HPRM: A Hierarchical PRM

Anne D. Collins
Department of Mathematics,
Stanford University
collins@math.stanford.edu

Pankaj K. Agarwal
Department of Computer Science,
Duke University
pankaj@cs.duke.edu

John L. Harer
Department of Mathematics,
Duke University
john.harer@duke.edu

***Abstract*— We introduce a hierarchical variant of the probabilistic roadmap method for motion planning. By recursively refining an initially sparse sampling in neighborhoods of the $\mathcal{C}$-obstacle boundary, our algorithm generates a smaller roadmap that is more likely to find narrow passages than uniform sampling. We analyze the failure probability and computation time, relating them to path length, path clearance, roadmap size, recursion depth, and a local property of the free space. The approach is general, and can be tailored to any variety of robots. In particular, we describe algorithmic details for a planar articulated arm.**

I. Introduction

One of the central problems in robotics is the *motion planning problem*: Given a robot $\mathfrak{R}$ and a workspace $\mathcal{W}$ containing a set $\mathcal{O}$ of obstacles, determine a collision-free motion between specified initial and final configurations of $\mathfrak{R}$ [16].

The set of configurations of a robot $\mathfrak{R}$ with d degrees of freedom can be represented by a d-dimensional *configuration space* $\mathcal{C}$. The *free space* $\mathcal{F} \subseteq \mathcal{C}$ is defined to be the set of configurations in which $\mathfrak{R}$ does not intersect any obstacle. The motion planning problem can be formulated as computing a path in $\mathcal{F}$ between two given configurations. The best known algorithm for motion planning, by Canny [4], takes time $n^{O(d)}$, where n measures the complexity of the obstacles, and the problem is known to be PSPACE-Hard [18]. Canny's algorithm and many other approaches compute a one-dimensional *roadmap* $\mathcal{R}$: a graph embedded in $\mathcal{C}$ that correctly captures the connectivity of $\mathcal{F}$, in the sense that any path in $\mathcal{F}$ is homotopic to a path in $\mathcal{R}$. See [10] for a survey on motion planning.

To get faster algorithms, we must sacrifice completeness, by allowing the planner to occasionally fail to return a collision-free path even though one exists. Recent research has focused on Monte Carlo approaches to motion planning, where the best we can hope for is "probabilistic completeness," an assurance that we find a solution with high probability. Our approach is a variant of the *probabilistic roadmap method* (PRM) [15]. The basic PRM constructs a roadmap $\mathcal{R}$ by sampling the free space uniformly and connecting a pair of sample configurations if a free path joining them can be easily generated by a *simple planner*, which tests, for example, whether the straight line joining them is free. A motion-planning query is then answered by connecting the initial and final configurations to nearby sample points, and searching in $\mathcal{R}$ for a path between them. This approach is particularly useful in situations in which a large number of paths are to be planned in the same environment. PRM variants seek to enhance the roadmap in narrow corridors that uniform sampling is likely to miss [3], [11], [10]. Although these planners perform well in practice, little work has been done on analyzing their performance [14], [13], [12].

Previous Work. Kavraki et al. [14] were the first to analyze the performance of PRM. They introduce the notion of an ε-good free space, one for which the simple planner connects any $x \in \mathcal{F}$ to at least a fraction ε of $\mathcal{F}$, and present a probabilistically complete planner for such spaces. But their algorithm relies on calls to an expensive complete planner. For ε-good free spaces that satisfy some extra assumptions, the so-called *expansive* free spaces of [12], a complete planner is no longer needed. However, the extra conditions are not very natural, and the algorithm is specifically tailored to "single-shot" planning, where only one query is asked. Answering multiple queries may require building a new roadmap each time.

An alternative approach is taken in [13], where the probability that two points are connected is related to the length and clearance of a path between them, and the number of points sampled. Since they sample points uniformly at a density determined by the narrowest part of $\mathcal{F}$, it oversamples high clearance regions. Hence the efficiency could be adversely affected by a tiny small-clearance region.

Our Results. We present a hierarchical variant of the probabilistic roadmap method for motion planning. Our approach combines a local property similar to that in [14] and the clearance results of [13], with a hierarchical sampling scheme that samples only as densely as the path clearance dictates, and does not rely on calls to a deterministic planner. The roadmap we generate is not only typically smaller than that obtained by uniform sampling, but also has a higher probability of success. The approach is general and can be tailored to any variety of robots. In particular, we provide algorithmic details for a planar articulated arm in Section V.

