Deterministic vs. Probabilistic Roadmaps

Michael S. Branicky, Steven M. LaValle, Kari Olson, Libo Yang

Abstract— *Within the popular probabilistic roadmap (PRM) framework for motion planning, we challenge the use of randomization. By applying quasi-random sampling techniques, we illustrate both experimental and theoretical advantages of using deterministic samples. Two quasi-random variants of PRM-based planners are proposed: 1) a classical PRM with quasi-random sampling, and 2) a quasi-random lattice-based Lazy roadmap. Both have been implemented, and are shown through experiments to offer some performance advantages in comparison to their randomized counterparts. Furthermore, our theoretical analysis shows that our approach leads to deterministic resolution completeness. We obtain the best possible asymptotic convergence rate, which is shown to be superior to that obtained by random sampling.*

I. Introduction

Over two decades of path planning research have led to two primary trends. By the 1980s, deterministic approaches provided both elegant, complete algorithms for solving the problem (e.g., [10], [38], [41]), and also useful approximate or incomplete algorithms (e.g., [14], [15], [23], [28]). The curse of dimensionality due to high-dimensional configuration spaces motivated researchers from the 1990s to the present time to develop randomized approaches which are incomplete, but capable of efficiently solving many challenging, high-dimensional problems [4], [22], [26], [28], [33]. A similar pair of trends occurred many years ago in the area of numerical integration and related optimization fields, where they were followed by a third trend: the development of *quasi-random* approaches that use deterministic sampling to achieve performance that is often superior to random sampling. Quasi-random sampling ideas have improved computational methods in many areas, including integration [44], optimization [35], image processing [18], computer graphics [42], and computational geometry [11]. In this paper, the term quasi-random can be considered synonymous with deterministic; the term exists to emphasize comparisons with random and pseudo-random concepts.

It is therefore natural to ask: Can quasi-random sampling ideas also improve path planning methods designed for high degrees of freedom? Is randomization really the key to solving high-dimensional problems? Contrary to current motion planning trends, we argue in this paper that randomization is not necessarily advantageous in solving high-dimensional planning problems. In some cases it might lead to simpler algorithms; however, randomization itself it is not the fundamental reason why path planning approaches have been successful in practice.

In this paper, we investigate the use of deterministic sampling in the context of the popular probabilistic roadmap (PRM) framework introduced in [26]. The main idea of the PRM is to generate samples at random in the collision-free subset of the configuration space, and then build a graph of collision-free paths by connecting pairs of samples that are within a specified distance threshold. The key novelty over naive grid-based searching algorithms is the use of random samples as opposed to using the tiling of samples and lattice structure provided by a grid. While the number of samples required for a grid is known to increase exponentially in dimension, it was argued that the PRM was “primarily developed for robots with many dofs” [26] by overcoming this difficulty through random sampling. Thus, the PRM framework represents an ideal context for evaluating deterministic sampling ideas.

Many extensions and variations of PRMs have been proposed in the literature [1], [2], [8], [20], [31], [39], [43], [50]. Most variations consider alternative strategies for generating samples using randomization. For example, the Visibility PRM generates samples at random but only keeps two kinds: 1) those declared as *guards* that are not able to connect to other guards, and 2) those declared as *connectors*, which connect to guards and bring together two or more connected components of the roadmap [43]. Creating nodes in narrow passages has been the main motivation of the enhancement step in [25], the generation of nodes near the configuration space obstacles in [1], the penetration of obstacles in [21], the Gaussian sampling in [8], the retraction to the configuration space medial axis in [50], and the use of the workspace medial axis in [20] and [39]. It is difficult to compare random sampling to deterministic sampling for each variation of the PRM. We speculate that randomization appearing in other sampling techniques could be safely replaced and possibly improved with deterministic sampling; however, case-by-case comparisons would be necessary, and are beyond the scope of this paper.

One PRM variation that considers an idea independent of sampling is the Lazy PRM [6]. In this case, the idea is to first construct a roadmap that ignores collision constraints, and then perform collision checking to validate edges only while searching for a solution. This results in dramatic reduction in preprocessing time. In the worst case, all edges will be checked, which is only as bad as the original PRM. Given that the idea is sampling independent, we chose to make and evaluate deterministic versions of the Lazy PRM in addition to the basic PRM.

At first glance, the progression from deterministic to randomized, and then back to deterministic might appear absurd; thus, some explanation is required. There appear to be two prevailing reasons for the preference of randomized

M.S. Branicky is with the Electrical Engineering and Computer Science Department, Case Western Reserve University, Cleveland, OH 44106 USA. E-mail: mb@ieee.org

S.M. LaValle (corresponding author) is with the Dept. of Computer Science, University of Illinois, Urbana, IL 61801 USA. E-mail: lavalle@cs.uiuc.edu, Phone: +1-217-265-6313

K. Olson is with Proctor and Gamble, Cincinnati, OH 45202 USA. Email: olson.kd@pg.com

L. Yang is with the Dept. of Computer Science, Iowa State University, Ames, IA 50011 USA. E-mail: lyang@cs.iastate.edu

methods over classical deterministic techniques: 1) they fight the curse of dimensionality by allowing a problem to be solved without prior, systematic exploration of all alternatives; 2) if the "problem maker" is viewed as an opponent in a game, then one can often avoid defeat by employing a random strategy (imagine defeating a deterministic strategy by designing a problem that causes worst-case performance).

The first reason is often motivated by considering that a grid with a fixed number of points per axis will require a number of points that is exponential in the dimension of the space. However, this result is not the fault of grids or even deterministic sampling. It was proven long ago by Sukharev [45] that *any* sampling method that constructs a good covering of the space requires an exponential number of samples.[1] We believe the explanation for good performance of path planners in solving challenging high-dimensional problems is that they are able to either exploit some greedy heuristics and/or find solutions to easier problems early by using low-resolution sampling. *These benefits are independent of the issue of randomization versus determinism.* We note that other researchers have argued that the PRM sampling method must appropriately adapt to the difficulty of the problem, as opposed to attempting a uniform covering (e.g., [21], [43]). We are in agreement with this idea, independently of whether samples are random or deterministic.

The second reason (defeating an opponent) might be valid in the case of "true" random numbers; however, any machine implementation generates a deterministic sequence of pseudo-random numbers. These numbers are designed to meet performance criteria that are based on uniform probability densities; however, once it is understood that these numbers are deterministic and being used to solve a particular task, why not design a deterministic sequence that can solve the task more efficiently, instead of worrying about statistical closeness to a uniform probability density? This motivates the design of quasi-random numbers. One could argue that it is difficult or impossible for an opponent to obtain the seed in a pseudo-random number generator; one could even attempt to construct a natural seed for number generation as considered in cryptography [40]. However, even if we suppose that true random numbers exist, it seems unlikely that practical examples drawn from applications will contain configuration spaces that are designed to break a specific deterministic sampling strategy. Furthermore, randomization can even be introduced back into a deterministic sampling strategy for precisely the reason of fooling an adversary while still maintaining quasi-random sample distribution properties that are superior to pseudo-random sampling [32].

In Section II we give an overview of quasi-random sampling methods and literature. Section III presents a PRM in which the pseudo-random samples have been replaced with quasi-random samples; several experimental comparisons are made that illustrate the advantages of deterministic sampling. Section IV presents a Lazy PRM that replaces pseudo-random samples with a deterministic lattice; we refer to this as the Lazy LRM. Our experiments show in this case that deterministic samples offer additional benefits, such as immediate knowledge of neighborhood structure and fast initial roadmap setup time. Section V presents some analysis, including deterministic guarantees that our planners succeed, which is not possible in the case of randomized planners. Finally, some conclusions are presented in Section VI.

Parts of the work presented here were presented in earlier form in [9].

II. Sampling Methods

Deterministic sampling techniques have been developed by numerous mathematicians over the past century. Excellent overviews of the subject include [32], [35]. A brief treatment, specific to our problem, is presented here.

A. Sampling Criteria

Let $X = [0,1]^d \subset \mathbb{R}^d$ define a space over which to generate samples.[2] Consider designing a set, P, of N d-dimensional sample points $\{p_0, \ldots, p_{N-1}\}$ in a way that covers X uniformly in some sense. If $d = 1$, the points may be evenly spaced in an obvious, uniform way, but for $d > 1$ the problem becomes very challenging. There are both the challenges of defining a useful criterion of uniformity and then designing a sample set that attempts to optimize the criterion.

A good criterion should measure whether the sample points appear reasonable in various regions of X. For example, the PRM repeatedly tries to connect to samples in a randomly-centered ball in X. It would be useful if the criterion measures how many samples will fall into these neighborhoods. With this in mind, define a *range space*, $\mathcal{R}$, as a collection of subsets of X. Let $R \in \mathcal{R}$ denote one such subset. Reasonable choices for $\mathcal{R}$ include the set of all axis-aligned rectangular boxes, the set of all balls, or the set of all convex subsets.

Let $\mu(R)$ denote the Lebesgue measure (or volume) of subset R. If the samples in P are uniform in some ideal sense, then it seems reasonable that the fraction of these samples that lie in any subset R should be roughly $\mu(R)$ (divided by $\mu(X)$, which is simply one). We define the *discrepancy* [49] to measure how far from ideal the point set P is:

$$D(P, \mathcal{R}) = \sup_{R \in \mathcal{R}} \left| \frac{|P \cap R|}{N} - \mu(R) \right| \qquad (1)$$

in which $|\cdot|$ applied to a finite set denotes its cardinality.

