Proceedings of the 2000 IEEE
International Conference on Robotics & Automation
San Francisco, CA • April 2000

Exact Cellular Decompositions in Terms of Critical Points of Morse Functions

Howie Choset, Ercan Acar, Alfred A. Rizzi, and Jonathan Luntz
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

Exact cellular decompositions are structures that globally encode the topology of a robot's free space, while locally describing the free space's geometry. These structures have been widely used for path planning between two points, but can be used for mapping and coverage of robot 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. In this paper, we derive a general framework for defining decompositions in terms of critical points and then give examples, each corresponding to a different task. All of the results in this paper are derived in an m-dimensional Euclidean space, but the examples depicted in the figures are two- and three-dimensional for ease of presentation.

1 Introduction

This paper presents some new examples of exact cellular decompositions whose cells are defined by critical points of Morse functions. Cellular decompositions have been widely used for planning a path between two points in the free space, but the motivating task for the work presented in this paper is coverage. Since the Morse functions define cells with "simple" structure, a planner can then use a cellular decomposition to achieve coverage by employing simple control strategies to cover each of the individual cells in the decomposition. A simple control strategy can be back-and-forth motions, resulting in a farming style pattern.

Our goal is to cover unknown spaces using different patterns. Varying the Morse function that defines the cells of the decomposition can change the pattern which is used to cover the free space. In this paper, we describe different functional definitions of cells that yield different patterns and comment on their utility. Since the cells are defined by critical points of Morse functions, detecting the critical points allows for coverage of unknown spaces. A companion paper describes a method to detect such critical points while covering an unknown space.

The sensor based coverage task is useful for many applications such as autonomous lawn mowing, floor cleaning, and snow removal, where the environment may not be known a priori or it is too cumbersome to input into the robot. However, the motivating application for the work presented in this and the companion paper is humanitarian de-mining, whose goal is to find and remove *every* land mine from a target region. The use of robots immediately bypasses the danger, reduces the cost, and potentially speeds the process.

2 Background

Exact cellular decompositions [7], [12] represent the free space by dividing it into non-overlapping 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 decomposition, where nodes represent the cells and edges connect nodes corresponding to adjacent cells. The problem of determining a path between two points, i.e. path planning, 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.

Perhaps the most widely known exact cellular decomposition is the trapezoidal decomposition [14] where each cell is a trapezoid or a triangle (Figure 1). Originally, this method assumes all obstacles in the environment are polygonal and uses the vertices of the polygonal obstacles to construct the decomposition. These vertices are classified as follows: in, out, floor, and ceiling. The *in* vertex is the (locally) left-most vertex whereas the *out* is the(locally) right-most. Of the remaining vertices, the *floor* vertices are ones on *top* of the polygonal obstacle and the *ceiling* vertices are the ones on *bottom.* At each floor vertex, a ray is drawn upward until it intersects an obstacle whereas at each ceiling vertex, a ray is drawn downward until it intersects an obstacle[1]. For both the in and out, two rays are drawn, one upward and one downward until an obstacle is encountered. To use this structure to derive a path between start and goal, we can examine the adjacency graph to derive a sequence of cells connecting start and goal cells (Fig-

[1] The terminology for floor and ceiling may seem to be reversed, but in actuality, the floor vertices describe the bottom of the cell which is the top of the obstacle, and the ceiling vertices the top of the cell.

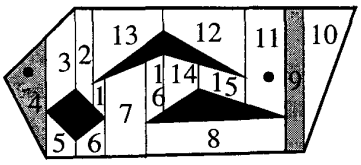

Fig. 1. Trapezoidal decomposition of bounded free space with polygonal obstacles. Cells 4 and 11 (shaded), contain the start and goal, respectively.

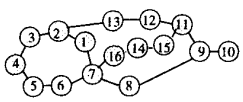

Fig. 2. Adjacency graph for the environment depicted in Figure 1.

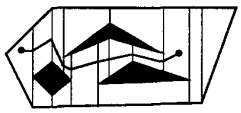

Fig. 3. Resulting path after searching the adjacency graph from Figure 2.

ure 2). The resulting path is piecewise linear, where each piece is a line segment that connects the midpoints of the boundaries of each cell in the sequence (Figure 3).

Cellular decompositions can be used to produce a path that *covers* the free space. Assuming each cell can be covered via a simple pattern, say simple back-and-forth motions, complete coverage is achieved simply by ensuring that the robot has identified and visited each cell in the region.

