Proceedings of the 2001 IEEE
International Conference on Robotics & Automation
Seoul, Korea · May 21-26, 2001

Quasi-Randomized Path Planning

Michael S. Branicky
EECS Dept.
Case Western Reserve Univ.
Cleveland, OH 44106 USA
mb@ieee.org

Steven M. LaV alle
Computer Science Dept.
Iowa State University
Ames, IA 50011 USA
lavalle@cs.iastate.edu

Kari Olson
EECS Dept.
Case Western Reserve Univ.
Cleveland, OH 44106 USA
kdo@po.cwru.edu

Libo Yang
Computer Science Dept.
Iowa State University
Ames, IA 50011 USA
lyang@cs.iastate.edu

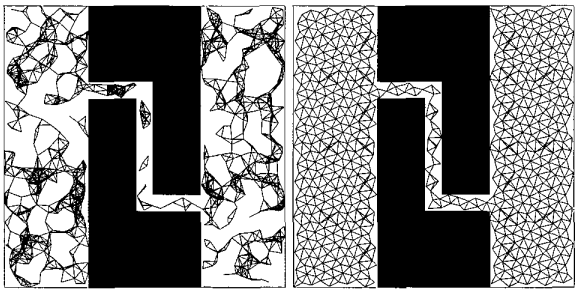
Abstract

We propose the use of quasi-random sampling techniques for path planning in high-dimensional configuration spaces. Following similar trends from related numerical computation fields, we show several advantages offered by these techniques in comparison to random sampling. Our ideas are evaluated in the context of the probabilistic roadmap (PRM) framework. Two quasi-random variants of PRM-based planners are proposed: 1) a classical PRM with quasi-random sampling, and 2) a quasi-random Lazy-PRM. Both have been implemented, and are shown through experiments to offer some performance advantages in comparison to their randomized counterparts.

1 Introduction

Over two decades of path planning research have led to two primary trends. In the 1980s, deterministic approaches provided both elegant, complete algorithms for solving the problem, and also useful approximate or incomplete algorithms. The curse of dimensionality due to high-dimensional configuration spaces motivated researchers from the 1990s to the present time to develop randomized approaches which are incomplete, but capable of efficiently solving many challenging, high-dimensional problems. A similar pair of trends occurred many years ago in the area of numerical integration and related optimization fields. These trends were followed by a third trend: the development of quasi-random approaches that use deterministic sampling to achieve performance that is often superior to random sampling. Quasi-random sampling ideas ha ve improved computational methods in man y areas, including integration [24], optimization [20], image processing [7], and computer graphics [23]. It is therefore natural to ask: Can quasi-random sampling ideas also improve path planning methods designed for high degrees of freedom? Is randomization really necessary?

In this paper, we take the first step towards answering these questions by illustrating some of the advantages of quasi-random sampling in the con text of the probabilistic roadmap (PRM) framew ork for path planning [1, 15]; a glimpse is given in Figure 1. W e present implemented, quasi-random variants of both the classical PRM [15] and the recent Lazy-PRM [4], and indicate some advantages over their randomized counterparts.



a. b.

Figure 1: a) A probabilistic roadmap based on random sampling; b) a quasi-random version. Each uses 1000 samples and the same connection radius.

At first glance, the progression from deterministic to randomized, and then back to deterministic might appear absurd; thus, some explanation is required. There appear to be two prevailing reasons for the preference of randomized methods over classical deterministic techniques: 1) they fight the curse of dimensionality by allowing a problem to be solved without prior, systematic exploration of all alternatives; 2) if the "problem maker" is viewed as an opponent in a game, then one can often avoid defeat by employing a random strategy (imagine defeating a deterministic strategy by designing a problem that causes worst-case performance). The second reason would be valid in the case of "true" random numbers; however, any machine implementation generates a deterministic sequence of pseudo-random numbers. These numbers are designed to meet performance criteria that are based on uniform probability densities; however, once it is understood that these numbers are deterministic and being used to solve a particular task, why not design a deterministic sequence that can solve the task more efficiently, instead of worrying about statistical closeness to a uniform density? This motivates the design of quasi-random numbers.

