On the Relationship Between Classical Grid Search and Probabilistic Roadmaps

Steven M. LaValle* Michael S. Branicky† Stephen R. Lindemann‡

Abstract

We present, implement, and analyze a spectrum of closely-related planners, designed to gain insight into the relationship between classical grid search and probabilistic roadmaps (PRMs). Building on the quasi-Monte Carlo sampling literature, we have developed deterministic variants of the PRM that use low-discrepancy and low-dispersion samples, including lattices. Classical grid search is extended using subsampling for collision detection and also the dispersion-optimal Sukharev grid, which can be considered as a kind of lattice-based roadmap to complete the spectrum. Our experimental results show that the deterministic variants of the PRM offer performance advantages in comparison to the original, multiple-query PRM and the single-query, Lazy PRM. Surprisingly, even some forms of grid search yield performance that is comparable to the original PRM. Our theoretical analysis shows that all of our deterministic PRM variants are resolution complete and achieve the best possible asymptotic convergence rate, which is shown to be superior to that obtained by random sampling. Thus, in surprising contrast to recent trends, there is both experimental and theoretical evidence that the randomization used in the original PRM is not advantageous.

1 Introduction

The purpose of this paper is to provide insight into fundamental questions that arise in the development of sampling-based motion planning algorithms. What factors lead to good computational performance in practice? Is randomization really important for breaking the curse of dimensionality? In recent times it has been widely believed that randomization is a necessary component for obtaining algorithms that solve high-degree-of-freedom problems. In practice, implementations have used pseudo-random number generators as a kind of "black box" that produces deterministic sequences that mimic the behavior of an elusive, true random source. Given that such sequences are deterministic, several more questions come to mind. Is is possible to open the black box, and design sequences that are better tailored to motion planning? Is it possible to preserve some of the desirable properties of pseudo-random sequences while adding new properties that are even more beneficial in motion planning? By recognizing that nearly all machine implementations of random numbers produce deterministic sequences, is it possible to design sequences that yield deterministic guarantees while performing at least as well as pseudo-random sequences in practice?

Such questions have led us to develop a spectrum of sampling-based planners. At one end of the spectrum sits classical grid search, which is commonly accepted as naive because the number of required samples grows exponentially in dimension, assuming the number of points per axis is held fixed. At the other end

*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

†M.S. Branicky is with the Dept. of Electrical Engineering and Computer Science, Case Western Reserve University, Cleveland, OH 44106 USA. E-mail: mb@cwru.edu

‡S.R. Lindemann is with the Dept. of Computer Science, University of Illinois, Urbana, IL 61801 USA. E-mail: slindema@uiuc.edu

of the spectrum sits the original probabilistic roadmap (PRM), which was "primarily developed for robots with many dofs" [29]. The present paper builds on our previous work [9], in which well-known deterministic sampling techniques were shown to offer advantages as replacements to pseudo-random sampling in the PRM context. We have chosen the original PRM because it eases the comparison: the samples from the pseudo-random number generator appear directly as vertices in the roadmap (except those in collision). Along the spectrum, we develop a sequence of planners that are minor variants of grids and PRMs, and utilize sampling techniques that have appeared in the quasi-Monte Carlo literature (see Section 3 for an overview). Using the spectrum, we provide a clear understanding of the relationship between various planners, ultimately relating grids to PRMs. Our theoretical analysis and experimental results lead to the remarkable conclusion that deterministic sampling methods, including some forms of grid search, are superior to the original and Lazy PRMs. One excellent idea in the original PRM is to allow two different sampling "resolutions": one is for sampling the configuration space, and the other is for careful collision checking (called the local planner phase). In the original PRM, this means that a *local planner* is used to connect vertices; in a grid, this means that few points-per-axis are needed, as long as high-resolution collision checking is performed to connect neighbors.

It might appear that grid search is doomed to failure because if we fix k points-per-axis, $N = k^d$ points are needed in d dimensions. Using pseudo-random sampling, the basis of the original PRM, it appears that N is chosen independently of dimension. By defining a consistent quality measure for comparisons, it can be seen, however, that all sampling methods are doomed to the same kind of failure. Section 3 defines the notion of *dispersion*, which can informally can be considered as the radius of the largest empty ball that can be placed in the space after the samples have been chosen. Keeping the points-per-axis fixed in a grid holds the dispersion fixed; one must do the same for pseudo-random sampling. It was shown in 1971 by Sukharev [50] that *any* sampling scheme will require an exponential number of points in dimension to maintain a fixed dispersion. Furthermore, due to results from multivariate order statistics [41], random samples yield worse dispersion than grids. Think about how a collection of n points chosen at random in $[0, 1]$ must leave a sizable gap in some location; if the largest gap (in one dimension, empty balls become intervals that contain no points) has length $1/(n+1)$ (the smallest possible), we would declare that the sequence is "too uniform" to be random!

Thus, random samples require some irregularity that is worse than pure uniformity, in the sense obtained by a grid. A collection of pseudo-random samples should have too many points in some places, and not enough in others (look ahead to Figure 2). If the samples are spread out too evenly, then statistical tests for randomness would fail. Is this required irregularity really important for generating good samples in motion planning? It should be the case that notions such as regularity and uniformity are preferred over the required irregularity of samples drawn from a uniform, random distribution. In another context, such as statistical simulation, the irregularity becomes important, but in motion planning it appears to be an unnecessary burden that complicates the determination of connectivity information. For example, it is especially unfortunate that the choice of pseudo-random samples prevents the efficient exploitation of regular neighborhood structure that exists in many deterministic sampling schemes, such as grids and lattices. It is also unfortunate that random sampling leads to large regions that are not covered by samples.

There are important qualifications to the assertions made in this paper:

- **Importance sampling is important.** Many recent probabilistic roadmap methods [1, 2, 8, 24, 35,

44, 47, 56] have demonstrated improved performance by concentrating samples in a nonuniform way, such as along C-space boundaries [2, 8], or the medial axis [24, 44, 56]. Our work can be viewed as complementary to importance sampling because there is no reason why deterministic samples cannot be used in the same way as random samples. Importance sampling in motion planning has been achieved by either *rejection* or *transformation* of the original samples. For methods based on rejection, we can immediately provide an alternative source of initial samples. An example of rejection appears in the Visibility PRM [47], in which samples are retained only if they can become *guards* or *connectors*. In this case, improvements in the Visibility PRM have already been observed by replacing random sampling with deterministic Halton points (which are covered in Section 3) [48]. Methods that involve transformations are significantly more complicated, especially because the transformations are based on heuristic parameters. Examples include the planner in [8], which transforms samples by constructing a Gaussian distribution, and [2], in which samples are perturbed onto the boundaries of the configuration space. In principle it is possible to transform uniform, deterministic sample sets into others that concentrate samples appropriately. We hope that this paper inspires ways to improve importance sampling, but this is beyond the current scope since such techniques would be specific to each planner.

- **A sampler does not have to be a "black box".** Although modest performance gains can be obtained by replacing the pseudo-random number generator with a deterministic "black box" that produces better samples, this should not be the only goal in future work. By carefully integrating the design of a sampler into a motion planner, we believe better approaches can be developed. For example, the original PRM requires neighbor searching to locate nearby samples; if the pseudo-random sampler is replaced by a regular lattice, the lattice could be used naively by performing the same kind of neighbor searching; however, a much faster algorithm may be obtained by exploiting the regularity of neighborhood structure.

- **We do not introduce the fastest planners.** Our intent is to gain insight into planning issues, as opposed to delivering the best planner. Therefore, we do not claim that the planners introduced in this paper are the fastest available, and do not perform comparisons with state-of-the-art planners.

- **Randomization is sometimes useful.** However, its value depends greatly on the paradigm within which it is used. For example, in decision theory, randomized (or mixed) strategies offer no advantages in games against nature; however, in some zero-sum games an improved value can be obtained in comparison to a deterministic (or pure) strategy. In the context of Canny's algorithm to compute roadmaps of semi-algebraic sets, the use of randomization reduces time complexity from $(n^k \log n)d^{O(k^4)}c^2$ to $(n^k \log^2 n)d^{O(k^2)}c^2$, in which n is the number of polynomials, d is their maximum degree, and k is the number of variables [10]. This improvement is small but not insignificant. Every paradigm of randomization warrants careful investigation. Other motion planning algorithms may be more challenging to derandomize. For example, randomization appears to reduce modeling requirements and improve robustness in manipulation planning [15]. As another example, with Rapidly-exploring Random Trees (RRTs) [33] the primary idea is Voronoi bias, which is obtained through random sampling without computation of Voronoi regions. There are many ways in which one could construct derandomized versions, but it is not yet conclusive that there are clear advantages to deterministic sampling in this context.

2 Grids and PRMs

This section introduces a few influential planners that will be discussed in detail in this paper, rather than providing a complete survey of techniques.