A minor short-coming of the trapezoidal decomposition for coverage is that many small cells, such as cell 9 (Figure 1), are formed that seemingly can be "clumped" into neighboring cells. Reorganizing the cells can result in a shorter (more efficient) path to cover the same area. To address this issue, the boustrophedon[2] cellular decomposition approach [9] was introduced. For a polygonal environment, the boustrophedon method is identical to the trapezoidal, except *no* rays are drawn at the floor and ceiling vertices, resulting in a decomposition with fewer and larger cells.

To generalize beyond polygons, we borrow ideas from Canny's work [4], [5], [6] who first applied a continuous slice method to motion planning. Recall that a *slice* is a line segment that is swept through the configuration space, $\mathcal{CS}$. We denote a slice as $\mathcal{CS}_\lambda$ where $\mathcal{CS} = \bigcup_\lambda \mathcal{CS}_\lambda$, i.e., the slices parameterized by λ and foliate the robot's configuration space $\mathcal{CS}$. Changing the value of λ sweeps a slice through the space. Let $\mathcal{FS}_\lambda$ be the slice contained in the free space, i.e. $\mathcal{FS}_\lambda = \mathcal{CS}_\lambda \bigcap \mathcal{FS}$. From Canny's work[4], [5], [6], we know that a change in the topology (connectivity) of $\mathcal{FS}_\lambda$ indicates that disconnected roadmap fragments ex-

[2]Boustrophedon literally means "the way of the ox." 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 cover (and thereby to plow) the entire field. The term was first used in English in 1699[1].

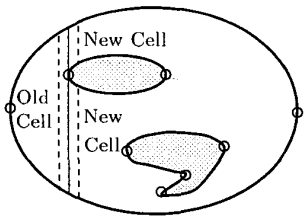

Fig. 4. In Event; as the slice moves left to right, its connectivity changes from one to two.

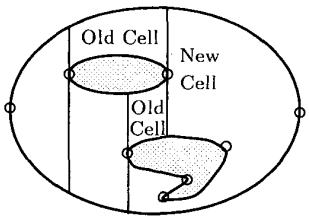

Fig. 5. Out Event; as the slice moves left to right, its connectivity changes from two to one.

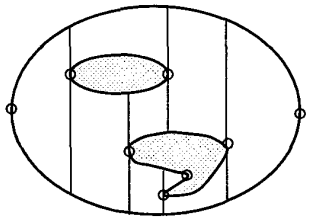

Fig. 6. Boustrophedon Decomposition

ist. Likewise, in this paper, we use a change in connectivity of the free space slice to indicate the boundary locations of the cells. When the connectivity increases, new cells are spawned (Figure 4). Conversely when connectivity decreases, multiple cells are completed and single cell is created (or cells disappear) (Figure 5). Figure 6 contains a simple example of the boustrophedon decomposition[3].

For coverage tasks, while the robot is performing back and forth motions, it is basically "slicing" its free space or "sweeping" a slice through its free space. While slicing the space, the planner looks for changes in connectivity of the slice to determine the cells and their adjacency relationships. Once the decomposition and adjacency graph are determined, the robot employs a simple graph search algorithm to compute a walk through the adjacency graph that visits all nodes, i.e., visits all cells. Since simple back-and-forth motions cover each cell, coverage is guaranteed to be complete.

3 Exact Cellular Decompositions in Terms of Morse Functions

In the previous section, we observed that changes in connectivity of a line-segment slice defines one type of cellular decomposition. In this section, we consider different "shapes" of slices that result in different decompositions. We vary the shape of the slice by changing its functional definition. A slice can be defined as the pre-image of a real valued function $h\colon \mathbb{R}^m \to \mathbb{R}$. For the boustrophedon decomposition, the slice function is

[3]The cells of this decomposition are identical to the *channels* introduced in [6] whose work focused on a new type of roadmap called the Opportunistic Path Planner.

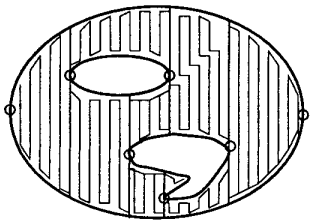

Fig. 7. Coverage pattern is back and forth motions like that of an ox plowing a field.

$h(x) = x^1$, where x^1 is the first coordinate of x, i.e., $\mathcal{CS}_\lambda = \{x \in \mathcal{CS} | h(x) = \lambda\}$. In this section, we will consider different definitions of h (and hence different shapes of slices) to yield different decompositions and perhaps coverage patterns of the target region.

