Ercan U. Acar
Howie Choset
Alfred A. Rizzi
Prasad N. Atkar
Douglas Hull

Carnegie Mellon University, USA

Morse Decompositions for Coverage Tasks

Abstract

Exact cellular decompositions represent a robot's free space by dividing it into regions with simple structure such that the sum of the regions fills the free space. These decompositions have been widely used for path planning between two points, but can be used for mapping and coverage of free spaces. In this paper, we define exact cellular decompositions where critical points of Morse functions indicate the location of cell boundaries. Morse functions are those whose critical points are non-degenerate. Between critical points, the structure of a space is effectively the same, so simple control strategies to achieve tasks, such as coverage, are feasible within each cell. This allows us to introduce a general framework for coverage tasks because varying the Morse function has the effect of changing the pattern by which a robot covers its free space. In this paper, we give examples of different Morse functions and comment on their corresponding tasks. In a companion paper, we describe the sensor-based algorithm that constructs the decomposition.

KEY WORDS—coverage, cellular decompositions, Morse functions

1. Introduction

Conventional motion planning techniques prescribe planners that determine paths between start and goal configurations (Latombe 1991). However, applications such as robotic demining and floor cleaning require a robot to pass over all points in its free space, i.e., follow a path to *cover* the space. Our approach to coverage is based on a hybrid topological and geometric structure called an *exact cellular decomposition*. Exact cellular decompositions represent the free space of a robot by dividing it into non-overlapping regions called *cells* such that the union of the cells fills the free space. Complete coverage is then reduced to ensuring that the robot visits each cell. In this paper, we introduce a new class of exact cellular decompositions, termed *Morse decompositions*, whose cells have simple structure which can be readily covered. Our decompositions apply only to the free spaces of circular or spherical robots.

We define the Morse decompositions using a "slicing method" inspired by Canny (1988a) where a slice is a co-dimension one manifold that foliates the free space. As the slice is swept through the target region, obstacles interact with the slice, either severing it into smaller pieces as the slice first encounters an obstacle or merging smaller pieces into larger pieces as the slice departs an obstacle. In other words, the connectivity of the slice changes as it is swept through the space. We use the connectivity changes to locate the cell boundaries. To characterize these changes, we define the slice as the pre-image of a real-valued function, termed a *slice function*, whose restriction to the boundary of the free space is a Morse function, i.e., has no non-degenerate critical points. At such critical points, we show that the connectivity of the slice changes. Morse theory assures us that between the critical points, "merging" and "severing" of slices do not occur, i.e., the topology of the slice remains constant. This means that the robot can trivially perform simple motions, such as back and forth motions, between critical points and guarantee complete coverage of a cell. Since we use the critical points of Morse functions to form the cells, we term the exact cellular decomposition as a Morse decomposition.

The slice function that determines the exact cellular decomposition also determines the resulting pattern by which the free space can be covered. This flexibility to generate a variety of coverage patterns within the same framework also allows us to address different tasks such as tether path planning. Some conventional planners such as the wavefront and brushfire planners can also be cast within the same framework, as well. Finally, we consider slice functions for applications such as material deposition and plane fuselage inspection that require the extension of planar coverage to coverage of two-dimensional surfaces embedded in $\Re^3$.

In this paper, we focus primarily on Morse decompositions and give an outline of their use for coverage. In Section 2, we overview the existing cellular decomposition methods and identify how our approach differs. We review the definitions

The International Journal of Robotics Research
Vol. 21, No. 4, April 2002, pp. 331-344,

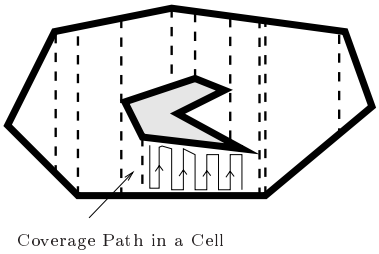

Fig. 1. Trapezoidal decomposition of an environment. Each cell is a trapezoid or triangle that can be covered by back and forth motions.

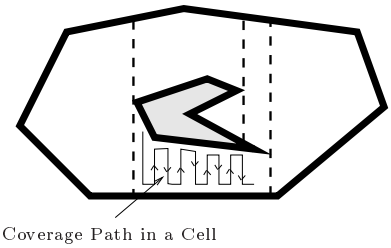

Fig. 2. The boustrophedon decomposition of the polygonal environment shown in Fig. 1. Each cell can be covered by performing back and forth motions parallel to the cell boundaries that are formed in the free space.

of critical points and how they are used to form cellular decompositions in Section 3. We outline a method that uses these decompositions to cover spaces in Section 4. In Section 5, we give a catalog of decompositions using a variety of Morse functions that are suitable for coverage tasks. For coverage in three-dimensional spaces we introduce the decomposition of compact two-dimensional surfaces embedded in three dimensions in Section 6. In a related paper (Acar and Choset 2002), we describe techniques that allow robots to incrementally construct cellular decompositions using only line-of-sight data. This enables a robot to perform coverage of previously unknown areas.

2. Background

Classical robot motion planning determines a path that a robot must follow to reach a goal configuration without passing through any obstacles. Three methods have dominated the motion planning field: potential functions (Khatib 1986; Rimon and Koditschek 1992), exact or approximate cellular decompositions (Chazelle 1984; Schwartz and Sharir 1983), and roadmaps (Canny 1988a). Our work is primarily built upon the ideas that have been used to develop roadmaps and exact cellular decompositions.

Perhaps the most widely known exact cellular decomposition is the trapezoidal decomposition, where each cell is a trapezoid or a triangle (Preparata and Shamos 1985). This method assumes all obstacles in the environment are polygonal and uses the vertices of polygonal obstacles to construct the decomposition. The trapezoidal decomposition is formed by extending a vertical line at each vertex up, down, or both ways until it intersects an obstacle boundary (Figure 1). Since each trapezoid and triangle can be covered by simple back and forth motions, complete coverage becomes trivial once the decomposition is formulated.

From a coverage perspective, a minor short-coming of the trapezoidal decomposition is that many small cells are formed that can seemingly be aggregated with neighboring cells. Reorganizing the cells can result in a shorter and more efficient path to cover the same area. To address this issue, the boustrophedon[1] cellular decomposition approach was introduced (Choset and Pignon 1997). The boustrophedon decomposition is formed by considering the vertices at which a vertical line can be extended both up and down in the free space (Figure 2). Straight line segments constructed at these vertices between the obstacle boundaries form the cell boundaries in the free space.

We generalize the boustrophedon decomposition beyond polygons by borrowing ideas from Canny's work (Canny 1988a, b; Canny and Lin 1993) who first applied a "slicing method" to motion planning. In the plane, a *slice* is a line segment that is swept through the configuration space, $\mathcal{CS}$. In higher dimensional spaces, the slice is a co-dimension one manifold denoted by $\mathcal{CS}_\lambda$. The slices are parameterized by λ (varying λ sweeps a slice through the space) and they foliate the robot's configuration space, $\mathcal{CS}$, i.e., $\mathcal{CS} = \bigcup_\lambda \mathcal{CS}_\lambda$. The portion of the slice in the free configuration space, $\mathcal{FCS}$, is denoted by $\mathcal{FCS}_\lambda$, i.e., $\mathcal{FCS}_\lambda = \mathcal{CS}_\lambda \bigcap \mathcal{FCS}$. Canny (1988a) used the connectivity changes of $\mathcal{FCS}_\lambda$ to ensure the connectivity of his roadmap which is a concise one-dimensional structure that can be used to determine a path between two configurations.