The path planning problem. Let $\mathcal{C}$ denote the configuration space (or *C-space*) of a d-DOF robot in a 2D or 3D world that contains static obstacles [37]. For convenience, assume that the configuration space is parameterized so that $\mathcal{C} = [0,1]^d \subset \mathbb{R}^d$, and some boundary points are identified to respect topology. Let $\mathcal{C}_{free}$ denote the set of all collision-free configurations. A path planning *query* is a pair of configurations, $\langle q_{init}, q_{goal} \rangle$. The path planning problem is to find a continuous path, $\tau : [0,1] \rightarrow \mathcal{C}_{free}$ such that $\tau(0) = q_{init}$ and $\tau(1) = q_{goal}$. It is assumed to be computationally prohibitive to construct an explicit representation of $\mathcal{C}_{free}$; however, a *collision detection* algorithm is available, which can quickly indicate whether a given configuration lies in $\mathcal{C}_{free}$ (distance information could also be computed).

Classical grid-based search. Grid-based search is considered by many to be the most straightforward form of path planning. Once configuration space representations were introduced [37], it became clear that each component of $\mathcal{C}$ is quantized, and a d-dimensional bitmap representation can be precomputed by iterating a collision detector over all quantized configuration values. An example of this approach appears in [34], in which a bitmap is constructed by rasterizing the Minkowski sums of robot-obstacle pairs, at discretized rotation values. We refer to such methods as "classical" because once the bitmap is constructed, classical AI search techniques may be applied to find a solution. A neighborhood structure must be defined, such as the set of $2d$ neighbors for each interior element of the bitmap ("up", "down", "left", "right" in the case of $d = 2$). A given query is quantized, and the bitmap can then be treated as a graph which is searched using algorithms such as dynamic programming, A^*, best-first, or bidirectional search, to connect q_{init} to q_{goal}. In fact, the bitmap could also be searched using recent path planning methods that are based on incremental search, such as randomized potential fields [4], multiple heuristics [30], Ariadne's clew [40, 39], RRTs [33], or the planner in [26, 45]. In some of these works [30, 4], a lazy bitmap is actually used; collision checking is only performed as needed during the search. It is well-known that only *resolution completeness* can be obtained, and that for a fixed resolution, the number of samples (bitmap size) increases exponentially in d.

The original PRM for multiple queries. The Probabilistic Roadmap (PRM) was introduced in [29] as a way to overcome the well-known curse of dimensionality that exists in grid search; it is similar to earlier work by Glavina [19]. The primary philosophy behind the PRM was to perform substantial preprocessing so that *multiple queries* for the same environment could be handled efficiently. This is analogous to the bitmap precomputation in classical grid based search. 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 1 constructs a PRM with N vertices. In Line 3, a random configuration in $\mathcal{C}_{free}$ is found by repeatedly picking a random configuration until one is determined by a collision detection algorithm to be in $\mathcal{C}_{free}$ (in practice, of course, pseudo-random configurations are generated). The NBHD function in Line 4 is a range query in which all vertices within a specified distance of q are returned, sorted by distance from q. In Line 5, it is sometimes preferable to add an extra condition, such as (**not** G.same_component(q, v)) or G.vertex_degree$(q) < K$, for some fixed K (e.g., $K = 15$). There have been

```
BUILD_PRM
 1   G.init();
 2   for i = 1 to N
 3       q ←RAND_FREE_CONF(q);
 4       G.add_vertex(q);
 5       for each v ∈ NBHD(q,G)
 6           if CONNECT(q, v) then
 7               G.add_edge(q, v);
```

Figure 1: The preprocessing phase: build a PRM.

conflicting recommendations in the research community regarding which method is superior; therefore, for our comparisons we have implemented one variant that uses a fixed radius, and another that uses K nearest neighbors. Both methods also use connected component analysis and the union-find algorithm to avoid unnecessary connections. See Section 6 for results. The CONNECT function in Line 6 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 incrementally stepping along using a collision detection algorithm. A heuristic, node-enhancement phase, described in [29], is not considered here.

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. This is analogous to the problem of insufficient resolution in classical grid search.

The Lazy PRM for single queries. 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 [29] to the *single query* philosophy which was used in some earlier planners [4, 17, 40].

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 1 is that the condition in Line 6 is dropped, and Line 7 is executed every time. This allows the PRM to be constructed quickly; however, more burden is placed on 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. Alternatively, it might be preferable to run the search only once on the initial graph, while validating edges during the search (instead of waiting for a solution and then validating it) [9]. 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. In classical grid search, this philosophy implies that the bitmap is not precomputed; collision checking is performed only as needed during the search.

3 A Spectrum of Sampling Techniques

A first step toward constructing a spectrum of planners from the original PRM to classical grid search is to characterize a spectrum of sampling techniques to cover $\mathcal{C}$, from pseudo-random sequences to grids. A brief introduction to sampling is given here, based primarily on the *quasi-Monte Carlo* literature [38, 41, 51]. The general philosophy is to view sampling as a deterministic optimization problem in which points are selected to optimize a criterion that is relevant to a particular problem.

3.1 Sampling Criteria

Motivated by applications in numerical integration and optimization, the most common criteria are discrepancy and dispersion. Each of these measures the amount of uniformity in a collection of points. This is different from statistical tests that might be applied to points generated by a pseudo-random number generator. The criteria are designed to match the problem to which the points are applied, as opposed to simulation of "randomness."

Let $X = [0,1]^d \subset \mathbb{R}^d$ define a space over which to generate points. Let $P = \{p_0, \ldots, p_{N-1}\}$ denote a finite set of N points in X. Let $\mathcal{R}$ be a collection of subsets of X that is called a *range space*. With respect to a particular point set, P, and range space, $\mathcal{R}$, the *discrepancy* [55] is defined as

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

in which $|\cdot|$ applied to a finite set denotes its cardinality, and μ is Lebesgue measure. Each term in the supremum considers how well P can be used to estimate the volume of R. For example, if $\mu(R)$ is $1/5$, then we would hope that about $1/5$ of the points in P fall into R. The discrepancy measures the largest volume estimation error that can be obtained over all sets in $\mathcal{R}$.

In most cases, $\mathcal{R}$ is chosen as the set of all axis-aligned rectangular subsets. In this case, the range space will be denoted as $\mathcal{R}_{aar}$. This is motivated primarily by applications to numerical integration. In this case, the Koksma-Hlawka inequality expresses the convergence rate in terms of a constant for bounded variation (in the sense of Hardy-Krause) of the integrand, and the bounded variation is expressed in terms of rectangular partitions [41].

For applications to optimization, the notion of *dispersion* was developed:

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

in which ρ denotes any metric. Cases in which ρ is the Euclidean metric or ℓ_∞ will be referred to as *Euclidean dispersion* and ℓ^∞ *dispersion*, respectively. Dispersion can be considered as the radius of the largest ball that does not contain an element of P. This represents a natural choice for many optimization problems because the error can be related directly to a Lipschitz constant for the function. For example, in one dimension, the maximum error is proportional to the longest interval that does not contain a sample point.

Since a large empty ball contains substantial measure, there is a close relationship between discrepancy, which is measure-based, and dispersion, which is metric-based. For example, for any P,

$$\delta(P, \ell^\infty) \leq D(P, \mathcal{R}_{aar})^{1/d},$$

which means low-discrepancy implies low-dispersion. Note that the converse is not true. An axis-aligned grid yields high discrepancy because of alignments with the boundaries of sets in $\mathcal{R}_{aar}$, but the dispersion is very low.

3.2 Finite Point Sets and Infinite Sequences

There are two important sampling families: 1) finite point sets, and 2) infinite sequences. We will use the term *point set* to refer to the first form, and *sequence* to refer to the second. For a point set, the number, N, is specified in advance, and a set of N points is chosen that optimizes discrepancy or dispersion. As for any set, there is no notion of ordering in a point set; however, for a sequence, the ordering of points becomes crucial.

Each of the two forms of sampling is suited for a different kind of algorithm. A grid is the most common example of a point set. Samples drawn using a pseudo-random number generator are a typical example of a sequence. If the appropriate N is known in advance, then a point set should be used. If the algorithm cannot determine N in advance, but is able to incrementally add more and more points, then a sequence is usually more appropriate. The availability of a sequence from pseudo-random number generators represents a major part of the past appeal of using randomization in motion planning; however, many other sequences are available.

Analysis of discrepancy or dispersion in the quasi-Monte Carlo literature is usually asymptotic [41]. The vast majority of sample construction algorithms are so simple and efficient that complexity analysis of the algorithm itself is not performed. The definition of asymptotic analysis for a sample sequence is straightforward; however, a point set must be converted into a sequence to perform the analysis. In this case, asymptotic analysis is actually performed over a sequence of point sets. Let P_N denote a point set in the sequence. For asymptotic analysis of point sets, P_{N+1} does not have to contain any of the points from P_N. For sequences, strict containment must always occur; each iteration adds a single point to the previous ones. Therefore, point sets often appear better than sequences in theory, but they might pose difficulties in practice because they may require the specification of N. A *low-discrepancy* sequence or point set is one that yields the best-possible asymptotic discrepancy, which is $O(N^{-1}\log^d N)$ for infinite sequences and $O(N^{-1}\log^{d-1} N)$ for finite point sets, under the common range space $\mathcal{R}_{aar}$. Note that the second bound is better, for reasons mentioned in the previous paragraph. A *low-dispersion* sequence or point set is one that obtains the lowest-possible asymptotic dispersion. For any ℓ^p metric, both sequences and point sets can achieve $O(N^{-1/d})$. The remainder of this section provides an overview of low-discrepancy and low-dispersion sampling.