We will demonstrate in the appendix (Lemma A.1) that critical points occur when the slice function gradient $\nabla h(x)$ is parallel to the surface normals of the obstacles (i.e. the slice is tangent to the obstacles), as can be seen in Figure 20. It is worth noting that the restriction of h to the boundary of the free space must be Morse which ensures critical points to be non-degenerate isolated points. Before we define cellular decompositions in terms of critical points of Morse functions, let us introduce $\mathcal{FS}_\lambda^j$ to be the jth connected component of $\mathcal{FS}_\lambda$. Now, $\mathcal{FS} = \bigcup_\lambda \bigcup_j \mathcal{FS}_\lambda^j$. Consider the space $X = \mathcal{FS} \backslash (\bigcup_\lambda \bigcup_j \mathcal{FS}_\lambda^j)$ for all $\mathcal{FS}_\lambda^j$ that contain a critical point. The connected components of X form the cells that form an exact cellular decomposition.

3.1 Boustrophedon Decomposition

Consider the function $h(x) = x^1$ where x^1 is the first coordinate of x. The slices, i.e. the pre-images of h, are $(m-1)$-dimensional hyper-planes. As seen earlier, in the plane, the coverage pattern is simple back and forth motions (Figure 7). The boustrophedon decomposition in three-dimensions has a planar slice which is swept through the space, forming cells at each critical point. Here, the topology of the slice changes each time a puncture appears or disappears in the slice plane. See Figure 8. Each three-dimensional cell in this decomposition is simply connected, but not contractable. At this time, we do not foresee an application for this particular decomposition, but included it here to demonstrate a decomposition in three dimensions.

To achieve coverage of the three-dimensional volume, each two-dimensional slice could be swept with a one-dimensional slice resulting in layers of back-and-forth patterns. This "double slicing" is reminscient of the approaches in [5], [15] that recursively call their algorithm on lower dimensional slices to connect disconnected roadmap fragments.

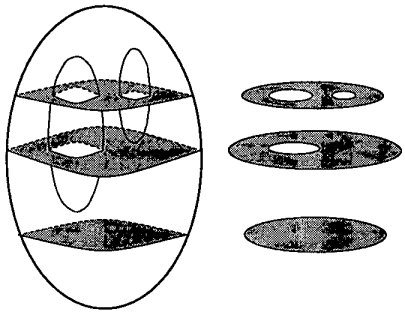

Fig. 8. Two-dimensional slice is swept from top to bottom changing topology at critical points. The grey slices are highlighted to note a change in the number of punctures between critical points.

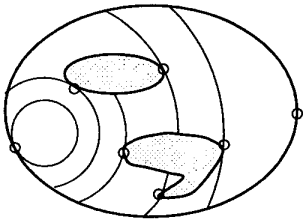

Fig. 9. Coverage pattern is concentric circles.

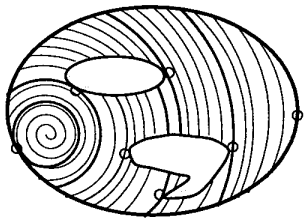

Fig. 10. Spiral pattern.

3.2 Spiral Pattern and Spike Pattern

The function $h(x) = \sqrt{\sum_i (x^i)^2}$ emits a pattern of concentric circles in the plane, spheres in $\mathbb{R}^3$, and S^{m-1} in $\mathbb{R}^m$. By Corollary A.2, critical points occur at points where the circle (sphere) is tangent to an obstacle. In the plane, this function yields a coverage pattern that 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. This occurs while simultaneously looking for critical points.

Note that in $\mathbb{R}^2$, instead of following a circle and stepping outward, the robot can follow a spiral pattern until it encounters critical points. The spiral pattern bypasses the need to step along the radial direction (Figure 10).

The function $h(x) = \tan(\frac{x^2}{x^1})$ induces a pattern that is orthogonal to the set of concentric circles. Using this pattern to perform coverage has the effect of covering more densely the region closest to the center of the pattern. This is useful if the likelihood of finding a desired

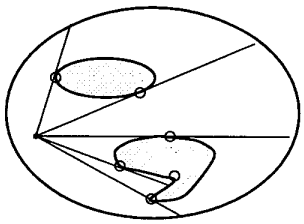

Fig. 11. Coverage pattern that slices the space like a pie.

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). This is similar to the pattern used by ants to home back to a desired location, like their home or discovered food. Some studies [17] have shown that when ants find food, they walk straight away from the food and then immediately return to the food. While going back and forth along these spikes, they record images of the food site. The result is a spiked pattern that helps ants to return discovered food.