0-7803-6475-9/01/$10.00© 2001 IEEE

2 Sampling Methods

The first investigations of sampling paradigms were developed in the context of numerical integration of high-dimensional functions. Deterministic grid-based quadrature formulas led to the same combinatorial explosion that was experienced years later in grid-based path planning methods. About 50 years ago [19], this frustration led to the development of numerical integration techniques (Monte Carlo) based on random sampling of the function domain, and convergence was not dependent on dimension. There are interesting parallels between later developments of specialized random sampling methods that improve performance for sharply-peaked integrands [12, 13, 17], and the recent development of specialized sampling methods for path planning [1, 5, 10].

In practice, random sampling methods require the construction of deterministic, pseudo-random samples. Thus, researchers began to question whether other deterministic samples could be designed which lead to better performance in numerical methods. The design of a good set of samples can be considered as an optimization problem in which the locations of the points are evaluated by a criterion of uniformity. One of the most common measures is the *discrepancy* [20], which can be defined for a set, P, of N d-dimensional sample points, $\{x_0, \ldots, x_{N-1}\}$ in $[0,1]^d$ as

$$D_N(P) = \sup_J \left| \frac{A(J)}{N} - \mu(J) \right|$$

in which J is any rectangular subset of $[0,1]^d$, $\mu(J)$ is its measure, and $A(J)$ is the number of points contained in $P \cap J$. The goal is to select a set P that reduces the discrepancy as much as possible. This measure of discrepancy can be thought of as the worst-case difference between the fraction of points in a box and the fraction of the box area, taken over the set of all possible boxes.

In the context of path planning, both the discrepancy and a related measure called dispersion appear to be useful. The *dispersion* of a sequence of points in a space $[0,1]^d$ is the maximum distance that any point in $[0,1]^d$ could be from the nearest point in the sequence. For any finite set P of N points in $[0,1]^d$, it is known that $d_N(P) \leq D_N(P)^{\frac{1}{d}}$, in which d_N is the dispersion (under the l^∞ metric) and D_N is the discrepancy [20, 25]. Hence, low-discrepancy point sets lead to low dispersion. The benefits of low discrepancy and low dispersion point sets is evaluated in the context of randomized path planning in Sections 3 and 4.

We describe four classes of quasi-random samples: 1) grids, 2) lattices, 3) closed sequences, and 4) open sequences. The first is a proper subset of the second, and the second is a proper subset of the third. In the last three classes, it has been possible to obtain low-discrepancy point sets that perform better than random sampling for numerical integration and optimization problems [8, 20, 24]. A *grid* corresponds to the usual uniform quantization of each of the coordinate axes. A *lattice* is a generalization that preserves the convenient neighborhood structure of a grid, but is generated by a collection of generally nonorthogonal basis vectors that lead to low discrepancy (more details on lattices appear in Section 4). A *closed* method does not require any neighborhood structure for the samples, and the only restriction is that the number of samples must be specified *a priori*. Finally, an *open* method does not require the number of samples. Generally, *closed* methods lead to lower discrepancy than open methods because the knowledge of the number of points aids in the optimization. In Section 3, we present a variant of the classical PRM by using general closed and open sampling methods (non-lattice). In Section 4, we define a variant of the Lazy-PRM, and exploit the convenient neighborhood structure of quasi-random lattices.

3 Quasi-random Roadmap (Q-PRM)

3.1 PRM Review

Probabilistic roadmap (PRM) planners use randomly sampled data to assist a robot with navigation through an obstacle-ridden area. The description of PRM presented here is based roughly on the initial algorithm described in [15]. It has inspired different versions of PRM planners that have been developed by different researchers (see [1, 3, 5, 15] and their references). When applying roadmap path planners, there are two phases. During the first phase, nodes are generated and connections between the nodes are added to the roadmap. In the second phase, the roadmap is queried with an initial and goal configuration. An outline of the roadmap generation portion of the PRM algorithm follows.