In the bounds above, N, is the variable considered in the asymptotic analysis (i.e., the limit is taken with respect to N). Therefore, the dimension, d, is treated as a constant. Thus, if there is a multiplicative factor which is a function of d, say $b(d)$, then it is not shown in an expression such as $O(N^{-1/d})$. However, it is important to understand that for many well-known cases, $b(d)$, and ultimately the dispersion or discrepancy, may increase dramatically with respect to dimension. Therefore, a substantial amount of recent research has been conducted to improve sample quality as the dimension increases. These issues will become important in our theoretical analysis, presented in Section 5.

3.3 Low-Discrepancy Sampling

Due to the fundamental importance of numerical integration, and the intricate link between discrepancy and integration error, most of the quasi-Monte Carlo literature has led to low-discrepancy sequences and point sets. Although motion planning is quite different from integration, it is worth evaluating these carefully-constructed and well-analyzed samples. Their potential use in motion planning is no less reasonable than using pseudo-random sequences, which were also designed with a different intention in mind.

Low-discrepancy sampling methods can be divided into three categories: 1) Halton/Hammersley sampling, 2) (t,s)-sequences and (t,m,s)-nets, and 3) lattices. The first category represents one of the earliest methods, based on the original ideas of van der Corput [52], who defined a low-discrepancy sequence of points in $[0, 1]$. Consider a binary representation of points in $[0, 1]$. A one-dimensional "grid" can be made with resolution 8, then samples are taken at: 0.000, 0.001, 0.010, 0.011, 0.100, etc. The problem with using such points in this order is that they scan from left to right, as opposed to moving around in a manner similar to that of pseudo-random sequences. The *van der Corput* sequence simply takes the binary counting above and reverses the order of the bits. During the original scan, the least significant bit alternates in every step, but this only yields a small change in value. By reversing bit order, the change is maximized, causing the coverage to be nearly uniform at every point in the sequence. After bit reversal, the sequence is: 0.000, 0.100, 0.010, 0.110, 0.001, 0.101, 0.011, 0.111. An infinite sequence is constructed by using reversed-bit representations of higher binary numbers. The next eight samples are obtained by reversing binary representations of 8 through 15.

The *Halton sequence* is a d dimensional generalization that uses van der Corput sequences of d different bases, one for each coordinate [20]. First, choose d relatively prime integers $p_1, p_2, \ldots, p_d$ (usually the first d primes, $p_1 = 2$, $p_2 = 3$, ...). To construct the i^{th} sample, consider the digits of the base p representation for i in the reverse order (that is, write $i = a_0 + pa_1 + p^2a_2 + p^3a_3 + \ldots$, where each $a_j \in \{0, 1, \ldots, p\}$) and 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 i^{th} sample in the Halton sequence is

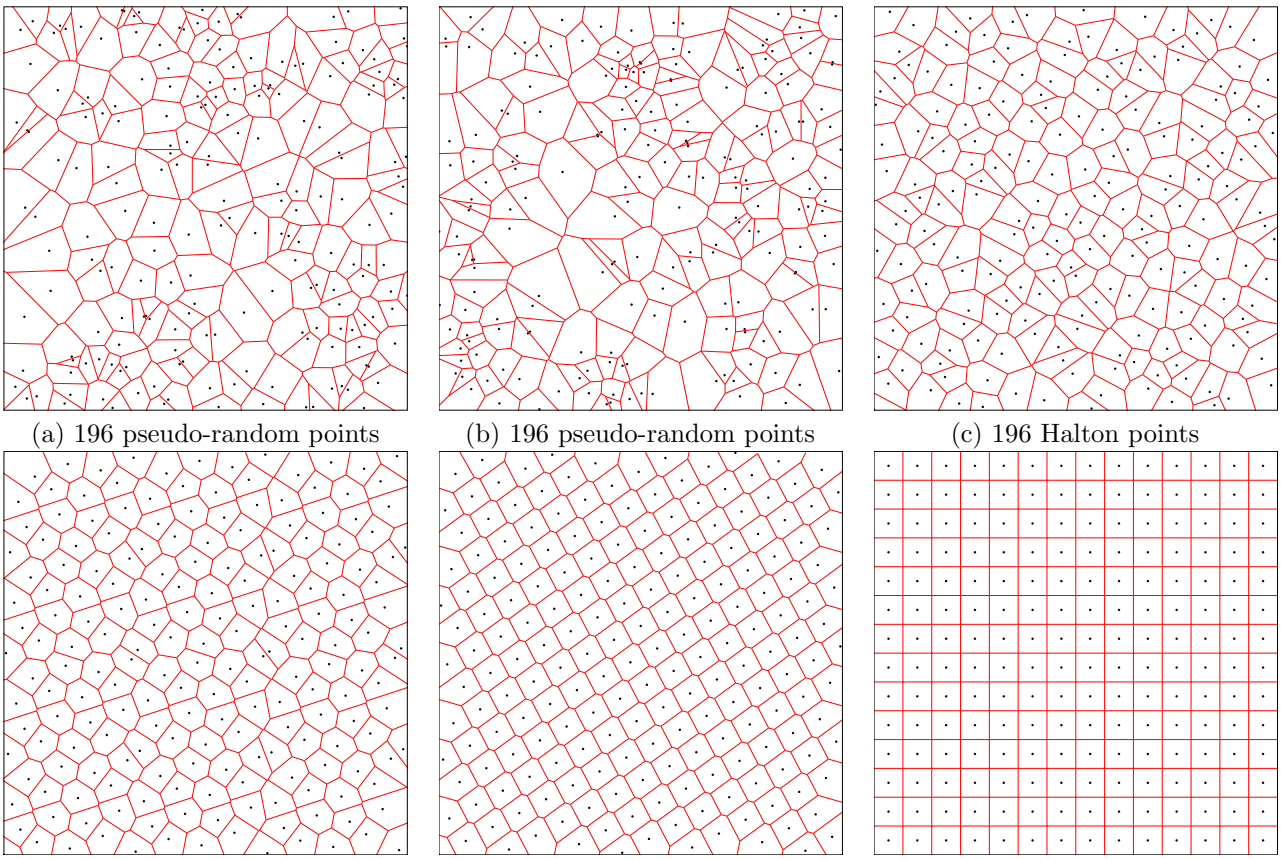
$$(r_{p_1}(i), r_{p_2}(i), \ldots, r_{p_d}(i)), \qquad i = 0, 1, 2, \ldots.$$

The *Hammersley* point set is an adaptation of the Halton sequence [21]. Using only $d - 1$ distinct primes, the i^{th} sample in a Hammersley point set with N elements is

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

The construction of Halton/Hammersley samples is simple and efficient, which has led to widespread application. However, the constant in their asymptotic analysis increases superexponentially with dimension [41].

Improved constants are obtained for sequences and finite points by using (t,s)-sequences, and (t,m,s)-nets, respectively [41]. The key idea is to enforce zero discrepancy over a particular subset of $\mathcal{R}_{aar}$ known as canonical rectangles, and all remaining ranges in $\mathcal{R}_{aar}$ will contribute small amounts to discrepancy. The most famous and widely-used (t,s)-sequences are Sobol' and Faure (see [41]). The Niederreiter-Xing



(a) 196 pseudo-random points (b) 196 pseudo-random points (c) 196 Halton points

(d) 196 Hammersley points (e) 196 lattice points (f) 196-point Sukharev grid

Figure 2: A sampling spectrum. The degree of uniformity in a sample set may be characterized informally as the amount of variation in the size and shape of Voronoi regions. Is the irregularity produced by pseudo-random sampling really advantageous in motion planning?

(t,s)-sequence has the best-known asymptotic constant, $(a/d)^d$, among all low-discrepancy sequences; in the expression, a is a small constant [42].

The third category is *lattices*, which can be considered as a generalization of grids that allows nonorthogonal axes [38, 49, 54]. As an example, consider Figure 2(e), which shows 196 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 i^{th} point according to $(\frac{i}{N}, \{i\alpha\})$, in which $\{\cdot\}$ denotes the fractional part of the real value (modulo-one arithmetic). In Figure 2(e), $\alpha = \frac{\sqrt{5}+1}{2}$, the Golden Ratio. 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 [38]. The i^{th} sample in the lattice is

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

Recent analysis shows that some lattice sets achieve asymptotic discrepancy that is very close to that of the best-known non-lattice sample sets [22, 51]. Thus, restricting the points to lie on a lattice seems to entail little or no loss in performance, but with the added benefit of a regular neighborhood structure that is useful for path planning. Historically, lattices have required the specification of N in advance, making them

examples of low-discrepancy point sets; however, there has been increasing interest in extensible lattices, which represent infinite sequences [23].

3.4 Low-Dispersion Sampling

Although dispersion has been given less consideration in the literature than discrepancy, it is more suitable for motion planning. Dispersion has been developed to bound optimization error; however, in a sampling-based motion planning algorithm, it can be used to ensure that any corridor of a certain width will contain sufficient samples. This will be shown in Section 5, which provides a connection between dispersion and motion planning that is similar to that between discrepancy and integration.