Finally, the meteorite-searching robot Nomad [2] plans to use a *sun-synchronous pattern* that maximizes the incident angle of light has on solar panels, to be installed in the future on the robot. Naturally, the closer this angle is to ninety degrees, the more efficient the power transfer from light to electricity. The sun-synchronous pattern looks like a set of spikes emanating from a center. As Nomad drives along a spike, the solar panel does not have to move. If properly timed, Nomad will be able to follow the entire pattern without moving the solar panel. Otherwise, Nomad moves the panel for each spike in the pattern.

3.3 Diamond and Squarel Pattern

The function $h(x) = \sum_i |x^i|$ is nonsmooth and produces cells that look like rotated squares or diamonds (Figure 13). Again, critical points occur when the diamond is tangent to the obstacle, but points where h is nonsmooth require careful consideration. These points do not have a conventional gradient, but instead possess a generalized gradient, $\partial h(x)$ [10]. For the sake of discussion, view the nonsmooth portion of a function as the common boundary the continuous portions (pieces) of a piecewise smooth function. Each piece has a conventional gradient associated with it. The generalized gradient is simply the convex hull of these gradients. For points where a function is smooth, the generalized gradient reduces to the conventional gradient.

Just as conventional nonsmooth analysis hold for functions that are "almost" smooth, i.e. a small perturbation away from a smooth function, we can consider a nonsmooth version of Morse theory to examine other interesting nonsmooth functions that are "almost" smooth. We can restate Lemma A.1 as follows: the point x is a critical point if and only if $\nabla h(x)$ is contained in $\partial m(x)$. In fact, we can apply the results of nonsmooth analysis to handle nonsmooth obstacles as well.

Using this nonsmooth analysis, we can now define the decomposition induced by $h(x) = \sum_i |x^i|$ (Figure 13). For coverage, instead of driving in concentric squares, we can direct the robot to spiral out while

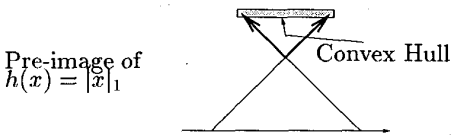

Fig. 12. Generalized gradient of nonsmooth function.

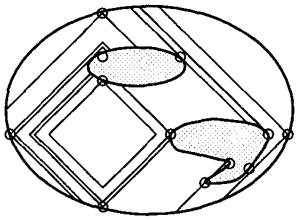

Fig. 13. Coverage pattern for $h(x) = |x|_1$.

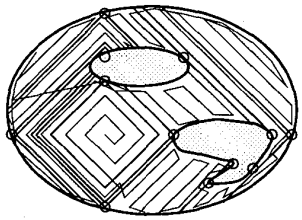

Fig. 14. Pattern using decomposition in Figure 13.

looking for critical points, hence the term *squarel*[4] Conventional differential drive or synchrodrive robots can often follow a squarel more easily than a spiral without accruing as much dead-reckoning error. The resulting pattern is shown in Figure 13. The function $h(x) = |x|_\infty = \max_i |x_i|$ emits a square pattern similar to the diamond pattern.

3.4 Brushfire Decomposition

As can be seen in Figure 16, the generalized Voronoi diagram also induces a decomposition. A common technique to construct the generalized Voronoi diagram is called the brush-fire method [12] where wavefronts emanate from all of the obstacles and collide on the Voronoi diagram.

Let $d_i(x)$ be the distance between a point x and a convex obstacle C_i. The function $D(x) = \min_i d_i(x)$ emits a decomposition termed the brushfire decomposition. The cells in the decomposition induced by $D(x)$ are not the Voronoi cells (Figure 16). Instead, consider each slice of $D(x)$, i.e. $D^{-1}(\lambda)$, to be a wave front where each point on the front has propagated a distance λ from the closest obstacle. As λ increases, the wave fronts progress. Cells of the brushfire decomposition are formed when these wavefronts initially collide.

Figure 17 contains a decomposition induced by D

[4]Thanks to Matt Mason who informed us of this term in an m-lab meeting.

