# Construction of C-space Roadmaps From Local Sensory Data
# What Should the Sensors Look For?

Elon Rimon
Dept. of ME, Caltech

John F. Canny
Dept. of CS, Berkeley

**Abstract** We propose a truly incremental exact navigation algorithm for general articulated robots: ***the robot may start with no apriori information about its environment, and is guaranteed to find the goal if it is reachable, or halt otherwise.*** The algorithm, termed the *incremental roadmap algorithm*, constructs a roadmap based on distance data collected on-line and encoded as repulsive potential field. The incremental behavior is achieved with two novel abstract sensors: a *critical point detector* and a *minimum passage detector*. We show by examples which environmental features must be measured by the detectors for a planar-body robot. A companion paper will discuss implementation of the detectors for navigation of 2D and 3D bodies. Actual engineering of the detectors, and their implementation for articulated robots, are open research problems.

## 1 Introduction

We are concerned with the *classical path planning* problem, where a robot has to navigate toward various goal configurations amidst stationary obstacles, while avoiding collision. There are currently four path-planning methodologies that are known to be general. Unfortunately, none of them can be considered practical. The first two possess an algorithm for computing the free path: Schwartz&Sharir's cell decomposition algorithm (1983) [15], and Canny's roadmap, or silhouette, method (1987) [2]. However, as of today both methods have been implemented only for robots with very small number of degrees of freedom. The other two are Yap's retraction method (1987) [16], a generalization of the Voronoi diagram to high dimensional configuration spaces, and Rimon&Koditschek's navigation functions (1992) [14], which are based on potential fields.

Potential fields are very attractive for sensory based navigation. A typical potential is constructed from a combination of an attractive potential centered at the goal and a repulsive potential based on distance from the obstacles that is measured on-line by the robot. In fact, to our knowledge the only commercially available path-planners that handle realistic articulated robots are based on heuristic formulation of potential-fields [1, 6]. These path-planners suffer, however, from the presence of undesired local minima, which are typically removed by human intervention or by random search.

The algorithm proposed here exploits sensory-based repulsive potentials to achieve collision avoidance and path safety. The problem of local minima is eliminated by careful construction of a roadmap, akin to Canny's network of silhouette curves [2]. Roughly speaking, the roadmap we construct consists of a collection of one-dimensional curves, related to the "ridges" of the repulsive potential, and termed *ridge curves*. The ridge curves are interconnected by *linking curves*, through two types of special points. The first kind are critical points where the connectivity of the free configuration space changes. The second kind are minimum clearance configurations, that always lie between adjacent ridge curves. Both types of points are detected by specialized sensors, the *critical point detector* and the *minimum passage detector*, respectively.

The idea of merging potential functions with the roadmap algorithm was originally proposed by Lin and Canny (1990) [9]. (We use terminology suggestive of the connection to their paper.) Their algorithm, however, relies on distance data collected *off-line* for the entire robot environment before navigation starts. Their algorithm additionally relies on a preprocessing step, in which the entire collection of critical points is computed. In contrast, this paper focuses on on-line sensory based navigation, where no apriori information about the environment is known. Moreover, since the incremental algorithm considers only the geometry along the path taken by the robot, it promises to be *output sensitive* i.e., fast in uncluttered environments while taking longer time to run in complicated ones.

From another perspective, there is the important open problem of sensory information [5]: *which sensory measurements must be collected on-line by the robot to achieve provably correct global navigation?* The off-line methodologies of Schwartz&Sharir and Canny presume that the geometry of the environment is first captured in terms of symbolic polynomial representation. General symbolic algebra tools are then used to extract what is essentially roots of polynomial systems, which in turn guide the construction of a free path. However, it is well known that current symbolic algebra tools are impractical even for very simple situations [8]. We propose here to circumvent such computational bottlenecks by direct sensory measurements. A companion paper now under preparation describes which environmental features must be measured by the detectors for 2D or 3D rigid-body navigation. The actual engineering of the detectors, as well as their implementation for articulated robots, are important open problems.

There are several attempts in the robotics literature to tackle the sensory information problem. Examples are: Lumelsky's work on two-dimensional configuration spaces [10], and Cox's work on three-dimensional configuration spaces [4]. In contrast, our algorithm is completely general. Moreover, unlike other general methodologies, its reliance on sensory measurements suggests that it may become practical as well.

The paper is organized as follows. First we review stratified sets, Morse theory, and bifurcation theory. As these theories form the foundations for our algorithm. Next, the naive version of the algorithm is described, where only the critical point detector is used. Then we show that the minimum passage detector must be added. A proof that the algorithm based on the two detectors always finds a path to the goal is then sketched. We conclude with a discussion of related open problems, such as the need for active perception.