Consider the configuration space $\mathcal{C}$ of the robot and the portion of that space which does not contain obstacles, $\mathcal{C}_{\text{free}}$. Let the roadmap $\mathcal{R}$ consist of a set of nodes $\mathcal{N}$ and paths between the nodes, $\mathcal{P}$. The roadmap construction phase is as follows:

1. $\mathcal{N} \longleftarrow 0, \mathcal{P} \longleftarrow 0$.
2. **Loop:**
3. $\quad n \longleftarrow$ a randomly chosen node in $\mathcal{C}_{\text{free}}$
4. $\quad \mathcal{N} \longleftarrow \mathcal{N} \cup n$
5. $\quad \mathcal{N}_n \longleftarrow$ a set of candidate neighbors of n
6. $\quad$ **For all** $n' \in \mathcal{N}_n$, in order of increasing distance from n do
7. $\quad\quad$ **if** a local path between n and n' is found **then**
8. $\quad\quad\quad P \longleftarrow P \cup p(n, n')$

An example of the result of this phase appears in Figure 1(a), which tried 1000 nodes. Note the characteristic "clumpiness" of the randomly-chosen nodes and the relatively large areas of free space that contain no samples. The figure also confirms the well-known fact that narrow passages in C-space are notoriously difficult to find at random. Indeed, several planners have been developed to address this issue. Creating nodes in narrow passages has been the main motivation of the enhancement step in [14], the generation of nodes near the configuration space obstacles in [2], the penetration of obstacles in [10], the Gaussian sampling in [5], the retraction to the configuration space medial axis in [2], and the use of the workspace medial axis in [9] and [22].

3.2 Q-PRM Overview

Monte Carlo methods, like PRM and its uniform random sampling cousins for integration and optimization, have been adopted for problems with high dimension to overcome the curse of dimensionality. Recently, quasi-Monte Carlo algorithms, which are identical to their random siblings except that they use deterministic point sets, have been shown to be both computationally efficient and accurate for a variety of applications including 360-dimensional integrations performed in finance and bounded optimization [26]

Simply put, Quasi-random Roadmap (Q-PRM) algorithms aim to replace specialized, quasi-random points for the randomly chosen points in Step 3 of the roadmap generation algorithm above. An example of this process appears in Figure 1(b). The same number of nodes (1000) were examined for roadmap inclusion as in Figure 1(a). Here, however, a path through the narrow passage has been found, there is no clumping of points, and every free-space point is fairly close to a graph node.

The quasi-random points used in Figure 1(b) are called *Hammersley* points. They are one of many point sets that have been designed to have *low discrepancy*. Here, "discrepancy" can be thought of as the degree of difference (measured by some criterion) of a set of points with respect to the properties of points that are uniformly distributed in space. There are many properties of uniform points and many criteria for measuring discrepancy that have been developed. One is based on "uniformly distributed with respect to axis-parallel rectangles." If a set of points, P, were *uniformly distributed* in the $[0,1]^2$ C-space of Figure 1, one might expect the number of points in a given rectangle $R = [a_1, b_1) \times [a_2, b_2)$ to be proportional to its volume $\mathrm{vol}(R) = (b_1 - a_1)(b_2 - a_2)$. Thus,

$$D(P, R) = n \cdot \mathrm{vol}(R) - |P \cap R|$$

is a measure of discrepancy for rectangle R. The discrepancy of point set P is obtained by finding the maximum such difference over all possible rectangles. Thus, one measure of the discrepancy that has been proposed (see [18] for a litany of others plus discussion) is the *L_∞-discrepancy* of the first N points of a d-dimensional sequence, $\mathbf{S}$, in $[0,1]^d$:

$$\sup_{[\mathbf{a},\mathbf{b})\subset[0,1]^d} \left| \frac{|\{\mathbf{s}_1, \ldots, \mathbf{s}_N\} \cap [\mathbf{a}, \mathbf{b})|}{N} - \prod_{i=1}^{d} (b_i - a_i) \right|$$

measured with respect to axis parallel boxes in $[0,1]^d$.