In this work, we use the topology changes of $\mathcal{FCS}_\lambda$, that is defined as the pre-image of functions whose restrictions to the obstacle boundaries are Morse, to decompose the free space for the purposes of coverage. However, Morse functions have already been used to partition spaces for other tasks. Edelsbrunner et al. (2001) consider a vector field induced by a Morse function defined on a piecewise linear two-dimensional manifold, essentially a triangulation of a smooth two-dimensional manifold. Their algorithm decomposes the triangulation into regions of uniform flow. They demonstrate the feasibility of their approach by analyzing triangulated

1. Boustrophedon literally means "the way of the ox" (Webster 1990). Typically, when an ox drags a plow in a field, it crosses the full length of the field in a straight line, turns around, and then traces a new straight line path adjacent to the previous one. By repeating this procedure, the ox is guaranteed to plow the entire field.

geographic terrain data. Their work follows Banchoff (1970) who introduces methods to find the critical points of a height function defined on a polyhedral surface by extending the results of Morse theory for smooth manifolds to polyhedral surfaces. His method examines a neighborhood of each vertex and calculates an index that identifies the type of the critical point, if it exists. Even though these results are applicable to finding the critical points in polyhedral domains, we are interested in methods that allow us to use sensors such as range sensors to locate the critical points. Forman (1998) introduces a discrete Morse theory for CW-complexes, general forms of simplicial complexes, e.g., triangulations. He uses a specific Morse function that has a single real number for each simplex, e.g., triangle. Even though his results are general, the Morse function he is using is not suitable to obtain a decomposition for coverage.

The level set method described by Sethian (1999) has similarities to the "slicing method" that uses Morse functions. The level set method computes and analyzes the motion of a curve/surface as it moves under a velocity field. It is stated that the topological changes of the curve while it is propagating are handled "naturally" but how to locate the points where the topological changes occur is not described. In our work, we are mainly interested in the topological changes that can be determined using sensors but not the propagation of the slice/curve by itself. Our approach differs in the sense that we need to locate these changes to form cells that have simple structure.

3. Morse Decompositions for Coverage

Inspired by Canny's work, we adapt the boustrophedon decomposition[2] approach to handle smooth and polygonal workspaces. Instead of analyzing the vertices of the workspace, we look for connectivity changes of the slice in the free space to locate the boundaries of the cells. When the connectivity increases, new cells are spawned. Conversely when connectivity decreases, multiple cells are completed and a single cell is created or no action is taken (Figure 3). Recall from the previous sections that we refer to critical points as those points where the connectivity of the slice changes. We also know from calculus that at the critical points the "first derivative" becomes zero. In this section, we will relate these two concepts to formulate the cellular decompositions. To do so, we define a slice as the pre-image of a real-valued function, termed a slice function, $h : \Re^2 \to \Re$, i.e., $\mathcal{CS}_\lambda = \{x \in \mathcal{CS} \mid h(x) = \lambda\}$, where $\lambda \in \Re$. Note that our slice function could have $\Re^m$ for $m \geq 2$ as its domain, but for the sake of discussion, for now we will limit ourselves to the plane. Therefore, we consider a compact smooth configuration space of a disk robot with finite boundary, $\mathcal{CS} \subset \Re^2$, populated by smooth configuration space obstacles, denoted as $\mathcal{CC}_i$ for $i \in \{1, \ldots, n\}$. However, we will later extend these results to non-smooth spaces by recourse to generalized gradients.

2. The cells of this decomposition are identical to the channels introduced in Canny and Lin (1993), whose work focused on a new type of roadmap called the Opportunistic Path Planner.

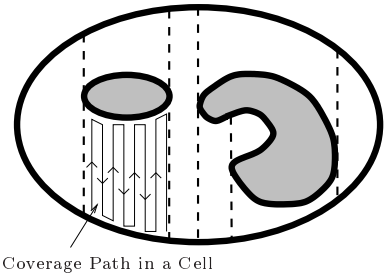

Fig. 3. The boustrophedon decomposition of a non-polygonal environment. As we sweep a straight line slice from left to right, its connectivity in the free space changes first from one to two, then two to one and so forth. At the points where these connectivity changes occur, we locate the cell boundaries in the free space.

3.1. Critical Points

From calculus, we know that a function takes its extrema values at its critical points where its first derivative vanishes or the boundary of its domain. Recall that the differential of a real-valued function $h : \Re^m \to \Re$ at $p \in \Re^m$ is $dh_p = \left[\frac{\partial h}{\partial x_1}(p) \quad \ldots \quad \frac{\partial h}{\partial x_m}(p) \right]$. A point $p \in \Re^m$ is called a critical point of h if $\frac{\partial h}{\partial x_1}(p) = \cdots = \frac{\partial h}{\partial x_m}(p) = 0$. A critical point p is non-degenerate if and only if its Hessian $(\frac{\partial^2 h}{\partial x_i \partial x_j}(p))$ is non-singular. If all the critical points of a function are non-degenerate, then the function is a Morse function (Milnor 1963).

To define our cells, we are interested in the critical points of $h|_{\partial \mathcal{FCS}}$, the restriction of $h(x)$ to the boundary of the free configuration space. Consider Figure 4, which depicts two sample critical points Cp_1 and Cp_2 on the boundaries of the obstacles, $\partial \mathcal{CC}_1 \subset \partial \mathcal{FCS}$ and $\partial \mathcal{CC}_2 \subset \partial \mathcal{FCS}$, respectively. We demonstrate that Cp_1 and Cp_2 are indeed critical points, i.e., $h|_{\partial \mathcal{FCS}}$ takes its local extrema at Cp_1 and Cp_2. Just as the function $h(x)$ can be viewed as measuring the distance between a point $x \in \mathcal{CS}$ and the x_2-axis, $h|_{\partial \mathcal{FCS}}$ measures the distance between points on the obstacle boundaries, $x \in \partial \mathcal{CC}$, and the x_2-axis. Consider a path on $\partial \mathcal{CC}_1$ that passes through Cp_1 as depicted in Figure 4. Moving along the path towards Cp_1 decreases the value of $h|_{\partial \mathcal{FCS}}(x)$. After passing through Cp_1, the value increases. In other words, $h|_{\partial \mathcal{FCS}}(Cp_1)$ is a local minimum of $h|_{\partial \mathcal{FCS}}$. Likewise, $h|_{\partial \mathcal{FCS}}(Cp_2)$ is a local maximum of $h|_{\partial \mathcal{FCS}}$. Since $h|_{\partial \mathcal{FCS}}(Cp_1)$ and $h|_{\partial \mathcal{FCS}}(Cp_2)$ are extrema of $h|_{\partial \mathcal{FCS}}$, Cp_1 and Cp_2 are critical points of $h|_{\partial \mathcal{FCS}}$.

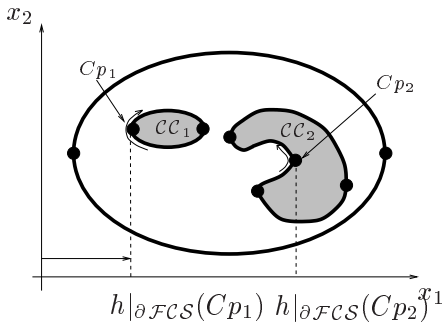

Fig. 4. The restriction of the slice function $h(x) = x_1$ to the obstacle boundaries $h|_{\partial\mathcal{FCS}}$ takes a local minimum at Cp_1 and a local maximum at Cp_2. The values $h|_{\partial\mathcal{FCS}}(Cp_1)$ and $h|_{\partial\mathcal{FCS}}(Cp_2)$ are plotted on the bottom. Since $h|_{\partial\mathcal{FCS}}$ takes its extrema at Cp_1 and Cp_2, they are the critical points.

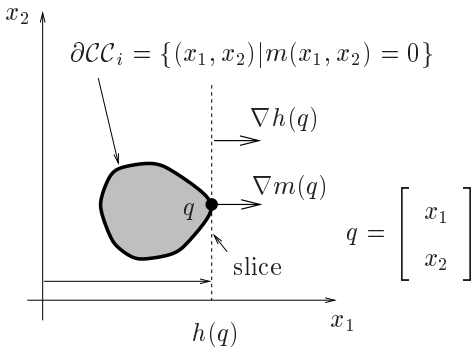

Fig. 5. At the critical point q, the gradient of the slice function $\nabla h(q)$ is parallel to the surface normal of the obstacle $\nabla m(q)$. The slice is tangent to the boundary of the obstacle $\mathcal{CC}_i$ at the critical point q (Corollary 2).

