[24] R. Shadmehr and F. A. Mussa-Ivaldi, "Adaptive representation of dynamics during learning of a motor task," *J. Neurosci.*, vol. 14, no. 5 (Pt. 2), pp. 3208–3224, 1994.
[25] D. J. Reinkensmeyer, J. L. Emken, J. Liu, and J. E. Bobrow. The nervous system appears to minimize a weighted sum of kinematic error, force, and change in force when adapting to viscous environments during reaching and stepping. presented at *Adv. Computat. Motor Control III*. [Online]. Available: http://www.bme.jhu.edu/acmc
[26] J. L. Emken and D. J. Reinkensmeyer, "Robot-enhanced motor learning: Accelerating internal model formation during locomotion by transient dynamic amplification," *IEEE Trans. Neural Syst. Rehab. Eng.*, vol. 13, no. 1, pp. 33–39, Jan. 2005.

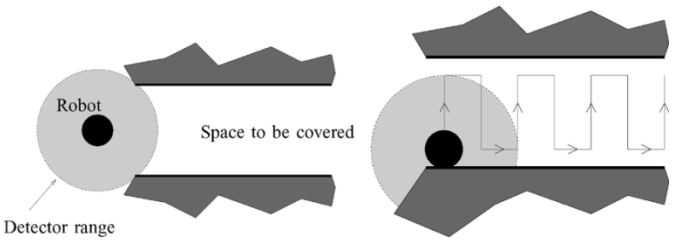

Fig. 1. Robot is required to pass the extended-range detector over all the points in the corridor. Even though the robot can perform back-and-forth motions to cover the space, these motions will not make efficient use of the available detector range.

Sensor-Based Coverage With Extended Range Detectors

Ercan U. Acar, Howie Choset, and Ji Yeong Lee

***Abstract*—Coverage path planning determines a path that passes a robot, a detector, or some type of effector over all points in the environment. Prior work in coverage tends to fall into one of two extremes: coverage with an effector the same size of the robot, and coverage with an effector that has infinite range. In this paper, we consider coverage in the middle of this spectrum: coverage with a detector range that goes beyond the robot, and yet is still finite in range. We achieve coverage in two steps: The first step considers vast, open spaces, where the robot can use the full range of its detector; the robot covers these spaces as if it were as big as its detector range. Here we employ previous work in using Morse cell decompositions to cover unknown spaces. A cell in this decomposition can be covered via simple back-and-forth motions, and coverage of the vast space is then reduced to ensuring that the robot visits each cell in the vast space. The second step considers the narrow or cluttered spaces where obstacles lie within detector range, and thus the detector "fills" the surrounding area. In this case, the robot can cover the cluttered space by simply following the generalized Voronoi diagram (GVD) of that space. In this paper, we introduce a hierarchical decomposition that combines the Morse decompositions and the GVDs to ensure that the robot indeed visits all vast, open, as well as narrow, cluttered, spaces. We show how to construct this decomposition online with sensor data that is accumulated while the robot enters the environment for the first time.**

***Index Terms*—Cell decomposition, coverage, Morse decomposition, sensor-based planning, Voronoi diagrams.**

I. Introduction

Coverage path planning determines a path that directs a robot to pass over all points in its free space. Integrating the robot's footprint (detector range) along the coverage path yields an area identical to that of the target region. We achieve coverage by introducing new types of *exact cell decompositions*. Exact cell decompositions [8], [16] represent the free space by dividing it into nonoverlapping regions called *cells*, such that adjacent cells share a common boundary, the interior of each cell intersects no other cell, and the union of all of the cells fills the free space. An adjacency graph encodes the topology of the cell decomposition, where nodes represent the cells and edges connect nodes corresponding to adjacent cells. Cell decompositions have conventionally been used to determine a path between two points which can be solved in two steps: first, identify the cells containing the start and the goal, and then search the adjacency graph for a sequence of cells that connect the start cell to the goal cell.

We use cell decompositions, however, to produce a path that *covers* the free space. We define cell boundaries such that each cell can be covered via a simple pattern, such as simple back-and-forth motions. A planner achieves complete coverage simply by ensuring that the robot identifies and visits each cell in the space, which is equivalent to finding an exhaustive walk through the adjacency graph.

In previous work, we defined a family of cell decompositions termed *Morse decompositions*, whose cells are defined by critical points of a Morse function. This work was motivated by Canny's roadmap work [5] where he uses critical points to ensure the connectivity of his roadmap. The critical points serve as landmarks because topologically meaningful events occur at them. Recall that Morse functions are real-valued functions with nondegenerate critical points. The Morse decompositions allow us to design sensor-based coverage algorithms that use simple motions, such as back and forth, for coverage with a robot-size detector. Varying the Morse function that defines the decomposition changes the pattern by which coverage is achieved.

Now, consider two ends of the coverage spectrum: coverage with detectors the same size of the robot and infinite-range detectors. The Morse decompositions are suitable for coverage with detectors that are the same size of the robot; we call these robot-size detectors. If the robot has an infinite-range detector, then it need not use Morse decompositions. Instead, the robot can follow the generalized Voronoi diagram (GVD) (sets of points equidistant to two obstacles [11], [19]) to completely cover a bounded free space.

In this paper, we address a problem in the middle of the extremes. We describe a new method to cover unknown spaces with detectors, such as infrared imaging systems or omnidirectional cameras, whose ranges are larger than the robot, but are still less than infinite. We term these *extended-range detectors.* In a sense, an extended-range detector has a "variable" effector size: in a vast, open space, the maximum range of the detector is the effector size. Covering a vast, open space with back-and-forth motions, as if the robot were as big as its maximum detector range, produces a "good" coverage path when the objective is to minimize path length with respect to the area covered.

In narrow spaces, however, the detector's range is limited by the presence of obstacles. As suggested in previous work by Hert *et al.* [14], and then by Lumelsky *et al.* [17], performing simple back-and-forth motions, again as if the robot were as big as its maximum detector range, does not produce an efficient path (Fig. 1). When the robot is in a cluttered or narrow space, its detector is effectively infinite, and thus can cover the space as if it had an infinite detector. See Fig. 2 for an overview of three modes of coverage: robot-size detector, extended-range detector, and infinite-range detector.

Manuscript received October 15, 2004; revised April 15, 2005. This paper was recommended for publication by Associate Editor G. Oriolo and Editor I. Walker upon evaluation of the reviewers' comments.

E. U. Acar was with the Robotics Institute, Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 USA. He is now with Intel Research, Hillsboro, OR 97124 USA (e-mail: acar@cmu.edu).

H. Choset is with the Robotics Institute, Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: choset@cs.cmu.edu).

J. Y. Lee is with the Korea Institute of Science and Technology, Seoul 136-791, Korea (e-mail: jiyeongl@andrew.cmu.edu).

Digital Object Identifier 10.1109/TRO.2005.861455

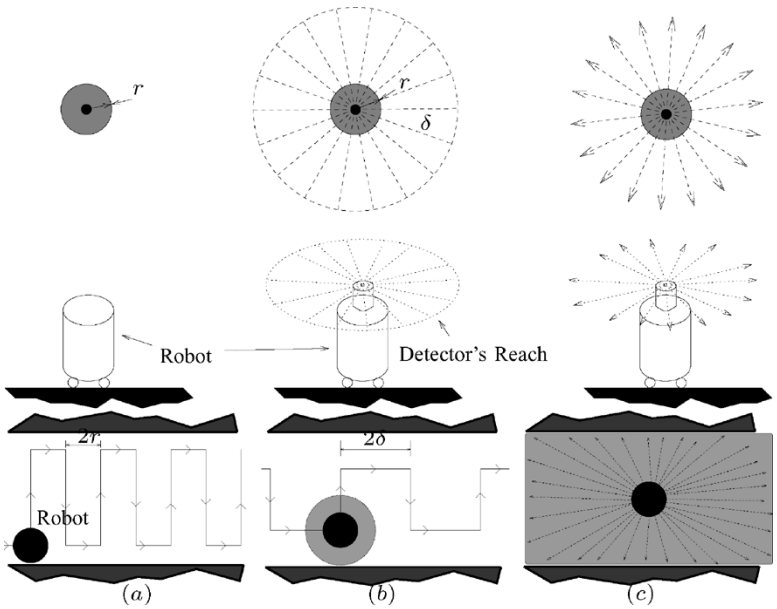

Fig. 2. (a) Robot with a detector as wide as the robot's width. (b) Robot with a detector that has extended range δ. (c) Robot with an infinite-range detector. The top row shows the top views of the robot and the detector range. The bottom row shows coverage paths with respective detector ranges for the same area. Note that in (c), the robot does not need to move to "detect" everything in this example.