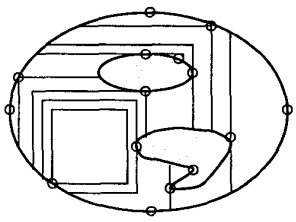

Fig. 15. Coverage pattern for $h(x) = |x|_\infty = \max_i |x^i|$

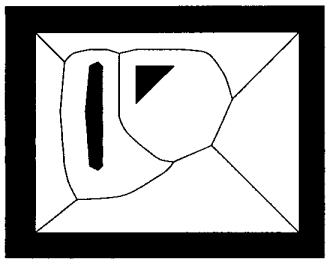

Fig. 16. Generalized Voronoi Diagram

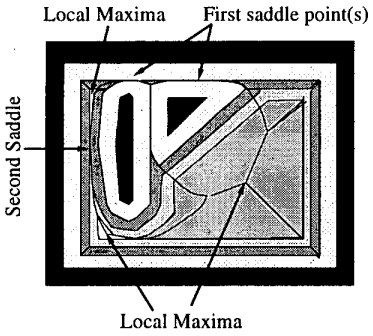

Fig. 17. Cells of the brush fire decomposition from Figure 16.

where regions of the same color represent a cell. To describe this figure, consider a wave front that originates on the boundary of the environment, which has three connected obstacles: the exterior, the vertical bar-like obstacle, and a triangle. These three wave fronts progress until they initially collide with each other, which occurs at a critical point. The light gray regions adjacent to the obstacle represent the three newly formed cells. The type of critical point that defines the gray regions in Figure 17 are saddle points. In fact, all of the cells are defined by saddle points of D. Since D is nonsmooth, its generalized gradient must be considered as well. For the sake of discussion, allow the "origin" of the generalized gradient be centered at the point x where distance is being measured. If the point x is contained in the boundary of the generalized gradient of D, then the point x is a saddle point [8].

For the sake of explanation, imagine that the wave front represents fire that is emanating from the obstacles that burns free space. At this point, the first three cells, described above, correspond to "burned" free space. The "unburned" regions include a little darker gray region top portion of the figure and a large portion of free space. Increasing λ, i.e. progressing the front forward until the wave fronts collide again, consumes the small unburned portion of free space in the top region of the figure and gives rise to another cell, which is dark gray and surrounds a large portion of unburned free space. At this point, a small medium gray region in the upper left portion of the figure is pinched off and a larger portion of free space remains unburned. This process is repeated until the entire free space is consumed by this imaginary fire.

It is worth noting that the "pinching" off of free space occurs at saddle points of D and that the wave fronts consume a region at local maxima of D. Roque and Choset [16], motivated by rudimentary brush fire analysis, demonstrated that if a Voronoi vertex is a local maximum of D, then a region surrounding it will eventually be separated by the rest of the free space, if the free space were set on fire by a very special isotropic fire. In essence, they applied the brush fire method to brush fires[5].

3.5 Wave Front 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 wave front potential where λ parameterizes the wave or level set of h emanating outward from the fixed point. For the wave front potential, the fixed point is the goal and when the front crosses the start, the planner can back-track a path from the goal to the start [11]. This function also emits a cellular decomposition, which is particularly useful for coverage by a tethered robot where the robot is incrementally fed and sweeps out curves each at constant tether length (Figure 18).

Critical points occur when the wave front becomes tangent to the obstacles as well as initially collide into each other. This first occurs in upper left portion of obstacle C_1 in Figure 18. Note how once the waves collide, they propagate as one wave with a nonsmooth cusp that originated at the critical point. In fact, this nonsmooth cusp traces the set of points of equal path-length to the goal for two classes of paths, one around to the right of the obstacle and one to the left. This critical point can be detected by measuring the length of an idealized tether that maintains tension with the robot. Unfortunately, we do not have a simple geometric condition, like parallel gradients, that detect these critical points, however.

4 Conclusion

In this paper, we described a formulation that provides a common framework for a variety of motion planning algorithms. We started with conventional slice algorithms where the slice is normally a flat plane (hyperplane in higher dimensions) which could be defined by the pre-image of a real valued function. Such slices are well suited to coverage with farming-style patterns and

[5]Thanks to Chuck Thorpe at Carnegie Mellon who first suggested this phrase.

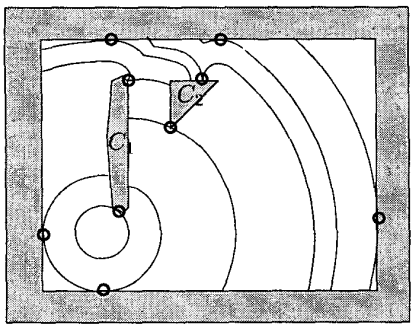

Fig. 18. Wave Front Decomposition.

material deposition in three-dimensions. However, by altering the slice function definition, and hence the slice geometry, we can achieve different tasks. Since these slice functions are Morse, we can describe a task, and then define its corresponding Morse function to ensure, with provable guarantees, completion of the task. We use the critical points of the Morse functions to provide these provable guarantees.