To calculate the critical points of $h|_{\partial\mathcal{FCS}}$, we use the gradient of $h(x)$, $\nabla h(x)$, and surface normals, $\nabla m(x)$, of the obstacles $\mathcal{CC}_i$. We assume that $\partial\mathcal{CC}_i$ is described by the pre-image of a continuous function $m : \Re^m \to \Re$, i.e., $\partial\mathcal{CC}_i = \{x \mid m(x) = 0\}$ (and thus $\nabla m(x)$ is the surface normal of $\partial\mathcal{CC}_i$). The following lemma states a well-known result.

LEMMA 1. (Lagrange Multiplier (Thorpe 1985)) Let S be an n-surface in $\Re^{n+1}$, $S = f^{-1}(c)$ where $f : U \mapsto \Re$ is such that $\nabla f(q) \neq 0$ for all $q \in S$. Suppose $g : U \mapsto \Re$ is a smooth function and $p \in S$ is an extreme point of g on S. Then, there exists a real number λ such that $\nabla g(p) = \lambda \nabla f(p)$. The number λ is called the Lagrange multiplier.

From the above, the following Corollary immediately follows.

COROLLARY 1. The point $x \in \partial\mathcal{CC}_i$ is a critical point of $h|_{\partial\mathcal{FCS}}$ if and only if $\nabla h(x)$ is parallel to $\nabla m(x)$.

It is worth noting that Canny generalized this result for multi-valued functions ($h : \Re^m \to \Re^n$, $m : \Re^m \to \Re^q$), i.e., $x \in \partial\mathcal{CC}_i$ is a critical point of $h|_{\partial\mathcal{FCS}}$, if and only if the following matrix loses its rank (Canny 1988a),

$$D(m,h)_x = \begin{bmatrix} \frac{\partial m_1}{\partial x_1}(x) & \cdots & \frac{\partial m_1}{\partial x_m}(x) \\ \vdots & \ddots & \vdots \\ \frac{\partial m_n}{\partial x_1}(x) & \cdots & \frac{\partial m_n}{\partial x_m}(x) \\ \frac{\partial h_1}{\partial x_1}(x) & \cdots & \frac{\partial h_1}{\partial x_m}(x) \\ \vdots & \ddots & \vdots \\ \frac{\partial h_q}{\partial x_1}(x) & \cdots & \frac{\partial h_q}{\partial x_m}(x) \end{bmatrix}$$

We depict the result of Corollary 1 in Figure 5, where at the critical point q of $h|_{\partial\mathcal{FCS}}$, the gradient of the slice function $\nabla h(q)$ and the surface normal $\nabla m(q)$ are parallel. Instead of considering parallel vectors, we can also consider the tangency of the slice to the obstacle boundaries at the critical points.

COROLLARY 2. The slice is tangent to an obstacle boundary at critical points.

The gradient of the slice function, $\nabla h(x)$, is orthogonal to the slice by definition. At a critical point, this vector is parallel to the surface normal of the obstacle. Therefore, the tangent spaces of the slice and obstacle boundary coincide at the critical point. In this work, we assume that $h|_{\partial\mathcal{FCS}}$ is a Morse function. In fact, we assume that obstacle boundaries have a structure that ensures $h|_{\partial\mathcal{FCS}}$ to be Morse. Using Sard's theorem, Fomenko and Kunii (1997) show that there always exists such a Morse function. In other words, it is always possible to find a real-valued function with non-degenerate critical points by perturbing the given space a small amount. This ensures that the slice and the boundaries of the obstacles never make a second order contact.

3.2. Connectivity Change of the Slice in the Free Space

In this section, we show that critical points of the restriction of the slice function induce changes in the connectivity of the slice in the free space. Note that Canny (1988b) establishes this result for semi-algebraic sets using Morse theory. Here, we consider general obstacle boundaries, including non-smooth ones. We prove the connectivity change in two steps. First, we demonstrate that $h|_{\partial\mathcal{FCS}}{}^{-1}(\lambda)$ changes topology as λ passes through a critical value, and then demonstrate how this change induces a change in the connectivity of the slice in the free space. Since we are considering a real-valued Morse function whose one-dimensional range is ordered, we can order the critical values of the Morse function. Assuming only one critical point per slice, adjacent critical points are

those whose critical values are "next" to each other. In other words, let Λ be the set of all critical values. The critical values $\lambda_1, \lambda_2 \in \Lambda$ are adjacent if for all critical values in Λ there does not exist a critical value $\bar{\lambda}$ such that $\lambda_1 < \bar{\lambda} < \lambda_2$. Morse theory asserts that between adjacent critical points of a Morse function, the topology of the manifold on which the Morse function is defined does not change (Milnor 1963). In the context of the slice function h, Morse theory states that there exists a diffeomorphism ϕ such that for all $\lambda_1, \lambda_2 \in (\lambda_*, \lambda^*)$, $\phi(h|_{\partial \mathcal{FCS}}{}^{-1}(\lambda_1)) = h|_{\partial \mathcal{FCS}}{}^{-1}(\lambda_2)$, where λ_* and λ^* are adjacent critical values of a real-valued Morse function.

LEMMA 2. The topology of the slice in the free configuration space changes only at critical points of the slice function restricted to the boundary of the configuration space obstacles.

Proof. Let $\mathcal{CC} = \bigcup_i \mathcal{CC}_i$ and $\mathcal{CC}_\lambda = \mathcal{CC} \bigcap h^{-1}(\lambda)$. Let $\mathcal{FCS} = \mathcal{CS} \backslash \bigcup_i \mathcal{CC}_i$ and $\mathcal{FCS}_\lambda = \mathcal{FCS} \bigcap h^{-1}(\lambda)$. Let $\partial \mathcal{FCS} = \bigcup_i \partial \mathcal{CC}_i$ and $\partial \mathcal{FCS}_\lambda = \partial \mathcal{FCS} \bigcap h^{-1}(\lambda)$. When λ is a regular value, $\partial \mathcal{FCS}_\lambda = \partial \mathcal{CC}_\lambda = \partial \mathcal{FCS}_\lambda$, i.e., $\partial \mathcal{FCS}_\lambda$ is a separator for the set $\mathcal{FCS}_\lambda \bigcup \mathcal{CC}_\lambda$. In the planar case, $\partial \mathcal{FCS}_\lambda$ comprises isolated points.

Morse theory assures us that the topology of $\partial \mathcal{FCS}_\lambda$ remains constant between critical points and only changes at critical points. Since $\partial \mathcal{FCS}_\lambda$ is a separator for the set $\mathcal{FCS}_\lambda \bigcup \mathcal{CC}_\lambda$, the topology of $\mathcal{FCS}_\lambda \bigcup \mathcal{CC}_\lambda$ must change when the topology of $\partial \mathcal{FCS}_\lambda$ changes. □

The above lemma can be interpreted in the planar case as follows. The pre-image of $h|_{\partial \mathcal{FCS}}$ are the end points of alternating intervals: an interval in the free configuration space, a subset of $\mathcal{FCS}_\lambda = \mathcal{FCS} \bigcap \mathcal{CS}_\lambda$, and an interval in the obstacle, a subset of $\mathcal{CC}_\lambda = \mathcal{CC} \bigcap \mathcal{CS}_\lambda$. Therefore, by Lemma 2, when the number of end points changes, so must the number of intervals. Moreover, these intervals must alternate between obstacle and free space. A change in the number of intervals corresponds to a change in connectivity of the slice in the free space (Figure 6).

3.3. Definition of Morse Decomposition

The result of Lemma 2 assures us that between "adjacent" critical points, the connectivity of the slice in the free space does not change. This means that between critical points, we can plan a simple routine coverage path that mainly comprises paths along the intervals of the slice. Essentially, using the critical points we can decompose the space into cells such that each one can be covered with back and forth motions.

