1 CAPTURING THE CONNECTIVITY OF HIGH-DIMENSIONAL GEOMETRIC SPACES BY PARALLELIZABLE RANDOM SAMPLING TECHNIQUES

David Hsu*, Lydia E. Kavraki†,
Jean-Claude Latombe*, and Rajeev Motwani*

Abstract: Applications such as robot programming, design for manufacturing, animation of digital actors, rationale drug design, and surgical planning, require computing paths in high-dimensional geometric spaces, a provably hard problem. Recently, a general path-planning approach based on a parallelizable random sampling scheme has emerged as an effective approach to solve this problem. In this approach, the path planner captures the connectivity of a space F by building a *probabilistic roadmap*, a network of simple paths connecting points picked at random in F. This paper combines results previously presented in separate papers. It describes a basic probabilistic roadmap planner that is easily parallelizable, and it analyzes the performance of this planner as a function of how well F satisfies geometric properties called ϵ-goodness, expansiveness, and path clearance. While ϵ-goodness allows us to study how well a probabilistic roadmap covers F, expansiveness and path clearance allow us to compare the connectivity of the roadmap to that of F.

*Computer Science Department, Stanford University, Stanford, CA 94305, USA.
†Computer Science Department, Rice University, Houston, TX 77005, USA.

1.1 INTRODUCTION

The path planning problem can be formulated as follows:

> Given:
>
> - A geometric and kinematic model of a rigid or articulated object, called the *robot*,
> - A geometric model of the *obstacles* in the physical space where the robot operates,
>
> Find a *path*, *i.e.*, a continuous sequence of collision-free *configurations* (*i.e.*, physical placements) of the robot, connecting two arbitrary input configurations q_b and q_e, called the *query* configurations, whenever such a path exists; otherwise indicate that no such path exists.

In this definition, a "robot" can be almost any type of moving object, *e.g.*, a robot manipulator, a digital actor, a surgical instrument, or a flexible molecule. Similarly, an "obstacle" can be any sort of forbidden region, *e.g.*, a hard object that a robot arm should not hit, a muddy terrain that a mobile robot should not traverse, or a critical tissue structure that a surgical instrument should not cut.

A classical way to look at the above path-planning problem is to represent a robot's configuration by n independent parameters, one for each of the robot's degrees of freedom (dofs). Thus, each configuration is a point in an n-D parameter space $\mathcal{C}$, called the robot's *configuration space*. The obstacles map into $\mathcal{C}$ as regions whose complement F is referred to as the *free space*. A solution path is a continuous curve segment in F connecting the two query configurations q_b and q_e. Such a path exists if and only if q_b and q_e lie in the same component of F.

Path planning is a provably hard computational problem [31]. There is strong evidence that solving it requires exponential time in the dimension n of $\mathcal{C}$. This result still holds for specific robots such as planar linkages consisting of links serially connected by revolute joints [16] and sets of rectangles executing axis-parallel translations in a rectangular workspace [10, 11]. Though general and complete algorithms have been proposed [6, 33], their high complexity precludes any useful application.

The dimension of $\mathcal{C}$ beyond which existing complete algorithms become practically useless is low, somewhere between 3 and 5. This means that they cannot be applied to rigid objects translating and rotating in 3-D workspaces, nor to 6-dof robot manipulators, two important cases in practice. On the other hand, applications tend to involve more degrees of freedom than ever before. For example, manufacturing workcells (*e.g.*, welding workcells for assembling car bodies) involving several robot manipulators having more than 30 dofs in total are no longer exceptions; by automatically computing collision-free trajectories, a path planner would drastically reduce the cost of programming the manipulators. In computer graphics, animation of synthetic actors to produce digital movies or video games requires dealing with several dozen dofs; using a path

planner may drastically reduce the number of input key frames. In molecular biology, a path planner could compute plausible docking motions of candidate drug molecules modeled as spatial linkages with many torsional dofs and help extracting promising leads from large databases of previously synthesized molecules.

Recently, a general path-planning approach based on a parallelizable random sampling scheme has emerged as a practical approach to compute paths in high-dimensional configuration spaces. Because computing an explicit representation of F is prohibitive, this approach only uses the implicit representation of F that is provided by a function `dist`(q), which computes the distance between the robot at configuration q and the obstacles in the Euclidean space $\mathbb{R}^2$ or $\mathbb{R}^3$. This function admits several reasonably efficient implementations (*e.g.*, [8, 9, 15, 22, 23, 24, 27, 30]). The planner samples $\mathcal{C}$ at random. Using `dist`, it retains the configurations in free space as *milestones* and, for every pair of milestones, it checks that a simple path between them (usually, the straight line segment in $\mathcal{C}$) is collision-free. The result is a graph R called a *probabilistic roadmap*. Given any pair of query configurations, the planner tries to connect each query configuration to a milestone of R. It outputs a path if it connects the two configurations to milestones in the same connected component of R. A number of probabilistic roadmap planners (PRMs) have been proposed [3, 4, 5, 14, 12, 17, 20, 29, 32]. None of them is complete in the strongest sense, but most achieve some form of probabilistic completeness, *i.e.*, if a path exists, the planner will find one with high probability after some amount of computation. In fact, PRMs have been remarkably successful in solving unusually difficult path-planning problems in high-dimensional configuration spaces. Moreover, the algorithms are easily parallelizable, though to our knowledge this feature has not yet been significantly exploited.

Section 1.2 describes a "basic" PRM that performs a uniform random sampling of the configuration space. Section 1.3 surveys more sophisticated sampling strategies proposed in the literature. A "good" probabilistic roadmap is one which provides adequate coverage of the free space, so that every query configuration can easily be connected to it, and whose connectivity conforms to that of the free space. In Sections 1.4 and 1.5 we formally analyze how large a roadmap needs to be (*i.e.*, how many milestones it should contain) in order to achieve adequate coverage and connectivity with high probability. This analysis shows that the probabilistic roadmap approach is efficient if the free space satisfies geometric properties called *ϵ-goodness* [19], *expansiveness* [14], and *path clearance* [17, 18]. Under these assumptions, the free space does not contain "narrow passages". Dealing with such passages is the main remaining issue in designing PRMs. In Section 1.6 we will briefly present ongoing efforts addressing this issue.