For a given range space, the goal is to select P to minimize (1). If N is fixed, then the set of samples is considered *closed.* An infinite sequence of samples is considered *open*, and it can be evaluated by applying (1) asymptotically as a function of N. By fixing N, lower discrepancy can generally be obtained. For example, if $N = 1$, P might contain

[1] For brevity, we call this the *Sukharev sampling criterion*; the "goodness" is in terms of point dispersion, which will be defined in Section II.

[2] This is without loss of generality because scalings, translations, and embeddings are easily accomplished.

one point at the center of X. Imagine that this is the first point in an open sequence. For $N = 4$, having the one point at the center is probably not optimal. Thus, with closed sample sets the points can be arranged in any way, but for open sets they are bound to the sequence. Even though closed sample sets can obtain lower discrepancy, an infinite sequence is often more useful in applications because it provides samples incrementally (which makes it easier to replace a pseudo-random sequence).

Whereas discrepancy is based on measure, a metric-based criterion, called *dispersion*, can be introduced:

$$\delta(P, \rho) = \sup_{x \in X} \min_{p \in P} \rho(x, p). \tag{2}$$

Above ρ denotes any metric, such as Euclidean distance or ℓ^∞. We refer to such variants as *Euclidean dispersion* and ℓ^∞ *dispersion.* Note that if ρ is a Euclidean metric, the dispersion yields the radius of the largest empty ball.[3] If $\mathcal{R}$ represents the set of all balls, then D is at least as large as this volume because $|P \cap R| = 0$. It is known that $\delta(P, \rho) \leq D(P, \mathcal{R})^{\frac{1}{d}}$, if ρ is the ℓ^∞ metric and $\mathcal{R}$ is the set of all axis-aligned rectangular subsets [35], [46]. (See those references and [32] for further results.) Thus, low discrepancy implies low dispersion. Dispersion is useful in the analysis of the PRM because it indicates whether samples can be connected for a given connection radius parameter.

For a fixed N, it is interesting to consider the best possible dispersion that can be obtained for any sample set. We refer to the following as the *Sukharev sampling criterion* [45]:

$$\delta(P) \geq \frac{1}{2\lfloor N^{\frac{1}{d}} \rfloor}, \tag{3}$$

which holds true for any point set P, when δ is the ℓ^∞ dispersion. Suppose we would like to place N points in $[0,1]^d$ so that the ℓ^∞ dispersion is a small as possible. Assume for convenience that $N^{\frac{1}{d}}$ is an integer. Solving (3) for a prescribed dispersion yields, $N \geq (1/2\delta)^d$, which means that the number of samples is exponential in dimension, regardless of how the points are placed! It was also shown in [45] that for any d and N, there exists a set, P, of N points such that (3) is an equality. This is achieved by arranging the points in a grid in which the discretization interval is roughly (due to the floor) $N^{-\frac{1}{d}}$ and the first point is shifted $\frac{1}{2}N^{-\frac{1}{d}}$ from the origin. This set represents the lowest possible ℓ^∞ dispersion; for Euclidean dispersion, it remains an open and challenging problem to find optimal point sets for general d and N.

B. Halton, Hammersley, and Other Sample Sets

Numerous low-discrepancy sample sets have been proposed. The choice of one set over others usually depends on several concerns: 1) the desired range space, $\mathcal{R}$, 2) theoretical bounds on the discrepancy, 3) the quality of the samples as observed in applications, 4) the difficulty of constructing the samples. The most common range space is the set of all axis-aligned rectangular subsets, which we denote by $\mathcal{R}_{aar}$. The second and third concerns are both included because current theoretical analysis is unable to completely characterize the practical value of a low-discrepancy sample set. For a well-studied sequence of samples, asymptotic bounds are usually given on the discrepancy, often expressed with an unspecified constant. When the constant is known, it is usually large, which produces a pessimistic bound. The fourth concern becomes important if the computational complexity or implementation difficulty outweigh the utility of the samples.

In an appendix of [32], the current upper and lower bounds on the best possible discrepancy for an open sequence, attainable for different range spaces are summarized. For example, the best known lower bound on D using $\mathcal{R}_{aar}$ is

$$O\left(\frac{1}{N} \log^{\frac{d-1}{2}} N \left(\frac{\log\log N}{\log\log\log N}\right)^{\frac{1}{2d-2}}\right);$$

however, if $\mathcal{R}$ represents the set of all balls, then the lower bound is $O(N^{-(d+1)/2})$.

The best known upper bound for open sequences and $\mathcal{R}_{aar}$, $O(\frac{\log^d N}{N})$, is achieved by the *Halton sequence*, which is constructed as follows [16]. Choose d distinct primes $p_1, p_2, \ldots, p_d$ (usually the first d primes, $p_1 = 2$, $p_2 = 3$, $\ldots$). To construct the ith sample, consider the digits of the base p representation for i in the reverse order: $i = a_0 + pa_1 + p^2a_2 + p^3a_3 + \ldots$, in which $a_j \in \{0, 1, \ldots, p\}$. Define the following element of $[0,1]$:

$$r_p(i) = \frac{a_0}{p} + \frac{a_1}{p^2} + \frac{a_2}{p^3} + \frac{a_3}{p^4} + \cdots.$$

The ith sample in the Halton sequence is

$$(r_{p_1}(i), r_{p_2}(i), \ldots, r_{p_d}(i)), \qquad i = 0, 1, 2, \ldots.$$

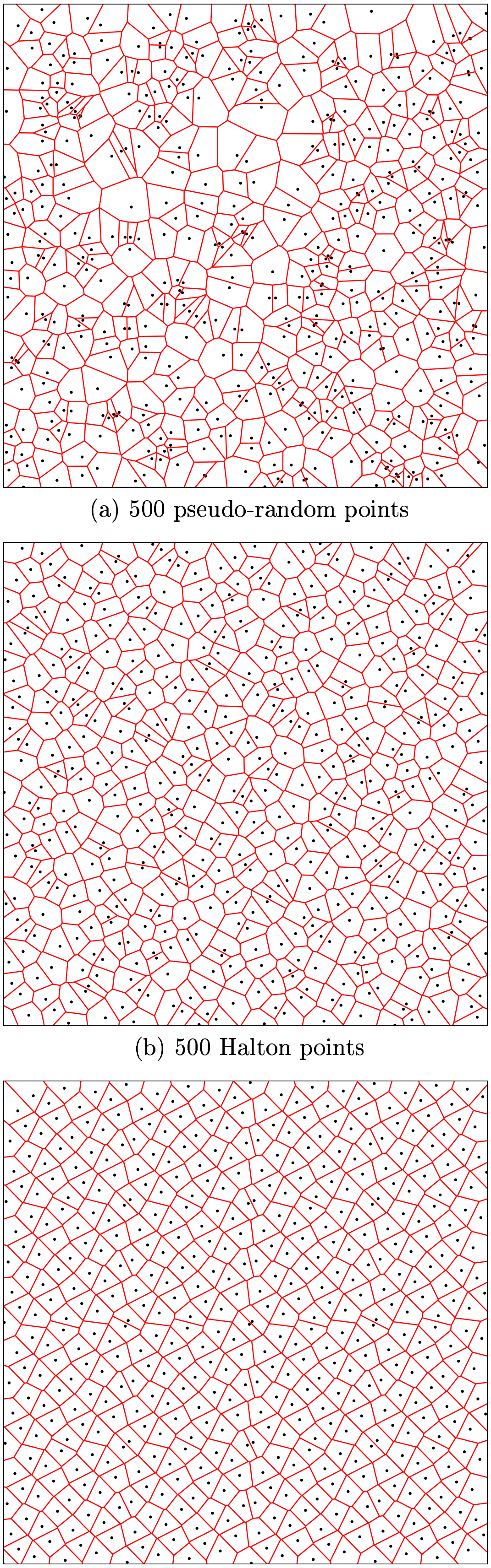
A *Hammersley sequence* is a closed-sequence variant of the Halton sequence that achieves even lower asymptotic discrepancy, $O(\frac{\log^{d-1} N}{N})$ [17]. Using only $d-1$ distinct primes, the ith sample in the Hammersley sequence is

$$\left(\frac{i}{N}, r_{p_1}(i), \ldots, r_{p_{d-1}}(i)\right), \qquad i = 0, 1, \ldots, N-1.$$

For the case of $d = 2$, the Hammersley sequence is often referred to as the Van der Corput sequence [48], which was introduced much earlier. Figure 1 compares sample sets in $X = [0,1] \times [0,1]$ and also shows the Voronoi diagram of the point set (i.e., in each region the representative sample is the closest among all samples). Notice that for pseudo-random points there is a large variation in region size and shape, which illustrates the nonuniformity of random samples. For Halton points, the Voronoi regions appear more regular, and the Hammersley points are even better because it is a closed sample set. The computed Euclidean dispersions for the three cases in order are: 0.0788, 0.0539, and 0.0413.[4] As expected, the dispersion is much better for Halton and Hammersley sets.

[3] Actually, for any metric, it gives the radius of the largest empty ball in that metric *restricted* to the domain X.

[4] We actually computed a variant of dispersion by computing the largest empty circle whose center lies in the convex hull of the points. The difference is negligible for the large number of samples.



(a) 500 pseudo-random points

(b) 500 Halton points

(c) 500 Hammersley points

Fig. 1. Shown are 500 pseudo-random, Halton, and Hammersley points, respectively, plus their associated Voronoi regions. Notice the regularity in the Voronoi diagram for the quasi-random sample sets.