We introduce a hierarchical decomposition for coverage with an extended-range detector. This decomposition combines Morse decompositions and GVDs. We divide the robot's free space into two regions: vast regions where obstacles are beyond the detector range of the robot, and narrow regions, where obstacles are within the detector range of the robot. For vast regions, we use a Morse decomposition of the space, but "pretend" that the footprint of the robot is the detector range. In narrow regions, we simply have the robot follow the GVD to cover the unknown space. We also introduce a method that uses sensor data to select appropriate coverage modes. This method allows the robot to locate vast/narrow regions while it is performing coverage in an unknown space.

We first review coverage for a robot-sized detector (Section II), and then for an infinite-sized detector (Section III). Then, we combine these results to achieve the main result of our paper, which is a hierarchical decomposition of the space into vast cells and narrow cells (Section IV). Sensor-based coverage is then achieved by covering the vast cells with Morse coverage techniques, and the narrow cells with Voronoi-based coverage approaches, and guaranteeing that the robot visits all vast and narrow cells. We present an incremental construction method for the hierarchical decomposition (Section V), and, finally, we prove the completeness of the approach (Section VI).

II. Coverage With a Robot-Size Detector

We believe that cell decompositions can be used to describe previous coverage work with robot-size detectors. Cao *et al.* [7] implicitly use a Morse decomposition, but they assume that all obstacles are convex. Butler *et al.* [4] present a coverage algorithm that uses decompositions of rectilinear spaces where all critical points are degenerate. In our work, we only consider nondegenerate spaces. Grid-based methods can be thought of as cell decompositions with very fine cells. Such a method is presented by Zelinsky *et al.* [22], where they generate a coverage path using a path transform that can be regarded as a numeric potential field.

We achieve coverage with a robot-size detector using a cell decomposition called the *boustrophedon decomposition* [9], where coverage in each cell can be achieved by performing back-and-forth motions with an interlap spacing equal to the robot's diameter. Since coverage in each cell is trivial, visiting each cell ensures complete coverage. Essentially, the boustrophedon decomposition provides the bookkeeping to ensure that back-and-forth motions indeed fully cover the free space.

Motivated by Canny's work [5], [6], we define the cells by passing a slice, the pre-image of a real-valued function $h(x, y) = x$. We term $h(x, y)$ a *slice function*, and define a slice as $\mathcal{CS}_\lambda = \{(x, y) \in \mathcal{CS} | h(x, y) = \lambda\}$ for some $\lambda \in \Re$ ($\mathcal{CS}$ denotes the configuration space[1] of a disk robot) through the free space. In our work, we use connectivity changes to locate the cell boundaries. The leftmost and rightmost boundaries of the cells occur at slices where the connectivity of the slice in the free space changes. These connectivity changes occur at the *critical points* [2] of $h|_{\partial \mathcal{CC}}$, the restriction of the slice function to the obstacle boundaries, $\partial \mathcal{CC}$, where $\mathcal{CC}$ represents the union of the configuration space obstacles. We assume that all the critical points of the function $h|_{\partial \mathcal{CC}}$ are nondegenerate (i.e., the critical points are isolated, and at a critical point, the Hessian of $h|_{\partial \mathcal{CC}}$ has a full rank), and thus $h|_{\partial \mathcal{CC}}$ is a Morse function [18]. At the critical points of $h|_{\partial \mathcal{CC}}$, the gradient $\nabla h(x, y)$ of h and the surface normal $N(x, y)$ of the obstacle boundary at the point (x, y) are parallel [2], [10].

We use the critical points to form cells. That is, at a critical point, the connected component of the slice in the free space that contains the critical points forms a cell boundary. The connectivity between the cells is represented by a graph representation, using the Reeb graph. In this graph, each node represents a critical point, and two nodes are connected by an edge if two corresponding critical points lie on the boundary of a common cell (Fig. 3).

This graph representation allows us to prescribe a planner to cover an unknown space by simultaneously covering the space and incrementally constructing the Reeb graph. To realize this procedure with

[1]Even though we present results in the configuration space of a disk robot, our results are valid in workspace, too. Essentially, we achieve coverage in the configuration space using workspace distance measurements.

[2]At the critical points of a function, its first derivative vanishes.

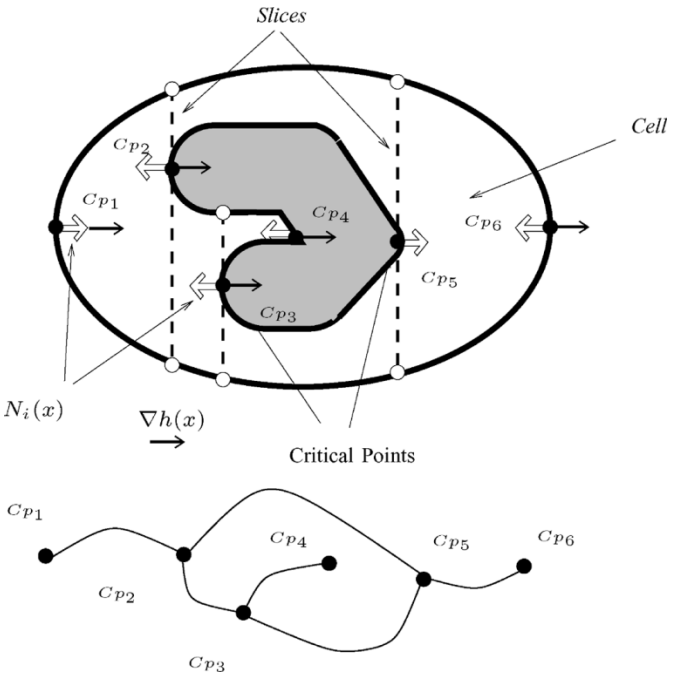

Fig. 3. Exact cell decomposition of a space and its Reeb graph. At the critical points of $h|_{\partial \mathcal{CC}}$, the gradient $\nabla h(x)$ and the surface normal $N(x)$ at a point $x \in \partial \mathcal{CC}$ are parallel. The cell boundaries have two parts, slices that contain critical points and portions of the obstacle boundaries. The Reeb graph with nodes as critical points and edges as cells represents the topology of the decomposition, and enables us to reduce the sensor-based coverage to an incremental construction procedure.

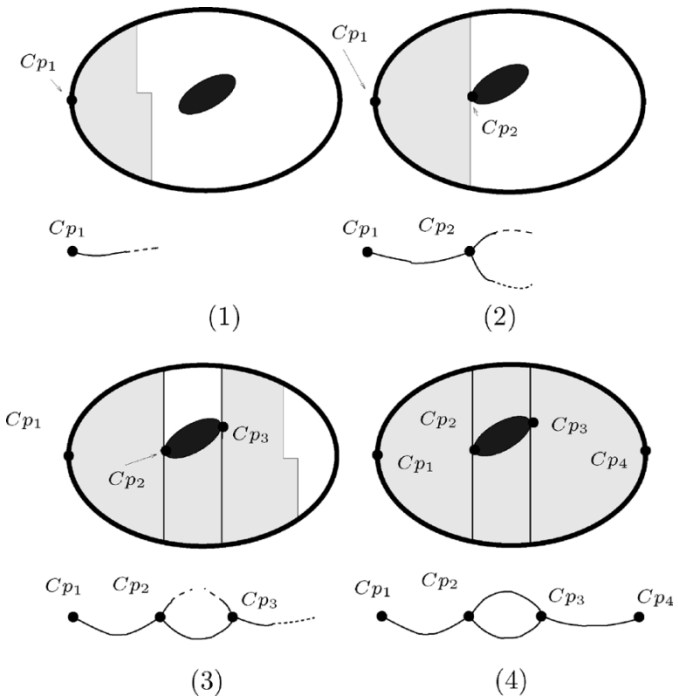

Fig. 4. Incremental construction of the graph while the robot is covering the space.

range sensors, we developed a critical-point sensing method and a coverage algorithm that guarantees that the robot will encounter all critical points in an unknown space (hence, achieving complete coverage) [1], [3]. This coverage algorithm consists of two modes: 1) back-and-forth motion parallel to the slice direction; and 2) boundary-following motion. By repeating the back-and-forth motion with interlap space equal to the diameter of the robot, most of the *workspace* is covered. During the boundary-following motion, the robot covers the points closest to the boundary and simultaneously looks for critical points. In [3], it is shown that using this algorithm, the robot can find all the critical points in the space, and thus can cover the complete space.