The boundaries of the cells have two parts: portions of obstacle boundaries and intervals whose closures contain critical points. Let the interval $\mathcal{FCS}_\lambda^j$ to be the jth open connected component of $\mathcal{FCS}_\lambda$. Now, $\mathcal{FCS} = \bigcup_\lambda \bigcup_j \mathcal{FCS}_\lambda^j$. We denote the set of intervals that contain a critical point by I, i.e., $I = \bigcup_{\Lambda_k} \bigcup_j \mathcal{FCS}_{\Lambda_k}^j$, where $\mathcal{FCS}_{\Lambda_k}^j$ contains a critical point Cp_k. Note that a critical point cannot be in the interior of an interval; it can only lie at the end points of an interval.

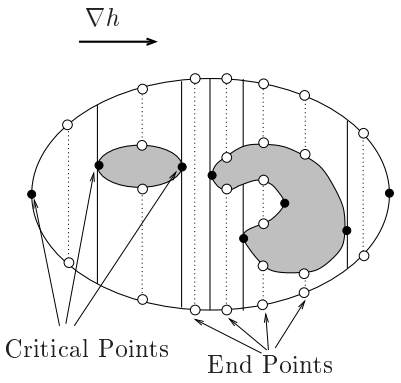

Fig. 6. The number of end points (white dots) does not change between adjacent critical points (black dots) and thus the connectivity of the slice does not change in the free space.

DEFINITION 1. (Morse Decomposition) Morse Decomposition is an exact cellular decomposition whose cells are the connected components of $\mathcal{FCS} \backslash I$.

Cells of a Morse decomposition are open subsets of the free space (Figure 7).

3.4. Non-Smooth Boundaries

The Morse theory that we use for decomposition is valid for the functions that are defined on a smooth compact boundary and it is not valid when the boundary is non-smooth. However in practice, we cannot avoid non-smooth boundaries. The main problem with a non-smooth boundary is that the surface normals at the non-smooth boundary points are not defined. However, we can use Clarke's generalized gradient to define the surface normals at these non-smooth points (Clarke 1990). The generalized gradient is the set of vectors within the convex hull of $\nabla m^1(x)$ and $\nabla m^2(x)$ where $\nabla m^1(x)$ and $\nabla m^2(x)$ are the surface normals of the adjacent smooth surfaces that are subsets of a piecewise smooth surface (Figure 8).

Degiovanni shows that Morse theory can be applied to the non-smooth boundaries using the generalized gradients as long as the slice function is Lipschitz[3] (Degiovanni 1997). In this work, we assume that all the smooth and non-smooth obstacle boundaries that we consider in the real-life applications lead to a Lipschitz slice function. This assumption requires that at the non-smooth points, the generalized gradient should not be an empty set. With this assumption, we can safely use Morse theory with the generalized gradient on non-smooth boundaries to achieve an exact cellular decomposition in terms of critical points. At a non-smooth point, if the gradient of the slice function $\nabla h(x)$ is contained within

3. A function is said to be Lipschitz at x, if for all y in the neighborhood of x, $|h(x) - h(y)| \leq K|x - y|$, where $K > 0$.

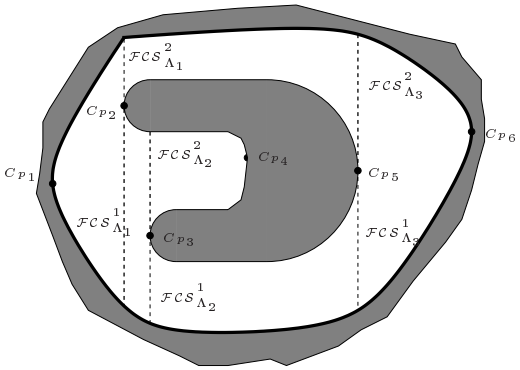

Fig. 7. $\mathcal{FCS}^j_{\Lambda_k}$'s are the intervals that contain a critical point in its boundary. Cells are formed by taking away all $\mathcal{FCS}^j_{\Lambda_k}$ from the free configuration space $\mathcal{FCS}$. Note that in this example, we use the function $h(x) = x_1$.

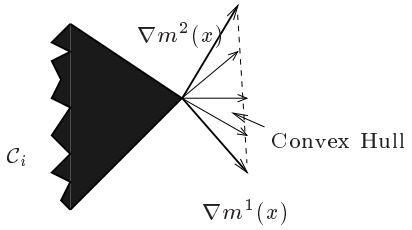

Fig. 8. The generalized gradient is the set of vectors within the convex hull of the surface normals $\nabla m^1(x)$, $\nabla m^2(x)$.

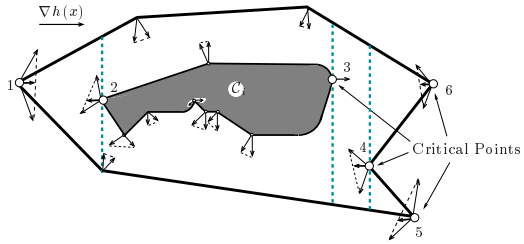

Fig. 9. At the non-smooth critical points 1, 2, 4, 5, 6, $\nabla h(x)$ or $-\nabla h(x)$ is contained within the convex hull. At the smooth critical point 3, the generalized gradient is the same as the surface normal.

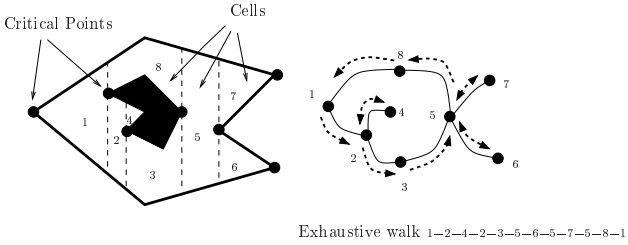

Fig. 10. The boustrophedon decomposition of a space and its adjacency graph. The nodes represent the cells and the edges indicate the adjacent cells. An exhaustive walk on the graph is generated by a depth-first search algorithm. Note that when the robot is done covering cells 4, 6, 7, it has to traverse the respective cell to reach the uncovered cells.

the convex hull of the generalized gradient, we can locate a non-smooth critical point. Note that at the smooth boundary points, the generalized gradient reduces to the conventional gradient (Figure 9).

4. Coverage Using Morse Decompositions

Once the decomposition is completed, the planner determines a coverage path in two steps. First, the planner determines an exhaustive walk through the adjacency graph (Figure 10). This list can be computed using a depth-first search-like algorithm. Once the ordered cell list is determined, the planner then computes the explicit robot motions within each cell. Note that the coverage path determined in the configuration space guides the robot in the workspace such that the robot itself passes over all the points in "reachable" the workspace. These motions are easy to determine because of the simple structure of the cells of which Morse theory assures us. A coverage path within a cell contains two parts: motion along the slice and motion along the boundary of the obstacles (Figure 11). A bulk of the coverage operation occurs with motions along the slices. Motion along a particular slice terminates when the robot encounters an obstacle. Motion along the boundary of the obstacle directs the robot to move "one width[4] over," i.e., increase its slice function value by one robot width while following the cell boundary along an obstacle. Note that it is possible to have two obstacles so close to each other such that a cell with a width smaller than the one robot width can be formed. To deal with such cases we introduce an algorithm in our related work (Acar and Choset 2002).

Since the adjacency graph is connected, the union of the cells corresponding to the nodes fills the free space and coverage within each cell is trivial, complete coverage reduces to finding an exhaustive walk through the adjacency graph. In general, each cell in the exhaustive walk may not be unique. When a robot passes through an already covered cell, instead of covering it again, it simply plans a path to the next cell in the exhaustive walk list using a conventional path planner.