This paper does not report on experimental results with implemented PRMs. Such results have been presented in several previous papers, with robots having from three to several dozen dofs. Complex practical applications of these planners include maintenance planning for aircraft engines [7], design for man-

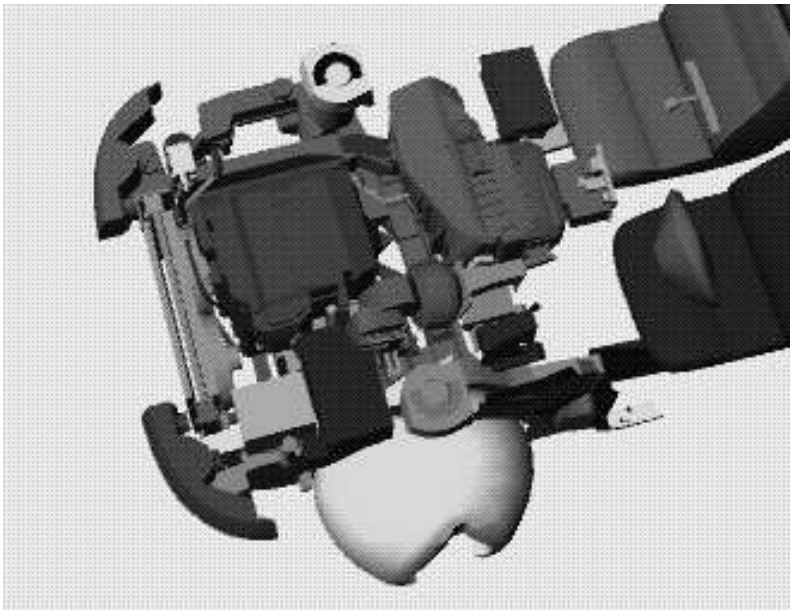

Figure 1.1 Car packaging model with 60,000 triangles [14]

ufacturing in the automotive industry [14], and graphic animation of human characters [21]. Figure 1.1 displays the CAD model of a car packaging; the path planner in [14] determines whether a subassembly designated by the user can be extracted without removing other parts. Figure 1.2 shows several snapshots of a path computed by the planner in [21] for a digital actor; only the configurations in the first and last snapshots where given to the planner.

1.2 BASIC PROBABILISTIC ROADMAP PLANNER

For simplification, we assume that the configuration space $\mathcal{C}$ is the Euclidean hyper-cube $[0,1]^n$. We say that two free configurations *see* each other if they can be connected by a straight-line path in the free space F.

The basic PRM is a simplified version of the planner described in [20]. It consists of two algorithms: `roadmap`, which precomputes a probabilistic roadmap, and `query`, which uses this roadmap to answer path-planning queries. Each query is defined by two configurations, q_b and q_e, in F.

1.2.1 Roadmap Construction

The procedure `roadmap` constructs a roadmap in two steps, as follows:

Procedure `roadmap`**:**

1. Pick s configurations uniformly at random in F. Call them *milestones* and let M be the set of milestones.

Figure 1.2 Animation of a digital actor [21]

2. Construct the graph $R = (M, L)$ in which L consists of every pair of milestones that see each other. Call R the *roadmap*.

The milestones are chosen at Step 1. The links between milestones are created at Step 2. Recall that `dist`(q) is a procedure that computes the Euclidean distance between the robot placed at q and the obstacles. Step 1 generates each milestone by picking successive configurations q in $[0,1]^n$, until one satisfies `dist`$(q) > 0$. Every q is obtained by choosing each of its coordinates uniformly at random in $[0,1]$. Step 2 checks the straight path between every two milestones for collision, by recursively decomposing it into two half segments and invoking `dist` at each segment endpoint. One can show that if a segment is short enough relative to the robot-obstacle distance computed at its two endpoints, the whole segment is guaranteed to be collision-free [4].

1.2.2 Query Processing

The query-processing algorithm is the following:

Procedure `query`**:**

1. For $i = \{b, e\}$ do:
   (a) If there exists a milestone m that sees q_i then $m_i \leftarrow m$,
   (b) Else
      i. Repeat t times
      pick a configuration q in F uniformly at random in a neighborhood of q_i
      until q sees both q_i and a milestone m.
      ii. If all t trials failed then return FAILURE, else $m_i \leftarrow m$.
2. If m_b and m_e are in the same connected component of the roadmap, then return a path connecting them, else return NO-PATH.

The procedure tries to connect each of the query configurations to a milestone of the roadmap, either directly (Step 1(a)), or through an intermediate configuration chosen in a neighborhood of the query configuration (Step 1(b)). The implementation of Step 1 makes use of the function `dist`. Each free configuration q at Step 1(b)i is obtained by picking successive configurations at random in a hyper-cube centered at q_i until one is collision-free. The `query` procedure returns NO-PATH if it connects the query configurations to two distinct components of the roadmap. This answer is correct whenever no two components of the roadmap lie in the same component of F. Moreover, the procedure outputs FAILURE if it cannot connect a query configuration to some milestone of the roadmap. Obviously, we would like the planner to rarely return FAILURE or an incorrect answer.

1.2.3 Parallelization

Constructing a good roadmap can be time consuming; but, when this is done, processing path-planning queries is very fast. Fortunately, the `roadmap` proce-

dure is amenable to parallelism. In fact, we expect that it will scale well both in shared memory and message passing architectures.

In a shared memory architecture with k processors, each processor can independently generate s/k milestones. Connections among these milestones can also be computed independently. Since we can roughly estimate the workload of each processor by counting the number of connections that are attempted by the processor, we can decide the assignment of tasks to processors off-line. A better load balancing scheme may be obtained by maintaining one or more queues with the connections that need to be checked and having the processors request tasks from that queue(s). Note that the selection and connections of milestones requires no interprocessor communication. However, some interprocessor communication is needed to compute the connected components of the resulting roadmap. The latter problem is well studied in the parallel processing literature and a discussion can be found in [1]. We simply note here that the computation of the roadmap components is far less expensive than the computation of the connections among the milestones and cannot significantly affect the overall scaling of our approach.

As far as message passing architectures are concerned, several schemes are possible. For example, each processor can generate s/k milestones and interconnect them locally. To minimize interprocessor communication, processors can periodically exchange a percentage of their nodes (milestones). When a processor learns about new nodes, it adds them to its roadmap, and attempts connections with the already existing nodes. The amount of data that needs to be communicated in this way is small compared to the size of the roadmaps built by the processors. Furthermore, our profiling in the uniprocessor case has shown that the computation of connections between milestones dominates running time. We thus expect that the time spent on communication among the processors will be a very small percentage of the overall running time. By communicating data during roadmap construction, we increase the probability that when the individual roadmaps are merged, the connectivity of the resulting roadmap will be the same as if the roadmap were generated by a single-processor machine.

1.3 OTHER SAMPLING STRATEGIES

The sampling strategy embedded in `roadmap` is extremely simple. As one would expect, more sophisticated sampling strategies have been proposed in the literature. The goal of most of them is to capture F's connectivity with the smallest possible roadmap, by performing a non-uniform sampling of the free space. Some strategies also attempt to avoid the relatively expensive precomputation of a roadmap, which is particularly desirable when only a few path-planning queries are to be made in the same free space. We review some of these strategies below.

Several papers describe PRMs that construct a roadmap in two stages. The first stage (similar to the above `roadmap` procedure) attempts to capture the connectivity of a free space F using no a priori knowledge, while the second

stage adds milestones in subsets of F chosen on the basis of the information revealed by the first stage. In [17] the second stage consists of adding new milestones in the neighborhoods of milestones that have been picked in the first stage and that see no or few other milestones. Several similar heuristics are discussed in [20]. Milestones poorly connected to other milestones at the end of the first stage tend to lie in "difficult" regions of the free space, and experiments reported in [17, 20] show that adding a resampling stage to the strategy is very effective at improving roadmap connectedness. In [13] the first stage computes a roadmap R' in a space F' obtained by dilating F. The second stage "pushes" the links of R' that do not fully lie in F into F, by adding new milestones around those links. By first widening the narrow passages in F, this strategy allows the planner to find paths through such passages more efficiently than a single-step strategy. Another multi-stage strategy is proposed in [2] which allows milestones to be connected by multiple types of paths.