Fig. 4 shows a simple space covered by this algorithm. In Fig. 4(1), the robot starts to cover the space at the critical point Cp_1, and it instantiates an edge with only one node. When the robot is done covering the cell between Cp_1 and Cp_2, it joins their corresponding nodes with an edge in the graph representation [Fig. 4(2)]. Now the robot has two new uncovered cells. Since the space is *a priori* unknown, the robot arbitrarily chooses the lower cell to cover. When the robot reaches Cp_3, nodes of Cp_2 and Cp_3 become connected with an edge, and the lower cell is completed [Fig. 4(3)]. At Cp_3, the robot decides to cover the cell to the right of Cp_3. When the robot senses Cp_4, it goes back to Cp_3 and starts to cover the upper cell. When it comes back to Cp_2, the robot determines that all the edges of all the nodes (critical points) have been explored [Fig. 4(4)]). Thus, the robot concludes that it has completely covered the space. This incremental construction method serves as a basis for the sensor-based coverage algorithm with a robot-size detector. Fig. 5 shows different stages of this incremental construction in *a priori* unknown 2.75×3.65 m^2 room with a Nomad mobile robot that has a ring of sonar sensors.

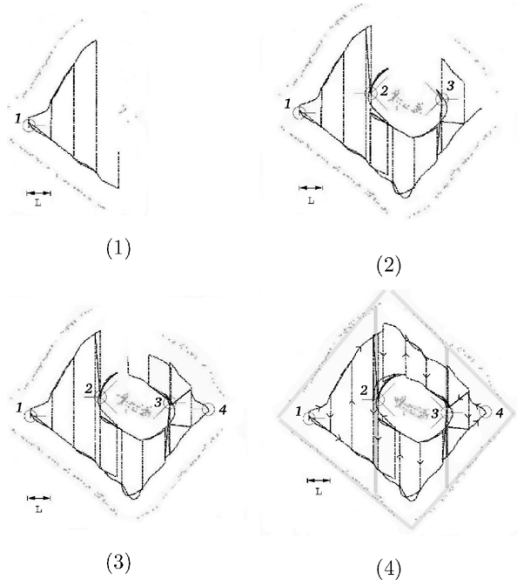

Fig. 5. Four stages of coverage in an unknown environment with a robot-size detector. The coverage path followed by the robot is shown by dotted black lines. We depict the critical points as circles with lines emanating from them. The lines represent the directions of the corresponding adjacent cells. The robot incrementally constructs the graph representation by sensing the critical points 1, 2, 3, 4, 3, 2 (in the order of appearance) while covering the space. In the final stage (4), since all the critical points have explored edges, the robot concludes that it has completely covered the space. For the sake of discussion, we outlined the boundaries of the obstacles and cells in (4). $L = 0.53$ m.

Finally, we would like to note that this paper mainly focuses on complete, but not optimal, coverage of unknown environments. For reference on optimal coverage algorithms (mainly for robot-sized detectors) that work in discrete or known spaces, see [13], [15], [20], and [21].

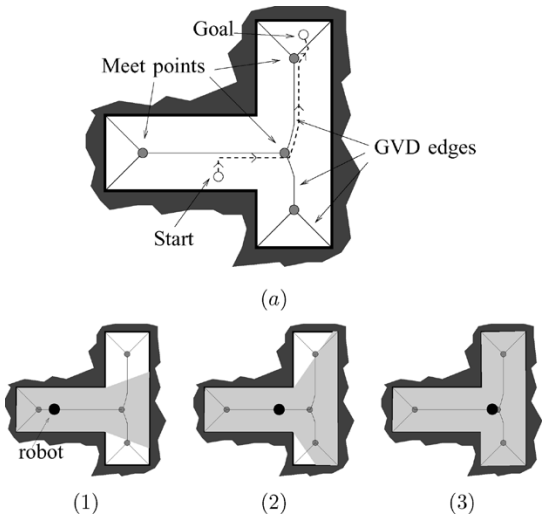

Fig. 6. (a) Robot uses the GVD to plan a path from a start to a goal by first planning a path from start to the GVD, then along the GVD edge to the vicinity of goal, and then from the GVD to goal. (1) The robot starts to cover the space with an infinite-range detector by following the GVD edges from right-hand side. In (2), almost two-thirds of the space is covered. In (3), the free space is completely covered.

III. Coverage With an Infinite-Range Detector: Generalized Voronoi Diagrams

To cover an unknown space with an infinite-range detector, we use the GVD. The GVD comprises GVD edges and nodes termed *meet points*. Each GVD edge is the set of points equidistant to two obstacles, i.e., $\{x \in \Re^2 | d_i(x) = d_j(x) \leq d_h(x)\ \forall h, \nabla d_i(x) \neq \nabla d_j(x)\}$, where $d_i(x)$ measures the distance to the closest point on obstacle C_i, and $\nabla d_i(x)$ is its gradient. At meet points where GVD edges "meet," there are more than two equidistant obstacles. In the context of conventional motion planning, the robot can use the GVD to plan a path from a start to a goal by first planning a path from the start to the GVD, then along the GVD edge to the vicinity of the goal, and then from the GVD to the goal. See Fig. 6 for the GVD of a simple space and a sample path between start and goal locations.

In the context of coverage, if the robot follows all of the GVD edges, then it is sure to come within the line-of-sight (LOS) for all points in the free space. In other words, by tracing the GVD, the robot is guaranteed to pass an infinite-range detector over all points in the free space. Since we can construct the GVD using the LOS information, we can guarantee the robot can cover an unknown region by incrementally constructing the GVD [11], [12]. Fig. 6 depicts stages of coverage with an infinite-range detector in a simple T-shaped room.

IV. Hierarchical Decomposition in Terms of Generalized Voronoi Diagrams and Morse Decompositions

In this and next two sections, we present an algorithm that combines the approaches described in Sections II and III to achieve coverage with the extended-range sensor. For this, we decompose the free configuration space into vast regions and narrow regions using workspace distance measurements. The vast regions are covered using the simple back-and-forth motion, and the narrow regions are covered by following the Voronoi diagram. In this section, we define the decomposition and the graph structure which encode the decomposition. We term the this graph the *hierarchical decomposition graph*.

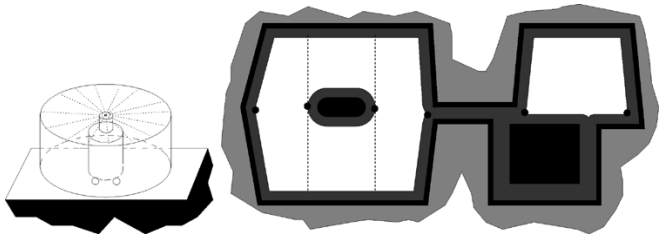

Fig. 7. Robot is assumed to be as big as the extended-range detector to cover vast, open spaces. The configuration space of the detector-size robot is disconnected, even though the configuration space of the robot itself is not.

A. Definition of Vast and Narrow Cells

We are considering a disk robot with radius r operating in a bounded, planar, 2-D workspace. The robot has a omnidirectional range sensor, with a limited sensing range of δ, placed at the center of the robot. Workspace is populated by workspace obstacles, whose boundaries are closed curves. We assume that there are no two obstacles whose boundaries are parallel to each other and separated by δ. Workspace is denoted as $\mathcal{WS} \subset \Re^2$, and the workspace obstacles are denoted as $\mathcal{C}_i$. Then free workspace, denoted as $\mathcal{FWS}$, can be defined as $\mathcal{FWS} = \mathcal{WS} \backslash \bigcup_i \mathcal{C}_i$. Let $D(x)$ denote the distance function that measures the distance in workspace between the center of the robot at a configuration x and the closest workspace obstacle. Now consider the configuration spaces $\mathcal{FCS}_r$ and $\mathcal{FCS}_\delta$ for the robot with radius r and the radius δ, respectively. Since we are considering the disk robot, $\mathcal{FCS}_r$ and $\mathcal{FCS}_\delta$ have the same dimensions as $\mathcal{FWS}$, and actually, they can be considered as subsets of $\mathcal{FWS}$. Therefore, the free configuration space $\mathcal{FCS}_r$ can be identified with $\{x \in \mathcal{FWS} | D(x) > r\}$. Without loss of generality, we assume that $\mathcal{FCS}_r$ is connected. Finally, let $\mathcal{FCS}_\delta$ be the free configuration space of the robot with radius δ, i.e., $\mathcal{FCS}_\delta = \{x \in \mathcal{FWS} | D(x) > \delta\}$. Note that even though $\mathcal{FCS}_r$ is connected, $\mathcal{FCS}_\delta$ may not be connected (see Fig. 7).

