Random Sampling in Geometric Applications

Mark Foskey

September 8, 2001

1 Introduction

This paper is an integrative survey of three works, one in the area of computational physics, one in computer graphics, and one in path planning. Each of these papers attacks a problem that could naturally be solved by sampling some high-dimensional space in a uniform grid. In each case, the relevant dimensionality is so high that uniform sampling is infeasible, and each paper relies on random sampling to overcome that difficulty. An additional connection linking these three papers is that each faces a problem for which the sampling must be *biased* in a systematic way.

The earliest of the three is Metropolis et al. [8] (henceforth $\mathrm{M(RT)}^2$), which proposes a general approach to estimating the integrals describing the equilibrium properties of fluids. The second paper, by Veach and Guibas [10] (henceforth VG), uses the approach of $\mathrm{M(RT)}^2$ to simulate the illumination of a scene by an integral method. Finally, the third paper, by Kavraki et al. [6] (henceforth KŠLO), randomly generates a graph of robot configurations for later searching when a path for the robot must be found.

While the three papers are united by their use of random sampling and a need for a bias in the sample distribution, there are especially noteworthy differences distinguishing KŠLO, on the one hand, from $M(RT)^2$ and VG, on the other. The latter two papers are both concerned with the computation of an integral, while the sampling in KŠLO is designed to generate a graph. As a consequence, for $M(RT)^2$ and VG, the desired probability density function for the sampling can be characterized exactly in terms of the function being integrated. For the motion planning problem of KŠLO, on the other hand, all that is known is that a greater density of samples is needed in "narrow passages" in configuration space. It is the goal of the present paper to investigate the possibility of more directly applying the ideas of $M(RT)^2$ and VG to the problem domain of motion planning.

The rest of the paper is organized as follows. Section 2 discusses Monte Carlo methods in general, and Section 3 describes the particular algorithm of $M(RT)^2$. Section 4 covers the "Metropolis Light Transport" algorithm of VG. Section 5 discusses the "Probabilistic Roadmap" algorithm described in KŠLO, and Section 6 discusses how the Metropolis algorithm might be applied in a planning setting. The paper concludes with Section 7.

2 Monte Carlo methods

Because Monte Carlo integration is a foundational method for two of the algorithms, I will begin by discussing that approach in general terms. This discussion is based on the presentation in Kalos and Whitlock [4]. Suppose we must numerically approximate

$$\int_R f(\mathbf{x})d\mathbf{x}$$

over some region R in $\mathbf{R}^n$. A straightforward approach is to use the n-dimensional analogue of the Riemann sum of elementary calculus. Suppose we can partition the region R into a collection of sub-regions R_i, each of which has n-dimensional volume v_i. For each subregion, we choose a representative point $\mathbf{x}_i \in R_i$. Then the value of the integral can be approximated as

$$\int_R f(\mathbf{x})d\mathbf{x} \approx \sum_i v_i f(\mathbf{x}_i).$$

When the dimension n is large, there are a number of problems with this approach:

- If R is an irregular region, partitioning it into sub-regions of known volume may be hard or impossible.
- Even if R is not irregular, the most natural partitioning of it may require a prohibitive number of sub-regions. For instance, if R is a cartesian product of intervals in $\mathbf{R}^n$, and each interval is subdivided into a fixed number of sub-intervals, then the total number of cells is exponential in n.
- Even if a clever partitioning of R into a manageable number of uniform regions is found, it is difficult to analyze how well the samples are capturing the variation in the integrand.

All of these problems may be solved by means of sampling. First, we recognize that computing integrals of functions can be reduced to computing their means. That is, if the mean of f over R is $\langle f \rangle$,

$$\int_R f d\mathbf{x} = \langle f \rangle \int_R d\mathbf{x}.$$

We can actually assume R is simply a box (i.e., a product of intervals) by setting $f(\mathbf{x}) = 0$ outside of the original region R, eliminating any difficulties in computing $\int_R d\mathbf{x}$.