II. Preliminaries

A roadmap of $\mathcal{F}$ is a graph $\mathcal{R} = (\mathcal{M}, \mathcal{E})$, where $\mathcal{M}$ is a set of sampled configurations, called *milestones*, of $\mathcal{F}$. Assume that we have a *simple planner*, which quickly constructs a free path of "simple shape" between two milestones if there exists one; for example, a common approach is to check whether the straight segment joining two milestones is free. We add an edge $(x,y) \in \mathcal{E}$ if the simple planer produces a free path from x to y. A simple planner thus trades efficiency with accuracy — it may not find a free path between two configurations even if there exists one. For technical reasons we assume that whenever there exists a free ball containing two configurations, the simple planner successfully finds a path between them.

In the following, we denote the ball in $\mathcal{C}$ of radius r centered at x by $B(r,x)$, and denote the volume of a subset $S \subseteq \mathcal{C}$ by $|S|$. The *clearance* $cl(x)$ of $x \in \mathcal{F}$ is the distance from x to the nearest point on $\partial\mathcal{F}$; the clearance of a path is the minimum clearance over all points on the path.

We next introduce a local property of the free space, which is similar to a localization of the ε-good property of [14], except that ours is not defined in terms of visibility.

Definition 2.1: $\mathcal{F}$ is *(δ,ε)-free* if for all $\delta' \leq \delta$ and $x \in \mathcal{F}$, at least a fraction ε of the points within δ' of x are free, that is,

$$|B(\delta',x) \cap \mathcal{F}| \geq \varepsilon \cdot |B(\delta',x)|.$$

For example, a square S of side length 1 is $(1,1/4)$-free since $|B(\delta,x) \cap S| \geq |B(\delta,x)|/4$ for any $x \in S$, $\delta \leq 1$. Intuitively, spaces with narrow passages can still have good values for (δ,ε), provided that the narrow regions are relatively "short", surrounded only by "thin" obstacles, or narrow in only a few dimensions.

Finally, we review the result of [13]: If configurations a and b can be connected by a path $\Gamma : [0,L] \to \mathcal{F}$ that has clearance at least r, then by uniformly sampling $\mathcal{F}$ at an appropriate density, we are guaranteed to connect a to b in the resulting roadmap with high probability. The argument proceeds by covering Γ with balls of radius $r/2$ whose centers are no more than $r/2$ apart. Milestones in consecutive balls will always be connected, since there exists a free ball of radius r which contains both $r/2$-balls. Therefore, if we sample $\mathcal{F}$ so densely that there is a milestone in each of these balls, the roadmap generated will connect a to b.

III. Hierarchical PRM

Let $\Gamma : [0,L] \to \mathcal{F}$ be a free path joining configurations a and b. If only a small portion of Γ has small clearance, then uniformly sampling $\mathcal{F}$ at the density determined by the minimum clearance is unnecessary; it suffices to sample so densely only in regions of small clearance. As

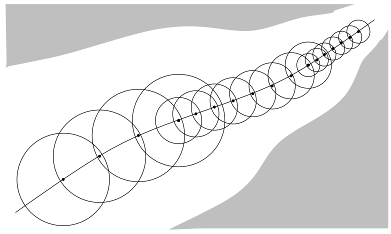


Fig. 1. Higher clearance parts of Γ are covered by larger balls.

in [13], we imagine covering Γ with balls, but we now let the *local* clearance of Γ dictate the radius of each ball; see Figure 1. We then sample $\mathcal{F}$ so that each ball contains a milestone.

Our sampling scheme proceeds as follows: Let

$$\mathcal{F}(r) = \{x \in \mathcal{F} \mid cl(x) \leq r\}$$

be the free r-neighborhood of $\partial\mathcal{F}$. We choose two parameters r_1 and N appropriately, and set $r_i = r_{i-1}/2$ for $i > 1$. We select the milestones in phases. We perform two steps in the ith phase of the algorithm.

(S1) We randomly sample a collection $\mathcal{M}_i$ of points, called *i-milestones*, in $\mathcal{F}(5r_i/2)$ at a uniform density $N/|B(3r_i,0)|$.

(S2) For each i-milestone x, we use the simple planner to decide which milestones in $\mathcal{M}_{i-1} \cup \mathcal{M}_i$ that lie within a distance $3r_i/2$ of x should be connected to x.

Uniformly Sampling $\mathcal{F}(r)$. In order to sample a neighborhood of the $\mathcal{C}$-obstacles without explicitly computing them, assume that we have available a *clearance oracle* that determines, given $x \in \mathcal{F}$ and a radius r, whether $cl(x) \leq r$. Note that we only need a bound on $cl(x)$, not the exact value. We describe such an oracle for a robot arm in Section V.