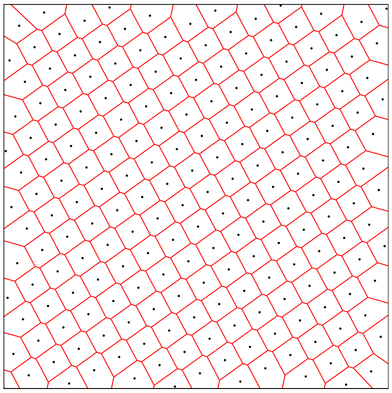


Fig. 2. 200 low-discrepancy lattice points generated by using the golden ratio, $\alpha = \frac{\sqrt{5}+1}{2}$. The Voronoi regions are also shown.

In Section III, we use Halton and Hammersley points because of their well-known performance in practice and their easy construction. There exist, however, sample sets with even better performance. Currently, the family of sample sets with the best known bounds (smallest constants in the asymptotic analysis) draw from powerful algebraic geometry techniques [37]. See [32], [35], [46] for many others.

C. Quasi-Random Lattices

Given the regularity observed for Hammersley points in Figure 1, it is natural to ask whether low-discrepancy closed sample sets exist that have grid-like properties, such as equal spacing between points and offsets to reach neighbors that are identical for any sample. Such sets do exist and are called quasi-random lattices. Recent analysis shows that some lattice sets achieve asymptotic discrepancy, $O(\frac{\log^{d-1} N}{N})$, for $\mathcal{R}_{aar}$, which is equivalent to that of the best known non-lattice sample sets [32]. Thus, restricting the points to lie on a lattice seems to entail little or no loss in performance [19], but with the added benefit of grid-like structure that is useful for path planning. Furthermore, as shown by Sukharev [45], the best possible ℓ^∞ dispersion is obtained by placing the points in a grid configuration.

As an example, consider Figure 2, which shows 200 lattice points generated by the following technique. Let α be a positive irrational number. For a fixed N (lattices are closed sample sets), generate the ith point according to $(\frac{i}{N}, \{i\alpha\})$, in which $\{\cdot\}$ denotes the fractional part of the real value (modulo-one arithmetic). This procedure can be generalized to d dimensions by picking $d-1$ distinct irrational numbers. A technique for choosing the α_k parameters by using the roots of irreducible polynomials is discussed in [32]. The ith sample in the lattice is

$$\left(\frac{i}{N}, \{i\alpha_1\}, \ldots, \{i\alpha_{d-1}\}\right), \qquad i = 0, 1, \ldots, N-1.$$

Many other possibilities exist for producing low-discrepancy lattices. One, which is applied in Section IV, is based on selecting d integers, z_1, …, z_d, and constructing the ith sample as follows [44]:

$$\left(\left\{\frac{iz_1}{N}\right\},\ldots,\left\{\frac{iz_d}{N}\right\}\right), \qquad i=0,1,\ldots,N-1.$$

A variety of techniques for selecting good z_k values are presented in [44].

It is well-known that the points in a lattice form an Abelian group with respect to addition, and that all points can be specified in terms of a collection of d linearly-independent basis vectors, $b_1,\ldots,b_d$, over the reals. For any lattice point, p, there exists a set of d integers, i_1, ..., i_d, such that

$$p=\sum_{j=1}^{d} i_j b_j,$$

assuming modulo-one arithmetic. Note that if the basis vectors are chosen as the rows of a $d \times d$ identity matrix, then a classical grid is obtained (which is known to have high discrepancy, but low dispersion).

By using generators, the grid-like neighbors can be obtained immediately by adding (or subtracting) one of the generators to the sample. For example, in the case of a 2D grid, the generators are $[1,0]$, $[0,1]$, which can be used to obtain the coordinates of standard four-neighbors, or eight-neighbors by allowing diagonals. This idea generalizes to any lattice for quick determination of neighboring samples, which is of critical importance in path planning algorithms such as the PRM.

One additional feature for some lattices is the ability to vary the resolution while preserving low discrepancy. Families of embedded lattices are described in [44], in which each lattice contains twice as many samples as the previous one, and each lattice includes all points from lower-resolution lattices. This feature is useful in planning for gradually increasing the resolution over time. One way to construct a family of embedded lattices is by

$$\frac{i\mathbf{z}}{N}+\frac{(k_1,\ldots,k_r,0,\ldots,0)}{p},$$

where $r \in \{0,\ldots,d\}$, each $k_i \in \{0,1\}$, the number of lattice points is $p^r N$, and p and N are relatively prime. In Section IV, we define a lattice-based variant of the Lazy PRM, and exploit the convenient neighborhood structure and embeddings of quasi-random lattices.

Summarizing, there are two popular measures of point uniformity: discrepancy and dispersion, both of which are relevant in the PRM context. Dispersion indicates whether there will be at least one neighbor for connection, while discrepancy tries to ensure there will be an appropriate number of alternatives for making attempted connections. Many low-discrepancy and low-dispersion quasi-random point sets have been proposed. One special subclass is lattices, which offer built-in neighborhood structures that are nicely suited for making PRM-like roadmaps.

III. Quasi-Random Roadmap (QRM)

A. The Basic PRM

We present a brief description of the path planning problem and the probabilistic roadmap (PRM) approach introduced in [26]. Let $\mathcal{C}$ denote the configuration space (or *C-space*) of a robot in a 2D or 3D world that contains static obstacles. Let $\mathcal{C}_{free}$ denote the set of all collision-free configurations. The path planning problem is to find a continuous path, $\tau : [0,1] \rightarrow \mathcal{C}_{free}$ such that $\tau(0) = q_{init}$ and $\tau(1) = q_{goal}$.

```
BUILD_PRM
1  G.init();
2  for $i = 1$ to $N$
3      $q \leftarrow$RAND_FREE_CONF($q$);
4      G.add_vertex($q$);
5      for each $v \in$ NBHD($q$,G)
6          if ((not G.same_component($q, v$)) and
7              CONNECT($q, v$)) then
8                  G.add_edge($q, v$);
```

Fig. 3. The preprocessing phase: build a PRM.

The primary philosophy behind the PRM was to perform substantial preprocessing so that *multiple queries* for the same environment could be handled efficiently. Section IV discusses a single-query version of the PRM. The approach for the basic PRM results in two phases to path planning. First, a *roadmap* encoded as an undirected graph, G, is constructed in a *preprocessing phase*. In a *query phase*, G is used to solve a particular path planning question for a given q_{init} and q_{goal}. Each vertex in G represents an element of $\mathcal{C}_{free}$, and each edge represents a collision free path between two configurations.

The algorithm outlined in Figure 3 constructs a PRM with N vertices. In Step 3, a pseudo-random configuration in $\mathcal{C}_{free}$ is found by repeatedly picking a pseudo-random configuration until one is determined by a collision detection algorithm to be in $\mathcal{C}_{free}$. The NBHD function in Line 5 is a range query in which all vertices within a specified distance of q are returned, sorted by distance from q (other variations are possible, of course). This is the step in which the range space concepts from Section II become relevant to planning: over a certain range space, it is important to have the points distributed so that NBHD contains a sufficient number of points. In Step 6, it is sometimes preferable to replace the condition (**not** G.same_component(q, v)) with G.vertex_degree$(q) < K$, for some fixed K (e.g., K = 15); this was done in our experiments. The CONNECT function in Line 7 uses a fast local planner to attempt a connection between q and v. Usually, a “straight line” path in $\mathcal{C}_{free}$ is evaluated between q and v by stepping along incrementally with a collision detection algorithm. In [26] a node enhancement phase is described which introduces some heuristic techniques that try to reduce the number of connected components by adding new nodes in critical places. We omit this phase here to facilitate sampling comparisons. A 2D example is shown in Figure 4(a) for which $N = 1000$. Note the characteristic clumping of the randomly-chosen nodes and the relatively large areas of free space that contain no samples. The figure also confirms the well-known fact that narrow passages in C-space are notoriously difficult to find at random [8], [21]

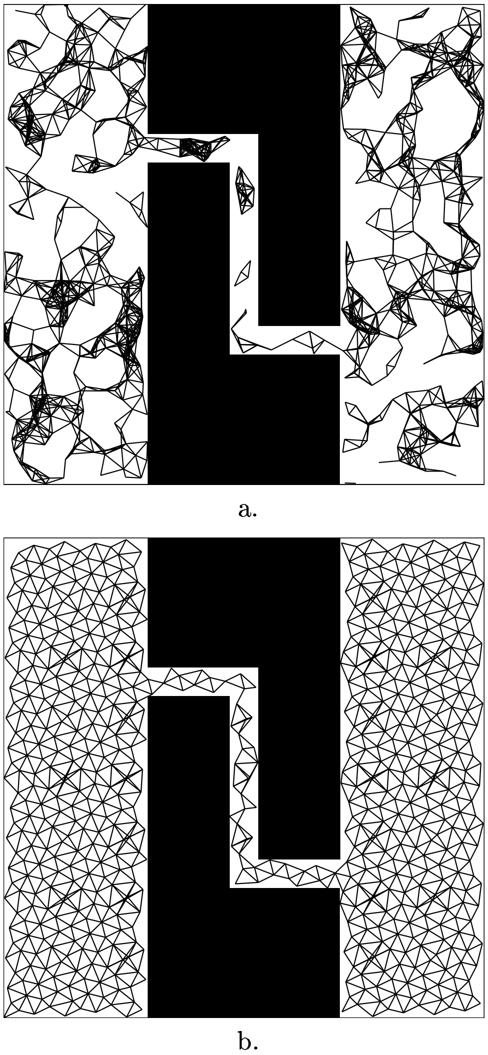


Fig. 4. a) A probabilistic roadmap based on traditional, pseudo-random sampling; b) a quasi-random version based on Hammersley points. Each uses 1000 samples and the same connection radius, 0.05.