The mean $\langle f \rangle$ can be estimated from a set of K samples by

$$\langle f \rangle \approx \frac{1}{K} \sum_{i=1}^{K} f(\mathbf{x}_i),$$

where the $\mathbf{x}_i$ are chosen uniformly at random from R. Actually, any choice of $\mathbf{x}_i$ will provide *some* estimate of $\langle f \rangle$, and a dense regular sampling will be equivalent to the Riemann sum approach above. The benefit of the random sampling is that one can characterize the quality of the estimate in terms of the statistical properties of f and the number of samples K. To be precise, the estimate $F = \frac{1}{K} \sum f(\mathbf{x}_i)$ is itself a random variable, and, under reasonable hypotheses, F has mean equal to the mean of f and variance given by

$$\mathrm{var}\{F\} = \frac{1}{K} \mathrm{var}\{f(x)\}.$$

Keeping in mind that a Monte Carlo computation yields a single instance of the random variable F, we see that $\mathrm{var}\{F\}$ says something about how likely it is for a given measurement to differ substantially from $\langle F \rangle = \langle f \rangle$. If K is large enough, F will be approximately Gaussian by the central limit theorem, and these probabilities can be quantified explicitly.

Because the reliability of the estimated integral is tied to $\mathrm{var}\{F\}$, it is important to be able to estimate it and, if possible, to make it small. We saw above that the variance is inversely proportional to K, and proportional to $\mathrm{var}\{f(x)\}$. The relationship to K is not problematic, but there are two difficulties involving $\mathrm{var}\{f(x)\}$.

The first problem is that $\mathrm{var}\{f(x)\}$ itself must be estimated, which involves computing another integral that is just as complicated as the first. However, our estimate of $\mathrm{var}\{f(x)\}$ need not be very precise, and we can estimate it using the same samples that we use to estimate $\langle f \rangle$.

The second and much larger problem is that the variance may be so large that the number of samples required may be very great. In the next section we discuss one solution to this problem.

3 The Metropolis Algorithm

$M(RT)^2$ is a seminal paper in the literature of Monte Carlo methods, so much so that Kalos and Whitlock write "In much of the literature of statistical physics, when a Monte Carlo calculation is mentioned what is meant is an application of the method of Metropolis et al." ([4], p. 78). The key benefit of the algorithm is that it works well in situations where nonuniform sampling is desired.

In this section we will briefly describe the Metropolis algorithm. As a motivating example, we will use the problem in computational thermodymics to which the algorithm was originally applied in $M(RT)^2$.

3.1 The Physical Problem

Imagine a fluid made up of molecules represented as a set of spheres, each of which repels the others. Let F be a physical quantity that depends on the state of the system. Applying standard principles of statistical mechanics, Metropolis et al. observe that the equilibrium value $\bar{F}$ of F can be computed as

$$\bar{F} = \frac{\int F(\mathbf{q}) e^{-E(\mathbf{q})/kT} d\mathbf{q}}{\int e^{-E(\mathbf{q})/kT} d\mathbf{q}}, \tag{1}$$

where $E(\mathbf{q})$ is the energy density at state $\mathbf{q}$, T is the thermodynamic temperature (that is, temperature in Kelvins), and k is Boltzmann's constant. This equation can be understood on the basis that $e^{-E(\mathbf{q})/kT}$ is an unnormalized density function for the probability of the system being in state $\mathbf{q}$.

The variable $\mathbf{q}$ represents a state in $3n$-dimensional state space, where n is the number of particles in the simulation. In a nontrivial simulation n could be several hundred, making Monte Carlo integration really the only feasible approach.

The difficulty is that most of the states will contribute very little to the integral. If the particles are densely spaced, which is the case of interest, then for most $\mathbf{q}$ the energy $E(\mathbf{q})$ will be large, making $e^{-E(\mathbf{q})/kT}$ quite small. If sampling for Monte Carlo integration were uniform, it would take a prohibitive number of samples to generate enough to sufficiently sample the significant portions of state space. In terms of the discussion above of Monte Carlo methods in general, the variance of $F(\mathbf{q})e^{-E(\mathbf{q})/kT}$ is likely to be quite large, requiring a large number of samples in order to achieve an estimate with an acceptably small variance. The same problem applies for the integral in the denominator.

3.2 The Solution

The solution is to arrange for the samples to be chosen with probability density proportional to $e^{-E(\mathbf{q})/kT}d\mathbf{q}$. For a set of samples $\{\mathbf{q}_i\}$ with that density, we can estimate $\bar{F}$ by

$$\bar{F} = \frac{\int F(\mathbf{q})e^{-E(\mathbf{q})/kT}d\mathbf{q}}{\int e^{-E(\mathbf{q})/kT}d\mathbf{q}} \approx \sum_{i=1}^{n} F(\mathbf{q}_i).$$

Notice that we need not perform two separate Monte Carlo integrations, one for the numerator and one for the denominator. The quotient in fact defines the very same weighted mean that is estimated by $\sum F(\mathbf{q}_i)$.