Our approach is based on the following straightforward idea, which extends to multiple sets in the obvious way.

Lemma 3.1: *The union $A \cup B$ of two sets A and B can be uniformly sampled by first uniformly sampling A at the desired density, then uniformly sampling B at the same density, but discarding those points that are also in A.*

Suppose that we have ordered the $(i-1)$-milestones, and let $X_j = \langle x_1, \ldots, x_j \rangle \subseteq \mathcal{M}_{i-1}$; note that any order will do, as it is for bookkeeping purposes only. Then step (S1) of HPRM proceeds as follows:

(S1a) For $x_j \in \mathcal{M}_{i-1}$, we uniformly sample a collection Y_j of N points in $B(3r_i,x_j)$.

(S1b) For each $y \in Y_j$, we add y to $\mathcal{M}_i$ if $y \in \mathcal{F}(5r_i/2)$ and if $B(3r_i,y) \cap X_{j-1} = \emptyset$.

That is, we sample N potential i-milestones around each $(i-1)$-milestone, but discard those which lie in the balls already covered by previously treated $(i-1)$-milestones. Lemma 3.1 implies that $\mathcal{M}_i$ is a uniform $(N/|B(3r_i,0)|)$-dense sampling of $\mathcal{F}(5r_i/2)$, provided that the balls $B(3r_i,x_j)$ do indeed cover $\mathcal{F}(5r_i/2)$.

IV. Analysis

In this section, we sketch a proof of the probabilistic completeness of HPRM. A number of details have been omitted from this abstract; see [7] for a complete proof.

Let r_1 be the initial radius parameter from Section III, and suppose that $\mathcal{F}$ is (r_1,ε)-free for some ε. Then, by Def. 2.1, for any $x \in \mathcal{F}$ and for any i, at least a fraction ε of the points within distance $r_i = r_1/2^{i-1}$ of x are free. The next lemma dictates the parameter N.

Lemma 4.1: *Let $x \in \mathcal{M}_{i-1}$, let Y be a set of N points selected uniformly from $B(3r_i,x)$, and let $S = Y \cap \mathcal{F}$. Choosing*

$$N \geq \max\left\{6^d \ln\left(\frac{1}{\gamma}\right), \frac{3^d}{\varepsilon}\ln\left(\frac{1}{\gamma}\right)\right\},$$

we have

(a) $z \in \mathcal{F} \cap B(2r_i,x) \implies \Pr[B(r_i,z) \cap S = \emptyset] \leq \gamma.$

(b) $z \in \mathcal{F} \cap B(2r_i,x)$ *and* $cl(z) \geq r_i/2$
$$\implies \Pr[B(r_i/2,z) \cap S = \emptyset] \leq \gamma.$$

The proof of Lemma 4.1 is straightforward. Roughly speaking, it says that for any $z \in \mathcal{F}$ within distance r_{i-1} of an $(i-1)$-milestone, the following two conditions hold with probability at least $1-\gamma$: (a) there is an i-milestone within r_i of z, regardless of the clearance of z; and (b) if, in addition, z has clearance at least $r_i/2$, then there is an i-milestone within $r_i/2$ of z.

Since the probability that a given ball contains an i-milestone depends upon the presence of a nearby $(i-1)$-milestone, which in turn depends on the presence of a nearby $(i-2)$-milestone, and so on, we can prove the following lemma.

Lemma 4.2: *If $cl(x) \geq r_i/2$, then the probability that we fail to sample an i-milestone in $B(r_i/2,x)$ is at most $i \cdot \gamma$.*

Let Γ be a collision-free path between two configurations a and b. Partition Γ into pieces, where Γ_i is the portion of Γ whose clearance is between r_i and r_{i-1}. Cover Γ_i with balls of radius $r_i/2$ whose centers lie on Γ_i and are no more than $r_i/2$ apart. If there is an i-milestone in each of the $2|\Gamma_i|/r_i$ balls of radius $r_i/2$ covering Γ_i, then our assumption that the simple planner connects two milestones contained in a free ball ensures that the resulting roadmap will connect a to b. Thus the probability that we fail is bounded by the probability that one of the balls covering Γ does not contain a milestone. Setting $L_i = |\Gamma_i|$, Lemma 4.2 implies that:

Theorem 4.3: *Let $\Gamma : [0,L] \to \mathcal{F}$ be a free path from a to b which has clearance at least r_k. Then the probability that* HPRM *fails to connect a and b is*

$$\Pr[\text{fail}] \leq 2\gamma \sum_{i=1}^{k} \frac{iL_i}{r_i}.$$

We next bound the time to generate the roadmap. Let τ (resp. σ) be the time taken by the clearance oracle (resp. simple planner). Let m_i be the number of i-milestones, and consider the ith phase of HPRM. For each of the $m_{i-1}N$ points x sampled in step (S1a), step (S1b) takes time τ to determine whether $x \in \mathcal{F}(5r_i/2)$ and another $O(m_{i-1})$ time to determine whether x lies in any of the previously sampled balls. In step (S2), the simple planner tests each i-milestone for connection with $O(m_{i-1}+m_i)$ $(i-1)$- and i-milestones.