## 2 Preliminaries

The robot configuration space, called *c-space*, is globally parametrized by a single copy of Euclidean space $\mathbb{R}^k$, where $k$ is the number of degrees of freedom of the robot. There is no loss of generality in making this choice, for coordinates representing rotational degrees of freedom are periodic in $2\pi$. Given a physical obstacle, its corresponding *c-obstacle* is the set of all configurations at which the robot intersects the obstacle. The boundary of a c-obstacle is exactly those configurations where the surfaces of the robot and an obstacle touch each other, while their interiors are disjoint. The free configuration space, called the *freespace* $\mathcal{F}$, is the space that remains after removing from c-space the interiors of the c-obstacles.

### 2.1 Stratified Sets

The freespace is typically a stratified set. A *regularly stratified* set $\mathcal{S}$ is a set $\mathcal{S} \subset \mathbb{R}^k$ decomposed into a union of disjoint smooth manifolds[1] called *strata*, satisfying the Whitney condition. The dimension of the strata varies between zero, which are isolated points, and $k$, which are open subsets of the ambient $\mathbb{R}^k$. The Whitney condition requires that the tangents of two neighboring strata "meet nicely." For our purposes suffices to say that this condition is almost always satisfied.

The boundary of $\mathcal{F}$ is a union of portions of the boundaries of the c-obstacles. Thus $\mathcal{F}$ is naturally a stratified set, consisting of portions of the individual c-obstacle boundaries, their respective intersection along lower dimensional manifolds, and the various connected components of the interior of $\mathcal{F}$. This is illustrated in Figure 1, for a planar body and three obstacles.

### 2.2 Morse Theory on Stratified Sets

Let $f$ be a smooth real-valued function defined on a smooth manifold $\mathcal{M} \subset \mathbb{R}^k$. A point $x \in \mathcal{M}$ is a *critical point* of $f$ if its derivative at $x$, $Df(x)$, vanishes there. A *critical value* of $f$ is the image $c = f(x) \in \mathbb{R}$, of a critical point $x$. $f$ is a *Morse function* if all its critical points are *non-degenerate* i.e., its second derivative matrix $D^2f(x)$ is non-singular at the critical points. When $f$ is defined on a stratified set $\mathcal{S}$, its critical points are the *union* of the critical points obtained by restricting $f$ to the individual strata. $f$ is a Morse function on $\mathcal{S}$ if it is, first of all, Morse in the classical sense on the stratum containing the critical point $x$. And, second, if $\nabla f(x) = Df(x)$ is *not* normal to any of the other strata meeting at $x$. For our purposes suffices to say that almost all the smooth functions on a given stratified set are Morse in the extended sense.

*Stratified Morse theory* is concerned with such Morse functions [7]. The theory guarantees that *as $c$ varies within the open interval between two adjacent critical values of $f$, the level-sets $\mathcal{S}_c = \{x \in \mathcal{S} : f(x) = c\}$ are topologically equivalent* (homeomorphic) *to each other.* In particular, the path-connectivity of the level-sets $\mathcal{S}_c$ is preserved between critical values. In our case $f$ is simply a linear functional, called the *sweep function* $\sigma$, that sweeps $\mathcal{F}$ with hyperplanes along the $k^{th}$ coordinate of the parametrization of c-space by $\mathbb{R}^k$: $\sigma(x_1, \dots x_k) = x_k$. (If $x_k$ is periodic, the sweep wraps around after a full period is complete.)

Any connectivity change of the slices $\mathcal{F}|_c = \{x \in \mathcal{F} : \sigma(x) = c\}$ must occur locally, in a neighborhood about a critical point of $\sigma$ in $\mathcal{F}$. In general, only some of the critical points correspond to change in the path-connectivity of the slices $\mathcal{F}|_c$. Of those, only the following two types are of interest to our algorithm. Critical points at which locally distinct connected components

[1] Recall that a manifold $\mathcal{M} \subset \mathbb{R}^k$ of dimension $d$ is a hypersurface that locally looks like $\mathbb{R}^d$, for a fixed $d$, $0 \leq d \leq k$.

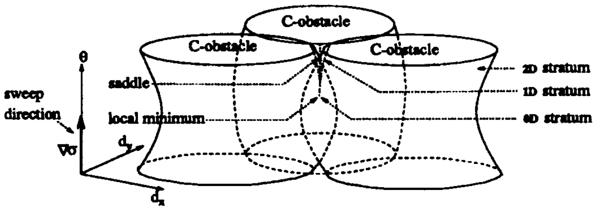


**Figure 1.** For a plane-sweep along $\theta$, only the saddle is an interesting critical point

of the freespace slices meet, called **join points**, and critical points where two connected components split, called **split points**. Their union is called the **interesting** critical points.