The key contribution of $\mathrm{M(RT)}^2$ is to devise an efficient way to choose samples with the desired probability density. Since the approach is quite general, consider a function $D(\mathbf{x})\colon \mathbf{R}^n \rightarrow R$, which need not integrate to 1. To generate a set of samples with density (in the limit) proportional to D, begin with an arbitrary sample $\mathbf{x}_0$ and somehow perturb it, getting a new sample $\mathbf{x}'_0$. Then define a probability $P(\mathbf{x}_0, \mathbf{x}'_0) = \min(1, D(\mathbf{x}'_0)/D(\mathbf{x}_0))$ and, with probability $P(\mathbf{x}_0, \mathbf{x}'_0)$, set $\mathbf{x}_1 = \mathbf{x}'_0$. Otherwise, set $\mathbf{x}_1 = \mathbf{x}_0$ (that is, $\mathbf{x}_0$ will be used twice in the simulation). In M(RT)^2, the samples are perturbed by adding a vector chosen uniformly at random from a box of the form $[-\delta, \delta]^n$.

In words, we can interpret the function D as a measure of the desirability of a sample. The above procedure automatically keeps a new sample if it is more desirable than the previous one. If the new sample is less desirable, the procedure keeps it with a probability proportional to the ratio of the new desirability to the old. Metropolis et. al prove that, with this procedure, the sample density converges over time to a density proportional to D.

4 Metropolis Light Transport

The problem of global illumination in computer graphics has motivated many references to the physics literature. In VG, Veach and Guibas formulate the problem as a high-dimensional Monte Carlo integration and apply the Metropolis algorithm. In this section we first discuss their integral formulation, which is different from the standard formulation in terms of an integral equation. Then, we discuss the reasons that uniform Monte Carlo integration works poorly in solving the integral. Finally, we discuss two procedures they use to improve the performance of the algorithm.

4.1 The Rendering Integral

Before considering true rendering, it will be helpful to consider the problem of the total flux arriving at a point $\mathbf{y} \in \mathbf{R}^3$ originating at a point light source $\mathbf{x}$, in the presence of an environment containing objects that may scatter or transmit light.

For a fixed n, let $(\mathbf{a} = \mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n = \mathbf{b})$ be a sequence of points such that each $\mathbf{x}_i$ with $0 < i < n$ lies on a surface in the environment. Such a sequence defines a path from $\mathbf{a}$ to $\mathbf{b}$ simply connecting each point to the next by a straight line segment. Let F_n be the total flux arriving from $\mathbf{b}$ at $\mathbf{a}$ via such n-hop paths. Then the total flux F will be given by

$$F = \sum_{n=0}^{\infty} F_n.$$

Each term F_n can be computed as the following integral:

$$F_n = \int_{\mathcal{M}^{n-1}} L_e(\mathbf{x}_0, \mathbf{x}_1) \prod_{i=0}^{n-1} G(\mathbf{x}_i, \mathbf{x}_{i+1}) \prod_{i=0}^{n-2} f_s(\mathbf{x}_i, \mathbf{x}_{i+1}, \mathbf{x}_{i+2}) dA(\mathbf{x}_0) \cdots dA(\mathbf{x}_n).$$

The integral is taken over the set $\mathcal{M}$ of all surfaces in the environment, and A is the area measure on $\mathcal{M}$. $L_e(\mathbf{x}_0, \mathbf{x}_1)$ is the light emitted from $\mathbf{x}_0$ towards $\mathbf{x}_1$, and f_s is the bidirectional scattering distribution function, or BSDF (Veach and Guibas use the term BSDF rather than BRDF to emphasize that transmission is modeled along with reflection). The function $G(\mathbf{x}, \mathbf{y})$ combines the visibility of $\mathbf{y}$ from $\mathbf{x}$ with the form factor for transmission intensity, and has the formula

$$G(\mathbf{x}, \mathbf{y}) = V(\mathbf{x}, \mathbf{y}) \frac{|\cos(\theta_{\mathbf{x}}) \cos(\theta_{\mathbf{y}})|}{\|\mathbf{x} - \mathbf{y}\|^2},$$

where $V(\mathbf{x}, \mathbf{y})$ is 1 if $\mathbf{x}$ and $\mathbf{y}$ are visible from each other, and 0 if not. If $\mathbf{x}$ or $\mathbf{y}$ is a point not on a surface (i.e., either $\mathbf{a}$ or $\mathbf{b}$), then the corresponding cosine factor is simply dropped from G.