Let $B_\delta(x)$ be the solid disk of radius δ centered at $x \in \mathcal{FWS}$. Then the vast region is defined as

$$\begin{aligned}\mathcal{VR} &= \{x \in \mathcal{FCS}_r | B_\delta(x) \subset \mathcal{FWS}\} \\ &= \{x \in \mathcal{FCS}_r | D(x) > \delta\}.\end{aligned}$$

Note that we are making a strong identification between $\mathcal{FWS}$, $\mathcal{FCS}_r$, and $\mathcal{FCS}_\delta$. In particular, a point $x \in \mathcal{FCS}_r$ is the location of the center of the robot in $\mathcal{FWS}$. With this identification, one can write $\mathcal{VR} = \mathcal{FCS}_\delta$ and, thus, for all x in the vast region, $D(x) > \delta$.

The narrow region $\mathcal{NR}$ is the complement of the vast region $\mathcal{VR}$, i.e., $\mathcal{NR} = \mathcal{FCS}_r \setminus \mathcal{VR}$. The vast cells are defined to be the connected components of the vast region. The narrow region is simply the complement of the vast region and the narrow cells are the connected components of the narrow region. See Fig. 8 for an example of the decomposition.

To cover the vast cells, the robot can perform back-and-forth motions as if its physical size were as big as the detector range whose radius is δ. In other words, the robot performs coverage in free configuration space of a 2δ-diameter disk using our coverage algorithms presented in Section II, but assumes an interlap spacing of 2δ as opposed to $2r$ (setting $r = \delta$) via lapping and wall-following motions.

The narrow cells can be covered by simply following the GVD. Note that for all points x in the GVD in the narrow cells, $D(x) \leq \delta$. Thus, when the robot is located in a narrow cell, at least one obstacle lies within the detector range δ. Therefore, the detector acts like an infinite-range detector in a narrow cell, and we can simply have the robot follow the GVD edges for complete coverage in the cell. See Section VI for a formal proof.

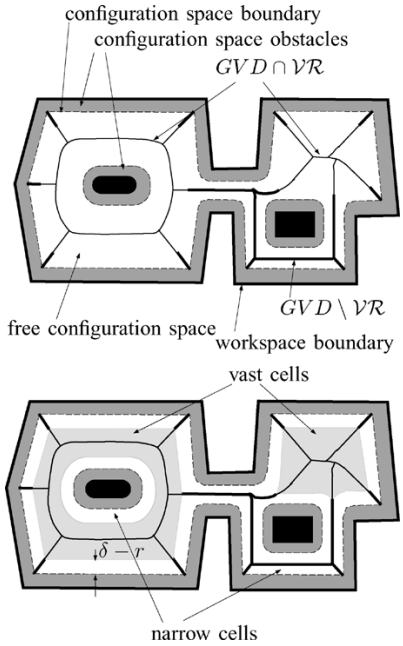

Fig. 8. Vast and narrow cells of the space. In the upper figure, the dark gray region (together with black region) represents the configuration space obstacle. The white region represents the free configuration space. Since the free configuration space is connected, the GVD is also connected. In the lower figure, the light gray region represents the vast region, and the white region represents the narrow region. The thin line represents the GVD in the vast region, and the thick line represents the GVD in the narrow region. There are two vast cells and two narrow cells in the given environment.

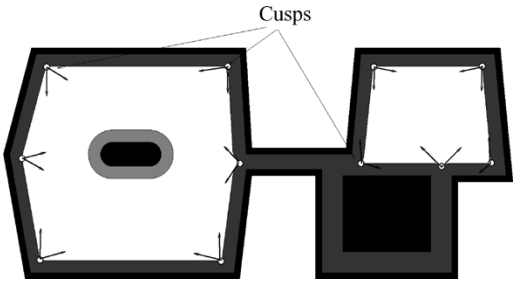

Fig. 9. Configuration space for a 2δ-diameter disk detector. The space becomes disconnected and the corridors are not accessible to the 2δ disk. At cusp points, the boundary of the configuration space is nonsmooth.

B. Transition Between Narrow and Vast Cells

In unknown spaces, the robot starts to cover the space in a narrow or vast cell, depending on the sensor reading, without knowing the existence of other cells. Therefore, while the robot is covering the space, it has to identify features of the cells that indicate the existence of new vast or narrow cells. For this purpose, we use cusp points. A cusp point is defined as a point x on the GVD with $D(x) = \delta$ (Fig. 9).

Recall that the robot covers a vast cell by performing boundary-following and lapping motions. While covering a vast cell, the robot needs to look for cusp points only during the boundary-following motions, since by definition, the robot traces all the points with $D(x) = \delta$ in the vast cell during the boundary-following motion. Thus, during the boundary-following motion, whenever the robot detects more than one closest obstacle (at a distance $D(x) = \delta$, then it has found a cusp point (Fig. 10). Since the robot completely traces the boundary of the free configuration space during the boundary following motion, the robot can find every cusp point in a vast cell.

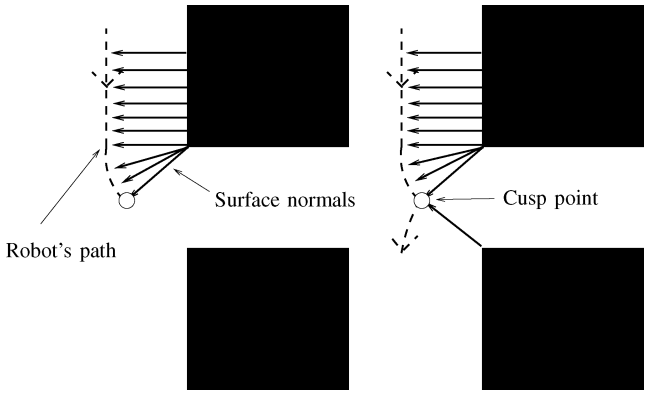

Fig. 10. Robot determines the location of the cusp point by sensing the jump in the surface normal while it is following the boundary of the obstacle.

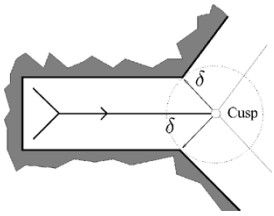

Fig. 11. Robot determines the location of the cusp point while it is following the GVD edge by sensing the increase in the minimum distance to the obstacles ($D(x) > \delta$).

To determine cusp points while covering narrow regions, the robot uses the *magnitude* of the distance measurements. Recall that the robot covers narrow regions by following the GVD edges, and thus the minimum distance to the obstacles is always less than the detector range. Therefore, in a narrow cell, the robot looks for an increase in the minimum distance. When the minimum distance to the obstacles is greater than the detector range, the robot locates the cusp point (Fig. 11). Since the GVD is connected in a connected space, by tracing the GVD in a narrow cell, the robot can find every cusp point in a narrow cell.

Recall that we are assuming that there are no two obstacles whose boundaries are parallel to each other and separated exactly by 2δ. This implies that the cusp points are isolated from each other, i.e., the set of cusp points is a 0-D set of isolated points. This makes the presentation of the proposed method clear. However, note that even if there are such obstacles, the robot still can cover the space by simply tracing the GVD.

C. Hierarchical Decomposition Graph

We encode the entire information that the robot needs for coverage with extended-range detectors in a *hierarchical decomposition graph*. It has two types of nodes: vast nodes and narrow nodes, representing vast cells and narrow cells, respectively. A vast node and a narrow node is connected by an edge if there is a cusp point between the corresponding vast cell and narrow cell in the free space. Actually, as we will see later, if a vast cell and a narrow cell share a common boundary, there must be a cusp point between them. Note that there can be multiple edges between a vast node and a narrow node, since there can be multiple cusp points between the corresponding cells (Fig. 12). Also note that there cannot be an edge between two vast nodes or two narrow nodes, because neither two vast cells nor two narrow cells can be adjacent to each other, by definition.