Our view is that physical properties and constraints inherent to the robot can help 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 brush fire decomposition. The brush fire 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 wall to guarantee uniform coverage (in the interior of the Voronoi cells), as opposed to using its odometry information.

The wavefront decomposition is well suited to inspection tasks that use tethered mobile robots or mechanisms that are incrementally inserted into the environment because each point on the wavefront slice has the same path-length in the free space to the point of feeding. These Morse functions can also be used to model phenomena such as ants leaving a spiked-pattern behind to guide other ants to food or their home. The Morse function that defines the spike pattern is also useful in guiding the Nomad robot so that its solar panels are maximally aimed at the sun. All of these tasks and phenomena have a common framework, only differentiated by its Morse function.

Other mathematical structures can define other classes of decompositions. For example, the nonsmooth points of D, the multi-object distance function which is commonly used in potential functions, defines the generalized Voronoi diagram. Future work considers other structures.

References

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

[2] D. Apostolopoulos, M. Wagner, and W. Whittaker. Technology and field demonstration results in the robotic search for antartic meteorites. In *International Conference on Field and Service Robots*, pages 185–190, August 1999.

[3] Glen E. Bredon. *Topology and Geometry.* Springer-Verlag, New York, NY, 1993.

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

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

[6] J.F. Canny and M.C. Lin. An Opportunistic Global Path Planner. *Algorithmica*, 10:102–120, 1993.

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

[8] H. Choset. Nonsmooth analysis, convex analysis, and their applications to motion planning. *Special Issue of the Int. Jour. of Comp. Geom. and Apps.*, 1998.

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

[10] F. H. Clarke. *Optimization and Nonsmooth Analysis.* Society of Industrial and Applied Mathematics, Philadelphia, PA, 1990.

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

[12] J.C. Latombe. *Robot Motion Planning.* Kluwer Academic Publishers, Boston, MA, 1991.

[13] J. Milnor. *Morse Theory.* Princeton University Press, Princeton, NJ, 1963.

[14] F.P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction.* Springer-Verlag, 1985. p198-257.

[15] E. Rimon and J.F. Canny. Construction of C-space Roadmaps Using Local Sensory Data — What Should the Sensors Look For? In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 117–124, San Diego, CA, 1994.

[16] W. Roque and H. Choset. Voronoi growth model applied to green island formation in forest fires. *Submitted to International Journal of Computational Geometry and Applications*, 1999.

[17] M. V. Srinivasan. Ants Match as They March. *Nature*, 392:660–661, 1998.

Appendix

A Critical Points of Morse Functions

In this paper, we looked for connectivity changes in a slice in the free space. In this section, we will demonstrate that these connectivity changes correspond to critical points of a function. From calculus, we know at critical points of a function the first derivative vanishes. Using a result from Morse theory [13], we are going to relate these two concepts. Recall that the "derivative" or Jacobian of a real valued function $h : \Re^m \to \Re$ at $p \in \Re^m$ is

$$dh_p = \left[\frac{\partial h}{\partial x_1}(p) \quad \cdots \quad \frac{\partial h}{\partial x_n}(p) \right]$$

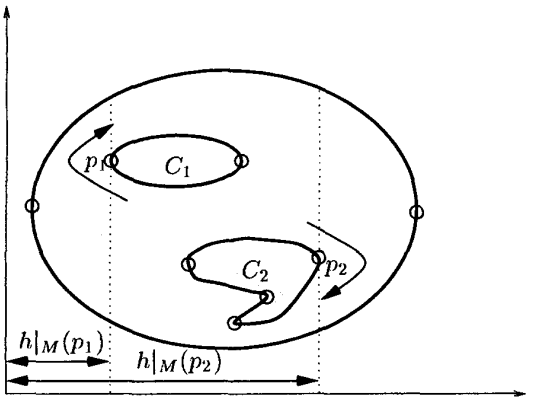

Fig. 19. Two ellipsoid shaped obstacles enclosed inside of a bounding ellipsoid. p_1 is the left-most critical point on the left-most obstacle and p_2 is the right-most critical point on the right-most obstacle. The values $h|_M(p_1)$ and $h|_M(p_2)$ are plotted on the bottom.