**Example:** In Figure 1 c-space is swept along the orientation axis $\theta$. The saddle is an interesting critical point, since two connected components of the freespace slices meet there and become a single component above it. *Non*-interesting critical points, such as the local minimum, may involve the appearance (or disappearance) of a new connected component of the freespace slices.

### 2.3 Bifurcation Theory

In addition to the sweep function $\sigma$, the algorithm uses a ***distance function***, $d : \mathcal{F} \to I\!R$, that serves as a repulsive potential. In general, it only has to be a smooth function, that is zero on the boundary of $\mathcal{F}$ and is strictly positive in its interior. In practice, however, it is more natural to use the distance of the robot from its environment, as measured by its sensors,

$$d(x) = \mathsf{dst}(\mathcal{A}(x), \mathcal{B}) = \min_{a \in \mathcal{A}(x), b \in \mathcal{B}} \{\|a - b\|\}, \qquad (1)$$

where $\mathcal{A}(x)$ is the set occupied by the robot when it is at configuration $x \in I\!R^k$, and $\mathcal{B}$ is the union of the physical obstacles. An extension of [3, Proposition 2.4.1] yields that $d$ of (1) is differentiable almost everywhere. We, however, shall treat $d$ as a smooth function.

We will need to evaluate $d$ on the individual level-sets of $\sigma$. Since $\sigma(x) = x_k$, the restriction of $d$ to the slice $\mathcal{F}|_{x_k=c}$ is simply $d_c(y) \triangleq d(\vec{y}, c)$, where $\vec{y} = (x_1, \ldots x_{k-1})$ and $x_k = c$. The algorithm constructs curves, called **ridge curves**, by tracing the *local maxima* of $d_c(y)$ as $c$ varies in $I\!R$. (The points $y$ are restricted to the slice $\mathcal{F}|_c$, while $d_c(y)$ is the distance between $\mathcal{A}(\vec{y}, c)$ and the entire obstacle set $\mathcal{B}$.) If a ridge-curve has endpoints, they lie on the boundary of $\mathcal{F}$, or they are ***bifurcation points*** that lie in the interior of $\mathcal{F}$. This is explained in the next paragraph. The solid curves in Figure 2(a) are the ridge-curves of a smooth potential on the planar freespace shown. For comparison, Figure 2(b) shows the ridge-curves generated by the non-smooth Euclidean distance (1).

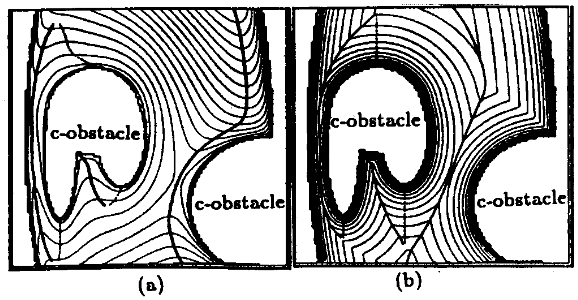


**Figure 2.** (a) Ridge curves for a smooth potential (b) and for the exact Euclidean distance

*Bifurcation theory* characterizes the behavior of the equilibria of one-parameter families of smooth gradient vector-fields $\nabla d_c(y)$, as $c$ varies in $I\!R$ [12]. It tells us the following facts. First, the loci of the equilibria of a generic one-parameter family $\nabla d_c(y)$ form a collection of smooth one-dimensional curves. This is shown in Figure 2(a), where the solid curves are the local maxima and the dashed curves are the local minima. Second, the equilibria curves meet each other at isolated points, called ***bifurcation points***. Third, a generic $\nabla d_c(y)$ has only one type of bifurcation, a *fold* or *saddle-node* bifurcation, where two equilibria curves meet each other smoothly at some parameter value $c = c_0$, and cease to exist for parameter values beyond $c_0$. Indeed, all the bifurcation points in Figure 2(a) are of this type.

## 3 Description of the Naive Algorithm

A *roadmap* for a robot freespace is a graph $\mathcal{R} \subset \mathcal{F}$ satisfying the following three properties: 1) *Connectivity*: every connected component of $\mathcal{F}$ contains exactly one component of $\mathcal{R}$; 2) *Accessibility*: $\mathcal{R}$ can be effectively reached from any start configuration; 3) *Departibility*: every goal configuration can be effectively reached from $\mathcal{R}$. Traditionally, when a roadmap is built off-line with complete information about the environment, there is no distinction between accessibility and departibility. In our case accessibility is achieved by uphill climb along the direction of increasing distance until a ridge-curve is reached. Departibility is achieved by invoking the incremental algorithm on a sequence of c-space slices of decreasing dimension.