Each vast node and narrow node contains subgraphs. A vast node contains a Reeb graph that represents the cell decomposition of the vast cell. Recall that in the Reeb graph, the cells correspond to edges, and the nodes correspond to the critical points. Note that within each vast

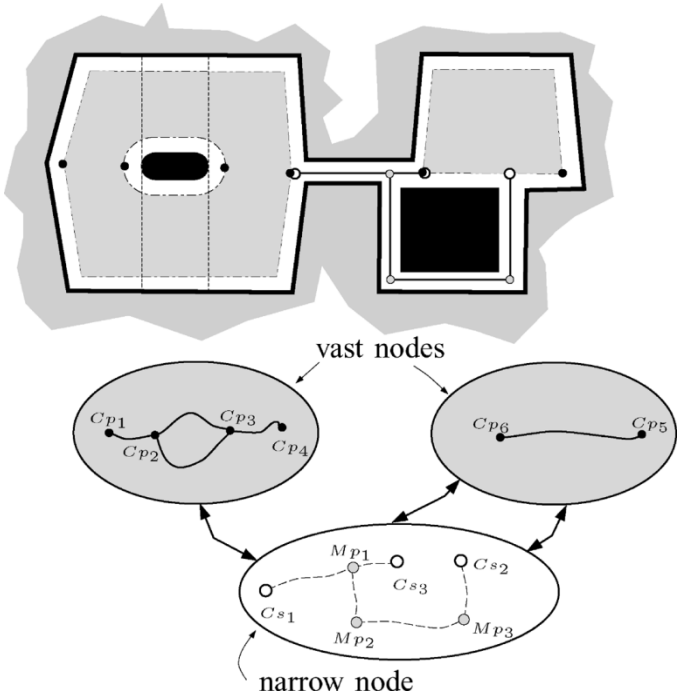

Fig. 12. In the graph (bottom figure), gray ellipses depict the vast nodes that contain vast subgraph represented as solid edges. Each vast subgraph has two or more associated critical points of $h|_{\partial CC}$ represented as black dots. Narrow node is represented by the white ellipse, and it contains the narrow subgraphs (i.e., a subset of the GVD), depicted as dashed edges. Hollow dots correspond to cusp points and gray dots represent the meet points. Double arrows show the links between the narrow nodes and their neighboring vast nodes. Note that there are two edges between the narrow node and the vast node on the right, since there are two cusp points between them. In this particular example, there are narrow nodes that are not shown in the figure, such as the ones at the corners of the environment. Since the subgraphs (i.e., GVD) associated with them are trivial (each subgraph has just one GVD edge in it), for the sake of clarity, we do not show them in the graph. Gray regions in workspace (top figure) represent the $\mathcal{FCS}_\delta$ in which the robot moves around to cover the vast cells. Thin solid lines represent a subset of the GVD, which is used to cover the narrow cells. Note that there are small narrow cells near every corner of the environment where the robot traces the GVD. The cells corresponding to those narrow regions are not depicted in the figure.

node, the Reeb graph is connected. A narrow node contains a GVD of the narrow cell.

Fig. 12 shows an example of a hierarchical decomposition graph for a sample environment. Gray ellipses represent the vast nodes, and white ellipses represent the narrow nodes. The cusp points are represented by hollow dots. The edges of the narrow subgraph (i.e., GVD edges) are depicted by dashed edges, and they are connected to each other via meet points, where there are more than two equidistant obstacles. Each meet point is represented by a gray dot.

V. Incremental Construction of the Hierarchical Decomposition

In this section, we describe an algorithm to incrementally construct the hierarchical decomposition, and then in Section VI, we prove completeness of the method.

Algorithm 1 shows the pseudocode for the coverage algorithm. Initially, the hierachical decomposition graph (N, E) is empty. At an arbitrary starting location, the planner creates a vast node or a narrow node, depending on the current distance measurements. Then, depending on the type of the node, the robot starts covering the cell. Whenever the robot detects a cusp point while covering the cell, the planner creates a new uncovered node and an edge between the current node and the new node. After covering the current cell, the planner retrieves an uncovered node from the graph, and then directs the robot to the uncovered node. Since the hierarchical decomposition graph is connected, by repeating the step described above, the robot is guaranteed to visit all the nodes in the graph, and thus completely cover the space.

Algorithm 1 Coverage using hierarchical decomposition graph
$N \leftarrow \emptyset, E \leftarrow \emptyset$
$x \leftarrow$ current robot configuration
$n \leftarrow$ CreateNode(x, $D(x)$)
mark n as uncovered
$N \leftarrow N \cup \{n\}$
repeat
  find a path from current position to n in (N, E)
  move the robot to $x(n)$
  if n is a vast node **then**
    MorseDecompCoverage(n, L, F)
  else
    TraceGVD(n, L, F)
  end if
  mark n as covered
  $N \leftarrow N \cup L, E \leftarrow E \cup F$
  $n \leftarrow$ FindUncoveredNode(N)
until $n == NULL$

We demonstrate incremental construction of the hierarchical decomposition while the robot is covering its free space using an example depicted in Fig. 13. The robot starts to cover the space in a vast cell by executing our prior coverage algorithm from leftmost corner that corresponds to the critical point Cp_1.

When the robot reaches the critical point Cp_2, which is on the leftmost side of the middle obstacle, it finishes covering the vast subcell between Cp_1 and Cp_2 [Fig. 13(1)]. At this point, since the space is unknown *a priori*, the robot arbitrarily chooses the lower vast subcell among the two new vast subcells to cover. The robot finishes the new vast subcell when it senses the critical point Cp_3 on the rightmost side of the middle obstacle, and starts to cover the vast subcell to the right side of the obstacle [Fig. 13(2)].

While the robot is covering this subcell, it locates the cusp point Cs_1 and instantiates a narrow node that contains a narrow subgraph [Fig. 13(3)]. The vast node (gray ellipse) and the narrow node (white ellipse) are connected via the cusp point represented by the double-arrow line. Note that in this example, the cusp point and the critical point Cp_4 occur at the same location, but, in general, this is not the case. The robot finishes the left room by covering the final vast subcell on the top of the obstacle and comes back to the cusp point Cs_1 [Fig. 13(4)]. Starting at Cs_1, the robot begins to follow the GVD edge along the corridor.

While the robot is following the GVD edge in the corridor, it encounters meet point Mp_1, shown as a gray dot in the graph that identifies two new edges (i.e., the GVD edges) of narrow subgraph. The robot chooses the lower edge, and follows it in the corridor until it reaches the other end of the corridor, which is the cusp point Cs_2.

At this point, the robot determines the existence of a new vast cell [Fig. 13(5)]. In the graph, it instantiates a vast node for this new vast cell, and a solid edge without nodes for the new vast subcell. Again, this new vast node is connected to the narrow node by an edge, since there is a cusp point between two cells corresponding to these nodes. The robot starts to follow the outer boundary toward the right starting