4. Here, width is determined by the size of the detector or end-effector that is being used.

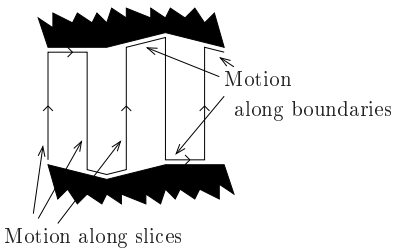

Fig. 11. To cover each cell, the robot performs simple back-and-forth motions. These motions are easy to determine because of the simple structure of the cells of which Morse theory assures us. The path contains two parts: motion along the slice and motion along the boundary of the obstacles.

5. Coverage with Different Patterns

All of the figures and examples in the previous section assumed a slice function $h(x, y) = x$ which emits the boustrophedon decomposition and its farming-like coverage pattern. In this section, we consider different "shapes" of slices that result in different decompositions. We vary the shape of the slice by changing the slice function to yield different decompositions and coverage patterns.

5.1. Spiral, Spike, and Squarel Patterns

The function $h(x) = \sqrt{\sum_i (x_i)^2}$ emits a pattern of concentric circles in the plane, spheres in $\Re^3$, and S^{m-1} in $\Re^m$. By Corollary 2, critical points occur at points where the circle or sphere is tangent to an obstacle. Critical points are then used to form annular-arc shaped cells and the adjacency graph (Figure 12). Just as before, a planner determines a coverage path in two steps: first it finds an exhaustive walk through the adjacency graph and then it plans the explicit coverage path in each uncleaned cell. Now, the coverage pattern within a cell has three parts: motion along a slice, motion orthogonal to the slice, and motion along the boundary of the cell. The slice here, however, is not a straight line segment, but rather a circle or subset of a circle. Therefore, in the plane, a planner initially directs the robot to circumnavigate a circle, move the inter-lap distance along the radius of the circle and circumnavigate a circle of a larger radius again. If the robot encounters an obstacle while circumnavigating a circle, the planner simply directs the robot to follow the obstacle boundary until the robot has moved an inter-lap distance in the radial direction of the decomposition. Note that we can view this method as growing a circle through a cell.

Note that in $\Re^2$, instead of following a circle and stepping outward, the robot can follow a spiral pattern until it encounters critical points (Figure 12). The spiral pattern bypasses the need to step along the radial direction. This yields a path that maximizes area covered per unit distance traveled in regions sparsely populated with obstacles.

The function $h(x) = \tan(\frac{x_2}{x_1})$ induces a pattern that is orthogonal to the set of concentric circles (Figure 13). Using this pattern to perform coverage has the effect of covering the region closest to the center of the pattern more densely. This is useful if the likelihood of finding a desired object is highest at the center of the pattern and the robot's detector experiences false negatives (something is under the detector but the detector does not sense it).

The function $h(x) = |x|_1$ produces cells that look like rotated squares or diamonds (Figure 14). For coverage, instead of driving in concentric squares, we can direct the robot to "spiral" out while looking for critical points, hence the term *squarel*. The resulting pattern is shown in Figure 14. The squarel pattern serves as an approximation to the spiral pattern that is easier to implement on differentially driven robots.

Note that $|x|_1$ is not smooth so we have to use the formulation given in Section 3.4. The squarel pattern has two parts: a flat line and a ninety-degree turn, as can be seen in Figure 14. Note that critical points occur when the flat portions of $\{x : |x|_1 = \lambda\}$ become tangent to an obstacle and at some of the ninety-degree turn points. At these points, the obstacle surface normal lies in the convex hull of the two flat portions that meet at the ninety-degree turn point.

5.2. Brushfire Decomposition

The brushfire algorithm (Latombe 1991) is a popular technique to construct the generalized Voronoi diagram (Figure 15). This algorithm has its name because in implementation, imaginary wavefronts emanate from each obstacle and collide at points on the generalized Voronoi diagram. By noting the location of the collision points, the algorithm constructs the Voronoi diagram. The algorithm also induces a decomposition that is *not* the Voronoi regions of the generalized Voronoi diagram. Instead, the decomposition models the topology of the wavefronts as they initially collide with each other and form or destroy new wavefronts. (Compare Figures 15 and 16.)

The distance function D that measures the distance between the point x and the nearest point c on the closest obstacle C_i, i.e., $D(x) = \min_{c \in \bigcup C_i} \|x - c\|$, admits a decomposition termed the brushfire decomposition. Each slice of $D(x)$ is a wavefront where each point on the front has propagated a distance λ from the closest obstacle. As λ increases, the wavefronts progress. Cells of the brushfire decomposition are formed when these wavefronts initially collide. Figure 16 contains a decomposition induced by D where regions of the same color represent a cell. Whereas for the boustrophedon decomposition we are essentially "pushing" a line segment through the cell, here we are "growing" a wavefront that originates on the boundary of the environment, which in Figure 16 has

Fig. 12. Cellular decomposition for $h(x) = (x_1)^2 + (x_2)^2$ and an associated spiral coverage pattern. The slices are the circles that are the pre-images of h. At the critical points, demarked with little circles (not to be confused with the slices), the circle-shaped slices become tangent to the obstacles. Rather than moving along circular paths and stepping outward, the robot follows a spiral pattern.

Fig. 13. Decomposition for $h(x) = \tan(\frac{x_2}{x_1})$ and a spiked pattern. The free space is sliced like a pie. At the critical points, the slices are tangent to obstacles. The robot can use this pattern to cover more densely the region closest to the center of the pattern.

Fig. 14. Decomposition for $h(x) = |x|_1$ and a coverage pattern. Squares are the slices and at the critical points the corner of a square touches an obstacle and/or side of it becomes tangent to an bstacle. Since it is easier for the robot to move along straight lines rather than circles, this pattern can be used to approximate the spiral pattern.

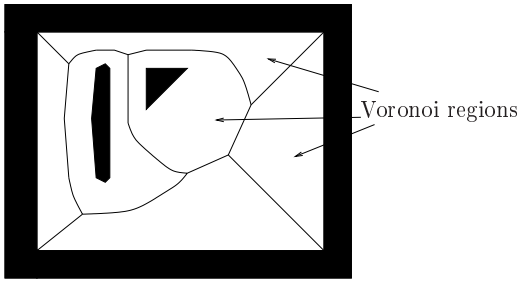

Fig. 15. Generalized Voronoi diagram of an environment.