The smallest possible value for the discrepancy of N points in d dimensions is $O(N^{-1}(\log N)^{(d-1)/2})$ [25]. Low discrepancy point sets have been proposed by a number of mathematicians, including Faure, Sobol, Hammersley, and Niederreiter [20]. Typically, these have discrepancies that are $O(N^{-1}(\log N)^d)$. Some are quite easy to generate. For example, Hammersley and Halton point sets can be generated in arbitrary dimensions using the following algorithm.

Definition 3.1 (Hammersley-Halton Sets [18])

- Hammersley points. *Choose $d-1$ distinct primes $p_1, p_2, \ldots, p_{d-1}$ (say the first $d-1$ primes, $p_1 = 2$, $p_2 = 3$, ...). The ith point of the set is given by*

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

  We call such points $\mathrm{HH}(1/N, p_1, \ldots, p_{d-1})$ *points.*

- Halton points. *Choose d distinct primes $p_1, p_2, \ldots, p_d$. The set's ith point is given by*

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

  We call such points $\mathrm{HH}(p_1, \ldots, p_d)$ *points.*

Here, the function $r_p(i)$ is obtained by writing the digits of the p-ary notation for i in the reverse order: for $i = a_0 + pa_1 + p^2a_2 + p^3a_3 + \ldots$, where $a_j \in \{0, 1, \ldots, p-1\}$, we set

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

Perhaps, a more useful notion than discrepancy is "dispersion." The *dispersion* of a point set P, $d(P)$, measures the fitness of a sequence for applications like optimization by considering the furthest distance between any point in the space and its nearest sample point. We can visualize the importance of the dispersion to roadmap analysis by placing a ball with radius equal to the dispersion at each sample point; the entire sample space is now covered. Obviously, the fewer of these balls we have, the smaller our roadmap is (in terms of number of nodes); the smaller these balls are, the easier it is to connect a query to the roadmap. Going back to Figure 1, one can see that the dispersion in (a) is bigger than that in (b). Unfortunately,

there are no known methods for calculating the dispersion of a point set exactly. The dispersion can be approximated by sampling the space in a finite number of places and performing the calculation on these samples, which produces a lower bound on the dispersion. Indeed, approximating the dispersion of a point set might just be a good candidate for Monte Carlo or quasi-Monte Carlo Methods! Another lower bound is to compute the nearest-neighbor distance for each node, and divide the maximum of these by two.

As stated in Section 2, discrepancy and dispersion are related by $d(P) \leq [D(P)]^{1/d}$. Thus, low-discrepancy point sets generally have low dispersions. For Hammersley points, $d(P) < (1+b) \cdot N^{-1/d}$, where b is the largest prime used; for a Halton sequence, $d(P) < b \cdot N^{-1/d}$. The best bounds are thus obtained using the first $d-1$ and d primes, respectively, in which case the Hammersley points always produce a lower upper bound. See [20, 25] for discrepancy and dispersion bounds for other low-discrepancy sequences.

We have used Hammersley, Halton, and Faure points as inputs to Q-PRM algorithms to solve a variety of planning problems in a range of dimensions. Broadly speaking, Q-PRM has performance better than or equal to its PRM counterpart. We present some theoretical advantages of quasi-random over random points, as well as some compelling experiments, in the next section. More results appear in [6, 21].