"Difficult" regions of the free space tend to lie close to F's boundary. For that reason, a number of papers propose sampling strategies that are directly aimed at generating a greater density of milestones near the boundary of the free space. In [3], when a configuration q is generated outside F, a number of rays are shot from q along random directions uniformly distributed in $\mathcal{C}$. For each ray, a binary search is used to identify a point near the boundary of F. In [28] a single ray is shot from q along a random direction; the procedure then simulates a walk of the robot along this direction, until it is in free space. In [12] new milestones are created near F's boundary to connect roadmap components that could not be connected by straight paths. A ray is shot from a milestone in one component along a direction picked at random. Using a technique similar to [28], a milestone is created where this ray encounters the free space boundary, and the ray is reflected in a random direction at this point to find another boundary point. All three references listed above observe that adding milestones near the free space boundary improves the planners' performance. The two-step strategy in [13] also produces milestone distributions that are denser near the free space boundary.

Precomputing a roadmap is advantageous when multiple path-planning queries are made in the same free space. To deal with single-query cases, some planners build a new roadmap for each new query. Knowing the query configurations q_b and q_e allows these planners to only sample the connected components of F that contain these configurations. The planner in [14] generates clouds of milestones picked at random in small neighborhoods of previously generated milestones, expanding from the two query configurations, until two clouds meet (bi-directional search). The planner in [5] uses a partially random sampling uni-directional strategy. Starting at q_b, it deterministically follows the steepest descent of a heuristic potential field U defined over $\mathcal{C}$ until it reaches a minimum of the potential. If this minimum is the goal (U is defined such that it has a global minimum at q_e), the planner stops; otherwise it tries to escape the basin of attraction of the local minimum by performing a series of random walks. Using a heuristic potential field allows the planner to solve some

very difficult problems efficiently. But, to be practically useful, the function U must be fast to compute; all known such functions U have pathological cases in which the planner behaves poorly.

The following two sections give a formal analysis of the performance of the procedures `roadmap` and `query`. A similar analysis for more complex strategies, such as those surveyed above, would be considerably more difficult. However, a noticeable attempt has been made in [19] to formally explain the improvements obtained with the two-stage strategies proposed in [17, 20].

1.4 ROADMAP COVERAGE

A first desirable property for a probabilistic roadmap is that it provides adequate coverage of the free space F. This means that the milestones should collectively see a large portion of F, so that any query configuration can easily be connected to one of them. Note that Step 1(b) of `query` allows for the case where a query configuration does not see any milestone. Indeed, it would be unrealistic to expect that a probabilistic roadmap provides complete coverage of F; in general, the probability of picking a new milestone that sees a portion of F not seen by previous milestones decreases and tends toward zero as the number of milestones grows.

In this section we establish that the milestones chosen by `roadmap` see a large portion of F with high probability if every point in F sees a significant portion of F (a property that we call ϵ-goodness). We also state that when the roadmap achieves adequate coverage of the free space, `query` efficiently connects query configurations to the roadmap.

For any subset $S \subseteq \mathcal{C}$, we let $\mu(S)$ denote its volume. For any $q \in F$, $\mathcal{V}(q)$ denotes the set of all free configurations seen by q; we call it the *visibility set* of q.

Definition 1. *Let ϵ be a constant in $(0, 1]$. A free configuration q is* ϵ-good *if $\mu(\mathcal{V}(q)) \geq \epsilon\mu(F)$. The free space is* ϵ-good *if every $q \in F$ is ϵ-good.*

There exist spaces that are not ϵ-good. Consider, for example, the simple 2-D free space shown in Figure 1.3. Every point q in this free space is ϵ-good for some $\epsilon \in (0, 1]$ that depends on q. However, as q tends toward the tangency point T, $\epsilon \to 0$. Hence, F is not ϵ-good. But any subset of F obtained by removing a small neighborhood of T is ϵ-good. More generally, the non-ϵ-goodness of F may be caused by the fact that two hyper-surfaces bounding F are tangent. In most cases, one can render the free space ϵ-good by removing small subsets of it.

Definition 2. *A set of milestones provides* adequate coverage *of an ϵ-good free space F if the volume of the subset of F not visible from any of these milestones is at most $(\epsilon/2)\mu(F)$.*

Note that, as ϵ increases, the coverage requirement grows weaker, *i.e.*, the portion of F that has to be visible by at least one milestone gets smaller. This

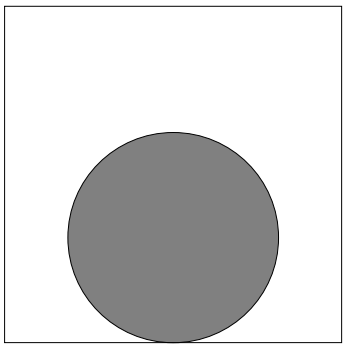

Figure 1.3 A free space that is not ϵ-good

comes from the fact that a greater ϵ will make it easier for `query` to connect query configurations to the roadmap. Naturally, the number of milestones needed becomes smaller.

Theorem 1. *Assume that F is ϵ-good. Let ϕ be a constant in $(0,1]$ and K be a positive real large enough that for any $x \in (0,1]$, $(1-x)^{(K/x)\log(2/x\phi)} \leq x\phi/2$. If s is chosen such that:*

$$s \geq \frac{K}{\epsilon} \log \frac{2}{\epsilon\phi},$$

then `roadmap` *generates a set of milestones that adequately covers F, with probability at least $1-\phi$.*

The proof of this theorem was established in [4, 19]. We reproduce it in Appendix A.

Theorem 1 does not allow us to compute s since we do not know the value of ϵ, except for simple spaces. Nevertheless, its significance is twofold. First, it tells us that although adequate coverage of the free space is not guaranteed, the probability that the milestones picked by `roadmap` does not cover F adequately decreases exponentially with the number of milestones. Second, the number of milestones needed increases moderately when ϵ decreases. Though it does not account for all aspects of the computation of a roadmap, s characterizes well the amount of work done by `roadmap`.

It now remains to establish that adequate coverage allows `query` to connect any query configuration to the roadmap, with high probability.

Theorem 2. *Let the maximum number of iterations t at Step 1(b)i of* `query` *be set to $\log(2/\psi)$, where ψ is a constant in $(0,1]$. If the milestones adequately cover F, then the probability that* `query` *outputs* FAILURE *is at most ψ. The expected number of iterations is at most 2.*

In other words, the failure probability of `query` decreases exponentially with the number t of iterations at Step 1(b)i. The proof of this theorem was es-

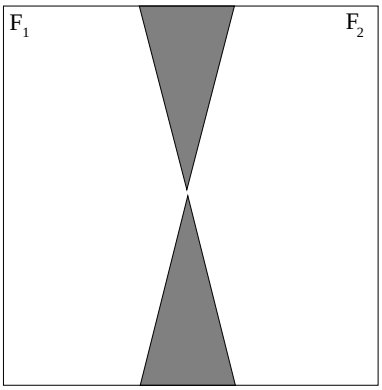

Figure 1.4 A narrow passage in an ϵ-good space

tablished in [19] and is reproduced in Appendix B. This proof assumes that Step 1(b)i of `query` samples the visibility set $\mathcal{V}(q_i)$ of q_i to find a configuration q that sees both q_i and a milestone m. Since $\mathcal{V}(q_i)$ is unknown, any implementation of `query` is only an approximation of the algorithm to which Theorem 2 applies strictly.