A point $p \in \Re^m$ is called a critical point of h if

$$\frac{\partial h}{\partial x_1}(p) = \cdots = \frac{\partial h}{\partial x_m}(p) = 0.$$

The real value $h(p)$ is called a critical value. A critical point is *non-degenerate* if its Hessian matrix

$$\frac{\partial^2 h}{\partial x_i \partial x_j}(p)$$

is non-singular. If the Hessian of all critical points of a function is non-singular, then the function is a *Morse function.*

A.1 Slice Function

The function that we are considering is really h restricted to the boundaries of the obstacles (we will show that this function is Morse at the end of this section). We denote the boundaries of the obstacles C_i as ∂C_i and the union of the boundaries as $M = \bigcup_i \partial C_i$. With this terminology in hand, let $h|_M$ be h restricted to M.

Figure 19 demarks two critical points: p_1, which is the left-most point of C_1 and p_2 which is the right-most point of C_2. The function $h(x)$ essentially measures the distance between a point x and the "y-axis." Likewise, $h|_M(x)$ measures the distance between points on the obstacle boundaries and the y-axis. Consider a path on ∂C_1 that passes through p_1 as depicted in Figure 19. Moving along the path towards p_1 decreases the value of $h|_M(x)$ and after passing through p_1, the value increases. In other words, p_1 is local minimum of $h|_M$. Likewise, p_2 is a local maximum of $h|_M$. These extrema are critical points of the function $h|_M$.

Now, lets look at the derivative of $h|_M$. Let $x \in \partial C_1 \subset \mathbb{R}^m$, i.e., x is a point in the boundary of C_1, but we are embedding this point in $\mathbb{R}^m$. Since $\nabla h|_M(x) = \Pi_{T_x \partial C_1} \nabla h(x)$, where $\Pi_{T_x \partial C_1}$ is the orthogonal projection operator onto the tangent space of ∂C_1, $h|_M(x)$ achieves a critical value when $\nabla h(x)$ is parallel to the surface normal of the obstacle C_1.

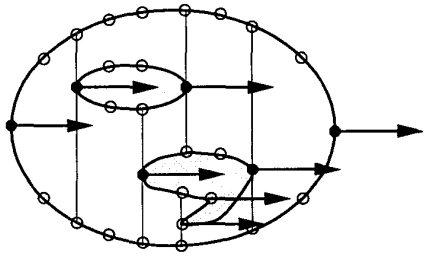

Fig. 20. Vectors denote $\nabla h(x)$ where they are perpendicular to the slice direction

The following lemma states this result more formally. Assume that the pre-image of a function $m\colon \mathbb{R}^m \to \mathbb{R}$ describes the boundary of an obstacle (and thus $\nabla m(x)$ is a surface normal), we have:

LEMMA A.1 *The point $x \in M$ is a critical point of $h|_M$ if and only if $\nabla h(x)$ is parallel to $\nabla m(x)$.*

Proof: Canny [5] showed that $x \in M$ is a critical point of $h|_M(x)$ if and only if the following matrix loses its rank,

$$D(m,h)_x = \begin{bmatrix} \frac{\partial m}{\partial x_1}(x) & \cdots & \frac{\partial m}{\partial x_m}(x) \\ \frac{\partial h}{\partial x_1}(x) & \cdots & \frac{\partial h}{\partial x_m}(x) \end{bmatrix}$$

Since neither row can be zero, this matrix loses its rank when the first and second rows are linearly dependent. Since we are considering Euclidean spaces, the first row of $D(m,h)_x$ is the transpose of $\nabla m(x)$ and the second the transpose of $\nabla h(x)$. $D(m,h)_x$ loses its rank when the gradient vectors $\nabla h(x)$ and $\nabla m(x)$ are parallel to each other (linearly dependent). Hence $\nabla h(x)$ is parallel to $\nabla m(x)$ at critical points (Fig. 20). ■

By definition, the following corollary immediately follows:

COROLLARY A.2 *The slice is tangent to an object at critical points.*

By definition, the slice vector is orthogonal to the slice. This vector is parallel to the surface normal of the object at a critical point. Therefore, the tangent spaces of the slice and obstacle boundary coincide at the critical point[6]. This means that the critical points of $h|_M$ are isolated, a necessary condition that must be satisfied for $h|_M$ to be Morse. We will assume that the slice and the boundaries of the obstacles do not make a second order contact. Effectively, [5] makes this transversality assumption.