3.3 Comparing PRM and Q-PRM

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

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

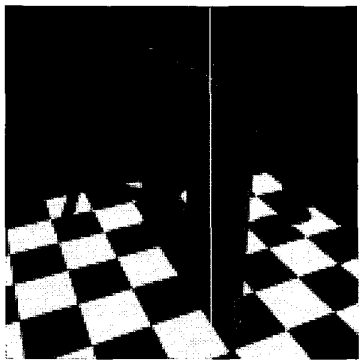


Figure 2: A 6-DOF planning problem.

Figure 2 shows a 6-DOF planning problem in which an elbow-shaped robot passes through a small opening. The Q-PRM solved the problem with 5908 nodes, and the PRM averaged 8020 nodes over 35 trials (the min and max for the PRM were 5551 and 14863, respectively).

In the above experiments, Hammersley points using the first $d-1$ primes were used in C-spaces of dimension d. In all our experiments, uniform random numbers were generated using the linear congruential generator found in MATLAB™ or the random source class of the LEDA/C++ library function. Hammersley-Halton low-discrepancy sequences were chosen using by coding the definitions in the previous subsection. The time to generate the quasi-random samples versus random samples was never a significant distinguishing factor.

We now turn to some theoretical comparisons of the performance of quasi-random versus random sampling. We begin here by exploring theoretical comparisons of discrepancy and dispersion from the literature. In the context of motion planning problems, discrepancy and dispersion are related to the uniform coverage of C-space. See Figure 4. Dispersion is also a direct measure of the maximum distance of any query to the roadmap and related to the ability to connect paths through narrow, twisting passages. Also, connection radii less than the dispersion of a point set may produce disconnected roadmap graphs.

The discrepancy for a set of N uniform random numbers in d dimensions is $N^{-1/2}$; that for low-discrepancy sequences is $O(N^{-1}(\log N)^d)$. Asymptotically (in the number of sample points N), the latter is smaller than the former. Since discrepancy is related to error in Monte Carlo integration, this has caused much interest in low discrepancy methods for that problem. However, if d is large and N is modest in size, the asymptotic bounds do not apply. Nevertheless, low-discrepancy points have still shown better performance than random samples in these exact situations [26].

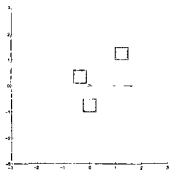


Figure 3: A two-link revolute manipulator.

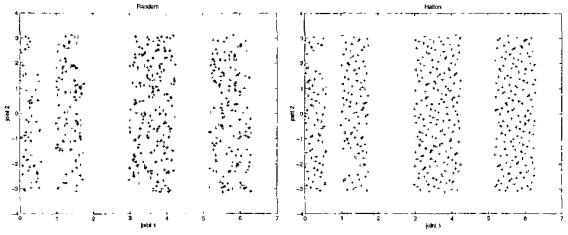


Figure 4: C-space coverage for the manipulator in Figure 3 for (left) PRM and (right) Q-PRM with Halton points. 900 nodes examined.

Dispersion bounds are also asymptotically superior for low-discrepancy points. The upper bound coming from discrepancy is $N^{-1/(2d)}$ for uniform random point sets. Compare this with $O((\log N)N^{-1/d})$ for low-discrepancy sequences. Typically, low-discrepancy point sets admit dispersion bounds (infinity norm) that are $O(N^{-1/d})$, as those reported for Hammersley and Halton points above. The dispersion of a random sequence (infinity norm) is $O(N^{-1/d}(\log N)^{1/d})$ almost surely [20]. Solving such formulas for N, it is clear that the number of points needed in the random setting becomes higher as the required dispersion decreases towards zero. See Figure 5.

It is interesting to note that in the table above, in 2 dimensions, we have $3(195)^{-1/2} \approx (464)^{-1/4}$ (which uses the discrepancy upper bound on dispersion for uniform random points; the constants in the a.s. bound are not known to us at this time). In the other dimensions, N is (sometimes many orders of magnitude) too small for the theoretical advantages of the quasi-random points to "kick in". Nevertheless, we empirically observe similar or better performance in terms of the number of nodes required to generate successful plans.