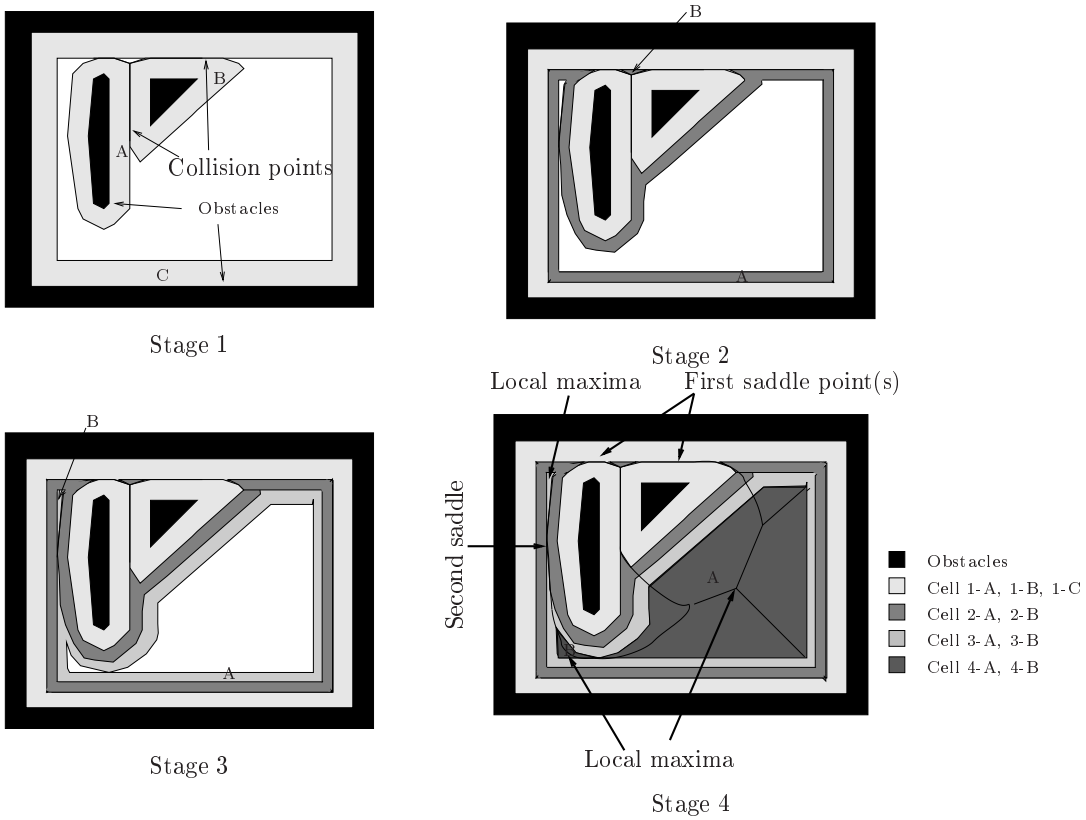

Fig. 16. Incremental construction of the cells of the brushfire decomposition. First, there are three cells denoted by 1-A, 1-B, and 1-C that all have the same range of λ values. These cells merge to form cells 2-A and 2-B. Then, there are two cells, 3-A and 3-B. Finally, there is one big cell 4-A and one small cell 4-B. The wavefronts collide with each other at the points located on the Voronoi diagram.

three disconnected obstacles: the exterior, the vertical bar-like obstacle, and a triangle. These three wavefronts progress until they initially collide with each other, which occurs at critical points. The light gray regions adjacent to the obstacles represent the three newly formed cells. The type of critical points that define the gray regions in Figure 16 are saddle points. In fact, all of the cells are defined by saddle points of D. Note that since D is non-smooth, its generalized gradient must be used as well.

To determine the coverage path, the planner again first derives the decomposition, finds an exhaustive walk through the adjacency graph, and then plans the coverage path within each cell. The coverage path within the cell consists of three parts: motion along the slice, motion orthogonal to the slice, and motion along the boundary of the cell (Figure 17). Motion along the slice is similar to before, except here the robot follows a path at a fixed distance to the nearest obstacle. The robot follows an obstacle boundary at a fixed distance until it returns to its starting point or a point where distance to two obstacles becomes the same. When the robot returns to its starting point, it simply moves away from the closest obstacle by one robot width and repeats following the obstacle boundary at the new fixed distance. When the robot becomes doubly equidistant, it is on the boundary of the cell at which point it follows the Voronoi diagram. The robot follows the Voronoi diagram until it reaches a point where the distance to the nearest obstacle is an integer multiple of the robot's diameter, at which point the robot then resumes obstacle boundary following at this fixed distance. Here, the boundary following motion is much different from before because cell boundaries lie exclusively in the free space with the exception of the first slice $\lambda = 0$.

The pattern induced by the brushfire algorithm is ideally suited for coverage with mobile robots experiencing dead-reckoning error but have long sensing range. A mobile robot can follow this pattern servoing off of the boundaries of the obstacles by moving forward and maintaining a fixed distance to the wall (Figure 17). Since the robot is servoing off of range readings to the obstacle boundary, this method is insensitive to dead-reckoning error. This benefit, however, requires that the robot can indeed measure distance to the obstacle which

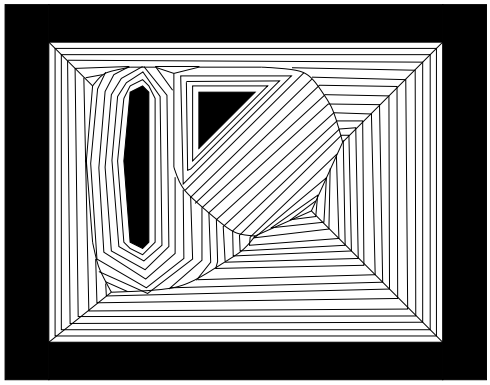

Fig. 17. Coverage pattern for the brushfire decomposition. To generate this pattern, the robot follows the boundaries of obstacles and thus it has a continuous robust reference for localization. Therefore this pattern is suitable for the robots that are prone to accrue dead-reckoning error. However, the robot heavily relies on long-range sensors.

could be far away from the robot. This contrasts the boustrophedon decomposition approach which requires only very limited sensing range but is sensitive to dead-reckoning error.

We would like to point out one difference between the Morse functions that we use to obtain the brushfire decomposition and those that we used in the previous sections. In the previous sections, the restriction of the slice functions to the obstacle boundaries were Morse functions. However, the slice function $D(x)$ for the brushfire decomposition, by itself, is a Morse function. We do not restrict $D(x)$ to the obstacle boundaries to obtain a Morse function. The critical points of $D(x)$ occur in the free space. Therefore, Lemma 2 no longer directly applies. However, using Morse theory for manifolds with boundaries (Handron 1999), we need to adapt Lemma 2 to rigorously analyze the brushfire decomposition. Note that at the critical points of $D(x)$, there are still topology changes. For example from stage-2 to stage-3 of Figure 16, the number of connected components of the boundaries (in fact slices) of the cells of A and B does not change, i.e., from two-connected to two-connected. However, the topology of the space changes. In other words, topology changes do not occur smoothly. One cell disappears and another cell appears. In general, Morse theory does not guarantee that the topology of a space changes at critical points. It guarantees, however, that the topology of the space remains fixed between adjacent critical points. Handron generalizes Morse theory for manifolds with boundaries and provides a necessary condition for topology changes at the critical points. This applies to brushfire and also in the next section to wavefront decompositions.

5.3. Wavefront Decomposition

Let $h(x)$ be the shortest path length of a point x to a fixed location. The level sets $h^{-1}(\lambda)$ foliate the free space where, for a given λ, the set of points in $h^{-1}(\lambda)$ are λ away from the fixed point in the free space. This particular function is sometimes termed the wavefront potential where λ parameterizes the wave or level set of h emanating outward from the fixed point. For the wavefront potential, the fixed point is the goal and when the front crosses the start, the planner can backtrack a path from the goal to the start (Jarvis 1985). This function also emits a cellular decomposition, which is particularly useful for coverage by a tethered robot where the robot's tether is incrementally fed and the robot sweeps out curves each at constant tether length.

Critical points occur both when the wavefront becomes tangent to the obstacles as well as when the wavefront arrives initially at the obstacle. This occurs first in upper left portion of obstacle C_1 in Figure 18. Note how, once the waves collide, they propagate as one wave with a non-smooth point that originated at the critical point. In fact, this non-smooth point traces the set of points of equal path-length to the goal for two classes of paths, one to the right of the obstacle and one to the left.

We would like to point out that continuous Dijkstra method introduced by Mitchell (1996) and later improved by Hershberger and Suri (1999) uses wavefront propagation to calculate the shortest paths between any point in the space and a "source" point. In this section, we demonstrated that the wavefront decomposition falls into the class of Morse decompositions that are suitable for coverage tasks.

5.4. Tethered Path Planning

Path planning for tethered robots (Xavier 1999; Sinden 1990) provides more challenges than conventional untethered planning because the tether path is history-dependent. We capture this path dependency by modifying the wavefront decomposition. The modified decomposition provides a rich representation of the tethered robot free space which can be used to generate a variety of paths.

We consider a circular mobile base with a tether assuming that the tether remains taut and cannot be extended beyond a specified maximum length but it may cross over itself. Let $\mathcal{WS} \subset \Re^2$ be a bounded workspace. The configuration space of the mobile robot without tether is also $\Re^2$ with punctures. Since there is a natural identification between $\mathcal{WS}$ and $\mathcal{CS}$ of this mobile robot, we model the mobile robot as a point in a modified workspace just for explanation purposes. The configuration space for a tethered robot can be envisioned as a plane having "flaps" cut out of it that wrap around obstacles. For the simple case of one obstacle in the workspace, the configuration space could have two flaps, one corresponding to paths that take the tethered robot left around the workspace obstacle and the other corresponding to the right (Figure 19). While points on different flaps may map to the same workspace points, traveling from one flap to another cannot be done directly.