Mathematically, any sum can be characterized as an integral over a discrete measure, and a sum of integrals can be formulated as a single integral over a modified measure. Veach and Guibas do this, writing the sum $\sum F_n$ as a single integral

$$F = \int_{\Omega} f(\bar{x}) d\mu(\bar{x}).$$

In this integral, Ω is the space of all paths $\bar{x} = (\mathbf{x}_0, \ldots, \mathbf{x}_k)$ as k ranges over the positive integers. For such a path, $f(\bar{x})$ is defined by

$$f(\bar{x}) = L_e(\mathbf{x}_0, \mathbf{x}_1) \prod_{i=0}^{n-1} G(\mathbf{x}_i, \mathbf{x}_{i+1}) \prod_{i=0}^{n-2} f_s(\mathbf{x}_i, \mathbf{x}_{i+1}, \mathbf{x}_{i+2}).$$

The symbol $d\mu(\bar{x})$ can be taken as notation for $dA(\mathbf{x}_0) \cdots dA(\mathbf{x}_k)$, with the value of k determined by $\bar{x}$. This formal procedure defines $\mu(\bar{x})$ as a measure on the space Ω.

The benefit of formulating an infinite summation of integrals into a single integral is that it provides a unified framework for performing a Monte Carlo integration. A single procedure can generate paths of varying length which all help estimate the integral.

The foregoing strictly involved the flux from a light source to a specific point. To render an image, we need to compute a similar integral for each pixel. We make the following changes:

- We no longer restrict $\mathbf{x}_0$ to a fixed point in space, but instead restrict it to $\mathcal{M}$ and assume that all light sources are a part of some surface in the scene.

- For each pixel we multiply $f(\bar{x})$ by an importance function $W^j(\bar{x})$, where j is the index of the given pixel. W^j in fact only depends on the final two points $\mathbf{x}_{n-1}$, $\mathbf{x}_n$ of $\bar{x}$, and it indicates what fraction of the light arriving at $\mathbf{x}_n$ from the direction of $\mathbf{x}_{n-1}$ makes a contribution to the scene. W^j will be nearly 1 if the ray from $\mathbf{x}_{n-1}$ to $\mathbf{x}_n$ passes through the center of the jth pixel on the image plane, rapidly decreasing to zero as the direction deviates.

Thus, for each pixel a different integral must be computed. However, in the Monte Carlo process all the integrals can be computed at the same time, as each path that makes a contribution to some pixel can be applied toward that pixel as it is generated.

4.2 Sampling Problems

For the rendering integral, true uniform sampling is in fact impossible, because the space Ω is infinite dimensional. However, there must be an upper limit on the path length in any case, so it is possible to imagine uniform sampling from the space of available paths. This still works poorly for a number of reasons.

- Since a ray of light loses energy with every bounce, most long paths will make little or no contribution to the scene and will thus be wasted.
- If the light source is hidden, so that the light it provides must first pass through a small opening to illuminate the scene, then very few of the paths generated will pass from the source, through the opening, and to the eye.
- In scenes involving such phenomena as caustics and highly reflective surfaces, certain families of light paths will have much more intensity than others. That is, the illumination function $f(\bar{x})$ will have a high variance, and hence uniform sampling will not do well.

Veach and Guibas solve these problems by using the Metropolis alorithm. They apply a number of refinements to improve the performance of the algorithm.

- In conventional Metropolis integration, the first few samples must be thrown away because they are biased by the location of the initial sample. Veach and Guibas resolve this problem by performing an initial estimate of the overall image brightness using a simpler sampling method, and then choosing a path approximately matching that brightness to seed the image.

- The authors combine several different strategies for generating samples, so as to improve the chances of getting enough samples in known problem areas. A benefit of the Metropolis approach is that it eliminates any bias that could otherwise be introduced by a specific sampling strategy.