The concept of ϵ-goodness has been extended in [19] by including multi-link paths in the definition of the mutual visibility of two milestones. This extension makes it possible to formally explain the improvements empirically observed with the two-stage sampling strategies of [17, 20]. However, ϵ-goodness and its extension are still too weak to guarantee that `roadmap` will construct a roadmap whose connectivity represents that of the free space. For example, the free space of Figure 1.4 is ϵ-good for $\epsilon \approx 0.5$. But a roadmap of moderate size constructed by `roadmap` will most likely consist of two connected components. In [19] we dealt with this issue by allowing `roadmap` to invoke a complete planner to try to connect the components of a roadmap. However, running such a planner can be totally impractical. In [14] we eliminated its need by introducing the notion of an expansive free space.

1.5 ROADMAP CONNECTEDNESS

1.5.1 Expansive Free Space

Let us now define precisely the kind of roadmap we would like `roadmap` to construct.

Definition 3. *Let F be an ϵ-good free space. A roadmap R is an* adequate representation *of F if its milestones provide adequate coverage of F and no two components of R lie in the same component of F.*

Let R be an adequate representation of F. Since F is ϵ-good, no component of F has volume less than $\epsilon\mu(F)$. Therefore, at least one milestone of R lies in

every component of F. Since no two components of R lie in the same component of F, there is a one-to-one correspondence between the components of R and those of F.

The notion of an expansive free space is directly related to the difficulty that `roadmap` has to connect milestones through narrow passages. The reason why it would require considerable time for this procedure to build a connected roadmap in the free space of Figure 1.4 is that a very small subset of points in F_1 (the half space on the left) see a large fraction of F_2 (the half space on the right); therefore, the probability that the planner picks a milestone in F_1 that sees a milestone in F_2 is small. By narrowing the passage between F_1 and F_2, one can make this probability arbitrarily small. Let us refer to the subset of points in a subset $S \subset F$ that can see a large portion of $F \backslash S$ as the lookout of S. If it is large enough, it is easy to connect any point in S to points outside S by picking points at random in S and $F \backslash S$.

Definition 4. *Let β be a constant in $(0, 1]$ and S be any subset of any connected component E of the free space F. The* β-lookout *of S is the set:*

$$\beta\text{-LOOKOUT}(S) = \{q \in S | \mu(\mathcal{V}(q) \backslash S) \geq \beta \mu(E \backslash S)\}.$$

Definition 5. *Let ϵ, α, and β be constants in $(0, 1]$. The free space F is* $(\epsilon, \alpha, \beta)$-expansive *if it is ϵ-good and, for every connected subset S that is a finite union of visibility sets, we have:*

$$\mu(\beta\text{-LOOKOUT}(S)) \geq \alpha \mu(S).$$

For simplification, we will abbreviate the term "$(\epsilon, \alpha, \beta)$-expansive" by "expansive".

The following comments may help apprehend the notion of expansiveness:

- In an expansive space, ϵ is uniquely defined, while α and β may take several values. For example, in Figure 1.4, the lookout of F_1 depends on the value of β. In general, α can only decrease as β increases. A natural choice for β is the one that yields the smallest bound on the number of milestones given by Theorem 3 stated below.

- There is no general relation between the order of magnitude of ϵ and that of α and β. For instance, ϵ is large in the free space of Figure 1.4, but α or β is small. Instead, in Figure 1.5, ϵ is small, but α and β are large. To see this, pick a point p in the narrow corridor (bottom left of the free space). The volume of $\mathcal{V}(p)$ is small relative to that of F, hence ϵ is small. But a large fraction of $\mathcal{V}(p)$ sees a large portion of F, hence α and β are both large.

Theorem 3. *Assume that F is $(\epsilon, \alpha, \beta)$-expansive. Let ξ be a constant in $(0, 1]$. If s is chosen such that:*

$$s \geq \frac{16}{\epsilon \alpha} \log \frac{8}{\epsilon \alpha \xi} + \frac{6}{\beta} + 2,$$

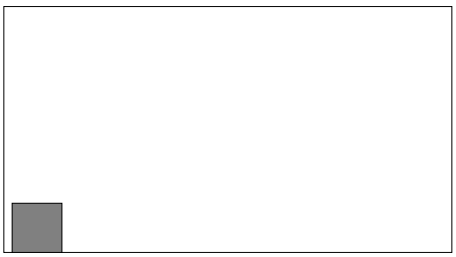

Figure 1.5 A free space with small ϵ and large α and β

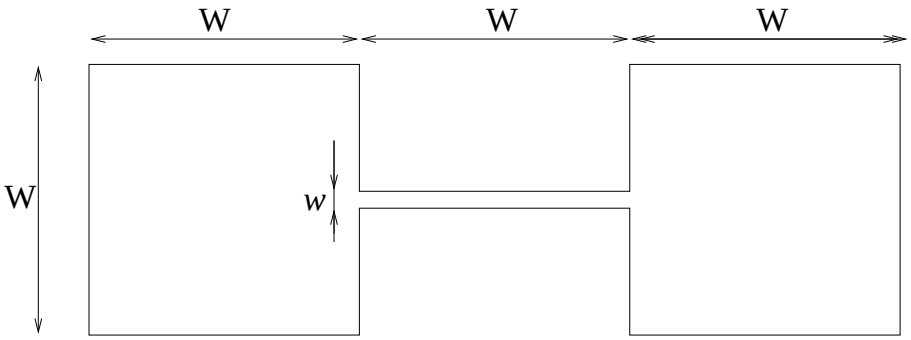

Figure 1.6 An $(\epsilon, \alpha, \beta)$-expansive free space where $\epsilon, \alpha, \beta \sim w/W$

then with probability at least $1 - \xi$, `roadmap` *generates a roadmap such that no two of its components lie in the same component of* F.

The proof of this Theorem was established in [14] and is reproduced in Appendix C.

Theorems 1 and 3 combined imply that with high probability, `roadmap` generates a roadmap that adequately represents F. Theorem 3 tells us that the probability that a roadmap does not capture F's connectivity decreases exponentially with the number of milestones, and that the number of milestones needed increases moderately when ϵ, α, and β decrease. One could be tempted to use a Monte Carlo technique to estimate the values of ϵ, α, and β in a given free space, and hence obtain an estimate of the number of milestones needed to get a roadmap that adequately represents F. But it seems that a reliable estimation would take at least as much time as building the roadmap itself.

Remark: Note that none of Theorems 1, 2, or 3 explicitly mention the dimension n of the configuration space. This comes from the fact that both ϵ-goodness and expansiveness are visibility properties whose definitions only refer to volumes of subsets of F. But the dependence on n may be hidden in the parameters ϵ, α, and β. To illustrate this point, consider the example of Figure 1.6. The free space consists of two squares whose sides have length W;

these cubes are connected by a rectangular narrow passage of length W and width w, where $w \ll W$. Up to a constant factor, each of the parameters ϵ, α, and β is on the order of w/W. Indeed, the volume of F is $W(2W + w) \approx 2W^2$. The points with the smallest ϵ-goodness are located in the narrow passage. Each such point sees only a subset of F of volume approximately $3wW$; hence, $\epsilon \approx 3w/2W \sim w/W$. A point near the top right corner of the left square sees this entire square; but only a subset of this square, of approximate volume wW, contains points that, each, see a set of volume $2wW$; hence, $\alpha \approx w/2W \sim w/W$ and $\beta \approx w/W$. In the n-D version of this example, two hyper-cubes, each having volume W^n, are connected by a hyper-parallelepipedic passage that has size w along k dimensions ($k \in [1, n-1]$) and size W along the $n-k$ other dimensions. Each of the parameters ϵ, α, and β is on the order of $(w/W)^k$. The worst case happens when $k = n-1$, that is, when the passage is narrow along $n-1$ dimensions.