It turns out that many low-discrepancy sequences are also low-dispersion (asymptotically optimal) sequences. This is true of Halton and Hammersley points, which helps explain why their Voronoi regions are more regular than those of pseudo-random samples in Figure 2. However, the constant in the asymptotic analysis is usually high, which is the price one must pay for worrying about alignments with elements of $\mathcal{R}$. Since dispersion is not concerned with such alignments, one can do much better.

While grids are a terrible choice for minimizing discrepancy over $\mathcal{R}_{aar}$, it turns out that some grids represent the best that can be done for dispersion. Consider Figure 2 once again. The Euclidean dispersions for the sample sets are: (a) 0.1166, (b) 0.1327, (c) 0.0912, (d) 0.0714, (e) 0.0664, and (f) 0.0505. As the amount of regularity in the samples increases, the dispersion decreases.

Determining the optimal sequence or point set (given d and N) in terms of Euclidean dispersion is closely related to the notoriously challenging problem of sphere packing [12]. Fortunately, the situation is much simpler for ℓ^∞ dispersion. Even though this metric might seem less natural than the Euclidean metric, a sequence that has optimal asymptotic ℓ^∞ dispersion will also have optimal asymptotic Euclidean dispersion. Thus, there is at least some insensitivity to the metric. Note, however, that the implied constant in the asymptotic analysis will vary with the use of a different metric. For the case of a Euclidean metric, it is extremely challenging to find a collection of samples that optimizes the implied constant [12]; however, for the ℓ^∞ metric, the problem is much more manageable, as shown by Sukharev.

The *Sukharev sampling criterion* [50] is a lower bound which states that for any point set P,

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

Thus, to keep dispersion fixed, the number of samples required grows exponentially in dimension. For a grid, this means holding the points per axis fixed; however, this bounds applies to *any* sequence! It turns out that for a fixed N, if $N^{\frac{1}{d}}$ is an integer, k, then the *Sukharev grid* yields the best possible dispersion, which is precisely $\frac{1}{2}N^{-1/d}$. This was shown for $N = 196$ and $d = 2$ in Figure 2(f). The grid is constructed by partitioning $[0, 1]^d$ into N cubes of width $1/k$ so that a tiling of $k \times k \times \cdots \times k$ is obtained, and a sample is placed at the center of each cube.

The Sukharev grid represents a point set. Nongrid, low-dispersion infinite sequences exist that have $\frac{1}{\ln 4}$ as the constant in the asymptotic convergence rate [41]. Also, extensible versions of these grids have been developed, which yield an infinite sequence [36].

Classical Grid Search (CGS)	Subsampled Grid Search (SGS)	Lattice Roadmap (LRM)	Quasi-random Roadmap (QRM)	Probabilistic Roadmap (PRM)
∘Many points per axis ∘Prohibitive in high dimensions ∘Optimal dispersion ∘Poor discrepancy	∘Few points per axis ∘Scales to moderate dimensions ∘Optimal dispersion ∘Poor discrepancy	∘Non-orthogonal axes ∘Regular neighborhood structure ∘Optimal dispersion ∘Near-optimal discrepancy	∘Irregular neighborhood structure ∘Optimal dispersion but has high constants ∘Optimal discrepancy	∘Irregular neighborhood structure ∘Non-optimal dispersion ∘Non-optimal discrepancy

Figure 3: A spectrum of planners, from classical grid search to the original PRM. As supported by the experiments in Section 6 and the analysis in Section 5, the worst planners in the spectrum are on the ends. Classical grid search suffers from too many points per axis, and the PRM suffers from a poor distribution of samples and irregular neighborhood structure.

4 A Spectrum of Planners

This section builds on the material from Sections 2 and 3 to obtain a spectrum of planners that range from classical grid search to PRMs. Figure 3 provides a summary of the five general families of planners that will be described in this section.

4.1 Multiple-Query vs. Single-Query

Before constructing a spectrum of planners, it is important to first recognize that multiple-query and single-query versions of each can be constructed. In the PRM context, a single-query, Lazy PRM was constructed by building a roadmap that initially ignores obstacles, and then performs collision checking only during the search associated with a single query (refer back to Section 2).

Let *CGS* refer to *classical grid search*, as described in Section 2. Does there exist a choice between multiple queries and single queries for classical grid search? One can definitely construct grid-based analogs to the PRM and Lazy PRM. First, consider the configurations represented in the grid as a collection of samples over $\mathcal{C}$. The standard grid neighborhood structure (in two dimensions, 4-neighbors, 8-neighbors, etc.) immediately determines which edges will be attempted in this "roadmap". For *multiple-query CGS*, one simply precomputes a bitmap over $\mathcal{C}$ that represents the collision regions. The standard neighborhood structure is then used to immediately infer which edges are to be used in search. During the search phase, no collision detection needs to be performed, just as in the multiple-query PRM. For *single-query CGS*, the grid resolution is simply declared (and the grid is perhaps allocated in memory); however, collision checking is performed only during the search for a particular query (either the multi-iteration searching approach in [6], or the single-iteration approach in [9] can be used). Note that there is no initial construction of a roadmap as in the Lazy PRM because the grid samples already have a predefined neighborhood structure. This is a big advantage of grids and lattices over pseudo-random samples and other irregular samples, such as Halton points.

In any of the methods described below, note that multiple-query or lazy, single-query versions may be constructed.

4.2 Improving the Original PRM

Here, two variants of the PRM are considered, a *quasi-random roadmap*, QRM, and a *lattice roadmap*, LRM, by directly applying the low-discrepancy sampling techniques described in Section 3. The QRM and LRM

were introduced in [9]. Moving from right to left in Figure 3, the first step derandomizes the PRM by using a deterministic low-discrepancy sequence or point set, such as Halton, Hammersley, or Niederreiter-Xing. This results in the QRM, which provides asymptotically-optimal discrepancy, which removes the large uncovered regions shown in Figures 2(a) and 2(b). Experiments in Section 6 will show that this yields modest performance improvements over the original PRM, and Proposition 6 in Section 5 shows that random sampling requires $O((\log N)^{\frac{1}{d}})$ times as many samples to reach the same quality level as Halton points, in terms of dispersion. Recall from Section 3 that even though many of the low-discrepancy samples (including Halton and Hammersley) yield asymptotically-optimal dispersion, the implied constants may grow quickly with dimension. This problem can eventually be tackled by using samples specifically designed to optimize dispersion, as opposed to "abusing" sequences that were designed to optimize discrepancy and hoping to achieve low dispersion. Note, however, in some applications, vertical alignments may be problematic, in which case it might make sense to focus on both discrepancy and dispersion.

The QRM simply treats sampling as a black box, and applies a superior deterministic substitute in the place of a pseudo-random number generator. Modest performance gains can be obtained; however, because the neighborhood structure is irregular, costly neighbor searching must still be performed. In numerical integration, for which most low-discrepancy sequences were designed, this is usually not an important concern; however, in motion planning, building connectivity information is critical. This motivates the development of an LRM, which in many ways appears most appealing within the spectrum. Since a lattice is a nonorthogonal grid, the regular neighborhood structure can be exploited to reduce or remove the costs of finding nearest neighbors. This is particularly advantageous, as stated previously, for single-query approaches.

By using lattices, it is possible to obtain near-optimal discrepancy and optimal dispersion. Some samples might yield better discrepancy at the expense of higher constants with respect to dimension for dispersion. By using grids, discrepancy becomes poor, but very low constants can be obtained for dispersion. One drawback of lattices and grids might appear to be that the number of points, N, must be specified in advance. While this is true for most lattices, this is not necessary. For the lattices in [49], the number of points can be iteratively doubled, while still maintaining good discrepancy [9]. There is also work on extensible low-discrepancy lattices, in which points may be added one-by-one, which results in a infinite sequence that may serve as a direct substitute for pseudo-random number generators in PRM-like planners [53]. Note, however, that this substitute maintains nice lattice structure.

4.3 Improving Classical Grid Search

The primary problem with classical grid search is that too many points per axis (e.g., 100) are typically required. The original PRM appears better because it makes clever use of a local planner. This allows collision-checking to be performed with high precision, while vertices in the PRM coarsely sample the space. In classical grid search, collision detection and configuration space sampling were lumped together. It therefore makes sense to adapt the local planner idea from the PRM to grid search. Due to the regular structure of grids, it is easy to view this as subsampling. Assume that the sampling rate for collision checking is much higher than the grid resolution. If we consider a motion from one grid point, q, to an adjacent grid point, q', then collision checking must be performed along a sequence of points obtained by linear interpolation (respecting topology) between q and q'. This is just like the local planner in a PRM. The grid can now be interpreted as a kind of trellis of paths. A similar idea was applied long ago for grid search

in [13]. The advantage of this approach is that we might be able to solve a query using a low resolution for the grid, yet still be able to check for collisions at the required level of resolution. Good performance can be obtained if there are no narrow corridors in the configuration space. If q_{init} and/or q_{goal} do not lie on the grid points, then attempts can be made to connect them (again by interpolation) to a set of nearby grid points. We will refer to this extension as *subsampled grid search, SGS*.