5 Motion Planning

In both $M(RT)^2$ and VG, a random sampling process is used to compute a numerical value arising in a high-dimensional geometric context. Motion planning is another geometric problem that is naturally formulated in higher dimensional space, and random sampling has been found useful for it as well. In this section we will give an overview of the problem, and a discussion of the probabilistic roadmap (PRM) approach, which is the approach that most closely resembles Monte Carlo integration.

5.1 Problem Overview

Consider an object $\mathcal{R}$ (the robot) which is free to move about an environment. We will assume $\mathcal{R}$ is 3-dimensional, and it may be articulated or rigid. The environment $\mathcal{E}$ contains various obstacles. The classical motion planning problem is to find an intersection-free path for $\mathcal{R}$ from a specified initial configuration (position and orientation) to a specified goal.

A configuration can be specified by giving the location of $\mathcal{R}$, its orientation, and the positions of its joints. This means that the space of configurations, known simply as configuration space (or C-space) is at least

6-dimensional, and more in the case of articulated robots. The obstacles in $\mathcal{E}$ determine forbidden regions of C-space, and the complement of the forbidden regions is known as the *free space*. The planning problem then reduces to finding a path for a point robot through the free space.

Provably correct motion planning algorithms have been devised. An example is the roadmap method of Canny [2] (called the "silhouette algorithm" by Latombe [7]). Unfortunately, all known examples take time exponential in the dimension of the C-space, so they are infeasible for problems with many degrees of freedom (dof).

5.2 Probabilistic Roadmap Planning

Two groups working independently, Kavraki and Latombe [5] and Overmars and Švestka [9], developed a randomized approach to planning that, while not guaranteed to succeed, works well in practice. The groups then collaborated to write KŠLO.

Their approach separates into a learning phase and a query phase. In the learning phase, a graph, the *roadmap*, is generated in the free space. In the query phase, a simple planning algorithm is used to link the initial and goal configurations to the graph, and then a standard graph search algorithm is performed to find the actual path.

The learning phase. The goal of the roadmap is to represent as faithfully as possible the connectivity of the free space. In KŠLO it is constructed by repeating the following steps:

- Generate a configuration $\mathbf{c}$ at random.
- For some fixed integer k, determine the nearest k nodes in the existing roadmap, measured using an appropriate distance metric on C-space.

- If there are any collision free line segments in configuration space connecting **c** to one of the k vertices, add **c** to the graph as a new vertex, and add one edge for each component of the roadmap for which **c** was successfully connected. That is, the k nearest nodes may belong to multiple connected components of the roadmap. For each such component, the shortest segment connecting **c** to one if its nodes is kept as a graph edge.

Over time, this procedure tends to produce a graph that is easily accessible from most portions of the free space, while at the same time beging likely to have a single connected component for each component of $\mathcal{F}$.

The query phase. This phase is quite simple. An attempt is made to link the initial and goal configurations to the roadmap using the same method by which nodes are added to the graph in the learning phase. If that attempt fails, a random walk is performed and a new attempt is made. This process may be repeated as long as time permits. Once the initial and goal configurations are linked, a path is found by a breadth-first search.

5.3 Narrow Passages

The most difficult problems for PRM planners are ones for which the robot must pass through a narrow passage to reach the goal. In such cases a great many samples must be generated to get a high enough density of samples in the narrow passage. Ideally, we would like to sample in a nonuniform way. This is directly analogous to the problem of indirect lighting through a small opening in global illumination, although the two problems do not map directly to one another. Each sample in Metropolis light transport is a distinct solution to a specialized, simple planning problem.

A number of approaches to nonuniform sampling in PRM planning have been proposed. In KŠLO, after the initial roadmap construction process, nodes are analyzed for their likelihood of being in a narrow passage (or other difficult area). They are considered to be in a difficult area if, over the course of the roadmap building process, a high proportion of the attempts to connect them to other nodes ended in failure. Such nodes are then "expanded" by performing a random walk, and then trying to link the endpoint of the random walk to other components in the same way that new nodes were added to the graph. Each random walk adds a new edge and a new node to the graph.

Hsu et al. [3] initially build the roadmap as if the obstacles had all been shrunk by a certain amount. Then configurations found to be in collision with the original obstacles are replaced with nearby free samples generated by localized sampling. This process has the effect of sampling more densely near object boundaries, and hence also more densely in narrow passages.

