly discuss nonconvex worlds.

3.1 Convex worlds

A planar line sweep decomposes a region into monotone subregions by adding a diving line at certain "events" as the sweep line moves across the region. (A subregion is monotone with respect to a sweep direction if a line perpendicular to the sweep direction intersects the region to form a connected set of points.) Figure 3 shows the four types of events in a line sweep. For convex worlds (i.e. where the perimeter and the holes are convex polygons), where will be one OPEN event, one CLOSE event for the perimeter, and there will be one SPLIT and one MERGE event for every hole. It is easy to see, as illustrated in Figure 4, that the sum of subregion altitudes is simply the altitude of the perimeter plus the altitude of all holes.

To determine the optimal sweep direction, we can express the sum of subregion altitudes as a function of sweep

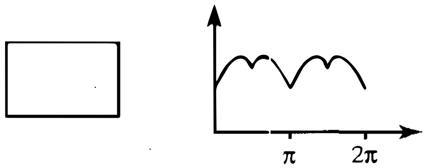

Figure 5: An example diameter function.

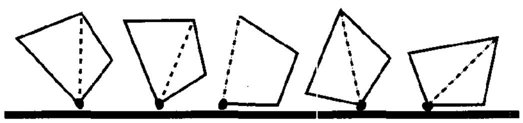

Figure 6: Creating the diameter function by rolling the polygon. The dotted line represents the chord used to determine the diameter or altitude of the polygon.

direction and then minimize this expression.

3.1.1 Altitude sum in terms of diameter functions

We can use the *diameter function* $d(\theta)$ to describe the altitude of the polygon along the sweep direction. For a given angle θ, the diameter of a polygon is determined by rotating the polygon by $-\theta$ and measuring the height difference between its highest and lowest point. An example diameter function is shown in Figure 5. The altitude of a polygon for a sweep direction at an orientation of α is $d(\alpha - \frac{\pi}{2})$.

We can then express the sum of subregion altitudes as:

$$S(\theta) = d_P(\theta) + \sum_i d_{H_i}(\theta) \tag{1}$$

where $d_P(\theta)$ is the diameter function of the perimeter, and $d_{H_i}(\theta)$ is the diameter function of hole i. The optimal decomposition is then determined by the the sweep direction $\alpha = \theta + \frac{\pi}{2}$ that minimizes S.

3.1.2 Form of diameter functions

The form of a diameter function can be understood by considering its height as it rolls along a flat surface. Assume we start with one edge resting on the surface. As illustrated in Figure 6, we can draw a "chord from the pivot vertex to another vertex of the polygon, and the height of the polygon will be determined by this vertex. Whenever the polygon has rolled on to the next side or when an edge at the top of the polygon becomes parallel to the surface, we will change to a different chord (from a different pivot vertex or to a different top vertex). Therefore, a diameter function (for an n sided polygon) has the following form:

$$d(\theta) = \begin{cases} k_1 \sin(\theta + \phi_1) & \theta \in [\theta_0, \theta_1) \\ k_2 \sin(\theta + \phi_2) & \theta \in [\theta_1, \theta_2) \\ \vdots & \\ k_{2n} \sin(\theta + \phi_{2n}) & \theta \in [\theta_{(2n-1)}, \theta_{2n}) \end{cases} \tag{2}$$

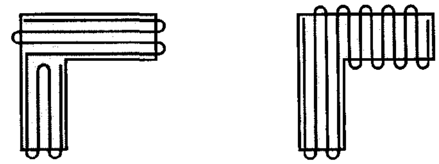

Figure 7: A simple example where assigning different sweep directions to subregions produces a better coverage path, i.e. one with fewer turns.

where $\theta_0 = 0$ and $\theta_{2n} = 2\pi$. The diameter function is piecewise sinusoidal; its "breakpoints" θ_i occur when an edge of the rotated polygon is parallel to the horizontal. This corresponds to when the sweep direction is perpendicular to an edge of the polygon. (The sweep line and the rows for covering the polygon would then be horizontal.)

Note that the diameter function only draws from the sine curve in the interval $\theta \in (0, \pi)$. Therefore, $d'' < 0$ everywhere except at the breakpoints.

3.1.3 Minimizing the altitude sum

The minimum of the function $S(\theta)$ must lie either at a critical point ($S'(\theta) = 0$) or at a breakpoint of one of its constituent diameter functions.

However, for any critical point in between breakpoints:

$$S''(\theta) = d''_P(\theta) + \sum_i d''_{H_i}(\theta) < 0 \tag{3}$$

which means that it corresponds to a maximum! Therefore, the minimum must lie at a breakpoint of one of the component diameter functions. Since these breakpoints correspond to when the sweep direction is perpendicular to an edge of a hole or the perimeter, the minimum can be determined by testing each of these sweep directions.