4 Quasi-random Lazy-PRM

A recent PRM variant called the Lazy-PRM has been proposed for the problem of answering single planning queries efficiently, as opposed to building an extensive roadmap prior to consideration of a planning query [4]. The resulting planner is sometimes very efficient in comparison to the original PRM.

The primary novelty of the Lazy-PRM is that the

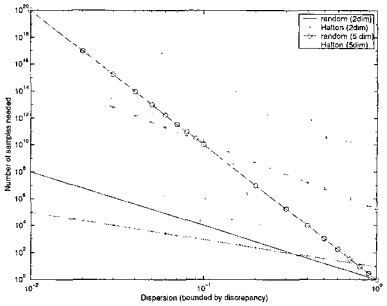


Figure 5: Comparison of number of points needed to achieve a desired dispersion, from discrepancy bounds.

roadmap is initially constructed without the use of a collision detector. All edges appear as if they were collision-free, regardless of the obstacles. Once an initial-goal query is given, the planner performs A^* search on the roadmap to find a solution. If any of the solution edges are in collision, they are removed from the roadmap, and the A^* search is repeated. Eventually, all edges may have to be checked for collision, but often the problem is solved well before this happens. If no solution is found, then more nodes may need to be added to the roadmap.

In addition to the low-discrepancy benefits of quasi-random sampling that were discussed in Section 3, another benefit arises in the case of a Lazy-PRM. In the first stage of the original Lazy PRM, a significant amount of time is spent on constructing a randomly-generated graph that may contain thousands of nodes and edges, and require thousands of nearest-neighbor queries; however, the roadmap encodes no true information because the obstacles are ignored. Using quasi-random sampling, it is possible to make a "lazier" PRM in which the initial graph is not even explicitly represented. Imagine, for example, if a grid was used as the PRM. The graph is essentially known in advance because all nodes, neighboring nodes, and edges are defined implicitly by the rules of the grid. If the grid is replaced with a low-dispersion set of quasi-random points that form a lattice, the benefits of both the grid-like structure and low-discrepancy can be exploited.

Lattices fall under the category of closed quasi-random methods, in which the number, N, of samples is fixed *a priori*. Suppose that the C-space is scaled so that $\mathcal{C} = [0,1]^d$. A sequence, $\{q_1, \ldots, q_N\}$ of N d-dimensional lattice points can be constructed by selecting the i^{th} point as $\frac{i\mathbf{z}}{N}$, in which $\mathbf{z}$ is a carefully-chosen integer vector with no components that have common factors with N, and the result is computed modulo one (only the fractional part is used). A procedure for choosing a value for $\mathbf{z}$ that yields low discrepancy, given N and d, is described in [24] (more elaborate lattice schemes are also given). It is well-known that the points in a lattice form an Abelian

group with respect to addition, and that all points can be specified in terms of a collection of d *generators*, $g_1, \ldots, g_d$. These can be considered as d independent basis vectors, from which all lattice points can be generated by taking integer linear combinations. By using generators, the location of all nearest neighbors can be obtained immediately by adding (or subtracting) one of the generators to the sample. For example, in the case of a 2D grid, the generators are $[1, 0]$, $[0, 1]$, which can be used to obtain the coordinates of standard four-neighbors. This idea generalizes to any lattice, and yields constant-time determination of neighbors.

It is actually possible to generate a family of low-discrepancy *embedded* lattices, in which each lattice contains twice as many samples as the previous one, and includes all of its samples. This is useful in the Lazy-PRM algorithm; when a solution is not found for a fixed N, a lattice of twice the "resolution" can be used, again without requiring explicit representation of the roadmap. The family of lattices is given by

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

where $r \in \{0, \ldots, d\}$, each $k_i \in \{0, 1\}$, the number of lattice points is $p^r N$, and p and N are relatively prime.