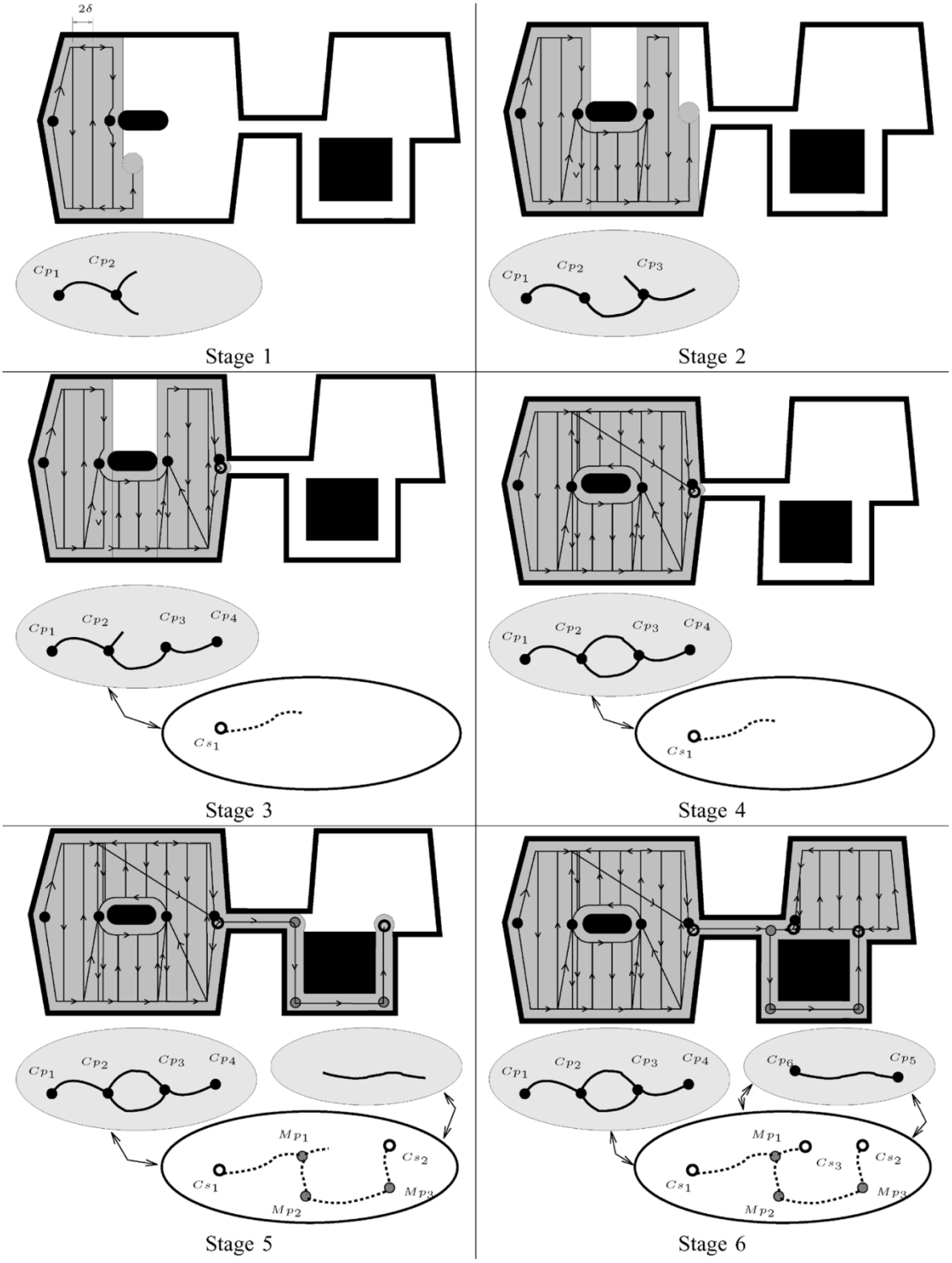

Fig. 13. Depiction of the stages of the incremental construction of the hierarchical decomposition while the robot is covering the space. Note that there are narrow cells near the corners of the environment, where the robot should trace the GVD as it does in the "corridors" in this example. The coverage of these corner regions is not depicted in the figure.

from the cusp point Cs_2 until it senses the critical point Cp_5. Then the robot starts to cover the vast subcell (toward left). When the robot senses the critical point Cp_6, which is also the cusp point Cs_3, the robot finishes covering the right room and starts to follow the corridor toward the left. When the robot comes back to Mp_1, the robot determines that it has visited all branches of the graph, and hence has passed the extended-range detector over all points in the space.

VI. Completeness of the Approach

Our goal is to develop an algorithm that can completely the cover workspace. However, it is, in principle, impossible to achieve this in some cases. Consider the environment shown in Fig. 14. In this example, the points around the corner cannot be covered by any coverage algorithm. In fact, those points are out of the sensor range from any

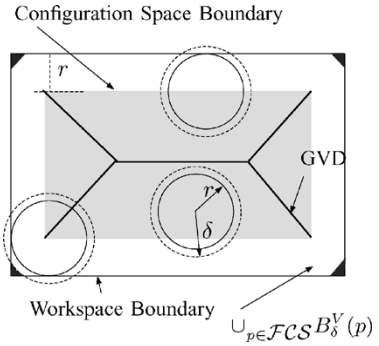

Fig. 14. Robot with radius r and sensor range δ in a rectangular environment. The white region (including the light gray region) $\cup_{p\in\mathcal{FCS}} B_\delta^V(p)$ represents the space that can be covered by the robot. The dark gray regions at the corners cannot be covered by any algorithm if the sensor range is not large enough. If the sensor range were large enough, the corners could have been covered by tracing the GVD, even though the robot cannot physically reach there.

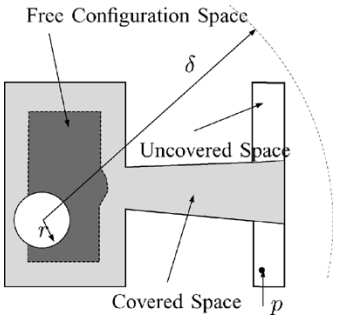

Fig. 15. Robot with radius r and with sensor range δ. Some of the points in the right room can be covered even if the robot cannot physically reach there. However, even with infinite sensor range, some point in the right room cannot be covered. That is, even if a point p lies in the disk $B_\delta(x)$ for some point x in $\mathcal{FCS}$, the point p might not be covered, if the line connecting the points p and x does not lie completely in free workspace.

point in the free configuration space. If the sensor range were large enough, these points could be reached from some point on the GVD.

Even with an infinite-range sensor, there could be some points that cannot be covered. In Fig. 15, the robot has infinite detector range, and thus, every point in the space lies within in the radius of sensor range. However, there could still be some points in the space that cannot be covered if the LOS between the robot and the point does not lie completely in free workspace.

Therefore, we need to define the completeness of the coverage as follows. Let $B_\delta^V(x)$ be the set of the points that are visible from the point x, and within the distance δ from the point x. That is $B_\delta^V(x) = \{p \in B_\delta(x) : \forall t \in [0,1],\ tp + (1-t)x \in \mathcal{FWS}\}$. Then the set $\mathcal{CVS} = (\cup_{x\in\mathcal{FCS}_r} B_\delta^V(x)) \cap \mathcal{FWS}$ is the set of the point that can be possibly covered by any coverage algorithm. Let $p(t), t \in [0,T]$, $T \in \mathbb{R}$ is the path (in $\mathcal{FCS}_r$) generated by the planner. Then we define the planner to be *complete* if

$$\left(\cup_{t\in[0,T]}(B_\delta^V(p(t))\right) \cap \mathcal{FWS} = \mathcal{CVS}$$

for the robot with radius r and sensing range δ. That is, if, by following the path $p(t)$, the robot can "see" the same amount of the space that is visible from at least one point in the space, then the planner is complete. As noted above, the set $(\cup_{x\in\mathcal{FCS}} B_\delta^V(x)) \cap \mathcal{FWS}$ may not be the same as the $\mathcal{FWS}$, and if there is a point that lies in $\mathcal{FWS}$ but not in $(\cup_{x\in\mathcal{FCS}} B_\delta^V(x)) \cap \mathcal{FWS}$, this point cannot be covered by any coverage algorithm.

Complete coverage with hierarchical decomposition relies on covering vast and narrow cells, and using cusp points to connect the cells. We have already prescribed complete coverage algorithms for vast cells [1]. To ensure complete coverage of narrow cells, the robot follows the GVD edges using the methods presented by Choset *et al.* [11]. Recall that by following the GVD edges, the robot with an infinite-range detector covers all points in the space. Since the narrow cells can be thought as the set of points in workspace whose distance to the nearest obstacle is less than δ, a robot following the GVD with detector range δ is also guaranteed to cover all points in the narrow cell, because a δ-ranged detector in a confined space, where the closest pair of points on any pair of obstacles is in the sensor range, is effectively an infinite-sized detector.

The proof of the completeness of the algorithm proceeds as follows. First, we show that each cell can be covered by either using the Morse decomposition method (described in Section II) or tracing the GVD (*Lemma 1*). Next, we show that a vast cell must intersect the GVD, and the boundary of a vast cell must contain a cups point (*Lemma 2*), and thus, the hierarchical decomposition graph (*Lemma 3*) is connected in the connected free configuration space. Finally, we conclude that the algorithm is complete (*Theorem 1*).

Recall that the Morse decomposition method is guaranteed to cover the complete space if the space is connected. By definition, a vast cell is connected, and thus, it is trivially true that the vast cell can be completely covered by the Morse decomposition. More precisely, if a point x lies in the set $\mathcal{WVR} = \{y \in \mathcal{FWS} : \dagger \in \mathcal{B}_\delta^{\mathcal{V}}(\ddagger)\{\imath\nabla\jmath\Updownarrow\rceil\ddagger \in \mathcal{VR}\}$, then the point x is covered by the Morse decomposition method. Now we show that the point that does not lie in the set $\mathcal{WVR}$ can be covered by following the GVD in the narrow region. This can be stated more precisely as follows.