The naive algorithm uses the *sweep function* $\sigma : \mathcal{F} \to I\!R$, the *distance function* $d : \mathcal{F} \to I\!R$, and a critical point detector (described below). It constructs a roadmap which consists of two types of curves. The first are the **ridge curves**, traced by the *local maxima* of $d_c(y)$ as $c$ varies in $I\!R$. The second are the **linking curves**. Each linking-curve passes through an interesting critical point of $\sigma$ in $\mathcal{F}$, and connects two ridge-curves. Let $\Sigma$ be the set of interesting critical points of $\sigma$ in $\mathcal{F}$, consisting of join and split points. Since $\nabla\sigma(x)$ never vanishes in the interior of $\mathcal{F}$, these points occur only on the boundary of $\mathcal{F}$. By definition, a point $x_c$ of $\Sigma$ is a common boundary point of two locally distinct connected components of $\mathcal{F}|_{x_c}$, where $\mathcal{F}|_{x_c}$ denotes the slice $\mathcal{F}|_{\sigma(x_c)}$. Assuming that $\mathcal{F}$ is bounded, $d_c(y)$ must attain a maximum on each connected component of $\mathcal{F}|_{x_c}$. Hence $x_c$ is also a common boundary point of two locally distinct basins of attraction of $\nabla d_c(y)$. The linking-curve is a union of two curves, each starting at $x_c$ and moving uphill in one of the two basins, until a ridge-curve is reached.

The robot must detect points $x_c$ of $\Sigma$ as it traverses a ridge-curve. This is achieved by means of the *critical-point detector*. It is required of this "sensor" that: *from a ridge-curve, it must detect all the interesting critical points associated with the ridge-curve's basin of attraction.* The latter term needs explanation. If $y^*(c)$ is a ridge-curve, its ***basin of attraction*** is the union of the individual basins of the points $y^*(c)$ with respect to the flow of $\nabla d_c(y)$, as $c$ varies in the domain of $y^*(c)$. It is convenient to make the requirement that the critical-point detector be *monotonic*: the order by which it detects the critical points from a given ridge-curve is identical to the order of their $k^{th}$ coordinate.

**Implementation of the critical-point detector:** For 2D or 3D single-body robots, points of $\Sigma$ correspond to passages between obstacles in the robot environment. This is exemplified in Figure 3, where an ellipse robot navigates in an environment whose obstacles are overlapping convex shapes. It is shown in the companion paper that for a sweep along the orientation axis, the interesting critical points are characterized by the following two properties. First, they occur only at configurations where the ellipse maintains simultaneous contact with two disjoint obstacles. Second, each critical point is associated with a direction vector $u$, such that the width of the ellipse in the direction $u$ is equal to the gap between the contacted obstacles along the same direction $u$. Figure 3(b) shows the ellipse in a critical configuration. The fixed-orientation slices of the freespace are disconnected just below the critical orientation, Figure 3(a), and comprise a single connected component just above it, Figure 3(c).

Given a detected point $x_c \in \Sigma$, the robot must depart from the ridge-curve and reach $x_c$, from which it continues with uphill motion to the other ridge-curve. To reach $x_c$, the robot first moves to a point $x = (\tilde{y}, c)$ on the ridge-curve such that $y$ lies in $\mathcal{F}|_{x_c}$. A curve from $x$ to $x_c$ is then constructed within $\mathcal{F}|_{x_c}$ by a recursive call with the start point $\hat{x}_s = x$, goal point $\hat{x}_g = x_c$, and $\hat{\mathcal{F}} = \mathcal{F}|_{x_c}$. Note that $x_c$ is guaranteed

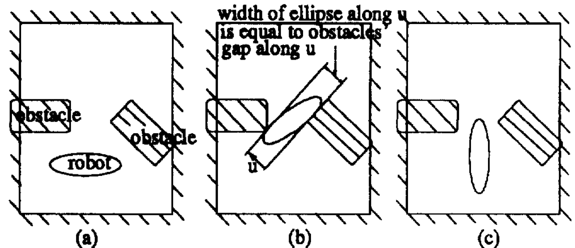

**Figure 3.** (a) freespace has two components (b) a critical configuration (c) freespace is connected

to be reachable from $x$ within $\mathcal{F}|_{x_c}$ since by construction $x_c$ belongs to the ridge-curve's basin of attraction. The recursion ends when the dimension of the slice becomes unity, for the roadmap of a one-dimensional slice is identical to the slice itself. This is a key characteristic of our algorithm: ***the departure from the roadmap toward a target outside it is achieved by construction of a roadmap on a lower-dimensional slice within which the target is guaranteed to be reachable.***

Points on the roadmap from which excursion to a target outside the roadmap begins are termed **departure points** (d-points). The goal, denoted $x_g$, is found by identifying d-points at points where the roadmap crosses the slice through the goal, $\mathcal{F}|_{x_g}$. Every such point $x$ serves as an intermediate starting point, from which a search after a path to $x_g$ is executed within $\mathcal{F}|_{x_g}$ by recursive call with $\hat{x}_s = x$, $\hat{x}_g = x_g$, and $\hat{\mathcal{F}} = \mathcal{F}|_{x_g}$. While $x_g$ is *not* automatically reachable from $x$ within $\mathcal{F}|_{x_g}$, it is shown in Section 5 that the algorithm does find one such $x$ from which $x_g$ is reachable. The recursion ends when the dimension of $\mathcal{F}|_{x_g}$ becomes unity. Here is a schematic description of the *naive* algorithm, starting with the required sensors.