Theorem 4.4: *Suppose that $\mathcal{F}$ is (r_1,ε)-free. Then the time spent on the ith phase of HPRM is*

$$O(\tau m_{i-1}N + m_{i-1}^2 N + \sigma m_i(m_{i-1}+m_i)),$$

where m_i is the number of i-milestones in $\mathcal{R}$, N is as in Lemma 4.1, and τ (resp. σ) is the time taken by the clearance oracle (resp. simple planner).

Remark 4.5: The running time can be improved in certain cases. Suppose we use the L_∞-norm on $\mathcal{C}$, as in Section V. Then balls are axis-parallel hypercubes, and the problem of finding the i-milestones in a given ball can be formulated as answering an *orthogonal range query*. Using a data structure known as a *kd-tree* [2], we can reduce some factors of m to $m^{1-1/d}$; see [7] for details.

Since they are stated in full generality, Theorems 4.3 and 4.4 are rather cumbersome. In order to compare HPRM to the uniform sampling scheme of [13], let us suppose that both L_i and $|\mathcal{F}(5r_i/2)|$ decrease as i increases. In particular, we assume that there exist two constants $0 < c_1, c_2 < 1$ such that:

(A1) $L_{i+1} \leq c_1 L_i$ for all i, and

(A2) $|\mathcal{F}(5r_{i+1}/2)| \leq c_2 |\mathcal{F}(5r_i/2)|$

Corollaries 4.6 and 4.7 bound the failure probability and roadmap size, respectively, subject to these assumptions. Proofs can be found in [7].

Corollary 4.6: *Let $\Gamma : [0,L] \to \mathcal{F}$ be a path from a to b with clearance at least r_k that satisfies* (A1) *with $c_1 \leq 1/2$. Then the probability that we fail to connect a to b in $\mathcal{R}$ is*

$$\Pr[\text{fail}] \leq (\gamma L/r_1) \cdot k^2.$$

For comparison, if we sample $\mathcal{F}$ at the k-milestone density, the argument in [13] gives $\Pr[\text{fail}] \leq (\gamma L/r_1) \cdot 2^k$; thus, our algorithm is more likely to find a path.

The value of c_1 in assumption (A1) which ensures that our hierarchical roadmap has a smaller failure probability than uniform sampling depends upon k, the recursion depth. In fact, it seems likely that the appropriate bound in Lemma 4.6 is $c_1 \leq 1/\sqrt[k-1]{4}$. Lemma 4.7 shows that a similar bound holds for the constant c_2.

COROLLARY 4.7: *Suppose that the free space $\mathcal{F}$ satisfies* (A2) *for some $c_2 \leq 1/\sqrt[k-1]{2}$. Then the total number of milestones in $\mathcal{R}$, to level k, is $|\mathcal{M}| = \sum m_i < u$, where u is the number of points needed to uniformly sample all of $\mathcal{F}$ at the k-milestone density.*

Thus HPRM generates a *smaller* roadmap for any free space that satisfies (A2). Note that in the limit as $i \to \infty$, we have $|\mathcal{F}(5r_{i+1}/2)|/|\mathcal{F}(5r_i/2)| \to 1/2$, so this is not a particularly strong condition; in fact, if $|\mathcal{F}(5r_{i+1}/2)|/|\mathcal{F}(5r_i/2)|$ is roughly $1/2$ for all i, then $|\mathcal{M}| \leq u/2^{k-1}$, and our roadmap is significantly smaller.