1.5.2 Path Clearance Assumption

Another analysis of the basic PRM can be done by explicitly considering the "width" of the passages in F [17, 18]. Let q and q' be two configurations in the same component of F and τ be a free path connecting them. Let ℓ be the Euclidean length of τ and σ be its distance to F's boundary. We call σ the *clearance* of the path.

Theorem 4. *Let ζ be a constant in $(0, 1]$ and a be the constant $2^{-n}\mu(\mathcal{B}_1)/\mu(F)$ where $\mathcal{B}_1$ denotes the unit ball in $\mathbb{R}^n$. If s is chosen such that:*

$$s \geq \frac{1}{a\sigma^n} \log \frac{2\ell}{\sigma\zeta},$$

then with probability at least $1-\zeta$ `roadmap` *generates a roadmap in which one component contains two milestones m and m' such that q sees m and q' sees m'.*

The proof of this theorem was given in [18] and is reproduced in Appendix C. Consistently with Theorem 3, it says that the probability that a roadmap fails to provide a path through a narrow passage decreases exponentially with the number of milestones. It also rightly suggests that the number of milestones may increase as $(2/\sigma)^n$. However, it is more conservative than Theorem 3. For instance, in the n-D version of the example shown in Figure 1.6, $\sigma = w/2$, even if the passage is narrow along a single dimension. While Theorem 4 suggests that the number of milestones increases as w^{-n}, Theorem 3 tells us that it only increases as w^{-k}, where $k \in [1, n-1]$ is the number of dimensions along which the passage is narrow. As another instance, consider the case where F is punctured by many holes uniformly distributed in a region of $\mathcal{C}$ (Figure 1.7). By increasing the number of holes and reducing their size, one can create many narrow passages such that ϵ, α, and β remain large while the clearance σ of any path connecting the left side of F to its right side becomes arbitrarily small. Theorem 3 then tells us that the basic PRM can easily deal with such passages.

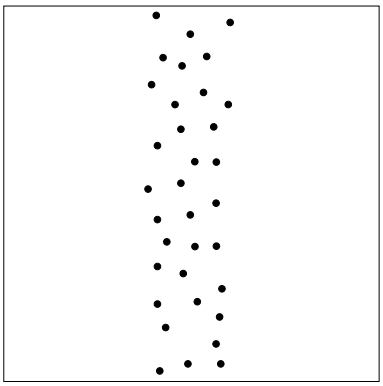

Figure 1.7 A space with small clearance, but large ϵ, α, and β

Note that a variant of the path-clearance assumption that may yield a slightly tighter bound than Theorem 4 is the σ-complexity assumption proposed in [32].

On the other hand, Theorem 4 may be more easier to exploit than Theorem 3. For example, in many robotics applications, only those paths whose clearance is greater than some predefined threshold can be reliably executed. This threshold can then be used to effectively estimate the size s of the roadmap.

1.6 CURRENT AND FUTURE WORK

During the last few years a number of path planning algorithms based on the construction of probabilistic roadmaps have been proposed and experimented with great success. This paper has described a basic probabilistic roadmap planner and has provided a formal analysis that explains its empirical success.

However, current probabilistic roadmap planners share the same relative inability to efficiently find paths through narrow passages. This inability has been experimentally observed, and it is formally explained by our results in expansive free spaces. Current research aims at dealing efficiently with free spaces that are poorly expansive.

The regions of F responsible for its poor ϵ-goodness and expansiveness necessary lie near the boundary of F. Consistently with this observation, several sampling strategies have been proposed which generate a greater density of milestones near the boundary of the free space, and these strategies have yielded empirically observed improvements. However, they are not sufficient to deal with narrow passages. The main reason is that they tend to increase the density of milestones everywhere near F's boundary, a space that still has high dimension $(n-1)$. Hence, they do not significantly increase the odds of placing milestones in narrow passages, which are the regions where the milestone distribution needs to be the densest.

The insights provided by the analysis of `roadmap` presented in this paper have recently led us to design a new two-stage strategy [13]. The first stage

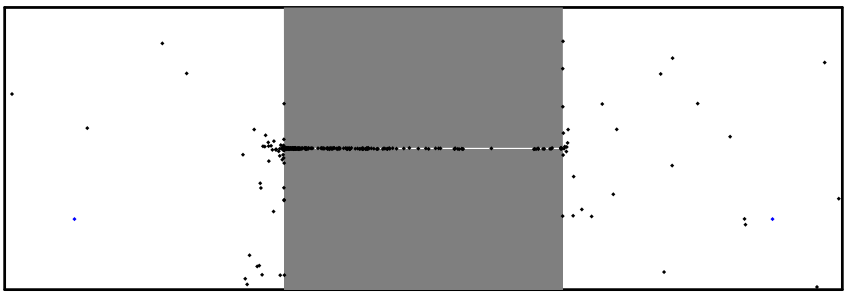

Figure 1.8 Milestone distribution obtained with a dilatation strategy

generates a roadmap R' in a dilated free space F' obtained by allowing some penetration distance of the robot into the obstacles [8, 24, 26, 27]. Dilating free space tend to increase its expansiveness, with the narrow channels benefiting much more than the already wide areas of the free space; hence, capturing the connectivity of R' is relatively easy. The second stage of the strategy "pushes" R' into the original free space F by picking additional milestones around the links of R' that do not lie entirely in F; hence, R' is used as a guide in the second stage to decide where to increase the density of the milestone distribution. The result is a roadmap R in F whose density increases near the boundary of F and becomes maximal inside narrow passages. For example, Figure 1.8 shows a milestone distribution obtained with this strategy in the simple 2-D free space of Figure 1.6, with $w/W = .001$. The efficiency of the strategy still needs to be empirically confirmed with experiments on realistic path-planning examples, *i.e.*, in high-dimensional and geometrically complex free spaces.

The simple sampling strategy of the procedure `roadmap` is easily parallelizable and we expect that it will scale well both on shared memory and message passing architectures. As more sophisticated sampling strategies are proposed to handle free spaces that are $(\epsilon, \alpha, \beta)$-expansive for small values of ϵ, α, and β, efficient parallelization will become less straightforward. However, as the computation of connections between milestones will most likely continue to dominate running time, we also expect that these strategies will scale well on parallel architectures. So far, parallelization has not been significantly exploited in PRMs. However, as more demanding applications of path-planning emerge, *e.g.*, in graphic animation, it is likely to become a key aspect of future PRMs.

Acknowledgments

This research was supported in part by ARO MURI grant DAAH04-96-1-007. D. Hsu is currently supported by a Microsoft Fellowship. L. Kavraki is partially supported by NSF CAREER Award IRI-970228. We thank GM R&D for providing us the model of Figure 1.1. We also thank James Kuffner and Stephen Sorkin for providing us Figures 1.2 and 1.8, respectively.