One final point of concern is: where should the grid be placed? It seems reasonable to align the grid axes with the coordinate axes, but the translation remains to be chosen. Typically, the "origin" of a grid is at the coordinate origin (i.e, a sample appears at $(0, 0, \ldots, 0)$). Let k denote the number of points per axis, and let d denote the dimension. Suppose $k = 3$ and $d = 2$. A classical grid would place one point in the center and the rest along the boundary (assuming no topological identifications). The maximum ℓ^∞ distance possible from a point $x \in [0, 1]^d$ to a grid point is $1/4$ (in a unit cube). Now, consider the Sukharev grid, which provides optimal dispersion. In this case, the maximum distance is only $1/6$. Using only two points per axis ($k = 2$), the Sukharev grid yields a maximum distance of $1/4$, which is equivalent to that of a classical grid for which $k = 3$. These differences might appear small; however, for high-dimensional problems for which only low-resolution sampling is possible, the difference is dramatic. For example, suppose $d = 10$. In this case, a Sukharev grid with $2^{10} = 1024$ points provides the same quality coverage (in terms of dispersion) as a classical grid with $3^{10} = 59049$ points. Of course, as k becomes large, the improvement diminishes; however, for large d, it is impractical to make k large. Thus, the difference remains significant. Note also that the improvement diminishes in other topological spaces. In fact, on a toroidal manifold, the two grids are equivalent. That said, the Sukharev grid is never worse than the classical grid, and it is usually much better. Careful grid placement, as opposed to naive grids, is a theme also evident in kinodynamic planning [14].

All forms of grid search ignore discrepancy, but perform very well with respect to dispersion. Note that SGS can be considered as a special kind of LRM in which the lattice is axis-aligned. This completes the spectrum of planners, which will be carefully analyzed and compared in Sections 5 and 6. One somewhat surprising result is that the Sukharev grid, using only a few points per axis, can solve the same kinds of problems as the original PRM! This confirms a prediction based on the Sukharev sampling criterion, (3), which is that an exponential number of samples in dimension are needed if dispersion is held fixed. If this is the best one can do, then it is expected that subsampled grid search performs at least as well as the original PRM. Note that just as in the case of lattices, one does not have to specify N in advance for grids. Incremental grid sampling, for which points can be added one-by-one while maintaining good sampling properties are introduced in [36].

5 Theoretical Considerations

The next step is to characterize the theoretical performance of the planning algorithms along the spectrum. Let *DRM* (for *deterministic roadmaps*) refer to all of the deterministic planners presented in Section 4 (QRM, LRM, CGS, and SGS), under the condition that they use asymptotically dispersion-optimal samples. Note that deterministic sampling enables all DRM planners to be *resolution complete*, in the sense that if it is possible to solve the query at a given sampling resolution, they will solve it. The resolution can be increased arbitrarily to ensure that any problem can be solved, if a solution exists. This is in contrast to the original

Problem	**Quality measure**	**Difficulty measure**	**Theoretical bound**
integration	discrepancy	bounded variation constant	Koksma-Hlawka Inequality
optimization	dispersion	Lipschitz constant	Niederreiter Inequality
planning	dispersion	corridor thickness	our analysis

Figure 4: Our analysis follows the spirit of that used in quasi-Monte Carlo literature.

PRM and other randomized variants, which are only probabilistically complete [32] (the probability tends to one that a solution will be found as the number of samples grows to infinity).

Figure 4 illustrates our motivation in the following analysis by making an analogy to analysis results from the integration and optimization literature. In quasi-Monte Carlo literature, the analysis of a sampling method is based on relating a difficulty measure for the problem at hand to the quality of the sampling technique in terms of this measure. Low-discrepancy samples were developed to perform better than random samples for numerical integration. As mentioned in Section 3.1, this relationship is given by the celebrated Koksma-Hlawka inequality, which expresses the convergence rate in terms of $D(P, \mathcal{R}_{aar})$ discrepancy and a constant that captures the bounded variation (in the sense of Hardy-Krause) in the integrand; see [41] for more information. Low-dispersion samples were developed to perform better than random samples in numerical optimization [50]. This is established by an inequality due to Niederreiter [41], which expresses the convergence rate in terms of dispersion and a Lipschitz constant for the function to be optimized. Inspired by these results, we have obtained a bound that expresses the convergence rate in terms of dispersion and the width of the narrowest corridor in $\mathcal{C}_{free}$. The corridor thickness appears to be a natural measure of difficulty, and has been considered previously in [3, 7, 25]. If there is no information regarding alignments of the obstacles in $\mathcal{C}$, it seems that dispersion is the most natural quality measure to use for motion planning. (In some particular contexts, however, it might be appropriate to consider various forms of discrepancy.)

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 planning analysis [3, 6, 25, 26, 31]. If a solution does not exist, our deterministic planners are 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, the solution might be impractical anyway.

We assume that for any PRM-like method, the radius parameter used to select neighbors in the NBHD function from Line 5 of Figure 1 is always sufficiently large. In theory, the radius can be made large enough so that an attempt is made to connect every vertex to every other vertex. In practice, this often becomes inefficient; therefore, a smaller value is typically 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.

Recall that $\mathcal{C} = [0, 1]^d \subset \mathbb{R}^d$, with boundary points possibly identified to respect topology. 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 [32].

Let a *tube*, $\mathcal{B}$, represent an uncountable collection of balls of equal radius whose centers are generated by a continuous path, $\tau : [0,1] \rightarrow \mathcal{C}_{free}$. 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 point $q \in \mathcal{C}_{free}$ (i.e., for each $q' \in V(q)$, $\lambda q + (1-\lambda)q' \in \mathcal{C}_{free}$ for all $\lambda \in [0,1]$, assuming that topology is respected in the interpolation). An alternative definition may be preferable here. The set $V(q)$ could be all configurations from which a gradient descent on a $\mathcal{C}$ metric converges to q. This would be preferable, for example, near cusps in the $\mathcal{C}_{free}$ boundary.

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 DRM planners. Therein, we only assume that sampling is accomplished using a set P of *low-dispersion points*, for which $\delta(P, \rho) < b(d)N^{-1/d}$, in which $b(d)$ is a constant that may depend on the dimension d. For Halton/Hammersley points, b is a function of the largest prime used in the construction. For many low-dispersion sequences, including a multiresolution Sukharev grid, $b(d) \equiv 1$, which is a small constant that is independent of dimension. If N is restricted to values such that $N^{1/d}$ is an integer, then $b(d) \equiv 1/2$. A low-dispersion sequence for which $b(d) \equiv \frac{1}{\ln 4}$ and no restriction is placed on N is given in [41] (surprisingly, it is not uniform).

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.

Proposition 1 *After N iterations, all DRM planners are 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

the asymptotic dispersion bound is $\delta(P, \rho) < b(d)N^{-1/d}$. 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 1 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 $\mathcal{C} = [0, 1]^d \setminus \sim$, 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 $\delta(P, \rho) < b(d)N^{-1/d}$ 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 $[0, 1]^d \setminus \sim$ by assuming $\mathcal{C}_{free} \in \Psi(4b(d)N^{-1/d})$ in the first step, which establishes the proposition. ∎

Proposition 1 can be reworked to bound the width of the query:

Proposition 2 *After N iterations, all DRM planners, for a query γ, either report a solution path or correctly declare 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 DRM planners are 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 a DRM or PRM planner might 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 1 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

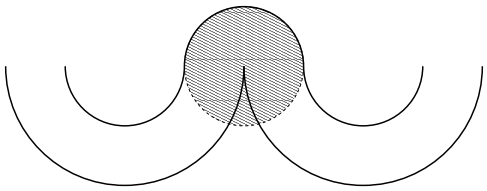


Figure 5: A narrow corridor in $\mathcal{C}_{free}$, used in the proof of Proposition 3.

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 corridor shown in Figure 5. Assume that 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 (including pseudo-random), 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, (3). ∎

We now consider asymptotic bounds for DRM planners. The next proposition indicates that DRM planners do the best possible, asymptotically.

Proposition 5 *The number of samples required by DRM planners to be complete for* $\Psi(\delta)$ *is asymptotically optimal.*

Proof: By Proposition 3, to be complete for $\Psi(\delta)$, the dispersion must be less than δ. Thus, the goal of a DRM algorithm should be to reduce δ using as few samples as possible. The low-dispersion sequences achieve the best possible asymptotic dispersion. Therefore, the number of samples used in DRM planners is asymptotically optimal. ∎

The following proposition gives some indication that random sampling does not yield the best possible asymptotic convergence in the PRM; in fact, it is significantly worse than using deterministic sampling.