V. CLEARANCE ORACLE FOR A PLANAR ARM

In this section, we discuss how to implement the clearance oracle for a d-link planar articulated arm $\mathfrak{R}$ with revolute joints, which has one of its endpoints anchored at the origin of the 2-dimensional workspace $\mathcal{W}$. $\mathfrak{R}$ has one degree of freedom for each link, and therefore a total of d degrees of freedom. Let $\phi = (\phi_1, \dots, \phi_d) \in \mathcal{C}$ correspond to the configuration of the arm in $\mathcal{W}$ where ϕ_i is the absolute orientation of link i, as measured with respect to the x-axis of $\mathcal{W}$. We permit $\mathfrak{R}$ to intersect itself. Note that $\mathfrak{R}$ can only reach points in the component of $\mathcal{W} - \mathcal{O}$ that contains the base, so any hole in a $\mathcal{W}$-obstacle can be ignored. We therefore assume that $\mathcal{O}$ is a collection of pairwise disjoint simple polygons. We assume that $\mathcal{W}$ is bounded by a rectangle. An obstacle $O \in \mathcal{O}$ maps to a $\mathcal{C}$-obstacle, corresponding to the set of configurations for which $\mathfrak{R}$ intersects O.

A key idea underlying our algorithm is to project a ball or path in $\mathcal{C}$ to the 2-dimensional workspace $\mathcal{W}$, and determine whether the image intersects the $\mathcal{W}$-obstacles $\mathcal{O}$. Although the details of how to compute the image in $\mathcal{W}$ of a $\mathcal{C}$-space ball or path depends upon the robot in question, the method we present for intersecting this $\mathcal{W}$-image with $\mathcal{O}$ is more general.

A configuration ϕ maps to $\mathcal{W}$ as follows: Parametrize link i by t, so that $t = 0$ at joint $(i-1)$ and $t = 1$ at joint i. If the length of link i is ℓ_i, then the location of the point t along link i when $\phi = (\phi_1, \dots, \phi_d)$ is given by the map $\Pi_i : \mathcal{C} \times [0,1] \to \mathcal{W}$, defined recursively by

$$\Pi_i(\phi,t) = \Pi_{i-1}(\phi,1) + (t \cdot \ell_i \cos\phi_i, t \cdot \ell_i \sin\phi_i),$$

where $\Pi_0(\phi,t) = (0,0)$. The portion of $\mathcal{W}$ occupied by link i in configuration ϕ is $\Pi_i(\phi,[0,1])$.

To determine whether $cl(\phi^0) \leq \rho$, we compute for each $i \leq d$,

$$\Lambda_i = \Pi_i(B(\rho,\phi^0),[0,1]),$$

the set of all points in $\mathcal{W}$ that link i will occupy for some $\phi \in B(\rho,\phi^0)$. Then $cl(\phi^0) \leq \rho$ if and only if one of these *link regions* Λ_i intersects $\mathcal{O}$.

Before proceeding, we note that some care must be taken when working in this non-Euclidean configuration space. In particular, the distance between two points on $\mathbb{S}^1$ is the length of the shorter of the two paths around the circle, and is therefore always less than π. For example, $\text{dist}(5\pi/3, \pi/3) = 2\pi/3$, and not $4\pi/3$. We use the L_∞ norm on $\mathcal{C}$, where the distance between ϕ and ϕ^0 is $|\phi - \phi^0| = \max_i\{\text{dist}(\phi_i - \phi_i^0)\}$.

COMPUTING THE LINK REGIONS Λ_i. Note that Π_i is independent of ϕ_j for $j > i$, so we can view Λ_i as the image under Π_i of the product of an i-dimensional ball with the interval $[0,1]$.

Λ_1 is easy to compute. $\Pi_1(\phi,t) = (t\ell_1\cos\phi_1, t\ell_1\sin\phi_1)$, and the image of $[\phi_1^0 - \rho, \phi_1^0 + \rho] \times [0,1]$ is a cone.

The projection for link 2 is $\Pi_2(\phi,t) = (x,y) \in \mathcal{W}$, where

$$\begin{aligned} x &= \ell_1 \cos\phi_1 + t\,\ell_2 \cos\phi_2, \\ y &= \ell_1 \sin\phi_1 + t\,\ell_2 \sin\phi_2. \end{aligned}$$

The link region Λ_2 is the image under Π_2 of the region

$$U = [\phi_1^0 - \rho, \phi_1^0 + \rho] \times [\phi_2^0 - \rho, \phi_2^0 + \rho] \times [0,1],$$

a subset of $\mathbb{S}^1 \times \mathbb{S}^1 \times [0,1]$. Clearly, the image of each edge of this cube is a candidate for a boundary curve of Λ_2. Unfortunately, these curves are not always sufficient.

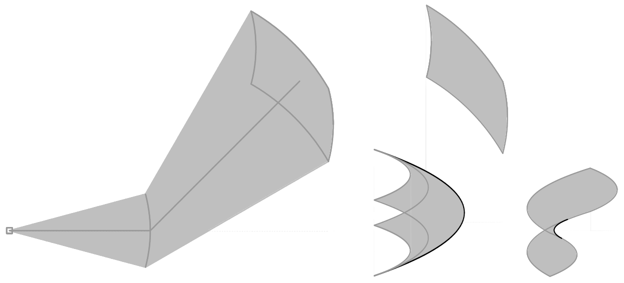
We recall the following result from differential topology; see any standard text, for example [9].