3.2 Nonconvex worlds

With a nonconvex perimeter and nonconvex holes, a planar line sweep still places dividing lines at SPLIT and MERGE events, but now the perimeter and any obstacle can both produce any of the four types of events, and the sum of subregion altitudes is greater than the sum of diameters of the holes and the perimeter.

We have shown that the same result for convex worlds also holds for nonconvex worlds — the line sweep which minimizes the sum of subregion altitudes is along a sweep direction perpendicular to one of the sides of the perimeter or of a hole. For more details, see Huang [7].

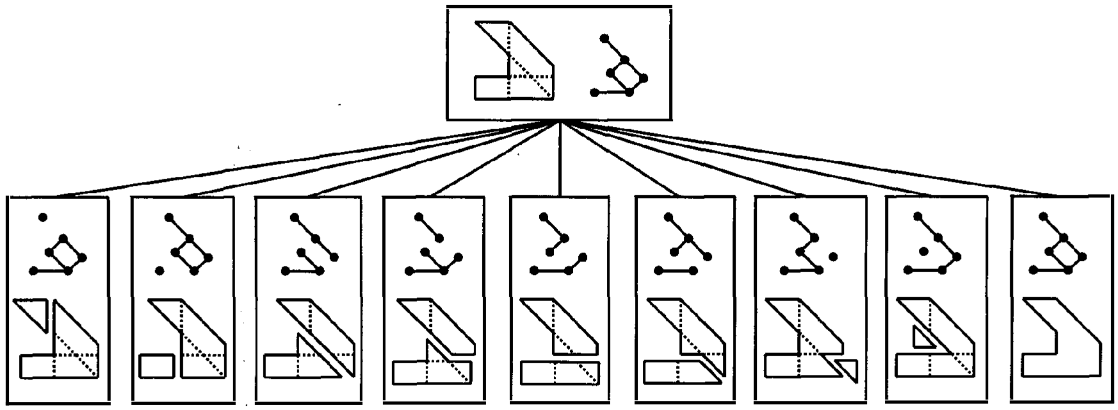

Figure 8: First stage of the dynamic programming problem decomposition. The top box shows a coverage region, its initial decomposition into cells, and the corresponding adjacency graph. There are 8 ways that this graph can be decomposed into two separate connected graphs; for each, the corresponding split of the coverage region is shown. The rightmost choice represents covering all cells as a single region.

4 MSA decomposition

Figure 7 shows a simple example where assigning a different sweep direction to each subregion results in a better decomposition. In this section, we address the general minimal sum of altitudes (MSA) decomposition: forming subregions and assigning sweep directions such that the sum of subregion altitudes (along their respective sweep directions) is minimized.

We propose an algorithm which is based on performing multiple line sweeps to decompose the coverage region into cells and then applying dynamic programming to combine cells into larger subregions and to assign a sweep direction for each subregion.

4.1 Decomposition of the coverage region

For each edge orientation (of the perimeter or a hole), we perform a line sweep using a sweep direction perpendicular to such edge. Each line sweep is done independently, but we overlay all decompositions, in effect taking the dividing lines introduced by all line sweeps. The resulting cells are monotone with respect to all sweep directions under consideration. We must additionally extend all nonconvex edges until they hit a boundary.

We hypothesize that the optimal MSA decomposition can be formed from combinations of the cells from this initial decomposition and now turn to combining these cells into subregions and assigning a sweep direction to each subregion.

4.2 Dynamic programming formulation

From the initial decomposition, we create an adjacency graph (each node represents a cell, and two nodes are connected if they share an edge). This adjacency graph may have cycles, even if there are no holes in the coverage region.

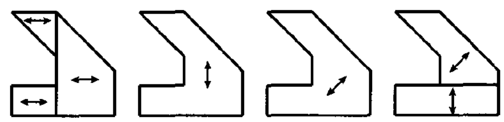

Figure 9: Figures (a) through (c) show the three line sweep decompositions of this region, of which (b) is the best with a subregion altitude sum of 7.0. Figure (d) shows an optimal MSA decomposition produced by our algorithm which has a sum of 5.5.

The basis of our dynamic programming formulation is to either split this graph in two, thus creating two smaller subproblems, or to try to unite all the cells corresponding to nodes in the graph and cover them as one large region.

At the start, we have a graph G which is the adjacency graph from the initial decomposition. If we split the graph, we create two (individually) connected subgraphs G_1 and G_2. These subgraphs together contain all the edges from G except those that connect a node from G_1 to a node in G_2.

We define the minimum sum of altitudes to be:

$$S(G) = \min\left\{C(G), \min_i S(G_1^i) + S(G_2^i)\right\} \quad (4)$$

where i iterates over all possible ways to split the graph G into two connected subgraphs and $C(G)$ returns the cost of covering all cells corresponding to nodes in G as one subregion. When there is only one node in the graph, $S(G) = C(G)$.