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

Proof: It was shown by Deheuvels [41] that ℓ^∞ dispersion for random samples is $O((\log N)^{\frac{1}{d}} N^{-\frac{1}{d}})$ with probability one. The asymptotic dispersion in a DRM achieves $O(N^{-\frac{1}{d}})$. The factor difference between the two is $O((\log N)^{\frac{1}{d}})$. ∎

Relating the PRM to grid search. We now discuss the implications of the previous propositions. Consider Propositions and 1 and 2. Using ℓ^∞ dispersion and selecting N such that $N^{1/d}$ is an integer, the Sukharev grid yields $b(d) \equiv 1/2$, which is the best possible performance that can be obtained. An exponential number of samples in d is required, but according to Proposition 4, this is unavoidable. (Note

that the PRM cannot even provide these deterministic guarantees.) The PRM at least has asymptotic analysis that establishes probabilistic completeness and convergence [3, 29]. However, Propositions 5 and 6 indicate that the asymptotic rate of convergence obtained by the best deterministic sequences (including the Sukharev grid) is asymptotically optimal, and the use of random samples is worse by a significant factor (with probability one). Proposition 6 quantifies the problems with random sampling. To satisfy statistical tests for uniformity, a pseudo-random number generator must clump too many points in some places, and not enough in others. If the distribution of points is "too uniform," then it will fail statistical tests. The deterministic samplers, on the other hand, carefully *disperse* the points without this wasteful concern.

One expectation that arises from the theoretical analysis is that a Sukharev grid with $\lceil N^{\frac{1}{d}} \rceil$ points per axis should perform at least as well as *any* collection of N samples (over a broad collection of problems). Thus, there is nothing inherently bad about grid sampling.

We expect that if the original PRM could solve a class of problems using N pseudo-random samples, the performance should be at least as good if a Sukharev grid with $\lceil N^{\frac{1}{d}} \rceil$ points per axis is used instead, since the dispersion must be at least as small. This expectation motivates our experiments in which problems of the level of difficulty considered in the original PRM are solved by Sukharev grids that only use a few points per axis. This will be demonstrated in Section 6, along with other experiments that show advantages of low-discrepancy sequences such as Halton points.

6 Experiments

Thus far, we have argued that there is a spectrum of planners between classical grid search and PRMs, which includes subsampled grid search, lattice roadmaps, and quasi-random roadmaps. Also, our theoretical results strongly suggest that low-dispersion sampling will asymptotically outperform random sampling for roadmap algorithms. However, it is also useful to see how significant these benefits are for typical motion planning problems. In addition to this general concern, there are several questions which are of particular experimental interest. These include: How do deterministic samples perform as dimension increases? Does using a large connection radius with random samples allow them to perform as well as more uniformly-distributed deterministic ones? Does axis-alignment work in favor of or against deterministic samples? In this section, we present five sets of examples, designed to answer these questions and illustrate different aspects of deterministic versus random sampling.

We implemented the algorithms using the Motion Strategy Library (msl.cs.uiuc.edu/msl/) in GNU C++ running Linux on 2.0 Ghz PCs. The PQP collision detection package from the Univ. of North Carolina was used. For any problems that involve 3D rotations, sampling was performed over the Euler angles (as opposed to using the Haar measure). Figure 6 depicts our five examples. First, we examine the problem of a point robot moving through a d-dimensional bent corridor (we show results for the cases $d = 2, 3, 6, 10$). This gives us an opportunity to see the performance of deterministic and random sampling techniques in low and moderate dimensions. Also, for some experiments we choose a random rotation to apply the sample set; this eliminates any axis alignments (the base example is axis-aligned). Second, we have the problem of extricating a rigid body (a fire truck, to be precise) from a cage. This is an example of a typical motion planning problem, and we use it to examine the effect of the connection radius on planner performance. Third, we take a similar problem: placing a feather in a cup. Fourth, we have the problem of moving an

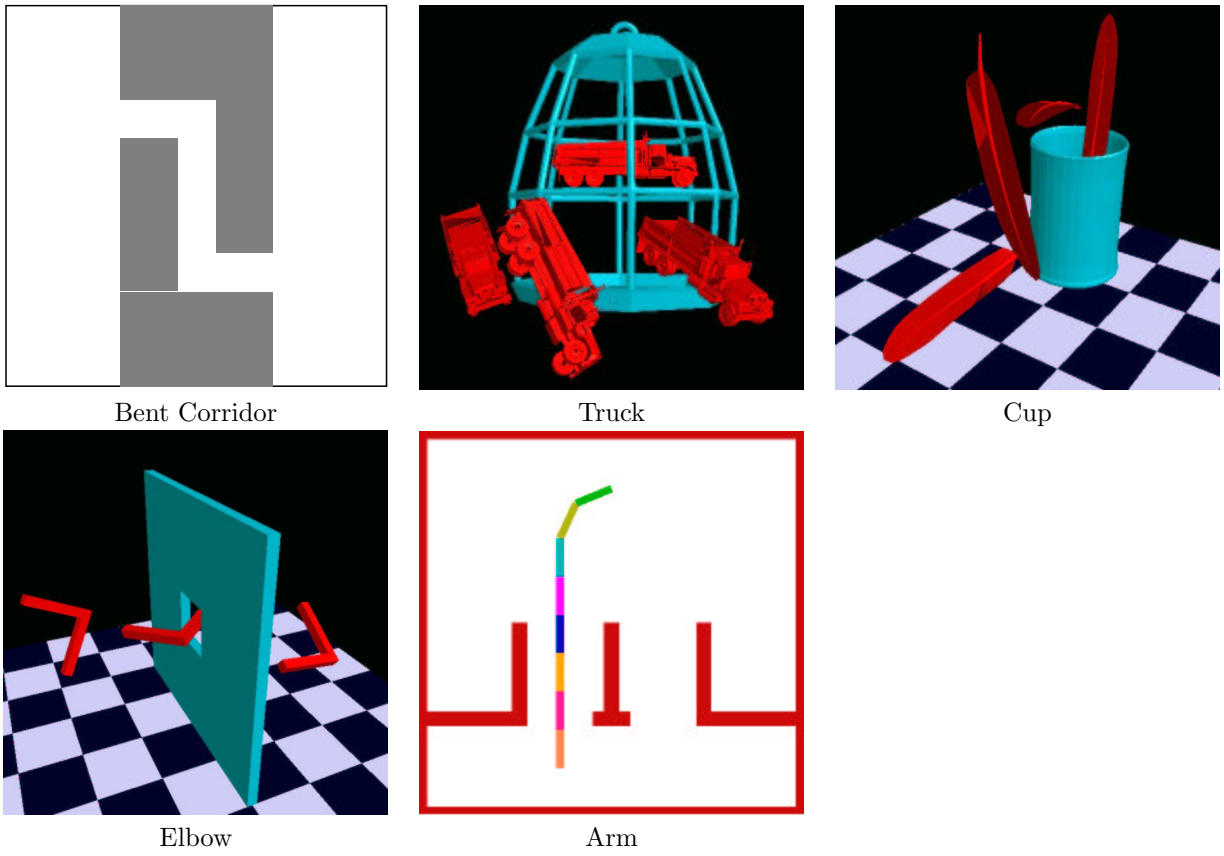


Figure 6: The first frame shows a 2D rendering of the bent corridor series of experiments. The truck, cup, and elbow are 6-DOF rigid-body problems, and the final frame shows an articulated robot in the plane.

L-shaped robot through a hole in a wall. By varying the position of the wall and the location of the hole, we obtain a notion of "expected performance" for deterministic samples analogous to that which naturally arises for random ones. Finally, we examine the case of an articulated robot in the plane. This is another example of a moderate-dimensional problem, and its configuration space has a different topology than the previous examples.

The bent corridor experiments presented here can be considered as a continuation of those in [9][1]. We examined random samples, Hammersley and Halton points, and Sukharev grids. For Sukharev grids, we give results for "first solution" and "guaranteed solution." The former indicates the minimum resolution Sukharev grid capable of solving the problem. The latter refers to the minimum resolution such that it and all higher-resolution grids solve the problem. (For example, in the 3-dimensional corridor with a large radius, the first solution uses 5 points per axis, and the guaranteed solution uses 11 points per axis.) Since Hammersley points constitute a point set rather than a sequence, there are "first" and "guaranteed" values for them as well; however, the differences are typically much less significant than with the Sukharev grid, and so we omit them. Figure 7 gives our results.

We see that Hammersley and Halton points outperform random sampling virtually across the board, and

[1]Regrettably, the original figure of bent corridor experiments contains a typographical error: the width values given are actually 1/2 of the actual width used.

Dim	Width	Rad	Rand	Ham	Hal	Suk_f	Suk_g
2	0.04	0.1	1058	335	962	121	625
		0.5	522	87	322	25	441
3	0.10	0.2	1592	630	943	729	1331
		0.6	913	232	943	125	1331
6	0.20	0.3	37069	37000	16958	15625	15625
		0.6	12613	6200	3064	729	15625
10	0.40	0.6	10541	5000	7300	59040	59049
		0.7	5415	4000	7300	59049	59049

Figure 7: Comparisons of the number of samples required to solve bent corridor problems, with small and large radii. Results for random sampling are averaged over 100 trials.