Appendix: A. Proof of Theorem 1

Theorem 1. *Assume that F is ϵ-good. Let ϕ be a constant in $(0,1]$ and K be a positive real large enough that for any $x \in (0,1]$, $(1-x)^{(K/x)\log(2/x\phi)} \leq x\phi/2$. If s is chosen such that:*

$$s \geq \frac{K}{\epsilon} \log \frac{2}{\epsilon\phi},$$

then `roadmap` *generates a set of milestones that adequately covers F, with probability at least $1-\phi$.*

Proof: Let M denote the set of the s milestones picked at Step 1 of `roadmap`. The volume H of points in F not visible from any of these milestones is:

$$H = \mu(\{q \in F | q \notin \cup_{m \in M} \mathcal{V}(m)\}).$$

Its expected value is:

$$E[H] = \int_{q \in F} \Pr[q \notin \cup_{m \in M} \mathcal{V}(m)] dq.$$

The ϵ-goodness of F entails that the probability that any given configuration is not visible from any of the s milestones is at most $(1-\epsilon)^s$. Thus:

$$E[H] \leq \mu(F)(1-\epsilon)^s \leq \mu(F)\epsilon\phi/2. \tag{1.A.1}$$

Given a random variable X assuming only non-negative values, the Markov inequality [25]:

$$\Pr[X \geq x] \leq E[X]/x$$

holds for all $x \in \mathbb{R}^+$. Using this inequality and the relation (1.A.1), we get:

$$\Pr[H \geq (\epsilon/2)\mu(F)] \leq \phi.$$

Hence, with probability $1-\phi$, H is at most $(\epsilon/2)\mu(F)$, in which case M provides adequate coverage of F. ∎

Appendix: B. Proof of Theorem 2

Theorem 2. *Let the maximum number of iterations t at Step 1(b)i of* `query` *be set to $\log(2/\psi)$, where ψ is a constant in $(0,1]$. If the milestones adequately cover F, then the probability that* `query` *outputs* FAILURE *is at most ψ. The expected number of iterations is at most 2.*

Proof: For any $q \in F$, the volume of the subset of $\mathcal{V}(q)$ visible from some milestone is at least:

$$\mu(\mathcal{V}(q)) - (\epsilon/2)\mu(F) \geq (\epsilon/2)\mu(F).$$

Therefore, for either query configuration q_i ($i \in \{b, e\}$), the probability that a random configuration chosen from $\mathcal{V}(q_i)$ is not visible from any milestone is at most $1/2$. The probability that `query` fails to connect q_i to a milestone on $\log(2/\psi)$ trials at Step 1(b)i is thus less than $\psi/2$. Since Step1(b)i is performed for both query configurations, the overall failure probability is at most ψ. Moreover, the expected number of executions of Step 1(b)i is at most 2. ∎

Appendix: C. Proof of Theorem 3

Let F be an $(\epsilon, \alpha, \beta)$-expansive free space. We begin by defining the linking sequence of a point $p \in F$:

Definition 6. *The* linking sequence *of a point $p \in F$ is a sequence of points $p_0 = p, p_1, p_2, \ldots$ and a sequence of sets $V_0 = \mathcal{V}(p_0), V_1, V_2, \ldots \subseteq F$ such that for all $i \geq 1$, $p_i \in \beta$-LOOKOUT(V_{i-1}) and $V_i = V_{i-1} \cup \mathcal{V}(p_i)$.*

Since the sets $V_0, V_1, V_2, \ldots$ are completely determined by the sequence $p_0, p_1, p_2, \ldots$, we will refer to just the sequence $p_0, p_1, p_2, \ldots$ as a linking sequence for p.

We now establish two lemmas. Lemma 1 says that any set of milestones M produced by `roadmap` is likely to contain a linking sequence of a given length for any milestone in M. Lemma 2 says that the sets associated with a linking sequence of this length span a large volume. The consequence is that the final sets determined by long-enough linking sequences for any two milestones p and q must intersect, since their volumes are large enough. In that case p and q will be connected by a path.

Lemma 1. *Let $w = 1/\alpha\epsilon$. Given any milestone $p \in M$, there exists a linking sequence in M of length t for p with probability at least $1 - we^{-(s-t-1)/w}$.*

Proof: Without loss of generality, let us assume that $\mu(F) = 1$. Let L_i be the event that there exists a linking sequence in M of length i for $p_0 = p$ and $\overline{L_i}$ be the event that there does not exist such a sequence.

$$\begin{aligned} \Pr(\overline{L_i}) &= \Pr(\overline{L_i} \mid \overline{L}_{i-1}) \Pr(\overline{L_{i-1}}) + \Pr(\overline{L_i} \mid L_{i-1}) \Pr(L_{i-1}) \\ &\leq \Pr(\overline{L_{i-1}}) + \Pr(\overline{L_i} \mid L_{i-1}). \end{aligned}$$

We would like to estimate $\Pr(\overline{L_i} \mid L_{i-1})$. That is, given that there exist a linking sequence $p_0 = p, p_1, p_2, \ldots, p_{i-1} \in M$, what is the probability that M contains no linking sequence of length i for p? All we need is that M contains no point lying in β-LOOKOUT(V_{i-1}). Note that $p_0, p_1, p_2, \ldots, p_{i-1}$ are conditioned and we cannot expect them to lie in β-LOOKOUT(V_{i-1}). However, the remaining $s - i$ points in M are unconditioned and chosen uniformly and independently from F. Since $\mathcal{V}(p) = V_0 \subseteq V_{i-1}$ and F is expansive, we have that:

$$\mu(V_{i-1}) \geq \mu(\mathcal{V}(p)) \geq \epsilon$$

and:

$$\mu(\beta\text{-LOOKOUT}(V_{i-1})) \geq \alpha\mu(V_{i-1}) \geq \alpha\epsilon = 1/w.$$

It follows that the probability that M does not contain a point in β-LOOKOUT(V_{i-1}) is at most:

$$(1 - 1/w)^{s-i} \leq e^{-(s-i)/w}.$$

Hence we have:

$$\Pr(\overline{L_i}) \leq \Pr(\overline{L}_{i-1}) + e^{-(s-i)/w}$$

and:

$$\Pr(\overline{L}_t) \leq \sum_{i=1}^{t} e^{-(s-i)/w} = e^{-(s-1)/w} \sum_{i=0}^{t-1} e^{i/w} = e^{-(s-1)/w} \frac{e^{t/w} - 1}{e^{1/w} - 1}.$$

Noting that $e^{1/w} - 1 \geq 1/w$, we obtain the desired bound:

$$\Pr(\overline{L}_t) \leq we^{-(s-t-1)/w}.$$

That is, with probability at least $1 - we^{-(s-t-1)/w}$, M contains a linking sequence of length t for p. ∎

Lemma 2. *Let $v_t = \mu(V_t)$ denote the volume of the tth set V_t determined by a linking sequence $p_0 = p, p_1, p_2, \ldots$ for a point $p \in E$, where E is some connected component of F. Then, for $t \geq \beta^{-1} \log 4 \approx 1.39/\beta$, $v_t \geq 3\mu(E)/4$.*

Proof: Let us scale up all the volumes so that $\mu(E) = 1$. Observe that since $V_i = V_{i-1} \cup \mathcal{V}(p_i)$, we obtain:

$$\begin{aligned} \mu(V_i) &= \mu(V_{i-1}) + \mu(\mathcal{V}(p_i) \setminus V_{i-1}) \\ &\geq \mu(V_{i-1}) + \beta\mu(E \setminus V_{i-1}). \end{aligned}$$

The last inequality follows by the definition of an expansive space. Observing that $\mu(E \setminus V_{i-1}) = \mu(E) - \mu(V_{i-1}) = 1 - v_{i-1}$, we have the recurrence:

$$v_i \geq v_{i-1} + \beta(1 - v_{i-1}).$$

The solution to this recurrence turns out to be:

$$v_i \geq (1-\beta)^i v_0 + \beta \sum_{j=0}^{i-1} (1-\beta)^j = 1 - (1-\beta)^i (1 - v_0).$$

Observing that $v_0 \geq 0$ and that $(1 - \beta) \leq e^{-\beta}$, we obtain:

$$v_i \geq 1 - e^{-\beta i}.$$

Clearly, for $t \geq \beta^{-1} \log 4$, we have $v_t \geq 3/4$. ∎

We are now ready to prove Theorem 3.