THEOREM 5.1: *Let M and N be two manifolds without boundary, and let $f : M \to N$ be a map from M to N. Suppose that f is a* submersion *at $p \in M$; that is, the derivative map $df_p : T_pM \to T_{f(p)}N$ has rank $n = dim(N)$. Then there exists a neighborhood of $f(p)$ in $f(M)$ that is diffeomorphic to $\mathbb{R}^n$.*

If f is not a submersion at p, we say that p is a *critical point* of f, and $f(p)$ is a *critical value*.

Let V be an open face of U. Since V is a manifold without boundary, Theorem 5.1 applies to the restriction of Π_2 to V. Now, a point on $\partial\Lambda_2$ does not have a neighborhood in Λ_2 which is diffeomorphic to $\mathbb{R}^2$. Therefore, $\Pi_2|_V$ must fail to be a submersion at any $(\phi,t) \in V$ for which $\Pi_2(\phi,t) \in \partial\Lambda_2$.

To illustrate, consider the open face V of U with $t = 1$: $V = (\phi_1^0 - \rho, \phi_1^0 + \rho) \times (\phi_2^0 - \rho, \phi_2^0 + \rho) \times \{1\}$. The



(a) (b)

Fig. 2. (a) $\mathcal{W}$-region $\Lambda_1 \cup \Lambda_2$ swept out by a 2-link arm if $\phi^0 = (0, \pi/4)$, $\rho = \pi/12$. (b) The region swept out by the end of link 2, for various ϕ^0. Critical curves, where interior points map to the image boundary, are drawn in black.

restricted map $\Pi_2|_V = \Pi_2(\cdot, 1) : V \to \mathcal{W}$ has derivative

$$d\Pi_2(\cdot,1)_\phi = \begin{pmatrix} -\ell_1 \sin\phi_1 & -\ell_2 \sin\phi_2 \\ \ell_1 \cos\phi_1 & \ell_2 \cos\phi_2 \end{pmatrix}.$$

Its rank is less than $dim(\mathcal{W}) = 2$ if and only if the determinant vanishes:

$$\ell_1 \ell_2 (\cos\phi_1 \sin\phi_2 - \cos\phi_2 \sin\phi_1) = \ell_1 \ell_2 \sin(\phi_2 - \phi_1) = 0.$$

If $\sin(\phi_2 - \phi_1) \neq 0$ for any point in V, then the image of V is a nice embedding of itself; however, if V contains critical points, its image appears folded, or twisted. See Figure 2.

Let Σ_i be the collection of critical curves obtained by projecting to $\mathcal{W}$ the locus of critical points in $B(\phi^0, \rho) \times [0,1]$ where Π_i, restricted to one of its open faces, is not a submersion. Then $\partial\Lambda_i \subseteq \Sigma_i$. Note that Σ_i always contains at least the images of the edges of $B(\phi^0, \rho) \times [0,1]$, since any map from a 1-dimensional manifold to the plane has rank at most 1. We omit the proof of the following lemma.

LEMMA 5.2: *Each critical curve in Σ_i is a line segment or a circular arc. Moreover, if $\rho < \pi/4$, then $|\Sigma_i| = \Theta(3^i)$.*

INTERSECTING Λ_i WITH $\mathcal{O}$. Finally, we test each link region Λ_i for intersection with the obstacles. If none of the link regions intersect any obstacle, then $cl(\phi^0) > \rho$.

We describe how to detect an intersection between Λ_i and $\mathcal{O}$. Without loss of generality assume that $i = d$. Λ_d intersects $\mathcal{O}$ if and only if a vertex of $\mathcal{O}$ lies inside Λ_d, a vertex of Σ_d lies inside $\mathcal{O}$, or an edge of Σ_d intersects an edge of $\mathcal{O}$. We first test whether any vertex of Σ_d lies in $\mathcal{O}$. We preprocess $\mathcal{W} - \mathcal{O}$ in $O(n \log n)$ time for planar point-location queries [8] and query it with each vertex of Σ_d in $O(\log n)$ time per vertex. If any vertex lies inside $\mathcal{O}$, then $cl(\phi^0) < \rho$ and we are done.