The function $C(G)$ must consider all the directions under consideration to determine the cost for covering the cells in G as a single region. For some (or possibly all)

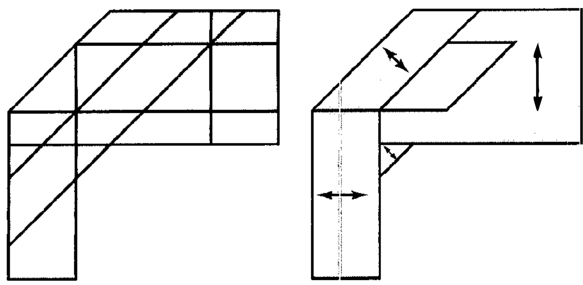

Figure 10: The left figure shows a coverage region and its initial decomposition. The right shows the MSA decomposition produced by our algorithm.

coverage directions, this region may not be monotone, in which case we assign the cost $+\infty$ for those directions. $C(G)$ returns the minimum over all sweep directions under consideration.

Figure 8 shows an example of the first level of decomposing a problem.

4.3 Results

We have implemented the MSA decomposition using the CGAL library [4]. Figure 9 shows the three line sweep decompositions and the optimal solution produced by our MSA decomposition algorithm. Figure 10 shows a more complicated example which took 15 minutes to generate on a Sun Ultra 10 workstation. The present algorithm is rather limited in the complexity of the environment input it can decompose in reasonable time.

There are two factors that indicate an exponential complexity for this algorithm. First, in creating the initial decomposition, each sweep direction contributes dividing lines that divide many other cells. This produces a large number of cells which results in an adjacency graph with many nodes.

Second, the dynamic programming phase must examine all connected subgraphs of 1 to n nodes. For a fully connected graph of n nodes, there are $2^n - 1$ such subgraphs, but it is not possible to have a fully connected adjacency graph (except in trivial cases). We may therefore expect many fewer than 2^n subproblems; however, the number of subproblems is most likely still exponential. The number depends not only on the geometry of the graph but also its maximum degree.

5 Conclusions

We have given a general overview of the coverage problem and described our assumptions and formulation for optimal coverage. We follow the basic approach of decomposing the coverage region into subregions, selecting a sequence of subregions, and generating a coverage path that covers each subregion in turn. This paper introduces a measure of optimality for coverage paths which we translated into a measure of optimality for the coverage region decomposition. The ultimate objective is to decompose a coverage region into subregions so that the sum of subregion altitudes is minimized.

We have shown that for polygonal worlds, the optimal line sweep decomposition uses a sweep direction perpendicular to an edge of the boundary. The more general minimum sum of altitudes (MSA) decomposition, however, can produce a better decomposition by allowing a different sweep direction to be assigned to each subregion. We have given an algorithm that performs multiple line sweeps to decompose the coverage region into cells, and then uses dynamic programming to combine these cells into larger subregions and to assign a sweep direction to each subregion.

References

[1] Z. Butler, A. Rizzi, and R. Hollis. Complete distributed coverage of rectilinear environments. In *Fourth International Workshop on the Algorithmic Foundations of Robotics*, 2000.

[2] H. Choset and P. Pignon. Coverage path planning: the boustrophedon cellular decomposition. In *Proceedings of the International Conference on Field and Service Robotics*, December 1997.

[3] H. Choset, E. Acar, A. Rizzi, and J. Luntz. Exact cellular decompositions in terms of critical points of Morse functions. In *IEEE International Conference on Robotics and Automation*, 2000.

[4] Computational Geometry Algorithms Library (CGAL). `http://www.cgal.org/`

[5] Y. Gabriely and E. Rimon. Spanning-tree based coverage of continuous areas by a mobile robot. Submitted to *Annals of Mathematics and Artificial Intelligence*.

[6] S. Hert, S. Tiwari, and V. Lumelsky. A terrain-covering algorithm for an auv. *Autonomous Robots*, 3(2–3):91–119, June–July 1996.

[7] W. Huang. The minimal sum of altitudes decomposition for coverage algorithms. Rensselaer Polytechnic Institute Computer Science Technical Report 00-3, June 2000.

[8] D. Kurabayashi, J. Ota, T. Arai, and E. Yoshida. Cooperative sweeping by mobile robots. In *IEEE International Conference on Robotics and Automation*, pages 1744–1749, 1996.

[9] G. Schmidt and C. Hofner. An advanced planning and navigation approach for autonomous cleaning robot operations. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 1230–1235, 1998.

[10] A. Zelinsky, R. A. Jarvis, J. C. Byrne, and S. Yuta. Planning paths of complete coverage of an unstructured environment by a mobile robot. *International Journal of Robotics Research*, 13(4):315–, 1994.