Once the PRM has been constructed, the query phase attempts to solve planning problems. Essentially, q_{init} and q_{goal} are treated as new nodes in the PRM, and connections are attempted. Then, standard graph search is performed to connect q_{init} to q_{goal}. If the method fails, then either more vertices are needed in the PRM, or there is no solution.

B. QRM Overview

Monte Carlo methods, such as the PRM and random sampling-based techniques for integration and optimization, have been adopted for problems with high dimension to overcome the curse of dimensionality. Recently, quasi-Monte Carlo algorithms, which are identical to their random siblings except that they use deterministic point sets, have been shown to be both computationally efficient and accurate for a variety of applications, including 360-dimensional integrations performed in finance and bounded optimization [47].

Our approach in the PRM context is to simply replace the pseudo-random sample generator that appears in Line 3 of Figure 3 with a low-discrepancy, deterministic sampling method. In Figure 4(b), we show a QRM constructed using 1000 Hammersley points, which is the same number as in Figure 4(a). Here, however, a path through the narrow passage has been found, there is no clumping of points, and every point in $\mathcal{C}_{free}$ is fairly close to a vertex.

We can visualize the importance of the dispersion to the roadmap by placing a ball with radius equal to the dispersion at each sample point; the entire sample space is now covered. Obviously, the fewer of these balls we have, the smaller our roadmap is (in terms of number of nodes); the smaller these balls are, the easier it is to connect a query to the roadmap. Going back to Figure 4, one can see that the dispersion in (a) is larger than that in (b). Of course, other methods (e.g., [8], [21]) can be used to address the narrow corridor problem; however, it is interesting to note that better performance can be obtained by merely replacing pseudo-random samples with deterministic samples. This seems to contradict the intuition that led to the PRM in the first place. It may also be possible to improve the performance of other PRM sampling techniques [1], [2], [8], [20], [31], [39], [43], [50] by using deterministic replacements for the samples.

We have used Hammersley, Halton, and generalized Faure points as inputs to QRM algorithms to solve a variety of planning problems in a range of dimensions. Broadly speaking, the QRM has performance better than or equal to its PRM counterpart. We present some comparative experiments in Section III-C. In Section V, theoretical advantages are considered, including resolution completeness, which is not possible for the original PRM.

C. Comparing PRM and QRM Experimentally

Figure 5 shows the results of experiments performed on narrow corridor problems that have the same geometry as the example in Figure 4, but in higher-dimensional spaces. The configuration spaces range from 2 to 10 dimensions, and each involves a corridor with two bends and a cubic cross section with its width indicated in the table (the entire C-space in d dimensions is $[0,1]^d$). The connection radius is given in the third column. The number of nodes required to find a path that travels through the corridor is shown for both the QRM and 100 averaged trials of the PRM. The final column indicates the improvement factor of quasi-random over random sampling, in terms of the number of nodes. We have also generally observed larger improvement factors as the corridor width narrows. With wide corridors in high-dimensional spaces, the performance of the methods appears to be comparable.

Figure 6 shows a 6-DOF planning problem that the QRM solved with 5908 nodes; the PRM averaged 8020 nodes over 35 trials (the min and max for the PRM were 5551 and 14863, respectively).

In the above experiments, Hammersley points using the first $d-1$ primes were used in C-spaces of dimension d. In all experiments, pseudo-random numbers were generated using the linear congruential generator

Dim.	Width	Rad.	PRM	QRM	Factor
2	.03	.10	464	195	2.38
3	.05	.25	828	579	1.56
3	.10	.40	106	26	4.08
6	.10	.40	12857	4052	3.17
10	.25	.60	1531	1506	1.02
10	.20	.60	6260	2101	2.98

Fig. 5. A comparison between the PRM and QRM for narrow corridor problems. The QRM improvement factor is shown in the final column. PRM results are averaged over 100 trials.

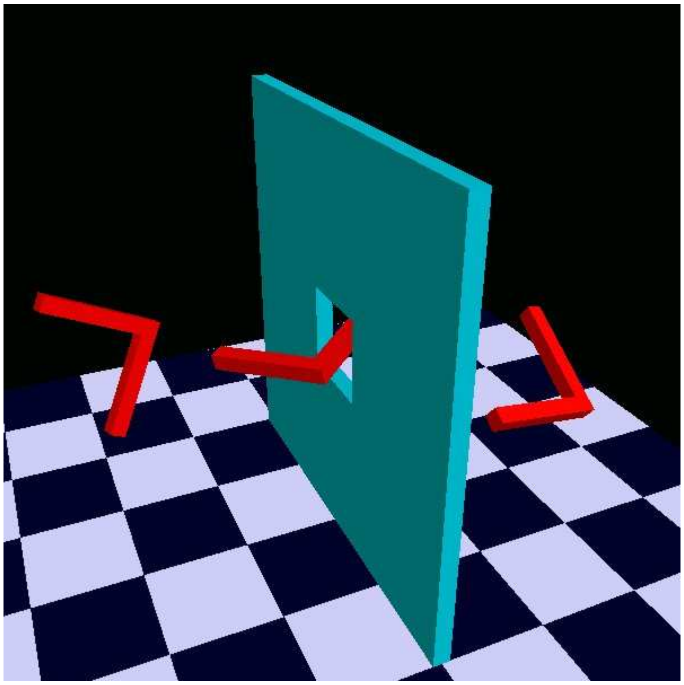


Fig. 6. A 6-DOF planning problem in which an elbow-shaped robot passes through a small opening.

found in MATLAB$^{\mathrm{TM}}$ or the random source class of the LEDA/C++ library. Hammersley-Halton low-discrepancy sequences were chosen by coding the definitions in Section II-B. The time to generate the quasi-random samples in comparison to pseudo-random samples was never a significant factor.

IV. Lazy Lattice Roadmap (Lazy LRM)

A. The Basic Lazy PRM

A recent PRM variant called the Lazy PRM has been proposed for the problem of answering single planning queries efficiently, as opposed to building an extensive roadmap prior to consideration of a planning query [6]. The resulting planner is sometimes very efficient in comparison to the original PRM. This represents a shift from the *multiple query* philosophy of the original PRM [26], and returns to the *single query* philosophy which was used in earlier planners [4], [15], [33].

The key idea in the Lazy PRM is to build the roadmap initially without the use of a collision detector. The difference with respect to the algorithm in Figure 3 is that the condition in Lines 6 and 7 is dropped, and Line 8 is executed every time. This allows the PRM to be constructed quickly; however, more burden is placed on the searching in the query phase. Once an initial-goal query is given, the planner performs A^* search on the roadmap to find a solution. If any of the solution edges are in collision, they are removed from the roadmap, and the A^* search is repeated. Eventually, all edges may have to be checked for collision, but often the problem is solved well before this happens. If no solution is found, then more nodes may need to be added to the roadmap. The advantage of the Lazy PRM is that the collision checking is only performed as needed. Thus, all edges do not have to be collision checked as in the case of the original PRM.

One of the difficulties with the Lazy PRM is that it constructs a large graph that does not represent anything particular to a given environment with obstacles. According to the experiments in [6], the initial graph construction time represents a significant fraction of the total running time. Substantial time is spent on constructing a randomly-generated graph that may contain thousands of nodes and edges, and require thousands of nearest-neighbor queries; however, the roadmap encodes no true information because the obstacles are ignored. Although the graph contains no problem-specific information, it cannot be compressed nor constructed implicitly because it is based on random sampling. Another difficulty is that without knowing the particular problem, it is difficult to determine how many vertices should appear in the roadmap. How many should be added if the query fails?

B. A Lazy LRM from Quasi-Random Lattices

The previously-mentioned difficulties with the Lazy PRM led us to investigate the use of quasi-random lattices in this context. These difficulties also motivated one of the Lazy PRM authors to recently make a grid-based variant [5]. Recall from Section II-C that lattices have a regular, well-defined neighborhood structure. This allows the initial roadmap to be implicitly defined with little or no precomputation because all vertices, neighboring vertices, and edges are defined implicitly by lattice rules. This is precisely what is obtained for classical grid search techniques. The grid can be declared in memory and neighborhoods are based on simple vector offsets, as opposed to the time-consuming NBHD range query in Line 5 of Figure 3. If memory limits are a problem, then a hashing scheme can be used to represent only those vertices that have been explored. In a sense, we can make a "lazier" PRM in which the initial graph is not even explicitly represented.

Given that lattices have discrepancy bounds that are as good as the best bounds for non-lattice sample sets, we can make a Lazy LRM (lattice roadmap) that simultaneously obtains the low-discrepancy benefits observed in the QRM and the dramatic reduction in precomputation time. Furthermore, embedded lattices, discussed in Section II-C, enable a multi-resolution approach to the problem. For example, if a lattice with N points fails to solve a query, the resolution can be doubled to $2N$ by implicitly declaring more points. Families of embedded lattices for which

Prob.	Min	Max	Avg	Lazy LRM
Elbow	1250	15250	4667	3963
Cup	2000	12000	4800	2152
Truck	5000	95000	35207	5138

Fig. 7. Comparisons of the number of nodes for the Lazy PRM vs. the Lazy LRM. PRM results are for 25 trials.