For nonsmooth obstacles, we can restate Lemma A.1 as follows: the point $x \in M$ is a critical point of $h|_M$

[6]Lemma A.1 and Corollary A.2 apply to non-flat slices as well. The proofs of the lemmas use the tangent spaces of M and $h^{-1}(\lambda)$

if and only if $\nabla h(x)$ is contained in $\partial m(x)$. In fact, we can apply the results of nonsmooth analysis to handle non-smooth obstacles as well. The result, whose proof is omitted due to space restrictions, is:

LEMMA A.3 *The point $x \in M$ is a critical point of $h|_M$ if and only if there exists a $v_h \in \partial h(x)$ that is parallel to some $v_m \in \partial m(x)$.*

A.2 Change in the Connectivity of the Slice

We will now show that critical points of the slice function induce changes in the topology (i.e., connectivity) of the slice in the free space. This is done in two steps. First, we demonstrate that $h|_M^{-1}(\lambda)$ changes topology as λ passes through a critical value, and then how this change induces a change in the topology (i.e., 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 $\bar{\lambda}$ 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 [13]. In the context of the slice function, Morse theory states that there exists a diffeomorphism ϕ such that for all $\lambda_1, \lambda_2 \in (\lambda_*, \lambda^*)$, $\phi(h|_M^{-1}(\lambda_1)) = h|_M^{-1}(\lambda_2)$, where λ_* and λ^* are adjacent critical values of a real valued Morse function.

In the planar case, this means that between adjacent critical points of $h|_M$, the number of points in the pre-image of the slice function remains the same. Likewise, in three-dimensions, between adjacent critical points of $h|_M$, the number of closed loops remains the same. Figure 21 displays the pre-images of different values of λ under the mapping $h|_M^{-1}(\lambda)$. For all $\lambda \in (\lambda_0, \lambda_2)$, $h|_M^{-1}(\lambda)$ has two points. The number of points in the pre-image increases to four as the slice passes through the critical value λ_2.

LEMMA A.4 *The topology of the slice in the free space changes only at critical points of the slice function restricted to the boundary of the obstacles.*

Proof:

Let $C = \bigcup_i C_i$ and $C_\lambda = C \bigcap h^{-1}(\lambda)$. Let $\mathcal{FS} = \mathcal{W} \backslash \bigcup_i C_i$ and $\mathcal{FS}_\lambda = \mathcal{FS} \bigcap h^{-1}(\lambda)$. $\mathcal{FS}_\lambda$ is the slice in the free space. Let $M = \bigcup_i \partial C_i$ and $M_\lambda = M \bigcap h^{-1}(\lambda)$. When λ is a regular value, $M_\lambda = \partial C_\lambda = \partial \mathcal{FS}_\lambda$, i.e., M_λ is a separator for the set $\mathcal{FS}_\lambda \bigcup C_\lambda$. In the planar case, M_λ comprises isolated points, and in $\mathbb{R}^3$ it contains closed curves.

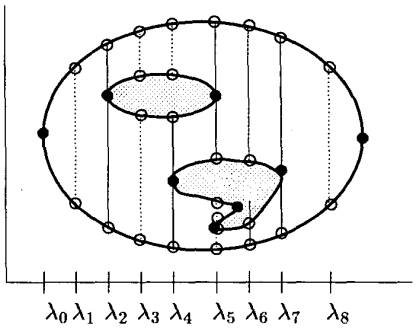

Fig. 21. Points in the pre-image of $h|_M$ for different values λ_i are plotted. Solid circles correspond to critical points and open circles are regular points.

Morse Theory assures us that the topology of M_λ remains constant between critical points and only changes at critical points. Since M_λ is a separator for the set $\mathcal{FS}_\lambda \bigcup C_\lambda$, the topology of $\mathcal{FS}_\lambda \bigcup C_\lambda$ must change when the topology of M_λ changes.

■

In the planar case, the pre-image of $h|_M$ are the end points of alternating intervals, an interval in the free space (a subset of $\mathcal{FS}_\lambda$) and an interval in the obstacle (a subset of C_λ). Lemma A.4 assures us that when the number of end points changes, so does the number of intervals. Since the intervals must alternate, a change in connectivity of the intervals occur only at critical points.

In three-dimensions, the pre-image of $h|_M$ are closed curves (the intersection of the obstacle with a slice plane). Lemma A.4 assures us that when the number of closed curves change, so does the connectivity, either path-connectivity (number of connected components) or simple connectivity (number of punctures), of the $\mathcal{FS}_\lambda$. In a $m+1$ dimensional space where the slices are m dimensional, the number of generators of the $m-1$st-homotopy group $\Pi_{m-1}(S^{m-1}; \mathcal{FS})$ [3] changes at critical points.