Lemma 1: Let x be a point in the set $\mathcal{CVS}\backslash\mathcal{WVR}$. Then the distance between the point x and the GVD is less than δ. Moreover, if the point p is the closest point on the GVD from the point x, the point x lies in the narrow region $\mathcal{NR}$.

Proof: The first part is shown by contradiction. Assume that there is a point x in the set $\mathcal{CVS} \setminus \mathcal{WVR}$ such that the distance from the point x to the GVD is bigger than δ. Let C be the closest obstacle from the point x, and c be the closest point on C from x. Since x lies in the $\mathcal{CVS} \setminus \mathcal{WVR}$, $D(x) < \delta$. Otherwise, $B_\delta(x)$ lies completely in the free space, and this implies that x lies in $\mathcal{WVR}$. Therefore, the $B_\delta(x)$ intersects the obstacle boundary ∂C, but, by assumption, it does not intersect the GVD. This means that the $B_\delta(x) \cap \mathcal{FWS}$ completely lies in the Voronoi region of the obstacle C. Consider the line defined by the points x and c. This line is normal to the boundary of C at the point c. Also, this line intersects the boundary of $B_\delta(x)$ in two points, and one of those points, labeled q, lies on free workspace. Clearly, the line segment connecting q and c is normal to ∂C at c, thus, the distance between q and C is $\delta + D(x) > \delta$. Since the point q lies in $B_\delta(x)$, and therefore, it is an element of the Voronoi region of C, the closest obstacle to q is also the obstacle C. That is, $D(q) > \delta$. Therefore, $B_\delta(q)$ lies completely in the $\mathcal{FWS}$, and thus q is in the vast region. Then, since $B_\delta(q)$ contains point x, x is also in the $\mathcal{WVR}$, which is a contradiction. Therefore, the distance between the points x and the GVD is smaller than δ.

The second part is trivial. Let p be the closest point on the GVD from the point x. If p does not lie in $\mathcal{NR}$, then $B_\delta^V(p)$ lies completely in free space, and thus x lies in $\mathcal{WVR}$, which is a contradiction. ■

The next step is to show that a vast cell must intersect the GVD.

Lemma 2: A vast cell V intersects the GVD. Moreover, if the GVD is not completely contained in V, the boundary point between the closure of the set GVD $\cap V$ and the closure of the set GVD $\setminus V$ is a cusp point.

Proof: Let p be a point in V, and C_i be the configuration-space obstacle closest from the point p, and c_i be the closest point on the obstacle C_i from p. Then, by the definition of the vast region, $D(p) > \delta$. Let L be the ray starting from the point c and passing through the point p. Then as the robot travels from the point p away from the obstacle C_i along the line connecting the points p and c_i, the distance from the obstacle C_i monotonically increases without bound. However, since we assume that the environment is bounded, at some point on L, the distance to some other obstacle, say C_j, becomes zero. Then, from the continuity of the distance function, this implies that there is a point q on L, where the distance to C_i and C_j is the same, that is, the point q is on the GVD. Then for all the points on the line segment p and q, the distance to the closest obstacle is bigger than δ. Thus, p and q lie in the same connected components of a vast cell, i.e., the vast cell V. Thus, V intersects the GVD.

Now, if the GVD is not a subset of the vast cell V, for all points p in GVD $\cap V$, $D(p) > \delta$, and for all points q in GVD $\setminus V$, $D(q) \leq \delta$. Then, since the GVD is connected, and the distance function is continuous, it follows that at a boundary point r between GVD $\cap V$ and GVD $\setminus V$, $D(r) = \delta$. That is, r is a cusp point. ■

Lemma 3: The hierarchical decomposition graph is connected.

Proof: From *Lemma 2* and the connectivity of the GVD, it follows that the union of the GVD and the vast region is connected. Then, since the narrow region is defined as the set GVD $\setminus \mathcal{VR}$, the union of the narrow region and the vast region is also connected. Moreover, again from *Lemma 2*, the set $\mathrm{cl}(\mathrm{GVD}) \cap \mathrm{cl}(\mathcal{VR})$ is the set of cusp points. Therefore, the hierarchical decomposition graph is also connected. ■

Finally, we show that the coverage with the hierarchical decomposition graph is complete.

Theorem 1: Coverage with the hierarchical decomposition graph is complete.

Proof: Since the robot can find every cusp point in the configuration space and the hierarchical decomposition graph is connected (*Lemma 3*), the robot can visit every node in the graph. Since each cell can be completely covered by either by tracing the GVD (*Lemma 1*) or using the Morse decomposition method [2], we conclude that the coverage with hierarchical decomposition is complete. ■

This establishes the completeness of the proposed method.

VII. Conclusion

Tasks such as looking for people with a thermal camera require a coverage algorithm that can handle variable-range detectors. We combined coverage with robot-size and infinite-range detectors to prescribe a hierarchical decomposition to achieve coverage with extended-range detectors. Our hierarchical decomposition consisted of two types of cells: vast and narrow. Coverage in the vast cells is achieved by using previous work with Morse decompositions. We formulate the Morse decompositions with conventional slice algorithms, where the slice is defined by the pre-image of a real-valued function, whose restriction to obstacle boundaries does not have any degenerate critical points, i.e., it is Morse. We used the critical points to form the cell boundaries such that the structure of each cell enables a planner to use simple motions, such as back-and-forth maneuvers, for coverage. We furthered this approach by altering the Morse function, and hence the slice geometry, to obtain a variety of coverage patterns. This work was originally prescribed for a robot to cover a target region with a robot-sized detector. Now, we treat the robot with its effective detector range as a "big robot," and use the conventional Morse algorithms to cover the vast cells.

We covered the narrow cells by following the Voronoi diagram. By using Morse decompositions and Voronoi diagrams for coverage with extended-range detectors, we were able to demonstrate that one can take advantage of the strengths of two different structures. Voronoi diagrams are more useful in narrow spaces than in vast, open spaces. Whereas in vast, open spaces, Morse decompositions are more useful than the Voronoi diagrams for coverage.

Previous work also prescribed the control laws to have a robot reactively cover a vast cell (from the Morse decomposition work), and cover a narrow cell (from the sensor-based control laws to generate GVD edges). In either case, the robot is mainly reactively sensing and then acting, only making high-level decisions at critical points of the Morse decomposition or meet points in the GVD. We switched between the two sets of reactive behaviors by looking for cusp points that lie on the boundaries of both types of cells. These cusp points are easy to sense with range sensors, such as sonars. Therefore, a mobile robot equipped with a range-sensing system, such as a sonar ring, can cover the space completely with extended-range detectors.

In this paper, we only considered coverage with omnidirectional detectors. As a part of future work, we are planning to develop coverage algorithms for detectors with a limited angle of view. Also, future work will consider effectors, other than sensors. Currently, we are looking at auto-body painting, where the effector is the footprint of the paint decomposition.

References

[1] E. U. Acar and H. Choset, "Sensor-based coverage of unknown environments: Incremental construction of Morse decompositions," *Int. J. Robot. Res.*, vol. 21, pp. 345–366, Apr. 2002.

[2] E. U. Acar, H. Choset, A. A. Rizzi, P. Atkar, and D. Hull, "Morse decompositions for coverage tasks," *Int. J. Robot. Res.*, vol. 21, pp. 331–344, Apr. 2002.

[3] E. U. Acar and H. Choset, "Critical point sensing in unknown environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, San Francisco, CA, 2000, pp. 3803–3810.

[4] Z. Butler, A. A. Rizzi, and R. L. Hollis, "Cooperative coverage of rectilinear environments," in *Conf. Robots Autom.*, Apr. 2000, pp. 2722–2727.

[5] J. F. Canny, *The Complexity of Robot Motion Planning*. Cambridge, MA: MIT Press, 1988.

[6] J. F. Canny and M. Lin, "An opportunistic global path planner," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1990, pp. 1554–1561.

[7] Z. L. Cao, Y. Huang, and E. Hall, "Region filling operations with random obstacle avoidance for mobile robots," *J. Robot. Syst.*, pp. 87–102, Feb. 1988.

[8] B. Chazelle, "Convex partition of polyhedra: A lower bound and worst-case optimal algorithm," *SIAM J. Comput.*, vol. 13, no. 3, pp. 488–507, 1984.

[9] H. Choset, "Coverage of known spaces: The boustrophedon cellular decomposition," *Auton. Robots*, vol. 9, pp. 247–253, 2000.