Prob.	Min	Max	Avg	PreCmp	Lazy LRM
Elbow	7.0	718	287	212	10.1
Cup	2.33	253	36.9	15.9	1.23
Truck	11.2	5480	935	800	18.5

Fig. 8. Comparisons of the running times for the Lazy PRM vs. the Lazy LRM. The first three columns give the query times for the Lazy PRM, and the fourth column is the time to construct the initial graph for the Lazy PRM. The final column gives the Lazy LRM query time (there is no initial graph construction because the lattice is implicitly defined).

each resolution has low-discrepancy are presented in [44]. The basic idea for increasing the resolution in one step is to reduce the length of a generator by a factor of two, which doubles the number of points. We chose to use these lattices in our implementation, which is described next.

C. Comparing the Lazy PRM and Lazy LRM

We now present an implementation of the Lazy LRM. The primary difference with respect to the method in [6] is that no initial roadmap is explicitly constructed: it is defined *implicitly* by the rules of a chosen lattice for fixed d and N. We also chose to run the A^* algorithm only once, and performed collision detection during the search. We had first implemented the iterative search and deletion scheme described in [6], but found it to be much less efficient for our computed examples (for both the lattice-based and pseudo-random Lazy roadmaps). The implementation is in LEDA/C++, and uses the PQP collision detection package from the Univ. of North Carolina. We used values of $\mathbf{z}$ given in the appendix of [44].

We performed dozens of experiments on each of several examples. Three of these examples are displayed in Figures 6, 9, and 10. These represent 3D environments that contain a 6-DOF robot. The rotation portion of the C-space is parameterized using yaw-pitch-roll angles. We compare our Lazy LRM implementation to our implementation of a (pseudo-random) Lazy PRM (note, however, that one can adjust many parameters that affect performance). For the Lazy PRM, we performed 25 trials on each example. The table in Figure 7 shows the minimum, maximum, and average number of nodes for the Lazy PRM in the first three columns. The final column shows the number of nodes used by the Lazy LRM. Figure 8 compares computation times. The Lazy LRM shows dramatic performance improvements, primarily because it exploits the neighborhood structure of the lattice to avoid the precomputation required by the Lazy PRM. Furthermore, the number of nodes often appears to be in favor of the Lazy LRM, for reasons discussed in Section III.

V. Theoretical Considerations

Deterministic sampling enables the QRM and Lazy LRM to be *resolution complete*, in the sense that if it possible to solve the query, they will eventually solve it. This is in contrast to the original PRM and other randomized variants, which are only probabilistically complete [28] (the probability tends to one that a solution will be found as the number of samples grows to infinity).

Furthermore, we exploit dispersion bounds to characterize the set of configuration spaces that can be solved. This characterization is in terms of a parameter that measures the narrowest corridor width, in a manner similar to that of [3], [7], [21]. We define a cylindrical tube, and the "width" of $\mathcal{C}_{free}$ is expressed in terms of the largest possible cross section of the tube, over all possible queries. Measuring this parameter may be as difficult as the planning problem; however, the expression of planner performance in terms of parameters that are difficult to measure is common in randomized planner analysis [3], [21], [22], [27]. If a solution does not exist, the QRM is able to declare that either the solution path must travel through a narrow passage that has a width smaller than a specified value, or there is no solution. Such a result might be useful in applications because once the corridor is known to be narrower than a reasonable precision level, it is essentially equivalent to not having a solution. Unfortunately, our result is based on dispersion upper bounds that are not as tight as the observed performance in practice. The implication is that many more samples will be needed than are really necessary before the planner can make conclusions about the corridor width.

We assume in our PRM analysis that the radius parameter used to select neighbors in the NBHD function from Line 5 of Figure 3 is always sufficiently large. In theory, the radius can be made large enough so that an attempt

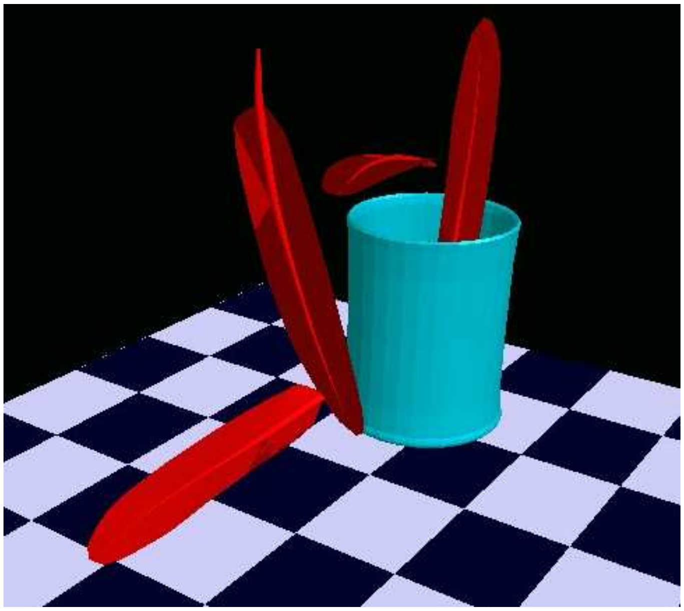


Fig. 9. Placing a feather (1184 triangles) into a cup (1632 triangles).

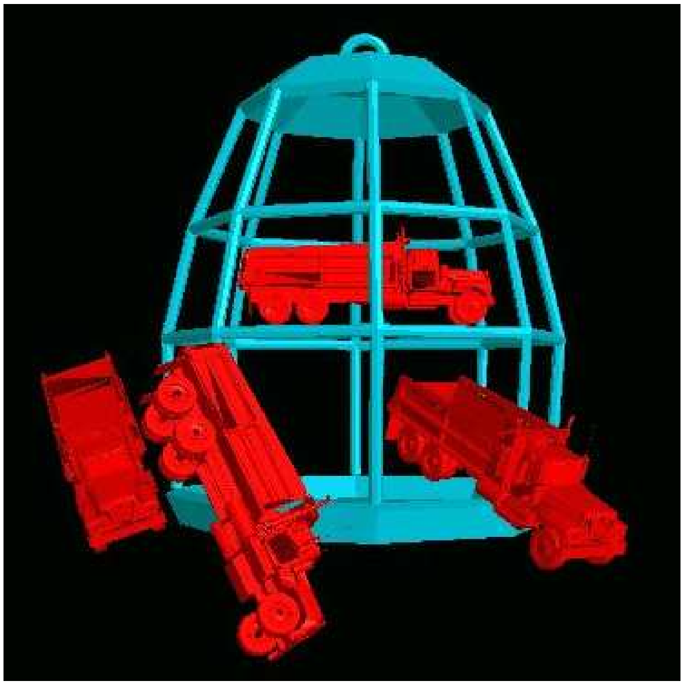


Fig. 10. Getting a truck (22284 triangles) out of a cage (1032 triangles).

can be made to connect every vertex to every other vertex. In practice, this becomes impractical; therefore, a smaller value is used.

Let $\gamma = \langle q_{init}, q_{goal} \rangle$ denote a *query.* The set, $\Gamma(\mathcal{C}_{free})$, of all queries in which $q_{init} \in \mathcal{C}_{free}$, $q_{goal} \in \mathcal{C}_{free}$, and $q_{init} \neq q_{goal}$, for a given $\mathcal{C}_{free}$ is called the *query space* of $\mathcal{C}_{free}$. Let $\Gamma_s(\mathcal{C}_{free}) \subseteq \Gamma(\mathcal{C}_{free})$ denote the set of all queries for which a solution exists.

Let $\mathcal{C}$ represent any d-dimensional configuration space, parameterized to yield $\mathcal{C} = [0,1]^d \backslash \sim$, in which $\backslash \sim$ denotes appropriate topological identifications along the boundary of the unit cube. Let Ψ represent the subset of the power set of $\mathcal{C}$ corresponding to all open subsets that can be constructed with algebraic constraints, as formulated in [28].

Let a *tube*, $\mathcal{B}$, represent an uncountable collection of balls of equal radius whose centers are generated by a continuous path, $\tau : [0,1] \to \mathcal{C}_{free}$. For each $s \in [0,1]$ there exists an open ball $B \in \mathcal{B}$ that is centered at $\tau(s)$ and has radius r, which is fixed for all $B \in \mathcal{B}$; let $B(s)$ denote the ball centered at $\tau(s)$. We call $2r$ the *width, $w(\mathcal{B})$, of the tube.*

Let $V(q)$ denote the set of all points visible from a set $q \in \mathcal{C}_{free}$ (i.e., for each $q' \in V(q)$, $\lambda q + (1-\lambda)q' \in \mathcal{C}_{free}$ for all $\lambda \in [0,1]$).

Suppose that a query $\gamma \in \Gamma_s(\mathcal{C}_{free})$ is given. Among all possible tubes, let $\mathcal{B}(\gamma)$ denote the tube with the largest width such that $B(0) \subset V(q_{init})$ and $B(1) \subset V(q_{goal})$. In other words, the entire first ball is visible from q_{init}, and the entire last ball is visible from q_{goal}. Denote this largest-width tube as the $\mathcal{B}(\gamma)$, and call its width the *width, $w(\gamma)$, of the query.* For any query $\gamma \in \Gamma(\mathcal{C}_{free}) \setminus \Gamma_s(\mathcal{C}_{free})$, we say that its width is zero because no tube exists.

Define the *width of $\mathcal{C}_{free}$* as

$$w(\mathcal{C}_{free}) = \inf_{\gamma \in \Gamma_s(\mathcal{C}_{free})} w(\gamma). \tag{4}$$

Let $\Psi(x)$ for $x \in (0,\infty)$ denote the set of all $\mathcal{C}_{free} \in \Psi$ such that $w(\mathcal{C}_{free}) \geq x$. Intuitively, this can be considered as the set of problems for which the width of the narrowest corridor is at least x.