We next present an implementation of the lattice-based Lazy-PRM. The primary difference with respect to the method in [4] is that no initial roadmap is explicitly constructed: it is defined *implicitly* by the rules of a chosen lattice for fixed d and N. We also chose to run the A^* algorithm only once, and performed collision detection during the search. We had first implemented the iterative search and deletion scheme described in [4], but found it to be less efficient for our computed examples (for both the lattice-based and randomized Lazy-PRM). The implementation is in LEDA/C++, and uses the PQP collision detection package from the Univ. of North Carolina. We used values of $\mathbf{z}$ given in the appendix of [24].

We performed dozens of experiments on each of several examples. Three of these examples are displayed in Figures 2, 6 (left), and 6 (right). These represent 3D environments that contain a 6-DOF robot. The rotation portion of the C-space is parameterized using yaw-pitch-roll angles. We compare our lattice-based Lazy PRM implementation to our implementation of a randomized Lazy-PRM (note, however, that one can adjust many parameters that affect performance). For the randomized Lazy-PRM, we performed 25 trials on each example. The table shows the minimum, maximum, and average number of nodes for the randomized Lazy-PRM in the first three columns. The final column shows the number of nodes used by the lattice.

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

The following table shows computation times, with an additional column that shows the amount of precomputation time used by the randomized Lazy-PRM to build the initial roadmap.

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

The lattice-based Lazy-PRM shows dramatic performance improvements, primarily because it exploits the neighborhood structure of the lattice to avoid the precomputation required by the randomized Lazy-PRM. Furthermore, the number of nodes often appears to be in favor of the lattice-based PRM, for reasons discussed in Section 3.

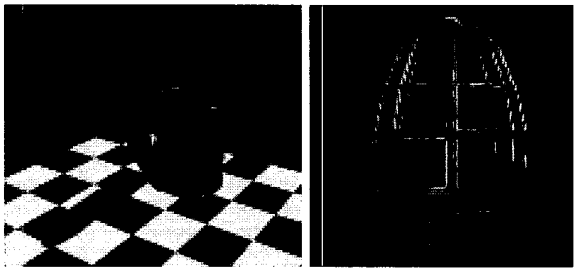


Figure 6: Left: Placing a feather (1184 triangles) into a cup (1632 triangles). Right: Getting a truck (22284 triangles) out of a cage (1032 triangles).

5 Discussion

We have investigated the use of quasi-random sampling for path planning by introducing deterministic quasi-random variants of both the original probabilistic roadmap (PRM) and the recent Lazy-PRM. Based on our experiments, quasi-random samples appear to offer performance improvements similar to those observed in other fields where Monte Carlo methods were replaced by quasi-Monte Carlo methods. The regular neighborhood structure of quasi-random lattices also led to performance benefits, in the context of a lazy evaluation. We emphasize the difficulty however, in providing conclusive experimental comparisons, given that there is no practical way to represent the distribution of problems on which these algorithms will be applied. Also, it is hard to compare deterministic, predictable methods to randomized methods, which yield varying results in multiple executions.

Another potential advantage of quasi-random sampling is that deterministic bounds on the performance of the planner can be derived. With random sampling, performance guarantees and completeness are measured probabilistically. When using deterministic samples, the planner is guaranteed to terminate in finite time with a solution, if a solution exists. The

proposed quasi-random PRMs are *resolution complete*, as opposed to *probabilistically complete*; however, determining performance bounds remains a topic of further research. Existing theoretical analysis of quasi-random sampling only shows asymptotic superiority to random sampling in terms of measures such as discrepancy and dispersion, which further complicates the problem of assuring superior performance for a small number of iterations. It is easy to design motion planning problems that require a certain level of discrepancy or dispersion for completion. Experimenting with them and relating these concepts to other notions, such as ϵ-goodness [10], is a topic of future study.