Otherwise, we walk along each arc of Σ_d and test for intersection with $\partial\mathcal{O}$. This can be accomplished by constructing a data structure for the so-called *circular-arc-shooting problem*, studied in [1], [6], in which one wishes to preprocess a set of polygons so that the first intersection of a query circular arc and the polygons can be determined as one follows the circular arc. Cheng et al. [6] have shown that a simple polygon can be preprocessed, in time $O(n \log^2 n)$, into a data structure of size $O(n \log n)$ so that a circular-arc-shooting query can be answered in $O(\log^2 n)$ time. However, since $\mathcal{W} - \mathcal{O}$ is not simply connected, we cannot directly use the algorithm of [6]. Instead, we proceed as follows.

If $\mathcal{O}$ has w (simply) connected components, we introduce w artificial edges to $\mathcal{W}$, $w-1$ of which join the components of $\mathcal{O}$ into one simply connected obstacle and the final edge extends to the bounding box of $\mathcal{W}$. Let $\tilde{\mathcal{O}}$ denote the union of $\mathcal{O}$ with these auxiliary edges. If we regard each edge as a thin rectangle, $\mathcal{W} - \tilde{\mathcal{O}}$ is a simple polygon and we preprocess it using the algorithm by Cheng et al. Let $\mathcal{H}$ be the resulting data structure.

Given a curve $\alpha \in \Sigma_d$ that starts in $\mathcal{W} - \mathcal{O}$, we determine the first intersection point ξ of α with $\mathcal{W} - \mathcal{O}$ (along α). If ξ lies on an edge of $\mathcal{O}$, then α intersects $\mathcal{O}$. We thus conclude that $cl(\phi^0) < \rho$, and we are done. Otherwise, α intersects one of the artificial edges at ξ, and we begin a new search in $\mathcal{H}$ to detect whether the remaining portion of the arc intersects $\mathcal{O}$.

By choosing the artificial edges carefully, we can limit the number of searches performed. Chazelle and Welzl [5] showed that for any set of w points in the plane, one can construct a spanning path with the property that every circle intersects $O(\sqrt{w})$ of its edges. Such a path is said to have *low stabbing number*. The algorithm described in [5] takes $O(w^3)$ time, but by generalizing the algorithm described in [17], the running time can be improved to $O(w^{3/2} \log w)$. We choose one representative point from each component of $\mathcal{O}$. Let W be the set of w representative points. We compute the above spanning path π on W and choose our artificial edges from among the $O(n\sqrt{w})$ pieces of π in $\mathcal{W} - \mathcal{O}$. After having computed π, the last step can be accomplished in $O(n \log n)$ time by a sweep-line algorithm [8]. Since α crosses $O(\sqrt{w})$ auxiliary edges, we spend $O(\sqrt{w} \log^2 n)$ time determining whether an edge of Σ_d intersects $\mathcal{O}$.

Finally, suppose that no curve of Σ_d intersects $\mathcal{O}$. Then Λ_d intersects $\mathcal{O}$ only if one of the components of $\mathcal{O}$ lies entirely within a face of the arrangement Σ_d. In this case, it suffices to determine whether any representative point in W lies inside Λ_d. We partition Λ_d into $2^{O(d)}$ pseudo-trapezoids, each of which is bounded by two vertical edges and two arcs (circular arcs or line segments) of $\partial\Lambda_d$, e.g., by computing the vertical decomposition of Λ_d [19]. We then determine whether any of these pseudo-trapezoids Δ contain a point of W. This can be done by computing $O(\sqrt{w})$ intersection points of π and $\partial\Delta$ in a total of $O(\sqrt{w} \log w)$ time. If any segment of π intersects $\partial\Delta$ an odd number of times, then a point of W lies inside

Δ. Hence, we can test whether any component of $\mathcal{O}$ is contained in Λ_d in time $O(2^{O(d)}\sqrt{w}\log w)$.

In summary, the clearance oracle proceeds as follows: (i) preprocess $\mathcal{W}-\mathcal{O}$ in $O(n\log n)$ time for point-location queries; (ii) choose a set W of w representative points, one from each polygon of $\mathcal{O}$, construct a spanning path π of W so that a circular arc intersects only $O(\sqrt{w})$ edges of π, and preprocess π for computing intersections between π and a circular arc; and (iii) preprocess $\mathcal{O}$ for circular-arc-shooting queries in $O(n\log^2 n)$ time using π and the data strucure of Cheng et al. [6]. A query can be answered in $O(\sqrt{w}\log^2 n)$ time. Now, given a configuration ϕ^0 and a radius ρ, and for each $i=1,\ldots,d$, we compute the $2^{O(i)}$ curves Σ_i that define the link region Λ_i, and test whether Λ_i intersects $\mathcal{O}$ in time $O(\sqrt{w}\,\log^2 n\,2^{O(i)})$. Since $\sum_{i=1}^{d} 2^{O(i)} = 2^{O(d)}$, we have the following theorem:

THEOREM 5.3: *For a planar articulated arm with d links, fixed base, and revolute joints, moving among w disjoint simple polygonal obstacles with a total of n vertices, we can determine whether the clearance of a configuration is within a specified bound in time* $\tau = 2^{O(d)}\sqrt{w}\log^2 n$ *after spending* $O(n\log^2 n + w^{3/2}\log w)$ *time in preprocessing the obstacles in* $\mathcal{W}$.