Suppose that the roadmap, G, is constructed for a particular $\mathcal{C}_{free}$. Then, the algorithm is said to be *complete for $\mathcal{C}_{free}$* if all queries in $\Gamma(\mathcal{C}_{free})$ are answered correctly in the query phase. A solution path must be reported if one exists; otherwise, failure is reported.

Our first two results establish the resolution completeness and complexity of all QRM-based algorithms (including the LRM introduced above). Therein, we only assume that sampling is accomplished using a set P of *low-dispersion points*, for which

$$\delta(P,\rho) < b(d) N^{-1/d},$$

where $b(d)$ is a constant that may depend on the dimension d. Propositions 1 through 3 below hold for any metric, provided that both tube width and dispersion are measured using the same metric, ρ. Propositions 4 through 6 hold for norms (which are all related by constants in subsets of $\mathbb{R}^d$) under the same provision.

Many such low-dispersion point sets and sequences exist, as introduced in Section II. For example, consider the Halton and Hammersley sequences generated by applying the first d or $d-1$ primes: p_i, with $p_1 = 2$, $p_2 = 3$, $p_3 = 5$, ... The Halton points satisfy [34], [35]

$$\begin{aligned} \delta(P,\ell^2) &< \sqrt{12 + \sum_{i=1}^{d} p_i}\; N^{-1/d}, \\ \delta(P,\ell^\infty) &< p_d\, N^{-1/d}, \end{aligned}$$

for the Euclidean and ℓ^∞ norms, respectively. The Hammersley points satisfy [34], [35]

$$\begin{aligned} \delta(P,\ell^2) &< \sqrt{20 + \sum_{i=1}^{d-1} p_i}\; N^{-1/d}, \\ \delta(P,\ell^\infty) &< (1 + p_{d-1})\, N^{-1/d}. \end{aligned}$$

The Euclidean constants can be improved by using the results in [34], however, we avoid any further refinements here. Improvements can also be obtained by using quasi-random samples for which tighter bounds exist [35], [37]. A grid constructed to achieve the Sukharev sampling criterion [45] yields $b(d) = \frac{1}{2}\sqrt{d}$ for ℓ^2 dispersion. If ℓ^∞ dispersion is considered, then $b(d)$ can even be made dimension independent. In this case, the Sukharev grid yields $b(d) \equiv \frac{1}{2}$, which is the tightest bound possible.

The following proposition characterizes, in terms of tube width, the set of $\mathcal{C}_{free}$ for which a QRM or LRM will correctly solve all queries after N samples are used.

Proposition 1 *After N iterations, the QRM is complete for all $\mathcal{C}_{free} \in \Psi(4b(d)N^{-1/d})$, in which N is the number of points, d is the dimension of $\mathcal{C}$, and $b(d)$ is a factor that depends on the sampling method.*

Proof: Suppose first that $\mathcal{C} = [0,1]^d$ (ignoring any identifications). Assume that $\mathcal{C}_{free} \in \Psi(2b(d)N^{-1/d})$. To show completeness, we establish that for any solvable query, a solution path will be found; let $\gamma \in \Gamma_s(\mathcal{C}_{free})$ be such a query. Because $\mathcal{C}_{free} \in \Psi(2b(d)N^{-1/d})$, there exists a tube, $\mathcal{B}$, of width at least $2b(d)N^{-1/d}$, such that $B(0) \subset V(q_{init})$ and $B(1) \subset V(q_{goal})$.

Let P denote the set of sample points, which is also the set of vertices in the roadmap, G. Each ball $B \in \mathcal{B}$ must contain at least one $q \in P$. This follows from the fact that N samples were generated, and it has been shown in [34] that for Euclidean dispersion, $\delta(P) < b(d)N^{-1/d}$. If any ball of radius $b(d)N^{-1/d}$ is empty, then the dispersion would violate this upper bound.

First, consider connecting q_{init} and q_{goal} to the roadmap. Since $B(0) \subset V(q_{init})$, all configurations found by linear interpolation between q_{init} and any point in $B(0)$ are collision free. Therefore, q_{init} will be connected to a configuration in P (either one contained in $B(0)$, or at least one in the same connected component of G as a configuration of P that lies in $B(0)$). Using a similar argument for $B(1)$, q_{goal} will also be connected to a roadmap vertex.

It finally remains to show that there exists a path in G between the two configurations in P to which q_{init} and q_{goal} are connected. Consider the balls of $\mathcal{B}$ as parameterized using $B(s)$ for $s \in [0,1]$. We construct a sequence, $q_0, \ldots, q_{k-1}$ of k configurations as follows. Let q_0 be any element of $P \cap B(0)$. Let $s_1 \in [0,1]$ denote the last point at which the ball $B(s)$ contains q_0, by starting with $B(0)$ and increasing s continuously. Let q_1 be any element of $P \cap B(s_1) \setminus \{q_0\}$. Note that the $B(s_1)$ must contain at least two points in P because q_0 lies on its boundary. Inductively, let q_i be any element of $P \cap B(s_i) \setminus \{q_{i-1}\}$, where s_i is the first point at which $B(s)$ does not contain q_{i-1}. Note that the induction is finite, and let q_{k-1} denote the final configuration in the sequence.

We argue that there must exist a path in G between each pair, q_i, q_{i+1}, of configurations for $i \in \{0, \ldots, k-2\}$. The point q_i must lie on the boundary of $B(s_{i+1})$; therefore, $B(s_{i+1})$ contains two points of P. Furthermore, all points between q_i and q_{i+1} via linear interpolation must be collision free. The algorithm in Figure 3 would either have produced an edge between them, or failed to because both were already part of the same connected component of G. Either way, there exists a path in G between q_i and q_{i+1}. By applying this for each configuration in the sequence, there exists a path in G between q_0 and q_{k-1}. Furthermore, q_0 is connected to q_{init}, and q_{k-1} is connected to q_{goal}. Therefore, the query is correctly answered by returning a solution path.

We now turn to the case in which $\mathcal{C} = [0,1]^d \backslash \sim$, in which $\sim$ denotes boundary identifications needed to appropriately reflect the topology of transformation groups that arise in motion planning: S^1, P^3, etc. For the dispersion measurements in $[0,1]^d$, balls near the boundary have to be contained entirely inside the unit cube. Once identifications are considered, some balls are allowed to overflow as long as their center lies in $[0,1]^d$. Since $\delta(P) < b(d)N^{-1/d}$ for $[0,1]^d$, these overflowing empty balls cannot have radius larger than $2b(d)N^{-1/d}$. Thus, the dispersion in the part of the proof for $[0,1]^d$ is simply scaled by two for the case of $[0,1]^d \backslash \sim$ by assuming $\mathcal{C}_{free} \in \Psi(4b(d)N^{-1/d})$ in the first step, which establishes the proposition. ■

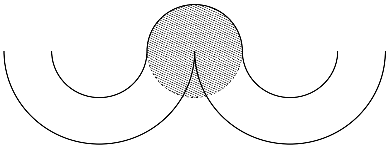


Fig. 11. A narrow corridor in $\mathcal{C}_{free}$, used in the proof of Proposition 3.

The previous proposition can be reworked to bound the width of the query:

Proposition 2 *After N iterations, the QRM for a query γ either reports a solution path or correctly declares that one of the following is true: there is no solution path, or $w(\gamma) < 4b(d)N^{-1/d}$.*

Proof: This follows directly from Proposition 1. Since the QRM is complete for $\Psi(4b(d)N^{-1/d})$, if no solution is found after N iterations, then $w(\mathcal{C}_{free}) < 4b(d)N^{-1/d}$ and $w(\gamma) < 4b(d)N^{-1/d}$. ■

The next proposition indicates that if the dispersion is at least δ, then it is possible for a PRM-based planner to miss solutions in corridors of width δ.

Proposition 3 *For any sample set, P, that has dispersion at least δ, no roadmap constructed using the algorithm in Figure 3 can be complete for $\Psi \setminus \Psi(\delta)$.*

Proof: We argue that completeness is lost by producing a $\mathcal{C}_{free}$ and query $\gamma \in \Gamma_s(\mathcal{C}_{free})$ that will be answered incorrectly. If the dispersion is δ, then there exists a ball, $B \subset [0,1]^d$ with radius δ such that $P \cap B = \emptyset$. Consider the narrow corridor in $\mathcal{C}_{free}$ that is shown in Figure 11. Assume that a no tube greater than width δ can be placed in the corridor. If $\mathcal{C}_{free}$ is chosen so that B is located as shown in the shaded area, then there will be no path in G that traverses the corridor. For any point on one side of the corridor outside of B, the straight-line path to any point on the other side of the corridor outside of B will intersect $\mathcal{C} \setminus \mathcal{C}_{free}$. Thus, a solution path will not be found. ■

From this the next proposition follows, which establishes that any PRM approach will require an exponential number of samples. It is assumed that the sampling scheme generates samples independently of the obstacle region.

Proposition 4 *Under any sampling scheme (randomized or deterministic), a roadmap requires a number of samples exponential in dimension, d, to be complete for $\Psi(\delta)$.*

Proof: This follows immediately from Proposition 3 and the Sukharev sampling criterion [45]. ■

We now consider asymptotic bounds for the QRM and PRM. The next proposition indicates that the QRM does the best possible, up to a constant of proportionality.