It is interesting to consider the spectrum from quasi-random sampling to random sampling. Intuitively, it appears as if random samples help to fight the curse of dimensionality by yielding methods that have little dependency on dimension. Quasi-random samples, such as the Hammersley sequence, appear to also fight the curse by offering advantages in terms of performance measures. Interestingly, the restriction of the samples to a lattice structure has little or no performance disadvantages in terms of discrepancy and dispersion [8]. This appears odd because lattices are a kind of skewed grid, and conventional wisdom indicates that grids are a poor choice because to obtain a fixed resolution, the number of samples must increase exponentially with dimension. The notion of resolution is comparable to dispersion, which guarantees each sample is within a prescribed displacement of other samples. The problem is that for *any* collection of N d-dimensional points in $[0,1]^d$, the dispersion, s, is bounded as $s \geq \frac{1}{2}N^{-\frac{1}{d}}$ [20]. Therefore, an exponential number of samples is needed to maintain a fixed dispersion or resolution, regardless of whether random sampling, quasi-random sampling, or grids are used.

Acknowledgments

We are grateful for the funding provided in part by NASA grant NAG2-1333 (Branicky) and NSF CAREER Award IRI-9875304 (LaValle). An earlier presentation of the Q-PRM appears in [6, 21]. Steve LaValle thanks Blake Hannaford for an enlightening discussion on sampling.

References

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

[2] N.M. Amato *et al.* OBPRM: An obstacle-based PRM for 3D workspaces. pp. 630–637. AK Peters, 1998.

[3] J. Barraquand *et al.* A random sampling scheme for path planning. *Int. J. Robotics Research*, 16(6):759–774, 1997.

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

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

[6] M.S. Branicky and K. Olson. Quasi-random roadmaps: Using deterministic low discrepancy point sets to break the curse of dimensionality in motion planning. Tech. Report. Ctr. Automation and Intelligent Systems Research, Case Western Reserve U., December 1999.

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

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

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

[10] D. Hsu *et al.* On finding narrow passages with probabilistic roadmap planners. In et al. P. Agarwal, ed., *Robotics: The Algorithmic Perspective*, pp. 141–154. A.K. Peters, Wellesley, MA, 1998.

[11] D. Hsu, J. C. Latombe, and R. Motwani. Path planning in expansive configuration spaces. In *IEEE Int. Conf. Robot. & Autom.*, pp. 2719–2726, 1997.

[12] M. Kahn and A. W. Marshall. Methods for reducing sample size in Monte-Carlo computations. *Oper. Res.*, 1:263–278, 1953.

[13] M.H. Kalos and P.A. Whitlock. *Monte Carlo Methods.* Wiley, New York, 1986.

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

[15] 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.

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

[17] S.M. LaValle, K.J. Moroney, and S.A. Hutchinson. Methods for numerical integration of high-dimensional probability densities with application to statistical image models. *IEEE Trans. on Image Processing*, 6(12):1659–1672, December 1997.

[18] J. Matousek. *Geometric Discrepancy: An Illustrated Guide.* Springer, Berlin, 1999.

[19] N.C. Metropolis and S.M. Ulam. The Monte-Carlo method. *J. Amer. Stat. Assoc.*, 44:335–341, 1994.

[20] H. Niederreiter. *Random Number Generation and Quasi-Monte-Carlo Methods.* SIAM, Phila., 1992.

[21] K. Olson. Quasi-random roadmap methods (QRM): Derandomizing probabilistic motion planning. Tech. Report. Ctr. for Automation and Intelligent Systems Research, Case Western Reserve U., July 2000.

[22] C. Pisula, K. Hoff, M. Lin, and D. Manocha. Randomized path planning for a rigid body based on hardware accelerated Voronoi sampling. *Proc. Workshop on Algorithmic Foundations of Robotics*, 2000.

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

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

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

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