Theorem 3. *Assume that F is $(\epsilon, \alpha, \beta)$-expansive. Let ξ be a constant in $(0, 1]$. If s is chosen such that:*

$$s \geq \frac{16}{\epsilon\alpha} \log \frac{8}{\epsilon\alpha\xi} + \frac{6}{\beta} + 2,$$

then with probability at least $1 - \xi$, `roadmap` *generates a roadmap such that no two of its components lie in the same component of F.*

Proof: Let $R = (M, L)$ be a roadmap constructed by `roadmap` in F. For each connected component F_j in F, let $M_j \subseteq M$ be the set of milestones belonging to F_j, and R_j be the subgraph of R containing the set M_j of vertices.

Suppose that we sample a total of $s = 2r + 2$ milestones from F. Consider any two points $p, q \in M_j$ for some j. Partition the rest of M into two sets, M' and M'', of r milestones each. It follows from Lemma 1 that any milestone in $\{p\} \cup M'$ has a linking sequence of length t in M' with probability at least $1 - we^{-(r-t)/w}$. The same holds for any milestones in $\{q\} \cup M'$. Let $V_t(p)$ and $V_t(q)$ be the final sets determined by the linking sequences of length t for the two milestones p and q, respectively. By Lemma 2, both sets have volume at least $3\mu(F_j)/4$ if we choose $t = 1.5/\beta$, and hence they must have a non-empty intersection with volume at least $\mu(F_j)/2 \geq \epsilon/2$. Since the r milestones in M'' are sampled independently at random, it follows that with probability at least $1 - (1 - \epsilon/2)^r \geq 1 - e^{-r\epsilon/2}$, there is a milestone $x \in M''$ that lies in the intersection. Note that both p and q have a path to x consisting of straight-line segments bending only at the linking sequence points, which of course belong to the set of milestones M_j. This means that there is a path from p and q to x using only the edges of the roadmap graph R_j.

Let B denote the event that p and q fail to be connected. We now calculate the probability $\Pr(B)$. Event B occurs if the sets in the linking sequences of p and q do not intersect or no points of M'' lie in the intersection. Hence, choosing $r \geq 2t$ and recalling that $w = 1/\alpha\epsilon$, we have:

$$\Pr(B) \leq 2we^{-(r-t)/w} + e^{-r\epsilon/2} \leq 2we^{-r/2w} + e^{-r/2w} \leq 3we^{-r/2w}.$$

The graph R_j will fail to be a connected graph if any pair of nodes $p, q \in M_j$ fail to be connected. The probability is at most:

$$\begin{aligned}
\binom{r}{2} \Pr(B) &= \binom{r}{2} 3we^{-r/2w} \\
&\leq 2wr^2 e^{-r/2w} \\
&\leq 2we^{-(r-4w\log r)/2w} \\
&\leq 2we^{-r/4w},
\end{aligned}$$

where the last inequality follows from the observation that $r/2 \geq 4w \log r$ for $r \geq 8w \log 8w$. Now if we also require that $r \geq 8w \log(8w/\xi)$, we have:

$$\begin{aligned}
2we^{-r/4w} &\leq 2we^{-2\log(8w/\xi)} \\
&\leq 2w\left(\frac{\xi}{8w}\right)^2 \\
&\leq \xi.
\end{aligned}$$

Clearly, it is sufficient to choose $r \geq 8w \log(8w/\xi) + 2t$. Substituting $w = 1/\alpha\epsilon$ and $t = 1.5/\beta$ into the expression for $s = 2r + 2$, we obtain the desired result. ∎

Appendix: D. Proof of Theorem 4

Theorem 4. *Let ζ be a constant in $(0,1]$ and a be the constant $2^{-n}\mu(\mathcal{B}_1)/\mu(F)$ where $\mathcal{B}_1$ denotes the unit ball in $\mathbb{R}^n$. If s is chosen such that:*

$$s \geq \frac{1}{a\sigma^n} \log \frac{2\ell}{\sigma\zeta},$$

then with probability at least $1-\zeta$ `roadmap` *generates a roadmap in which one component contains two milestones m and m' such that q sees m and q' sees m'.*

Proof: We assume the existence of a path $\tau : u \in [0,\ell] \mapsto \tau(u) \in F$ connecting q_b to q_e, where u stands for the arc length from q_b and ℓ denotes the total length of the path. Let σ designate the infimum of the Euclidean distance between $\tau(u)$ and the boundary of F, when u spans the interval $[0,1]$. Given any two configurations $q = \tau(u)$ and $q' = \tau(u')$ on τ, let $l(q,q')$ denote the path length $|u-u'|$.

Let $\mathcal{B}_r(x)$ designate the ball of radius r centered at $x \in \mathbb{R}^n$. We pick $k = \lceil 2\ell/\sigma \rceil$ configurations on τ, denoted by $q_0 = q_b, q_1, \ldots, q_k = q_e$, such that $l(q_i, q_{i+1}) \leq \sigma/2$, for all $i \in [0, k-1]$. We have that:

$$\mathcal{B}_{\sigma/2}(q_{i+1}) \subset \mathcal{B}_\sigma(q_i).$$

For any two points $p_i \in \mathcal{B}_{\sigma/2}(q_i)$ and $p_{i+1} \in \mathcal{B}_{\sigma/2}(q_{i+1})$, the straight-line segment connecting p_i and p_{i+1} lies entirely in F; indeed, the above relation implies that p_{i+1} also lies in $\mathcal{B}_\sigma(q_i)$. So, a sufficient condition for `query` to find a path is that each ball $\mathcal{B}_{\sigma/2}(q_i)$, $i = 1, \ldots, k-1$ contains at least one milestone. The probability that a ball of radius r lying entirely in F contains none of the s milestones is $(1-\mu(\mathcal{B}_r)/\mu(F))^s$. In $\mathbb{R}^n$ we have $\mu(\mathcal{B}_r) = r^n\mu(\mathcal{B}_1)$. Therefore, the probability that the planner does not find a path is at most:

$$\left(\lceil \frac{2\ell}{\sigma} \rceil - 1\right) \left(1 - \frac{\mu(\mathcal{B}_{\sigma/2})}{\mu(F)}\right)^s,$$

which is itself no greater than:

$$\frac{2\ell}{\sigma} \left(1 - 2^{-n} \frac{\mu(\mathcal{B}_1)}{\mu(F)} \sigma^n\right)^s.$$

Hence, choosing s such that the above quantity is at most $\zeta \in (0,1]$ guarantees that the planner will find a path with probability at least $1-\zeta$. ∎

References

[1] S. Akl. *Parallel Computation: Models and Methods*, Prentice Hall, Englewood Cliffs, NJ, 1997.