**Sensors:** 1) A sensor that measures the robot configuration $x$; 2) A sensor that measures the distance between the robot and its environment, $d(x)$; 3) A *critical-point detector*, that detects from a ridge-curve the interesting critical points associated with the ridge-curve's basin of attraction.

**Input:** start and goal configurations, $x_s$ and $x_g$.

**Output:** If $x_g$ is reachable: a path from $x_s$ to $x_g$. Otherwise: a roadmap for the connected component of $x_s$.

**Data structure:** $\mathcal{R}$—the explored roadmap.

**Incremental Roadmap Algorithm:**

1. Connect $x_s$ to a ridge-curve by following $\nabla d_c(y)$, where $c = \sigma(x_s)$. Put $x_s$, the curve, and its ridge-curve endpoint in $\mathcal{R}$ (no other curves emanate from $x_s$).

2. **Repeat:** Move to an explored node of $\mathcal{R}$ that has an unexplored curve emanating from it. Explore the curve until its endpoint $x$ is found according to one of the following events.

A departure point encountered:

(a) An interesting critical point $x_c$ is detected: Move on the ridge-curve to a point $x$ in $\mathcal{F}|_{x_c}$. Make $x$ a node of $\mathcal{R}$. Decide whether to make a recursive call with $\hat{x}_s = x$, $\hat{x}_g = x_c$, $\hat{\mathcal{F}} = \mathcal{F}|_{x_c}$; or to continue exploring the ridge-curve first. If the first option is taken, after $x_c$ is reached, complete the linking-curve by following $\nabla d_c(y)$ to the other ridge-curve. Add the linking-curve and its other endpoint to $\mathcal{R}$.

(b) The goal slice $\mathcal{F}|_{x_g}$ is encountered at $x$: Make $x$ a node of $\mathcal{R}$. Decide whether to make a recursive call with $\hat{x}_s = x$, $\hat{x}_g = x_g$, and $\hat{\mathcal{F}} = \mathcal{F}|_{x_g}$; or to continue exploring the ridge-curve first. If a recursive call is made, STOP if $x_g$ is found.

A ridge-curve endpoint encountered:

(c) A fold-bifurcation encountered at $x$: Make $x$ a node of $\mathcal{R}$ (no other curves emanate from $x$).

(d) A boundary point of $\mathcal{F}$ is encountered at $x$: Make $x$ a node of $\mathcal{R}$ (typically no other curves emanate from $x$, but the ridge-curve might be continuing through $x$ if $x$ is a single-point stratum of $\mathcal{F}$).

(e) An explored node is encountered again: a full circle along $\mathcal{R}$ has been completed (e.g. when $x_k$ is a periodic coordinate). Add the explored curve to $\mathcal{R}$.

Until the explored roadmap contains no nodes with unexplored curves. In that case $x_g$ is NON-REACHABLE.

## 4 The Complete Algorithm

We now show that the naive algorithm must be augmented with a *minimum passage detector*.

### 4.1 The Naive Algorithm can Fail

The naive algorithm employs two means for departing from a ridge-curve: recursive calls in $\mathcal{F}|_{x_c}$ and recursive calls in $\mathcal{F}|_{x_g}$. It might happen that all the interesting critical points lie beyond the closure of the current ridge-curve basin. It might also happen that the same ridge-curve doesn't cross $\mathcal{F}|_{x_g}$. In Figure 4, the freespace is the interior of two vertical pipes joined by a horizontal pipe. There is an inner c-obstacle in the shape of a plate "hiding" the entrance to the connecting pipe. For the $\theta$ sweep direction, there are three ridge-curves: $R_1$, $R_2$ (associated with the plate), and $R_3$. Paths from the basin of $R_1$ to the basin of $R_3$ must pass through the basin of $R_2$. However, the interesting critical points $x_1$ and $x_2$ lie between the basins of $R_2$ and $R_3$ i.e., outside the closure of the basin of $R_1$. The algorithm, after exploring the entirety of $R_1$, erroneously concludes that the goal is not reachable. This false conclusion persists even if we make local changes in the sweep direction.