Sukharev grids often do as well. Except for the 10-dimensional case, the first Sukharev solution has fewer samples than the number required by random sampling; often, however, the guaranteed solution has as many or more. This illustrates a fundamental disadvantage of sampling using point sets: it is often not possible to know how many points are required to solve the problem. If one chooses an "unlucky" value of N, then one may fail to solve the problem even though many samples have been taken. There are several ways one can construct an infinite sequence from finite point sets, such as iterative deepening or some multiresolution approach [36]. These experiments strongly suggest that effective grid sampling requires an incremental approach. As expected, random sampling seems to gain versus deterministic sampling when the radius is increased. However, this does not seem to be significant. It may be that Hammersley and Halton points also benefit from a larger radius, more so than Sukharev grids (because for grids, neighborhoods are very clearly defined by the points' regular spacing).

Also, we give results for the cases with rotated samples, for the problems of 2, 3, and 6 dimensions. As seen in Figure 8, the differences are minor. This indicates that the deterministic sequences are not benefitting unduly from axis alignments.

Figure 9 gives the results for the problem of moving a truck outside of a cage and for placing a feather into a cup. No specific comments need to be made; as in the previous example, Hammersley and Halton points perform the best. Sukharev grids are impaired by the fact that there are limited choices for resolution (4, 5, and 6 points per axis yield 4096, 15625, and 46656 samples, respectively). Once again, incremental grid-based approaches should offset this disadvantage or perhaps eliminate it entirely. For these examples, we tried two different connection approaches: first, we used a fixed radius (as in the other experiments); second, we tried connecting to a node's K nearest neighbors. Note that one must be careful when taking K nearest neighbors on a grid, since many neighbors can be equidistant. This issue can be resolved by either randomized techniques (choosing K at random from the set of equidistant), or deterministic techniques, which could guarantee greater uniformity though at greater cost.

For random samples, it is natural to consider expected performance, rather than performance for a particular trial. This is essential, because the performance of random sampling can vary significantly from trial to trial. However, the performance of deterministic sequences does not change from run to run; is it possible to have a notion for deterministic sequences similar to expected performance for random samples? For problems of numerical integration, theorists consider the expected error of deterministic sample sequences over a set of problems (the set is equipped with a probability measure). There does not seem to be any

Dim	Width	Rad	Rand	Ham	Hal	Suk_f	Suk_g
2	0.04	0.5	511	217	266	25	316
			485	214	92	164	216
			475	117	179	41	332
			536	236	362	132	296
			485	226	123	56	253
3	0.10	0.6	678	760	294	265	563
			679	268	1744	569	569
			687	383	257	216	216
			667	254	822	269	541
			751	830	858	440	440
6	0.20	0.6	6487	2239	3321	8855	8855
			7472	4512	3960	10765	10765
			6960	2293	6786	477	9285
			7546	8744	4504	9119	9119
			6402	8637	3425	9149	9149

Figure 8: Comparisons of the number of samples required to solve bent corridor problems. A rotation matrix was constructed by generating random Euler angles, and the rotation was applied to the sample set; five tests are shown. Note that for the Sukharev grid, $N^{\frac{1}{d}}$ is not necessarily an integer because samples that lie outside of $[0,1]^d$ after rotation are discarded.

Prob	Conn	Rad/K	Rand	Ham	Hal	Suk_f	Suk_g
Truck	Rad	0.4	8352	6600	7925	46656	46656
		0.6	3009	1300	1100	15625	15625
	K-Near	20	8293	1300	6224	46656	46656
		100	2507	1300	2105	15625	15625
Cup	Rad	0.4	4751	1400	4500	15625	15625
		0.6	1920	100	2500	729	729
	K-Near	20	3672	100	2500	729	729
		100	1449	100	1500	729	729

Figure 9: Comparisons of the number of samples required to solve two 6-DOF problems, with small and large radii. Results for random sampling are averaged over 100 trials.

Hole	Rand	Ham	Hal	Suk_f	Suk_g
0.2	136	72	123	729	729
0.1	2989	2444	2419	10949	10949

Figure 10: Comparisons of the average number of samples required to solve the problem from Figure 6(d). The wall was placed at $x = 0.2, 0.3, 0.4, 0.5$ and the y and z coordinates of the center of the hole varied from 0.3 to 0.5 (in increments of 0.1), leading to 36 total experiments. Results are shown for two different hole sizes (the height and length of the robot was 0.3, and the width was 0.03). The connection radius used was 0.6.

Rad	Rand	Ham	Hal	Suk_f	Suk_g
0.5	476	200	223	256	256
0.9	442	200	223	256	256

Figure 11: Comparisons of the number of samples required to solve the articulated robot problem.

meaningful analogue for motion planning problems (i.e., how would one define a probability measure over all motion planning problems?) However, our fourth example attempts to capture the spirit of "expected performance" by taking a set of similar problems and examining performance over the entire set. The basic problem we consider is that of moving an L-shaped robot through a hole in a wall. The wall is aligned with the yz-plane; however, its position varies along x-axis. Also, the position of the hole varies in the yz-plane.

In Figure 11, we give the results of experiments with an articulated robot (8 links with a fixed base, yielding 8 DOF). The arm was extracted from one opening and inserted into another.

All of the experiments so far have considered multiple-query versions of the planners. In [9], a single-query Lazy LRM was evaluated by using the extensible lattices presented in [49]. For the case of single-query planning, the lattice structure could be exploited to avoid virtually all of the precomputations that were required to set up the initial PRM. For the sake of completeness, we reproduce results from [9] regarding the performance of lattice point sets in lazy planners for the Elbow, Cup, and Truck problems. The first column gives the problem name. The next three columns (N_min, N_max, and N_avg) give the minimum, maximum, and average numbers of nodes needed in the Lazy PRM, averaged over 25 trials. The column labeled Lattice shows the number of nodes for the Lazy LRM. The next three columns (T_min, T_max, T_avg) give the minimum, maximum, and average computation times in seconds, including both precomputation and searching. The PreCmp column indicates how much of the average computation time was devoted to setting up the initial roadmap (before collision checking is performed). The final column gives the total amount of time in the Lazy LRM. These times are much smaller because there is essentially no precomputation time due to the fact that the neighborhood structure is immediately inferred from the lattice. Thus, for single-query problems, the advantages of deterministic sampling become even greater.

In conclusion, our experimental results show that in addition to being asymptotically optimal, low-dispersion sampling methods perform well for ordinary motion planning problems. Halton and Hammersley points outperformed random sampling for nearly all of our experiments, and Sukharev grids were often competitive. In cases where grids were not competitive, it is likely that some kind of incremental grid-based sequence would be. Since our experiments represent a fairly broad range of problems, we believe that our results will hold for a large class of motion planning problems.

Prob.	N_min	N_max	N_avg	Lattice	T_min	T_max	T_avg	PreCmp	Lattice
Elbow	1250	15250	4667	3963	7.0	718	287	212	10.1
Cup	2000	12000	4800	2152	2.33	253	36.9	15.9	1.23
Truck	5000	95000	35207	5138	11.2	5480	935	800	18.5

Figure 12: Comparisons of the number of nodes and time required to solve several 6-DOF problems using the Lazy PRM and the Lazy LRM (lattice-based roadmap). Results for the randomized version are over 25 trials.

7 Discussion

We have provided a spectrum of planners that ranges from classical grid search to the probabilistic roadmap. Surprisingly, the original PRM does not appear to be advantageous over deterministic approaches, including grid search, according to both our experiments and our theoretical analysis. Since the "probabilistic" theme of PRMs does not seem advantageous, it is perhaps better to refer to these, grid search, RRTs, and other related planners as instances of *sampling-based motion planning*. This distinguishes them from exact or combinatorial motion planning algorithms (e.g., [11, 43, 46]). Within sampling-based motion planning, one can imagine subclasses based on whether an approach is multiple-query or single-query. For multiple-query planning, both probabilistic and deterministic roadmaps could fall under a category of *sampling-based roadmaps*.

Based on our work, we believe one of the main factors for the success of the original PRM was the excellent use of subsampling, which for grid sampling, enables challenging problems to be solved with only a few points per axis. In the original PRM, this idea also contributed greatly to the performance. Although there appears at first to be no exponential dependency on dimension because N is chosen directly, the exponential dependency reveals itself once again if we try to hold dispersion fixed. The notion of fixing the dispersion is general enough to mean keeping the resolution the same for a grid, while also applying to any sampling scheme. Using (3), the Sukharev sampling criterion, $N^{\frac{1}{d}}$ can be considered as the best possible "points per axis" for any sampling scheme, whether or not it is a grid, and also whether it is random or deterministic.

Very high dimensions. We note that our experiments focused mostly on six dimensions, with some examples in [9] up to ten dimensions. This includes many problems of interest in robotics, but examples exist in robotics and computational biology in which there are dozens or hundreds of dimensions. Obviously, even a Sukharev grid with two points per axis would be impossible to manage for some problems, while pseudo-random samples would appear to have no trouble. One straightforward way to use deterministic sampling would be to use a non-lattice sequence in the place of pseudo-random samples. In very high dimensional problems we expect, however, that the performance differences between using pseudo-random sampling and deterministic sampling would be negligible for most problems. This is due primarily to the fact that spaces of this dimension will be severely undersampled, regardless of the sampling scheme used (i.e., the "points per axis" for any sampling scheme would be approximately one). For example, in d dimensions, there are 2^d quadrants, one of whose interior must be empty if less than 2^d samples are used.