[10] H. Choset, E. Acar, A. Rizzi, and J. Luntz, "Exact cellular decompositions in terms of critical points of Morse functions," in *Proc. IEEE Int. Conf. Robot. Autom.*, San Francisco, CA, 2000, pp. 2270–2277.

[11] H. Choset and J. Burdick, "Sensor-based motion planning: Incremental construction of the hierarchical generalized Voronoi graph," *Int. J. Robot. Res.*, vol. 19, no. 2, pp. 126–148, Feb. 2000.

[12] ——, "Sensor-based motion planning: The hierarchical generalized Voronoi graph," *Int. J. Robot. Res.*, vol. 19, no. 2, pp. 96–125, Feb. 2000.

[13] Y. Gabriely and E. Rimon, "Spanning-tree based coverage of continuous areas by a mobile robot," in Springer Science & Business Norwell, MA, Paper 1573–7470, Mar. 2001, vol. 31, pp. 77–98, to be published.

[14] S. Hert, S. Tiwari, and V. Lumelsky, "A terrain-covering algorithm for an AUV," *Auton. Robots*, vol. 3, pp. 91–119, 1996.

[15] W. H. Huang, "Optimal line-sweep-based decompositions for coverage algorithms," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 1, Seoul, Korea, 2001, pp. 27–32.

[16] J. C. Latombe, *Robot Motion Planning*. Boston, MA: Kluwer, 1991.

[17] V. J. Lumelsky, S. Mukhopadhyay, and K. Sun, "Dynamic path planning in sensor-based terrain acquisition," *IEEE Trans. Robot. Autom.*, vol. 6, no. 4, pp. 462–472, Aug. 1990.

[18] J. Milnor, *Morse Theory*. Princeton, NJ: Princeton Univ. Press, 1963.

[19] C. Ó'Dúnlaing and C. K. Yap, "A "Retraction" method for planning the motion of a disc," *Algorithmica*, vol. 6, pp. 104–111, 1985.
[20] V. S. Spires and S. Y. Goldsmith, "Exhaustive geographic search with mobile robots along space-filling curves," in *Proc. 1st Int. Workshop Collective Robot.*, Paris, France, Jul. 1998, pp. 1–12.
[21] I. A. Wagner and A. M. Bruckstein, "Cooperative cleaners: A study in ant-robotics," Center for Intell. Syst., The Technion, Haifa, Israel, Tech. Rep. CIS-9512, 1995.
[22] A. Zelinsky, R. A. Jarvis, J. C. Byrne, and S. Yuta, "Planning paths of complete coverage of an unstructured environment by a mobile robot," in *Proc. Int. Conf. Adv. Robot.*, Tokyo, Japan, Nov. 1993, pp. 533–538.

Feedback Control of a Cable-Driven Gough–Stewart Platform

Lu Yingjie, Zhu Wenbai, and Ren Gexue

***Abstract*—This paper introduces control tests on a feed positioning system for a large radio telescope. The system is an integration of three subsystems, including a cable trolley subsystem, with a two-degrees-of-freedom orienting mechanism mounted on the trolley, which points a Gough–Stewart mechanism, the third subsystem. The first two subsystems provide a coarse control of the motion of the upper platform of the Gough–Stewart mechanism, while the Gough–Stewart mechanism gives a finer control of the feed-mounted lower platform by actively isolating the vibrations from the upper platform. Standard proportional-integral-derivative controllers with direct terminal feedback from the optical sensors are employed to control the system. Delay-induced dynamic instabilities encountered in the experiments are addressed. A simple error analysis on control with flexible cable is carried to validate the control of the cable-drive architecture. Tests are carried out on a field model of dimension 50 m $\times$ 50 m $\times$ 12 m. Typical experiments manifest remarkable positioning ability of the proposed system.**

***Index Terms*—Cable-drive architecture, feedback control, Gough–Stewart platform, radio telescope.**

I. Introduction

A 500-m aperture spherical telescope (FAST) is proposed to be built in the unique karst area of southwest China [1]. The astronomical receivers of the telescope are required to move on a focus surface halfway from the main reflector, see Fig. 1, with a root-mean-square (RMS) error of 4 mm. The receivers are to be positioned by a cable-driven cabin or trolley in the platformless feed-support systems [2], [3], see Fig. 2. However, the estimated wind-induced vibration of the trolley supported by $600 \sim 700$-m-long cables might be as high as 0.5 m, much higher than the specification. A Gough–Stewart platform [4] oriented by a mechanism mounted on the trolley is used to isolate the lower payload platform from the vibration induced by wind and other disturbances on the system, see Fig. 3.

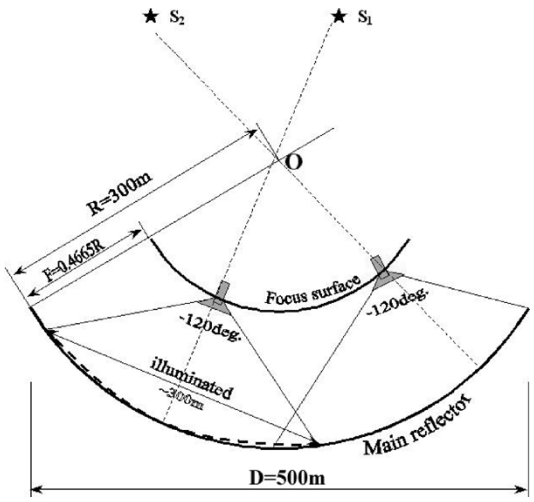

Fig. 1. Optical geometry of FAST.

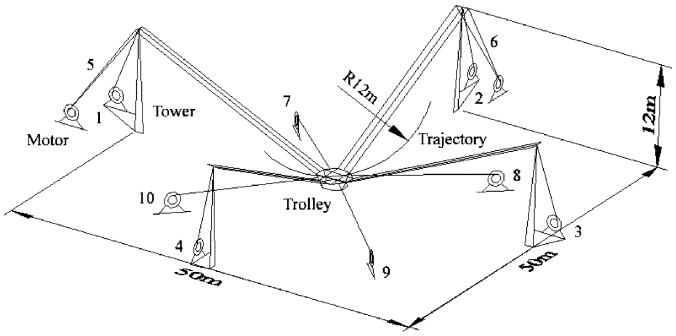

Fig. 2. Overall structure of the field model. The numbers refer to cable identifications.

As shown in Figs. 2–3, the trolley is driven by cables, numbered 1–4, and suspended by cables, numbered 5 and 6. Four pretensioned cables, numbered 7–10, are used to adjust the stiffness of the structure. All the cables can be fed in/out individually by motors. The drive cables and the pretensioned cables are pulled through the fixed trolleys, while the suspension cables run through the fixed pulleys on the towers and those on the trolley.

In the positioning system, the cable-drive architecture is used to realize large movement of the trolley, the orienting mechanism with orthogonal hinged axes is used to adjust the orientation of the upper platform to trace the normal of the spherical focal surface, and the Gough–Stewart platform is used to suppress the vibration of the payload.

The dynamics and control of the Gough–Stewart platform have been studied extensively; however, in most of the previous work, the base platform is fixed on the ground [5]–[7]. Geng and Haynes [8] and Graf and Dillmann [9] have investigated the use of the Gough–Stewart platform for vibration isolation, where vibration of relatively small amplitude is concerned. There are also many publications on the design and control of cable-suspended robots [10]–[14]. In this paper, the Gough–Stewart platform is used in combination with the large-scale cable-drive architecture to provide a high-precision motion platform. Dynamics of such a mechanical system has been studied using analytical and numerical methods [15], [16].

Manuscript received April 3, 2004; revised January 17, 2005 and April 24, 2005. This paper was recommended for publication by Associate Editor W. F. Chung and Editor F. Park upon evaluation of the reviewers' comments. This work was supported in part by the National Astronomical Observatories of CAS under the Research Fund for Large Radio Telescopes, in part by Tsinghua University under the Fundamental Research Fund Grant JC1999031, and in part by the NNSF under Grants 10172049 and 10433020.

L. Yingjie and R. Gexue are with the Department of Engineering Mechanics, Tsinghua University, Beijing 100084, China (e-mail: lu-yj04@mails.tsinghua.edu.cn; Rengx@mail.tsinghua.edu.cn).

Z. Wenbai is with the National Astronomical Observatories, Chinese Academy of Science, Beijing 100012, China (e-mail: wbzhu@bao.ac.cn).

Digital Object Identifier 10.1109/TRO.2005.861459