### 4.2 The Minimum Passage Detector

Although many issues concerning the critical-point detector are still open, it is unreasonable to require that

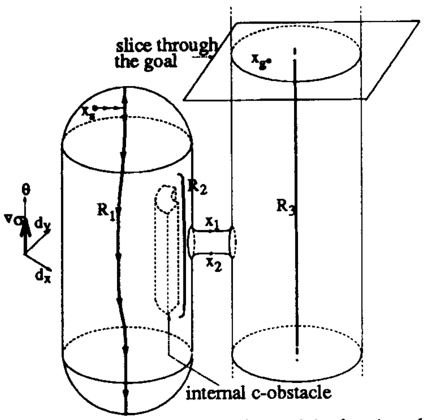


**Figure 4.** There are no interesting critical points between the basins of $R_1$ and $R_2$

it would be able to detect critical points beyond the closure of the current ridge-curve basin. The minimum passage detector we now describe enables the robot to move between adjacent basins whose common boundary is not marked by interesting critical points.

Let $d(x)$ be a general smooth repulsive potential. Bifurcation theory (Section 2) guarantees that the equilibria of a generic $\nabla d_c(y)$ form smooth curves. Let the *saddle curves* be the ones traced by saddles of $d_c(y)$, as $c$ varies in $\mathbb{R}$. Analogous to interesting critical points, only the following type of saddles is of interest to our algorithm. They are described in terms of the *Morse index*. Given a Morse function $f$, its Morse index at a critical point $y_0$ is the number, $\lambda$, of negative eigenvalues of its second derivative matrix $D^2 f(y_0)$.

**Definition 1** *Let $\mathcal{X} \subset \mathbb{R}^m$ be a stratified set with non-empty interior, and let $f$ be a Morse function on $\mathcal{X}$. A saddle point of $f$ in the interior of $\mathcal{X}$*[2] *is an* **interesting saddle** *if $\lambda = 1$ or $\lambda = m - 1$.*

In our case $f(y) = d_c(y)$, $\mathcal{X} = \mathcal{F}|_c$, and $m = k - 1$. An interesting saddle would be, using the case $\lambda = m - 1 = k - 2$ for example, a local maximum along $k - 2$ directions, and a local minimum along the remaining single direction in $\mathcal{F}|_c$. Curves traced by an interesting saddle are termed interesting saddle curves. It is shown in Section 5 that between any two adjacent ridge-curve basins lies at least one interesting saddle curve.

It is required of the minimum passage detector that: *from a ridge-curve, it must detect the local minima of $d(x)$ on the interesting saddle curves associated with the ridge-curve's basin of attraction.* (Points $x$ are restricted to the saddle curve, while $d(x)$ is the distance between the robot $\mathcal{A}(x)$ and the entire obstacle set $\mathcal{B}$.) The local minima of $d(x)$ along the interesting saddle curves are termed *minimum passage points* (Figure 6).

[2] This can be extended to interesting saddles on $\mathcal{X}$'s boundary.

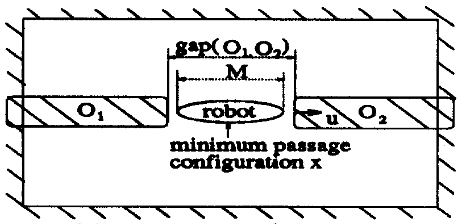


**Figure 5.** minimum passage configuration

**Implementation of the minimum passage detector:** Consider Figure 5 as an example. The robot is an ellipse with major axis $M$. The repulsive potential is the Euclidean distance $d(x) = \text{dst}(\mathcal{A}(x), \mathcal{B})$. For a sweep along the orientation axis, the minimum passage points are characterized as follows. Consider a pair of disjoint obstacles, $\mathcal{O}_1$ and $\mathcal{O}_2$ shown in the figure. Let $\text{gap}(\mathcal{O}_1, \mathcal{O}_2)$ be the minimal distance between them, and let $u$ be the direction of the line along which $\text{gap}(\mathcal{O}_1, \mathcal{O}_2)$ is attained. $\text{gap}(\mathcal{O}_1, \mathcal{O}_2)$ is larger than $M$, hence the c-obstacles corresponding to $\mathcal{O}_1$ and $\mathcal{O}_2$ are disjoint. The ellipse orientation that maximizes its width along $u$ is the one where its major axis is collinear with $u$. Let $c_0$ be the corresponding ellipse orientation. Further, let $x_0 = (\vec{y}_0, c_0)$ be the ellipse configuration where its center is at the center of the gap.

Intuitively, $y_0$ is a local minimum of $d_{c_0}(y)$ for translations perpendicular to $u$, and is a local maximum for translations parallel to $u$. Hence $y_0$ is an interesting saddle point. Moreover, $x_0$ is a local minimum of $d(x)$ along the saddle curve containing $x_0$. To see this, consider translations of the ellipse with its orientation fixed to $c$ close to $c_0$. Consider, in particular, translations perpendicular to $u$, such that the ellipse crosses the gap from one side to the other, while maximizing its distance from both obstacles. It can be intuitively observed that the minimum value of $d$ along this motion is larger than $d(x_0)$. Hence $x_0$ is a local minimum of $d(x)$ on the saddle curve. Research now under progress will make this statement precise, as well as extend it to solid bodies.