Note that the dependence on d and n is completely decoupled, and that τ is sublinear in n.

REMARK 5.4: The projection technique desribed above can also be used as a simple planner for a d-link planar arm. Given a pair of configurations u and v, it either reports that it cannot find a collision-free path between u and v, or it returns a collision-free path consisting of at most d line segments, each of which is parallel to an axis (i.e., only one ϕ_i changes along each edge). The running time is $O(d^2\sqrt{w}\log^2 n)$. We omit the details from this extended abstract. Unlike the most commonly used simple planner, which determines whether a path is free by sampling points along the path and checking whether all the sampled points lie in the free space, this planner never returns a path that is not collision free.

VI. REFERENCES

[1] P. K. Agarwal and M. Sharir. Circle shooting in a simple polygon. *J. Algorithms*, 14(1):69–87, 1993.

[2] P.K. Agarwal and J. Erickson. Geometric range searching and its relatives. Advances in Discrete and Comput. Geom. (B. Chazelle, J. Goodman, and R. Pollack, eds.), AMS, Providence, 1998.

[3] N.M. Amato, O.B. Bayazit, L.K. Dale, C. Jones, and D. Vallejo. OBPRM: An obstacle-based prm for 3d workspaces. In *Proc. of the Workshop on Algorithmic Foundations of Robotics*, pages 155–168, March 1998.

[4] J.F. Canny. *The Complexity of Robot Motion Planning*. M. I. T. Press, Cambridge, 1988.

[5] B. Chazelle and E. Welzl. Quasi-optimal range searching in spaces of finite VC-dimension. *Discrete and Computational Geometry*, 4:467–490, 1989.

[6] S.-W. Cheng, H. Everett, O. Cheong, and R. van Oostrum. Hierarchical vertical decompositions, ray shooting, and circular arc queries in simple polygons. In *Proc. 15th Symp. Comp. Geom.*, pages 227–236, 1999.

[7] A.D. Collins. *Configuration Spaces in Robotic Manipulation and Motion Planning*. PhD thesis, Duke University, 2002.

[8] Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, Berlin, 1997.

[9] V. Guillemin and A. Pollack. *Differential Topology*. Prentice-Hall, 1974.

[10] D. Halperin, L. E. Kavraki, and J.-C. Latombe. Robotics. In Jacob E. Goodman and Joseph O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, chapter 41, pages 755–778. CRC Press LLC, Boca Raton, FL, 1997.

[11] D. Hsu, L. Kavraki, J.-C. Latombe, R. Motwani, and S. Sorkin. On finding narrow passages with probabilistic roadmap planners. In *3rd Workshop on the Algorithmic Foundations of Robotics*, 1998.

[12] D. Hsu, J.-C. Latombe, and R. Motwani. Path planning in expansive configuration spaces. *Int. J. of Comp. Geom. and Apps*, 9(4 & 5):495–512, 1999.

[13] L. Kavraki, M.N. Kolountzakis, and J.-C. Latombe. Analysis of probabilistic roadmaps for path planning. *IEEE Trans. Rob. Aut.*, 14(1):166–171, 1998.

[14] L. Kavraki, J.-C. Latombe, R. Motwani, and P. Raghavan. Randomized query processing in robot motion planning. In *Proc. 27th ACM Symp. on Theory of Computing*, pages 353–362, Las Vegas, NV, 1995.

[15] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars. Probabilistic roadmaps for path planning in high dimensional configuration spaces. *IEEE Trans. on Robotics and Automation*, 12(4):566–580, 1996.

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

[17] J. Matousek. More on cutting arrangements and spanning trees with low stabbing number. Technical Report B-90-2, Freie Universitat Berlin, 1990.

[18] J.H. Reif. Complexity of the mover's problem and generalizations. In *Proc. of the 20th IEEE Symp. on Foundations of Comp. Sci.*, pages 421–427, 1979.

[19] M. Sharir and P.K. Agarwal. *Davenport-Schinzel Sequences and Their Geometric Applications*. Cambridge University Press, 1995.