Proposition 5 *The number of samples required by the QRM and Lazy LRM to be complete for $\Psi(\delta)$ is asymp-*

totically optimal.

Proof: By Proposition 3, to be complete for $\Psi(\delta)$, the dispersion must be less than δ. Thus, the goal of the PRM algorithm should be to reduce δ using as few samples as possible. The Hammersley-Halton sequences achieve the best possible asymptotic dispersion [34]; therefore, the number of samples used in the QRM is asymptotically optimal. For the Lazy LRM, asymptotic optimality can be obtained by using the Sukharev grid [45]. ■

The following proposition gives some indication that random sampling does not yield the best possible asymptotic convergence in the PRM.

Proposition 6 *For a fixed d, the PRM with random sampling requires* $O((\log N)^{\frac{1}{d}})$ *times as many samples (with probability one) as the QRM to achieve the same* ℓ^∞ *dispersion.*

Proof: It was shown by Deheuvels [35] that ℓ^∞ dispersion for random samples is $O((\log N)^{\frac{1}{d}} N^{-\frac{1}{d}})$ with probability one. The asymptotic dispersion for Hammersley-Halton sequences is optimal by reaching the Sukharev sampling criterion, $O(N^{-\frac{1}{d}})$. The factor difference between the two is $O((\log N)^{\frac{1}{d}})$. ■

It is important to note that there is a well-known gap between the theoretical bounds for quasi-random sets and their good behavior observed in practice. If d is large and N is modest in size, then the theoretical results derived here will not be useful in practice. Nevertheless, low-discrepancy points have still shown better performance than random samples in these exact situations [47]. See Figure 12; it is interesting to note that in 2 dimensions, $3(195)^{-1/2} \approx (464)^{-1/4}$ (which uses the discrepancy upper bound on dispersion for uniform random points; the constants in the a.s. bound are not known to us at this time). In the other dimensions, N is (sometimes many orders of magnitude) too small for the theoretical advantages of the quasi-random points to take effect. Nevertheless, we empirically observe similar or better performance in terms of the number of nodes required to generate successful plans, as indicated in Sections III and IV. One way to improve the gap is to use low-discrepancy sets that have lower proven bounds than the Hammersley-Halton sets [37].

VI. Discussion: Where are we Going?

Based on the proposed planners in Sections III and IV we conclude that replacing pseudo-random sampling with deterministic sampling offers both practical and theoretical advantages in the PRM context. Based on our experiments, quasi-random samples appear to offer performance improvements similar to those observed in other fields where Monte Carlo methods were replaced by quasi-Monte Carlo methods. The regular neighborhood structure of quasi-random lattices led to further performance benefits, in the context of lazy evaluation. We emphasize the difficulty, however, in providing conclusive experimental comparisons, given that there is no practical way to represent the distribution of problems on which these algorithms will be applied. Also, it is hard to compare deterministic, predictable methods to randomized methods, which yield varying results in multiple executions. We do, however, obtain deterministic converge analysis. Our QRM and Lazy LRM planners can determine whether or not a solution exists at a specified resolution, which is not possible with their randomized counterparts.

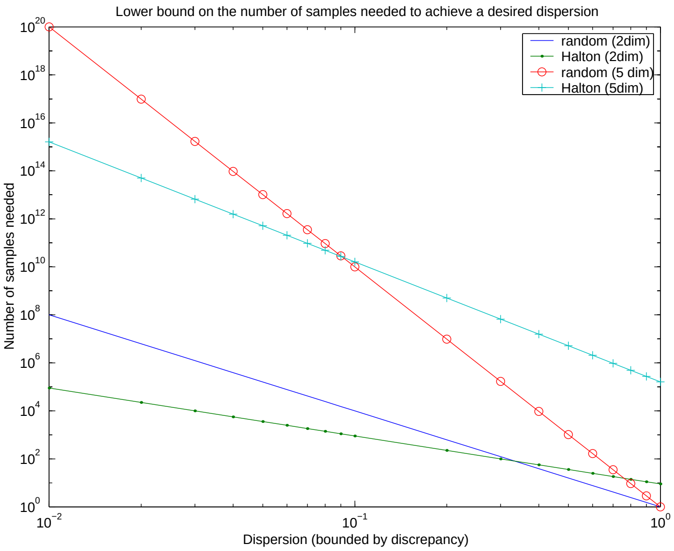


Fig. 12. Comparison of number of points needed to achieve a desired dispersion, from discrepancy bounds.

Note that we do not claim that our methods here are the fastest available. Our goal was simply to illustrate the power of deterministic sampling techniques in precisely the place where randomization is currently believed to be of great value. Many exciting directions for future research and discussion exist, especially in deterministic analysis of derandomized path planning algorithms and in considering possible derandomized versions of other randomized path planning algorithms, both PRM variants and others.

We have not yet considered the implications of the C-space topology on the sampling criteria. One of our motivations for choosing the lattices of [44] was their design for rapid integration of periodic functions (the integration occurs over a torus). Although the general theory encompasses arbitrary measures in topological spaces [36], well-known sample sets such as the Hammersley-Halton sequences are designed for low-discrepancy over $[0,1]^d$. There are exceptions, such as the low-dispersion sequence given for a d-dimensional torus in [12] (page 115). If the boundary identifications are taken into account, it should be possible to take advantage of sample sets that have even lower discrepancy. The problem of designing low-discrepancy samples for different topological spaces is yet to be investigated in the context of motion planning.

We believe the implications of our work put a new twist on the understanding of various path planning algorithms. For a fresh perspective, consider the following methods in light of our work:

- **Grid search:** Modern planning algorithms have attempted to improve on well-known grid-based search methods in which the C-space is partitioned into a grid. If collision checking is performed during search, this classic approach can be considered as a Lazy LRM in which the lattice generators are simply the grid directions. A^* search is typically used, but any graph search method is applicable.

- **Potential field methods:** Both randomized and deterministic potential field methods attempt to improve search by using heuristics to escape local minima [4], [15], [24]. For the randomized potential field planner in [4], the search is even performed over an implicit grid. Thus, it can be considered as a Lazy LRM on a grid lattice, with A^* search replaced by the potential field method.
- **PRM:** The basic PRM proposed the use of substantial preprocessing to construct a roadmap based on random sampling in an attempt to reduce the exponential number of samples needed for a grid-based approach. Given the Sukharev sampling criterion, however, we know that an exponential number of samples is needed in any case. Furthermore, random sampling actually achieves slower theoretical convergence than deterministic sampling [35].
- **Lazy PRM:** The Lazy PRM provides a kind of missing link between classical grid search and the basic PRM. The spirit is much like classical search (assuming only one execution of the A^* algorithm), but occurs over a randomly-generated graph, instead of an implicitly defined lattice.
- **QRM:** This appears to an improvement over the PRM by merely using quasi-random samples. Based on sampling theory, we observe that one must generate an exponential number of samples whether the PRM uses pseudo-random or quasi-random sampling. The QRM performs asymptotically-optimal sampling. We could also construct a Lazy QRM, which we would expect to be an improvement over the Lazy PRM.
- **Lazy LRM:** We are back to a method that appears very similar to classical grid based search, except that the grid is generalized to a lattice (which is essentially a nonorthogonal grid). Yet surprisingly, the method is an improvement over the Lazy PRM (and we believe over a Lazy QRM), which is in turn an improvement to the PRM, which in turn was designed to avoid the pitfalls of grids. The main reason why the circular implication falters is that no sampling method can avoid the need for an exponential number of samples.
- **Other methods:** Of course many other planning approaches can be considered. Incremental searching methods, such as the randomized potential field planner [4], RRTConCon [30], or the planner in [22], can be considered as alternatives to A^* search in any roadmap approach. For example, the Lazy LRM could be searched using a randomized potential field planner instead of A^*.

Given these observations, what characteristics are important for sampling in high-dimensional spaces? We believe that the ability of a planner to vary the "resolution" of the sampling during execution is crucial. It must be able to solve easier queries early at a low level of resolution. Of course, this idea can be traced back to classic quad-trees and other multi-resolution ideas in planning (e.g. [14]). Methods such as bidirectional search based on Rapidly-exploring Random Trees [29], [30] allow solutions to be found at a low level of resolution before systematically exploring everything in the worst case. In this way, easier planning queries can be resolved quickly. In the sampling-based roadmap context, this means that interleaving roadmap construction and path queries is important. It might be possible to find a path with a roadmap that contains only a few samples. Upon failure, more samples should be added, and the query can be tried again. In this case, the planner can answer easier queries quickly before resorting to a high-level of resolution.

We also recognize the difficulty of naming recent path planners. The name "roadmap" was introduced in classical path planning [10], [28], [38] to represent a network of paths that are accessible from anywhere in $\mathcal{C}_{free}$, and the connectivity of the roadmap must preserve the $\mathcal{C}_{free}$ connectivity. The original PRM captures this spirit by converging to the correct connectivity and accessibility probabilistically. However, the use of grids and lattices in the Lazy approaches, or other approaches to achieve accessibility and connectivity, does not seem to capture the spirit of original roadmaps because the space is densely covered by uniform sampling. Given our progression of terminology, one can imagine moving from a PRM, to QRM, to LRM, and finally to grid roadmap. This is far from the economy of paths obtained by a method such as retraction [38]; it instead corresponds to a quantization effect. Also, single-query planners such as those in [4], [22], [30], [33] do not attempt to provide accessibility and connectivity; they perform variants of classical heuristic search to obtain a solution path for a single query, which is quite unlike the roadmap framework. Thus, as non-probabilistic, non-roadmap approaches show their advantages, the PRM term, and even the roadmap term in general, may be less useful for characterizing recent, practical approaches to motion planning.