[2] N. Amato, O.B. Bayazit, L.K. Dale, C. Jones, and D. Vallejo. OBPRM: An Obstacle-Based PRM for 3D Workspaces. *Proc. Workshop on Algorithmic Foundations of Robotics (WAFR)*, Houston, TX, March 1998.

[3] N. Amato, and Y. Wu. A Randomized Roadmap Method for Path and Manipulation Planning. *Proc. IEEE Int. Conf. on Robotics and Automation*, Minneapolis, MN, pp. 113-120, 1996

[4] J. Barraquand, L.E. Kavraki, J.C. Latombe, T.Y. Li, R. Motwani, and P. Raghavan. A Random Sampling Scheme for Path Planning. *Int. J. of Robotics Research*, 16(6):759-774, 1997.

[5] J. Barraquand and J.C. Latombe. Robot Motion Planning: A Distributed Representation Approach. *Int. J. of Robotics Research*, 10(6):628-649, 1991.

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

[7] H, Chang and T.Y. Li. Assembly Maintainability Study with Motion Planning, *Proc. IEEE Int. Conf. on Robotics and Automation*, Nagoya. pp. 1012-1019, 1995.

[8] D.P. Dobkin, J. Hershberger, D.G. Kirkpatrick, and S. Suri. Computing the Intersection Depth of Polyhedra. *Algorithmica*, 9:518-533, 1993.

[9] E.G. Gilbert, D.W.Johhson, and S.S. Keerthi. A Fast Procedure for Computing the Distance Between Complex Robots in Three-Dimensional Space. *IEEE Tr. on Robotics and Automation*, 4:193-203, 1988.

[10] J.E. Hopcroft, J.T. Schwartz, and M. Sharir. On the Complexity of Motion Planning for Multiple Independent Objects: PSPACE-Hardness of the 'Warehouseman's Problem'. *Int. J. of Robotics Research*, 3(4):76-88, 1984.

[11] J.E. Hopcroft and G.T. Wilfong. Reducing Multiple Object Motion Planning to Graph Searching. *SIAM J. on Computing*, 15(3):768-785, 1986.

[12] T. Horsch, F. Schwarz, and H. Tolle. Motion Planning for Many Degrees of Freedom - Random Reflections at C-Space Obstacles. *Proc. IEEE Int. Conf. on Robotics and Automation*, San Diego, CA April 1994, pp. 3318-3323.

[13] D. Hsu, L.E. Kavraki, J.C. Latombe, R. Motwani, and S. Sorkin. On Finding Narrow Passages with Probabilistic Roadmap Planners. *Proc. Workshop on Algorithmic Foundations of Robotics (WAFR)*, Houston, TX, March 1998.

[14] D. Hsu, J.C. Latombe, and R. Motwani. Path Planning in Expansive Configuration Spaces. Proc. *IEEE Int. Conf. on Robotics and Automation*, Albuquerque, NM, 1997, pp. 2719-2726. An extended version of this paper will appear in *Int. J. of Computational Geometry and Applications*.

[15] P. Jiménez, F. Thomas, and C. Torras. Collision Detection Algorithms for Motion Planning. *Robot Motion Planning and Control*, J.P. Laumond (ed.), Lecture Notes in Control and Information Sciences, 229, Springer, New York, NY, 1998, pp. 305-343.

[16] D.A. Joseph and W.H. Plantiga. On the Complexity of Reachability and Motion Planning Questions. *Proc. 1st ACM Symp. on Computational Geometry*, pp. 62-66, 1985.

[17] L. Kavraki. *Random Networks in Configuration Space for Fast Path Planning*. Ph.D. Thesis, Rep. No. STAN-CS-TR-95-1535, Department of Computer Science, Stanford Univ., Stanford, CA, 1995.

[18] L. Kavraki, M. Kolountzakis, and J.C. Latombe. Analysis of Probabilistic Roadmaps for Path Planning. *Proc. IEEE Int. Conf. on Rob. and Aut.*, Minneapolis, MN, pp. 3020-3025, 1996.

[19] L. Kavraki, J.C. Latombe, R. Motwani, and P. Raghavan. Randomized Query Processing in Robot Motion Planning. *Proc. ACM SIGACT Symposium on the Theory of Computing (STOC)*, Las Vegas, Nevada, 1995, pp. 353-362.

[20] L. Kavraki, P. Švestka, J.C. Latombe, and M. Overmars Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566-580, August 1996.

[21] Y. Koga, K. Kondo, J. Kuffner, and J.C. Latombe. Planning Motions with Intentions. *Proc. of SIGGRAPH'94*, ACM, pp. 395-408, 1994.

[22] M. Lin and J.F. Canny. A Fast Algorithm for Incremental Distance Computation. *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Sacramento, CA, 1994, pp. 602-608.

[23] M. Lin, D. Manocha, J. Cohen, and S. Gottschalk. Collision detection: Algorithms and applications. *Algorithmic Foundations of Robotics*, Goldberg et al. (Eds), A K Peters, Ltd., 1995, pp. 129-141.

[24] B. Mirtich. *V-Clip: Fast and Robust Polyhedral Collision Detection*. Tech. Rep. TR97-05, Mitsubishi El. Res. Lab., Cambridge, MA, 1997.

[25] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, Cambridge, UK, 1995

[26] C.J. Ong. On the Quantification of Penetration between General Objects. *Int. J. of Robotics Research*, 16(3):400-409, 1997.

[27] C.J. Ong and E.G. Gilbert. Growth Distances: New Measures for Object Separation and Penetration. *IEEE Tr. on Robotics and Automation*, 12(6):888-903, 1996.

[28] M. Overmars. A random Approach to Motion Planning. Technical Report, RUU-CS-92-32, Department of Computer Science, P.O. Box 80.089, 3508 TB Utrecht, The Netherlands, 1992.

[29] M. Overmars and P. Švestka. A Probabilistic Learning Approach to Motion Planning. *Algorithmic Foundations of Robotics*, K. Goldberg et al. (eds.), A.K. Peters, Wellesley, MA, 1995, pp. 19-37.

[30] S. Quinlan. Efficient Distance Computation Between Non-Convex Objects. *Proc. IEEE Int. Conf. on Robotics and Automation*, San Diego, CA, pp. 3324-3330, 1994.

[31] J. Reif. Complexity of the Mover's Problem and Generalizations. *Proc. IEEE Symp. on Foundations of Computer Science*. IEEE, pp. 421-427, 1979.

[32] P. Švestka and M. Overmars. Probabilistic Path Planning. *Robot Motion Planning and Control*, J.P. Laumond (ed.), Lecture Notes in Control and Information Sciences, 229, Springer, New York, NY, 1998, pp. 255-304.

[33] Schwartz, J.T. and Sharir, M. 1983. On the 'Piano Movers' Problem: II. General Techniques for Computing Topological Properties of Real Algebraic Manifolds. *Advances in Applied Mathematics*. 4:298-351.