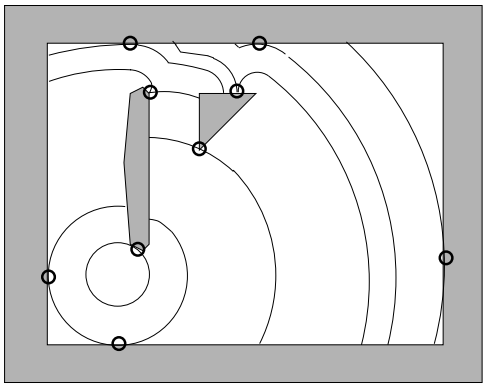

Fig. 18. Wavefront decomposition. Cusp points on the wavefronts originate from the critical points, e.g., the cusp point on the upper boundary of the obstacle located on the left.

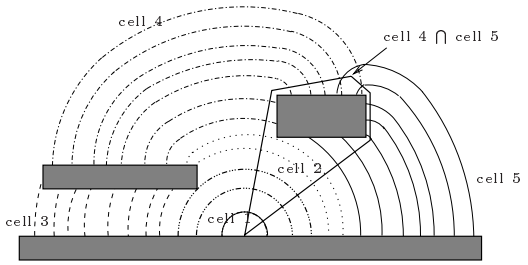

Fig. 19. Coverage path for a tethered robot. In each cell, paths are denoted with different style curves. Points in the intersection of the projected cells can be reached by more than one path.

The decomposition for a tethered robot will bear many similarities to the wavefront decomposition. Recall from the wavefront decomposition that when wavefonts meet, they partially annihilate each other and form a new front. Now, we allow the wavefronts to "seemingly" flow through each other in the workspace, i.e., the same goal configuration can be reached with two different tether paths. In reality, the wavefronts are not flowing through each other but rather each wavefront is flowing through its own flap in the configuration space. Whereas the Morse function used to define the wavefront decomposition was the shortest path length to a start point from any point in the workspace, the Morse function used to define a decomposition for tethered robots will be the shortest path in the configuration space of the tethered robot. We define the decomposition using critical points of the tether length function, defined as $tl_u : x \in \mathcal{CS} \rightarrow \Re$. The value $tl_u(x)$ is the length of the tether when it is taut between the points x and u where u is the fixed point of the tether.

Just like the previous planning methods, path planning can be achieved by first searching through a cell list. Then within each cell, motion along a slice is constrained by the tether length. Whenever the robot is done moving along a slice, the tether length is extended by a fixed amount and the robot resumes the motion along another slice (Figure 19).

6. Cell Decomposition for Surface Deposition

In this section, we take the first step towards lifting the results of the previous sections into three dimensions for applications such as inspection of non-planar surfaces, material deposition, material removal, and CNC tool path planning. The Morse function that we use for decomposition is $h : M \rightarrow \Re, h(p) = \lambda,\ p \in M$ where M is an oriented, compact two-dimensional

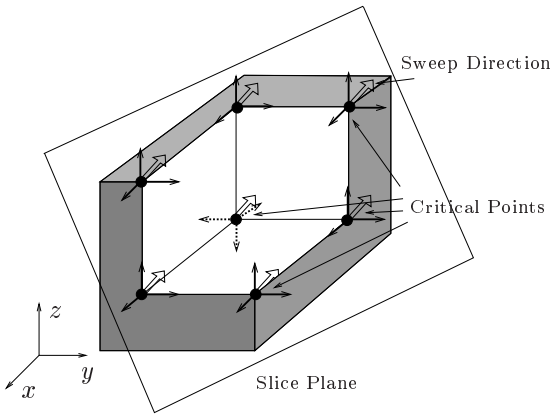

Fig. 20. The critical points are located at the corners where the sweep direction lies in the positive or negative span of the generalized gradients.

surface that is embedded in $\Re^3$. The pre-images of h are the slice planes that are swept through the space as the value of λ changes. Just as in the previous sections, critical points occur when the gradient of h becomes parallel to the surface normal of the surface. If the surface is non-smooth, we use the generalized gradients in three dimensional space. Critical points occur when the sweep direction lies in the positive or negative span of the generalized gradients (Figure 20).

However, since the end-effector must follow paths at a distance away from the surface M for applications such as painting, we must locate the critical points on an "offset surface". The details of how to compute the critical points on this offset surface are not pertinent to this paper. However, we would like to point out that the mapping of the critical points on the surface M to the ones on the offset surface is in general not one-to-one.

In Figure 21, we depict the cellular decomposition using the canonical example of Morse theory literature. The cells

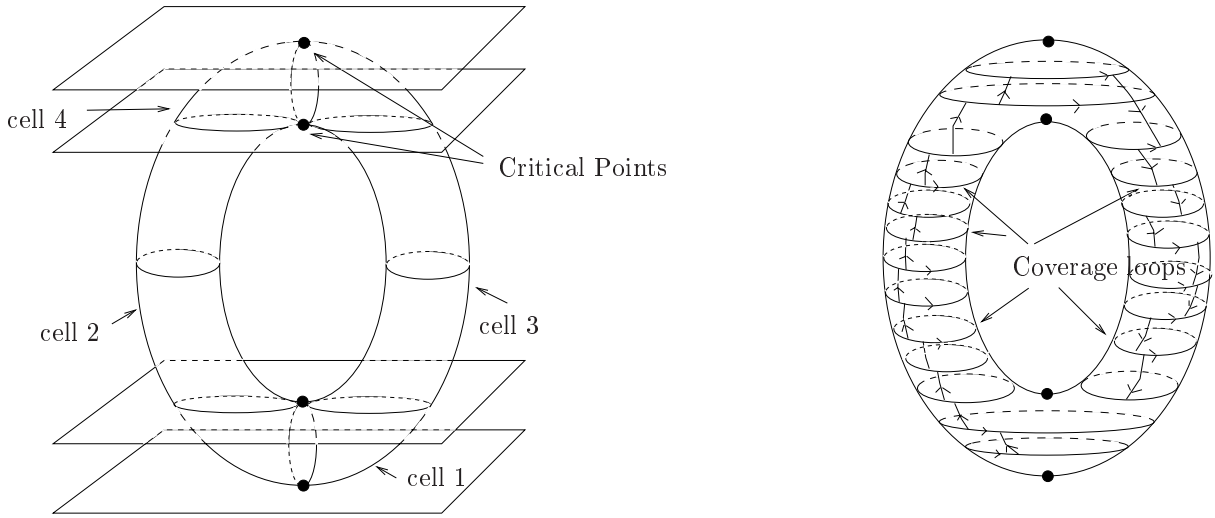

Fig. 21. The cellular decomposition of the torus, its adjacency graph and the coverage paths. To cover a cell, the planner guides an end effector along the loops. Whenever a loop is completed, the end ffector is shifted to an upper or lower loop.

lie between the planes that pass through the critical points. The cellular decomposition is encoded using an adjacency graph to keep track of the covered cells. As before, a planner first generates a sequence of cells to be covered. Then within each cell, the planner guides an end effector along the loops. Essentially, this motion corresponds to motion along the slices. Just like before, we are pushing a slice (loop) through the surface of the object. Whenever a loop is completed, the end effector is shifted to an upper or lower loop. In Figure 22 coverage paths on an object with a hole in it and on a mock plane fuselage generated by simulation are shown.