Although this paper has indicated advantages of deterministic sampling methods, we do not claim that randomized approaches offer no advantages. For the PRM and Lazy PRM, it is straightforward to make a derandomized variant because the samples directly become vertices in the graph. In general, however, derandomization of algorithms is a challenging task that often leads to a much more complicated solution [11]. Also, randomization has been found useful for overcoming uncertainties and improving robustness in contexts such as manipulation planning and execution [13]. Significant work remains to evaluate tradeoffs between randomization and determinism for other motion planning algorithms and for robotics in general.

Acknowledgments

We are grateful for the funding provided in part by NASA grant NAG2-1333 (Branicky) and NSF CAREER Award IRI-9875304 (LaValle). Steve LaValle thanks Robert Bohlin, Francesco Bullo, Blake Hannaford, and Shai Sachs for helpful discussions.

References

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

[2] N. M. Amato and Y. Wu. A randomized roadmap method for path and manipulation planning. In *IEEE Int. Conf. Robot. & Autom.*, pages 113–120, 1996.

[3] J. Barraquand, L. Kavraki, J.-C. Latombe, T.-Y. Li, R. Motwani, and P. Raghavan. A random sampling scheme for robot path planning. In G. Giralt and G. Hirzinger, editors, *Proc. of the 7th International Symposium on Robotics Research*, pages 249–264. Springer, New York, NY, 1996.

[4] J. Barraquand and J.-C. Latombe. A Monte-Carlo algorithm for path planning with many degrees of freedom. In *IEEE Int. Conf. Robot. & Autom.*, pages 1712–1717, 1990.

[5] R. Bohlin. Path planning in practice; lazy evaluation on a multi-resolution grid. In *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, 2001.

[6] R. Bohlin and L. Kavraki. Path planning using lazy PRM. In *IEEE Int. Conf. Robot. & Autom.*, 2000.

[7] R. Bohlin and L. Kavraki. A randomized algorithm for robot path planning based on lazy evaluation. In J. Reif S. Rajasekaran, P. Pardalos and J. Rolim, editors, *Handbook on Randomized Computation*. Kluwer Academic, 2001.

[8] V. Boor, N. H. Overmars, and A. F. van der Stappen. The Gaussian sampling strategy for probabilistic roadmap planners. In *IEEE Int. Conf. Robot. & Autom.*, pages 1018–1023, 1999.

[9] M. Branicky, S. M. LaValle, K. Olson, and L. Yang. Quasi-randomized path planning. In *IEEE Int. Conf. Robot. & Autom.*, pages 1481–1487, 2001.

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

[11] B. Chazelle. *The Discrepancy Method*. Cambridge University Press, 2000.

[12] J. H. Conway and N. J. A. Sloane. *Sphere Packings, Lattices, and Groups*. Springer-Verlag, Berlin, 1988.

[13] M. A. Erdmann. *On Probabilistic Strategies for Robot Tasks*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1989.

[14] B. Faverjon. Obstacle avoidance using an octree in the configuration space of a manipulator. In *IEEE Int. Conf. Robot. & Autom.*, pages 504–512, 1984.

[15] B. Faverjon and P. Tournassoud. A local based method for path planning of manipulators with a high number of degrees of freedom. In *IEEE Int. Conf. Robot. & Autom.*, pages 1152–1159, 1987.

[16] J. H. Halton. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numer. Math.*, 2:84–90, 1960.

[17] J. M. Hammersley. Monte-Carlo methods for solving multivariable problems. *Ann. New York Acad. Sci.*, 86:844–874, 1960.

[18] B. Hannaford. Resolution-first scanning of multi-dimensional spaces. *CVGIP: Graphical Models and Image Processing*, 55:359–369, 1993.

[19] F. J. Hickernell. Lattice rules: How well do they measure up? In P. Bickel, editor, *Random and Quasi-Random Point Sets*, pages 109–166. Springer-Verlag, Berlin, 1998.

[20] C. Holleman and L. E. Kavraki. A framework for using the workspace medial axis in PRM planners. In *IEEE Int. Conf. Robot. & Autom.*, pages 1408–1413, 2000.

[21] D. Hsu, L. E. Kavraki, J.-C. Latombe, R. Motwani, and S. Sorkin. On finding narrow passages with probabilistic roadmap planners. In et al. P. Agarwal, editor, *Robotics: The Algorithmic Perspective*, pages 141–154. A.K. Peters, Wellesley, MA, 1998.

[22] D. Hsu, J.-C. Latombe, and R. Motwani. Path planning in expansive configuration spaces. *Int. J. Comput. Geom. & Appl.*, 4:495–512, 1999.

[23] Y. K. Hwang and N. Ahuja. Gross motion planning–A survey. *ACM Computing Surveys*, 24(3):219–291, September 1992.

[24] Y. K. Hwang and N. Ahuja. A potential field approach to path planning. *IEEE Trans. Robot. & Autom.*, 8(1):23–32, February 1992.

[25] L. Kavraki and J.-C. Latombe. Randomized preprocessing of configuration space for path planning. In *IEEE Int. Conf. Robot. & Autom.*, pages 2138–2139, 1994.

[26] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. & Autom.*, 12(4):566–580, June 1996.

[27] F. Lamiraux and J.-P. Laumond. On the expected complexity of random path planning. In *IEEE Int. Conf. Robot. & Autom.*, pages 3306–3311, 1996.

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

[29] S. M. LaValle. Rapidly-exploring random trees: A new tool for path planning. TR 98-11, Computer Science Dept., Iowa State University, Oct. 1998.

[30] S. M. LaValle and J. J. Kuffner. Rapidly-exploring random trees: Progress and prospects. In B. R. Donald, K. M. Lynch, and D. Rus, editors, *Algorithmic and Computational Robotics: New Directions*, pages 293–308. A K Peters, Wellesley, MA, 2001.

[31] P. Leven and s. Hutchinson. Toward real-time path planning in changing environments. In *Workshop on the Algorithmic Foundations of Robotics*, pages 393–406, 2000.

[32] J. Matousek. *Geometric Discrepancy*. Springer-Verlag, Berlin, 1999.

[33] E. Mazer, G. Talbi, J. M. Ahuactzin, and P. Bessière. The Ariadne's clew algorithm. In *Proc. Int. Conf. of Society of Adaptive Behavior*, Honolulu, 1992.

[34] R. A. Mitchell. Error estimates arising from certain pseudorandom sequences in a quasi-random search method. *Mathematics of Computation*, 55(191):289–297, 1990.

[35] H. Niederreiter. *Random Number Generation and Quasi-Monte-Carlo Methods*. Society for Industrial and Applied Mathematics, Philadelphia, USA, 1992.

[36] H. Niederreiter and L. Kuipers. *Uniform Distribution of Sequences*. John Wiley and Sons, New York, 1974.

[37] H. Niederreiter and C. P. Xing. Nets, (t,s)-sequences, and algebraic geometry. In P. Hellekalek and G. Larcher, editors, *Random and Quasi-Random Point Sets, Lecture Notes in Statistics, Vol. 138*, pages 267–302. Springer-Verlag, Berlin, 1998.

[38] C. O'Dunlaing and C. K. Yap. A retraction method for planning the motion of a disc. *Journal of Algorithms*, 6:104–111, 1982.

[39] C. Pisula, K. Hoff, M. Lin, and D. Manoch. Randomized path planning for a rigid body based on hardware accelerated voronoi sampling. In *Proc. Workshop on Algorithmic Foundation of Robotics*, 2000.

[40] M. Rabin. Transaction protection by beacons. *J. Computation Systems Science*, 27(2):256–267, 1983.

[41] J. T. Schwartz and M. Sharir. On the piano movers' problem: II. General techniqies for computing topological properties of algebraic manifolds. *Communications on Pure and Applied Mathematics*, 36:345–398, 1983.

[42] P. Shirley. Discrepancy as a quality measure for sample distributions. In *Proc. Eurographics*, pages 183–194, Vienna, 1991. Elsevier Science.

[43] T. Simeon, J.-P. Laumond., and C. Nissoux. Visibility-based probabilistic roadmaps for motion planning. *Advanced Robotics Journal*, 14(6):477–494, 2000.

[44] I. H. Sloan and S. Joe. *Lattice Methods for Multiple Integration*. Oxford Science Publications, Englewood Cliffs, NJ, 1990.

[45] A. G. Sukharev. Optimal strategies of the search for an extremum. *U.S.S.R. Computational Mathematics and Mathematical Physics*, 11(4), 1971. Translated from Russian, *Zh. Vychisl. Mat. i Mat. Fiz.*, 11(4):910–924, 1971.

[46] S. Tezuka. *Uniform Random Numbers: Theory and Practice*. Kluwer Academic Press, Boston, 1995.

[47] J. F. Traub and A. G. Werschulz. *Complexity and Information*. Cambridge University Press, Cambridge, 1998.

[48] J. G. van der Corput. Verteilungsfunktionen I. *Akad. Wetensch.*, 38:813–821, 1935.

[49] H. Weyl. Über die Gleichverteilung von Zahlen mod Eins. *Math. Ann.*, 77:313–352, 1916.

[50] S. A. Wilmarth, N. M. Amato, and P. F. Stiller. MAPRM: A probabilistic roadmap planner with sampling on the medial axis of the free space. In *IEEE Int. Conf. Robot. & Autom.*, pages 1024–1031, 1999.