Search issues. Graph searching is one of the oldest and most fundamental issues in motion planning. Although most PRM-based planners use A^* search, it is important to revisit some of the basic search issues in this context. Given the close connection drawn between classical grid search and the PRM in this paper, it

should be clear that classical search issues which apply to grids should also apply to PRMs, QRMs, Sukharev grids, etc. that have a large number of samples. If a PRM has thousands of nodes in a high-dimensional space, then the common local minima problems associated with search most likely exist. Many single-query motion planning algorithms were developed precisely to handle this problem. Potential field approaches have improved search by using heuristics to perform a greedy search [4, 17, 27]. For the randomized potential field planner in [4], the search is even performed over an implicit grid, and randomization was used to try to escape local minima. Thus, it can be considered as a lazy, single-query CGS, with A^* search replaced by the potential field method. The planner in [13] can also be considered as a lazy CGS planner. Of course many other planning approaches can be considered. Incremental searching methods, such as Ariadne's clew [40], bidirectional RRTs [33], or the planners in [26, 45], can be considered as alternatives to A^* search in any roadmap approach. As the implicit grid resolution in a lazy CGS planner becomes higher, it is clear that the sampling itself is not as important. The key is how to search in a bounded metric space.

Multiresolution approaches. Our work helps shed some light on the relationship between the PRM and multiresolution search (e.g., [16, 28, 32]). One disadvantage of the Sukharev grid is that for each increment of the points per axis, the total number of points increases dramatically for higher-dimensional problems. This was why in some experiments, the Sukharev grid performed poorly. The PRM relies on a pseudo-random infinite sequence which yields a reasonable coverage of the configuration space, even though only the first N points in the sequence are used. Also, one can easily add more PRM vertices by taking the next points in the sequence. In general, it is desirable to have a deterministic sequence that behaves in a similar manner, but also provides optimal dispersion and/or discrepancy. The Halton sequence achieves this, which makes it an attractive replacement to the random samples in the PRM. It is possible, however, to use an infinite sequence that maintains lattice structure while incrementally yielding low dispersion and/or discrepancy. Low-discrepancy extensible lattices that are infinite sequences have been introduced recently in [23]. In [5], a trellis is iteratively maintained (similar to SGS), and a hyperplane of samples is added in each iteration before search is performed again. In [36], extensible grid-based sequences have been introduced. In this case, the sequence could serve as a replacement to the pseudo-random number generator, or it could be used as a basis for a multiresolution planner. Conceptually, there is not much difference between adding more points to a PRM or adding more points to an extensible lattice. Thus, the two are closely related.

Selection of quality measures. If nothing is known about the space and a lazy, single-query approach is used, then dispersion seems to be a natural measure of sample quality. If it is known that alignment problems exist, then it is better to use discrepancy with a range space that is aligned. In our experiments, we have found that if the boundary of $\mathcal{C}_{free}$ is axis-aligned and a grid-based sampling is used, the average computation time over many examples appears about the same, but the variance is higher. In some cases, the alignment helps, and in others it hurts. For multiple-query problems, it might seem that measures such as whether samples lie within mutual visibility regions would be more appropriate. This could ensure that far-away samples could connect to each other when building the roadmap. However, it is not clear how this could be achieved since it is assumed that the obstacle region is only implicitly represented, and thus only revealed by the sampling process itself. Thus, it seems difficult, if not impossible, to develop a general-purpose sampling theory that is based on visibility. Note that rejection schemes, such as that used in the Visibility PRM [47], can be easily combined with the sample sequences described this paper (this has already been successfully done [48]). It would be interesting to determine whether lattice structure could be

similarly exploited.

Bringing back randomization. A recent trend in quasi-Monte Carlo literature is to consider randomized versions of Halton sequences, lattices, and (t,s)-sequences. There exist simple techniques that preserve the low-discrepancy/low-dispersion properties of deterministic sequences while at the same time each sample is uniform randomly distributed.[2] An elegant example of this work appears in [53], in which a randomized Halton sequence is obtained by cleverly constructing the first element at random, and computing the appropriate continuations of classical van der Corput sequences for each coordinate. Some randomization was added to Halton points and was observed to offer some improvements in [18]. Randomized low-dispersion sequences can be used in motion planning to obtain *both* probabilistic analysis and the deterministic guarantees from Section 5!

Acknowledgments

We are grateful for the funding provided in part by NASA grant NAG2-1333 (Branicky), NSF awards 9875304, 0208891 (LaValle), NSF awards 0208891 and 0208919 (Branicky, Frazzoli, and LaValle). This work has benefited greatly from discussions and/or debates with Nancy Amato, Robert Bohlin, Francesco Bullo, Bruce Donald, Mike Erdmann, Leo Guibas, Blake Hannaford, Seth Hutchinson, Lydia Kavraki, Kari Olson, Mark Overmars, Shai Sachs, Nicola Simeon, and Libo Yang. We thank Eric Larsen, Stefan Gottschalk, Ming Lin, and Dinesh Manocha for making the PQP collision detection package available.

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 evalaution. In S. Rajasekaran, P. Pardalos, J. Reif, and J. Rolim, editors, *Handbook on Randomized Computation*. Kluwer Academic, 2001.

[2] However, the samples in the sequence are not uncorrelated, as in the case of a uniform, random sequence.

[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. S. Branicky, S. M. LaValle, K. Olson, and L. Yang. Quasi-randomized path planning. In *Proc. IEEE Int'l Conf. on Robotics and Automation*, pages 1481–1487, 2001.

[10] J. Canny. Computing roadmaps of general semi-algebraic sets. *Computer Journal*, 36(5):504–514, 1995.

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

[12] J. H. Conway and N. J. A. Sloane. *Sphere Packings, Lattices, and Groups*. Springer-Verlag, Berlin, 1999.

[13] B. R. Donald. A search algorithm for motion planning with six degrees of freedom. *Artif. Intell.*, 31:295–353, 1987.

[14] B. R. Donald, P. G. Xavier, J. Canny, and J. Reif. Kinodynamic planning. *Journal of the ACM*, 40:1048–66, November 1993.

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

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

[17] 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.

[18] R. Geraerts and M. H. Overmars. A comparative study of probabilistic roadmap planners. In *Proc. Workshop on the Algorithmic Foundations of Robotics*, December 2002.

[19] B. Glavina. Solving findpath by combination of goal-directed and randomized search. In *IEEE Int. Conf. Robot. & Autom.*, pages 1718–1723, May 1990.

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

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

[22] 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.

[23] F. J. Hickernell, H. S. Hong, P. L'Ecuyer, and C. Lemieux. Extensible lattice sequences for quasi-monte carlo quadrature. *SIAM Journal on Scientific Computing*, 22:1117–1138, 2000.

[24] 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.

[25] 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.

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

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

[28] S. Kambhampati and L. S. Davis. Multiresolution path planning for mobile robots. *IEEE J. of Robot. & Autom.*, 2(3):135–145, 1986.

[29] 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.

[30] K. Kondo. Motion planning with six degrees of freedom by multistrategic bidirectional heuristic free-space enumeration. *IEEE Trans. Robot. & Autom.*, 7(3):267–277, 1991.

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

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

[33] 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.

[34] J. Lengyel, M. Reichert, B. R. Donald, and D. P. Greenberg. Real-time robot motion planning using rasterizing computer graphics hardware. *Computer Graphics*, 24(4):327–335, August 1990.

[35] P. Leven and S. Hutchinson. Real-time motion planning in changing environments. In *Proc. International Symposium on Robotics Research*, 2000.

[36] S. R. Lindemann and S. M. LaValle. Incremental low-discrepancy lattice methods for motion planning. In *Proc. IEEE International Conference on Robotics and Automation*, 2003.

[37] T. Lozano-Pérez. Spatial planning: A configuration space approach. *IEEE Trans. on Comput.*, C-32(2):108–120, 1983.

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

[39] E. Mazer, J. M. Ahuactzin, and P. Bessière. The Ariadne's clew algorithm. *J. Artificial Intell. Res.*, 9:295–316, November 1998.

[40] 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.

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

[42] 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.

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

[44] 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.

[45] G. Sánchez and J.-C. Latombe. A single-query bi-directional probabilistic roadmap planner with lazy collision checking. In *Int. Symp. Robotics Research*, 2001.

[46] 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.

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

[48] Thierry Simeon, 2002. Personal communication.

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

[50] 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.

[51] S. Tezuka. Quasi-monte carlo: The discrepancy between theory and practice. In K.-T. Fang, F. J. Hickernell, and H. Niederreiter, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2000*, pages 124–140. Springer-Verlag, Berlin, 2002.

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

[53] X. Wang and F. J. Hickernell. Randomized Halton sequences. *Math. Comp. Modelling*, 32:887–899, 2000.

[54] X. Wang and F. J. Hickernell. An historical overview of lattice point sets. In K.-T. Fang, F. J. Hickernell, and H. Niederreiter, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2000*, pages 158–167. Springer-Verlag, Berlin, 2002.

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

[56] 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.