7. Conclusion

In this paper, we have described a formulation that provides a common framework for a variety of coverage tasks. We started with conventional slice algorithms where the slice is normally a flat plane defined by the pre-image of a real-valued function whose restriction to obstacle boundaries is Morse. We showed that the topology of the slice in the free space changes at the critical points of the Morse function. We used these critical points to form the cell boundaries such that the structure of each cell enables a planner to use simple control strategies such as back-and-forth motions for coverage. We furthered this approach by altering the Morse function, and hence the slice geometry, to accomplish different tasks.

Our view is that physical properties and constraints inherent to the robot can help to guide the construction of Morse functions to achieve a task. For example, if a robot can reliably measure distance to the nearest obstacles, one can derive a function based on range information, such as the distance function used in the brushfire decomposition. The brushfire decomposition is particularly well-suited to coverage by a robot that experiences dead-reckoning error because the robot can servo off of range information to the nearest obstacle to guarantee coverage in the interior of the free space. If we are planning coverage paths for a tethered robot, then the modified wavefront decomposition is a well-accommodated decomposition of the free workspace.

By varying the domain of the Morse function, we lifted this approach into three dimensions. We introduced a cellular decomposition to cover a closed, orientable, and connected surface embedded in $\Re^3$. A slice in this case is a two-dimensional sheet that intersects the surface; these intersections form the paths that cover the surface. This work will serve as the first step towards our goal of robotic surface deposition.

The rigorous formulation of critical points has enabled us to describe methods by which we can sense critical points using conventional range sensors. With this technique, we can readily prescribe methods by which we can construct a cellular decomposition on-line using sensor data, which enables us to cover unknown spaces. In a related paper (Acar and Choset 2002), we demonstrate the details of sensor-based construction of cellular decompositions for the purposes of provably complete coverage in unknown spaces.

Acknowledgments

We would like to thank Dr. Jonathan E. Luntz, Dr. Teruko Yata, and members of M-Lab of Carnegie Mellon's Robotics Institute for their invaluable suggestions through out our work. The authors gratefully acknowledge the support of the Office of Naval Research Young Investigator Program Grant #N00014-99-1-0478, Tom Swean, and the Naval Explosive Ordnance Division for support of this work.

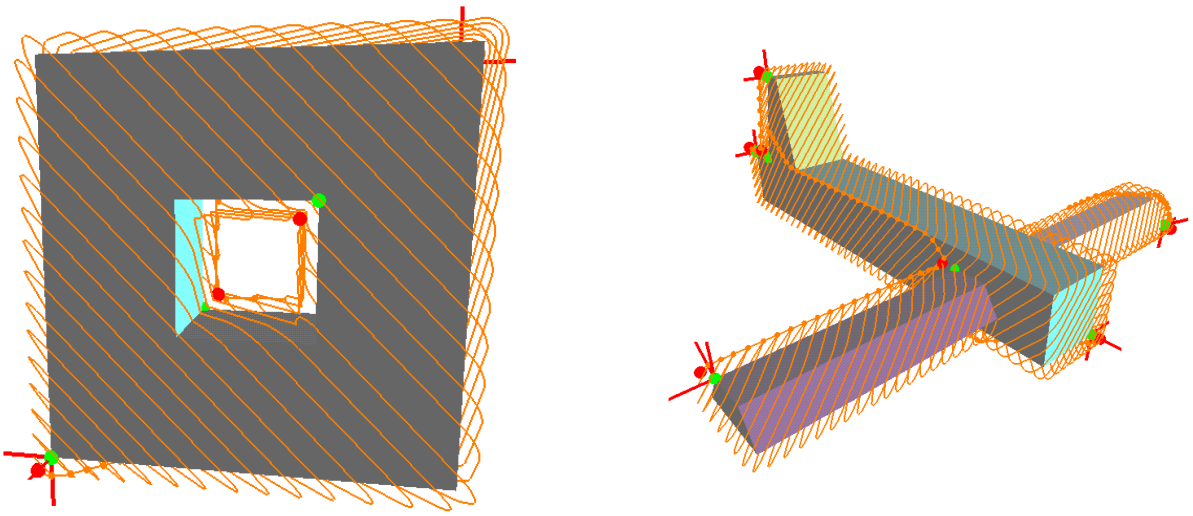

Fig. 22. Coverage loops on an object that is homeomorphic to the torus and on a mock plane fuselage. Coverage loops are shown at a fixed distance away from the surface for visualization purposes. The darker dots represent the critical points on the offset surface and the lighter ones depict the critical points on the surface itself.

References

Acar, E. U., and Choset, H. 2002. Sensor-based coverage of unknown environments: Incremental construction of morse decompositions, *The Int. J. Robotics Research* 21(4):345–366.

Banchoff, T. F. 1970. Critical points and curvature for embedded polyhedral surfaces. *American Mathematical Monthly*, 77:475–485.

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

Canny, J. F. 1988. Constructing roadmaps of semi-algebraic sets I: Completeness. *Artificial Intelligence* 37:203–222.

Canny, J. F., and Lin, M. 1993. An opportunistic global path planner. *Algorithmica* 10:102–120.

Chazelle, B. 1984. Convex partition of polyhedra: A lower bound and worst-case optimal algorithm. *SIAM Journal on Computing* 13(3):488–507.

Choset, H., and Pignon, P. 1997. Coverage path planning: The boustrophedon decomposition. In *Proceedings of the International Conference on Field and Service Robotics*, Canberra, Australia.

Clarke, F. H. 1990. Optimization and nonsmooth analysis. *Society of Industrial and Applied Mathematics*, Philadelphia, PA.

Degiovanni, M. 1997. Nonsmooth critical point theory and applications. *Nonlinear Analysis, Theory, Methods and Applications* 30:89–99.

Edelsbrunner, H., Harer, J., and Zomorodian, A. 2001. Hierarchical morse complexes for piecewise linear 2-manifolds. *ACM Symposium on Computational Geometry*, Medford, MA.

Fomenko, A. T., and Kunii, T. L. 1997. *Topological Modeling for Visualization.* Tokyo: Springer-Verlag.

Forman, R. 1998. Morse theory for cell complexes. *Advances in Mathematics* 134:90–145.

Handron, D.G.C. 1999. *Generalized Billiard Paths and Morse Theory for Manifolds with Corners*, Ph.D. thesis, Rice University, Houston, Texas.

Hershberger, J., and Suri, S. 1999. An optimal algorithm for euclidean shortest paths in the plane. *SIAM Journal on Computing* 28(6):2215-2256.

Jarvis, R. A. 1985. Collision free trajectory planning using distance transforms. *Mech. Eng. Trans. of the IE Aust.* ME10:197–191.

Khatib, O. 1986. Real-time obstacle avoidance for manipulators and mobile robots. *Int. J. Robotics Research* 5(1):90-98.

Latombe, J. C. 1991. *Robot Motion Planning*. Boston, MA: Kluwer Academic Publishers.

Milnor, J. 1963. *Morse Theory*. Princeton, New Jersey: Princeton University Press.

Mitchell, J.S.B. 1996. Shortest paths among obstacles in the plane. *Int. J. Computational Geometry and Applications* 6(3):309–332.

Preparata, F. P., and Shamos, M. I. 1985. *Computational Geometry: An Introduction*. Springer-Verlag.

Rimon, E., and Koditschek, D. E. 1992. Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation* 85:501–518.

Schwartz, J. T., and Sharir, M. 1983. On the "piano movers" problem II: general techniques for computing topological properties of real algebraic manifolds. *Advances in Applied Mathematics* 4(1):298–351.

Sethian, J. A. 1999. *Level Set Methods and Fast Marching Methods*. Cambridge University Press.

Sinden, F. W. 1990. The tethered robot problem. *Int. J. Robotics Research* 9:122–133.

Thorpe, J. A. 1985. *Elementary Topics in Differential Geometry*. Springer-Verlag.

Webster. 1990. *Webster's Ninth New Collegiate Dictionary*. Springfield, MA: Merriam-Webster, Inc.

Xavier, P.G. 1999. Shortest path planning for a tethered robot or an anchored cable. In *Proc. of IEEE ICRA*, Detroit, MI, 2:1011–1017.