The complete algorithm is identical to the one described in Section 3. Only that the sensor list is augmented by the minimum passage detector, and step 2(a) is augmented with recursive calls for construction of linking-curves through the minimum passage points.

### 4.3 Connection Between the Two Detectors

The following lemma establishes a connection between interesting critical points and minimum passage points.

**Lemma 4.1 ([13])** *Let $d(x)$ be a smooth repulsive potential on $\mathcal{F}$* (not necessarily the Euclidean distance). *The* **interesting critical points** *are exactly the* **minimum passage points** *that occur on the boundary of* $\mathcal{F}$ (Figure 6(b)).

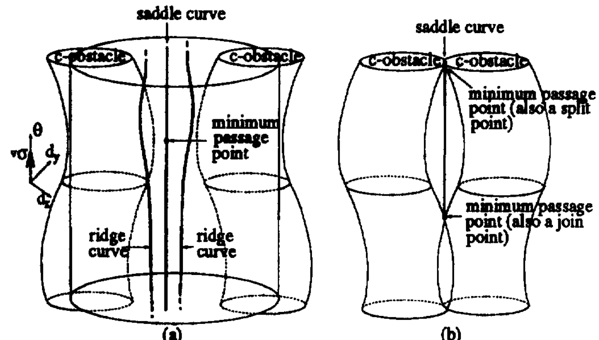


**Figure 6.** (a) minimum passage point in the interior of $\mathcal{F}$ (b) and on the boundary of $\mathcal{F}$

It follows from the lemma that *the minimum passage detector generalizes the critical point detector*. However, the interesting critical points (equivalently, the minimum passage points on the boundary of $\mathcal{F}$) correspond to local change in the connectivity of $\mathcal{F}$ along the sweep direction and depend only on $\mathcal{F}$ and on the sweep function $\sigma$. The interior minimum-passage points additionally depend on the specific repulsive potential $d$.

## 5 Correctness of the Algorithm

We are concerned with the following generic situation. First, $\mathcal{F}$ is assumed to be a regularly stratified set (Section 2). Second, $\mathcal{F}$ is assumed to be a regular set i.e., a set that is equal to the closure of its interior. This ensures that each stratum of $\mathcal{F}$ is either an open set in the ambient $\mathbb{R}^k$, or a lower dimensional manifold on the boundary of such a set. Next, the repulsive potential $d$ is assumed to be a Morse function on $\mathcal{F}$. It is known that almost all the smooth functions on a given stratified set are Morse. We also assume that $\mathcal{F}$ has no special symmetries, so that the one-parameter family $\nabla d_c(y)$ is generic (Section 2). A curve traced by an equilibrium of $\nabla d_c(y)$ of a specific type, as $c$ varies in $\mathbb{R}$, is termed *equilibria curve*. For instance, ridge-curves are equilibria curves of local maxima.

**Lemma 5.1 ([13])** *Under the generic assumptions made above, and for a bounded robot workspace, there are finitely many* **equilibria curves** *in $\mathcal{F}$.*

We are ready to show that the algorithm terminates.

**Proposition 5.2 ([13])** *Let the robot be an articulated $k$ degrees-of-freedom mechanism. Then the incremental roadmap algorithm* **terminates** *on every bounded workspace.*

**Sketch of proof:** The main step is to show that there are finitely many nodes in $\mathcal{R}$. According to Lemma 5.1, there are finitely many ridge-curves. Hence there are finitely many ridge-curve endpoints. Lemma 5.1 also asserts that there are finitely many interesting saddle-curves, and according to Lemma 4.1 every interesting critical point is the endpoint of an interesting saddle-curve. Hence there are also finitely many interesting critical points. Finally, we show that every interior minimum-passage point is a critical point of $d(x)$ in the ambient $\mathbb{R}^k$. There are finitely many such points since $d(x)$ is a Morse function. □

We will need the following mountain pass theorem, adapted for our purposes from [11]. Let $\mathcal{X} \subset \mathbb{R}^m$ be a stratified set with non-empty interior, which is closed in $\mathbb{R}^m$. Let $f$ be a Morse function on $\mathcal{X}$, such that $f$ is zero on the boundary of $\mathcal{X}$ and is strictly negative in its interior. Let $\mathcal{B}_1, \mathcal{B}_2 \subset \mathcal{X}$ be two distinct basins of attraction of the flow of $-\nabla f$, each attracted to a local minimum of $f$. Let $\overline{\mathcal{B}}_i$ denote the closure of $\mathcal{B}_i$, for $i = 1, 2$. The set $\mathcal{V} = \overline{\mathcal{B}}_1 \bigcap \overline{\mathcal{B}}_2$, if non-empty, represents a "mountain range"—a set separating $\mathcal{B}_1$ from $\mathcal{B}_2$ on which $f$ attains higher values.

**Theorem 1 (Mountain pass theorem)** *If $\mathcal{V}$ is non-empty, $f$ has an* **interesting saddle point** *in $\mathcal{V}$.*

In our case $f(y) = -d_c(y)$ and $\mathcal{X} = \mathcal{F}|_c$. $\mathcal{B}_1$ and $\mathcal{B}_2$ are two "mountains" separated by "valleys". The interesting saddle occurs along the path connecting the two mountain tops that minimizes the amount of descent. The following theorem is the main result of this paper.

**Theorem 2 ([13])** *Let the robot be an articulated $k$ degrees-of-freedom mechanism. If $x_g$ is reachable from $x_s$, the incremental roadmap algorithm finds at least one* **d-point** *in $\mathcal{F}|_{x_g}$ from which $x_g$ is* **reachable.** *Through recursive invocation of the algorithm from d-points in $\mathcal{F}|_{x_g}$,* **a path to $x_g$ is found.**

**Sketch of proof:** First we show that every point in $\mathcal{F}$ lies in the closure of some ridge-curve basin. Then we invoke the mountain pass theorem to show that an interesting saddle curve lies between every two adjacent ridge-curve basins. The critical-point detector and the minimum-passage detector locate at least one point on each interesting saddle curve. Hence linking-curves are constructed to every ridge-curve reachable from $x_s$. □

## 6 Discussion and Open Problems

The feasibility of the algorithm hinges on the need to understand which environmental features correspond to the c-space features sought by the *critical point detector* and the *minimum passage detector*. In planar body navigation, for instance, we have seen that these features correspond to passages between obstacles in the robot immediate environment. A companion paper now under preparation will discuss this issue in detail, for a rigid body navigation. We have not looked into the more complex case of kinematic chains. But even with rigid-body navigation we encountered the following important problem of active perception. As the robot moves along a ridge-curve, some obstacles in the robot immediate environment may occlude its view of features that mark interesting critical points or minimum passage points. Therefore, the associated problem is: *given a sensor specification, how to augment the incremental algorithm with active excursions away from a ridge-curve, in a way that guarantees coverage of the entire ridge-curve basin.*

## References


[1] J. Barraquand and J.C. Latombe, *A Monte-Carlo Algorithm for Path-Planning With Many degrees of Freedom*, ICRA 1990, 1712-1717

[2] J. F. Canny, *The complexity of robot motion planning*, MIT Press, 1988

[3] F. H. Clarke, *Optimization and Nonsmooth Analysis*, SIAM Publication, 1990

[4] J. Cox and C.-K. Yap, *On-line Motion Planning: Case of a Planar Rod*, Annals of Math. and Artificial Intelligence, 1991, 3:1-20

[5] B. R. Donald, *Information Invariants in Robotics: I-State, Communication, and Side-Effects*, ICRA 1993, 1:276-283

[6] B. Faverjon and P. Tournassoud, *A Practical Approach to Motion Planning for Manipulators with Many Degrees of Freedom*, ISRR 1989, 65-73

[7] Goresky and Macpherson, *Stratified Morse Theory*, Springer-Verlag, 1980

[8] D. Kapur and Y. N. Lakshman, *Elimination Methods: An Introduction*, J. Symb. Computation, 1992, 11:42-58

[9] M. C. Lin and J. F. Canny, *An Opportunistic Global Path Planner*, ICRA 1990, 1554-1558

[10] V. Lumelsky and A. Stepanov, *Path Planning Strategies for Point Automaton Moving Amidst Unknown Obstacles*, Algorithmica 1987 2:403-430

[11] R. S. Palais and C.-L. Terng, *Critical Point Theory and Submanifold Geometry*, Springer-Verlag, 1988

[12] T. Poston and I. Stewart, *Catastrophe Theory and its Applications*, Pitman, 1978

[13] E. Rimon and J. Canny, *Construction of C-space Roadmaps Using Local Sensory Data*, Tech Report, ME Dept., Caltech, 1992 (revised June 1993)

[14] E. Rimon and D. E. Koditschek, *Exact Robot Navigation Using Artificial Potential Functions*, IEEE Trans. on Robotics&Automation, 1992 8:501-518

[15] J. T. Schwartz and M. Sharir, *On the piano movers problem II.*, Advances in Applied Math., 1983, 4:298-351

[16] C. Yap, *Algorithmic Motion Planning*, Lawrence Erlbaum Assoc., 1987, 95-143