Alternatively, Wilmarth, Amato, and Stiller [11] retract each sample, whether or not is initially in $\mathcal{F}$, to approximately the nearest point on the medial axis of $\mathcal{F}$. The medial axis is defined in terms of a convenient metric on C-space.

6 Metropolis Planning?

All of the known algorithms for nonuniform PRM sampling use heuristics to increase the sampling density. There is no formal measure of what the density should be in terms of the geometry of free space, and thus no attempt to sample from some specified distribution. Given a sensible density function that could be evaluated efficiently, the Metropolis algorithm could be used to generate samples with that density. In this section, we outline such an approach and discuss its potential benefits and difficulties.

6.1 Defining a Density Function

Criteria. We would like to sample so as to produce a graph with the following properties:

- **Connectivity:** It should have one connected component for each component of $\mathcal{F}$.
- **Coverage:** Every point in $\mathcal{F}$ is visible (in the sense that there is a line of sight in C-space) from some node.
- **Parsimony:** There are as few nodes as possible.

Since we are using a randomized approach, we cannot guarantee that any of the above desiderata are satisfied. Instead we view them as goals at which to aim.

The goals of coverage and parsimony suggest that each node should be able to see a roughly constant number of other nodes. Strictly speaking, this condition of *constant neighbor density* is neither necessary nor sufficient for full coverage, but it will tend to cause the nodes to spread through the scene in a way that improves coverage.

The goal of connectivity-with-parsimony is harder to pin down. I mention three qualitative observations that have appeared in the literature. First, there is the simple observation that samples should be denser in narrow passages, which is made in KŠLO and elsewhere. Second, many authors have noted that it can help if sampling is denser near the boundary of $\mathcal{F}$ [1, 3]. The boundary of $\mathcal{F}$ (denoted $\partial\mathcal{F}$) serves as a lower dimensional structure that reflects the connectivity of $\mathcal{F}$. (A connected component of $\mathcal{F}$ can have a boundary with multiple connected components. However, if connected components of $\partial\mathcal{F}$ that are visible to each other are all linked, the resulting structure will have the same connectivity as $\mathcal{F}$.) Finally, another subdimensional structure that respects the connectivity of $\mathcal{F}$ is its medial axis, which is employed by Wilmarth, Amato, and Stiller as mentioned above.

A possible density function. As an example of how a Metropolis-based sampling scheme might work, I propose the following density function: for a given configuration $\mathbf{c}$, define $f(\mathbf{c})$ to be the reciprocal of the (n-dimensional) volume of the subset of $\mathcal{F}$ visible from $\mathbf{c}$. This density yields approximate constant neighbor density, and it will cause denser sampling in narrow passages. It does not, however, lead to greater density near boundaries in open areas.

To evaluate this function, let S^{n-1} be the $n-1$ sphere in $\mathbf{R}^n$, viewed as a space of directions in C-space. Let the function $\rho_{\mathbf{c}}\colon S^{n-1} \to \mathbf{R}_{\geq 0}$ be defined so that, for $t \in S^{n-1}$, $\rho_{\mathbf{c}}(t)$ is the distance from $\mathbf{c}$ to $\partial\mathcal{F}$ along the direction through t. Then the volume visible from $\mathbf{c}$ is proportional to

$$\int_{S^{n-1}} \rho_{\mathbf{c}}(t)^n dt$$

Since the Metropolis algorithm only yields a density proportional to the given function, we need not concern ourselves with constant factors. Furthermore, it will be easier to estimate the expected value $\langle(\rho_{\mathbf{c}}(t)^n)\rangle_{S^{n-1}}$ than the original integral. Thus, let us denote that expected value by $V(\mathbf{c})$, and define $f(\mathbf{c}) = 1/V(\mathbf{c})$.

We can then estimate $V(\mathbf{c})$ by the following algorithm. Before the algorithm can be applied, a moderate number of samples are generated uniformly at random, and the locations of samples found to be outside $\mathcal{F}$ (the *rejected samples*) are recorded. Then, for a given $\mathbf{c}$, $V(\mathbf{c})$ may be estimated as follows.

- Subdivide C-space into a fixed number of volumes fanning out approximately uniformly from **c**. The simplest way to do this is to subdivide by axis-aligned planes passing through **c**, for instance yielding octants if the C-space is 3-dimensional. Since this yields 2^n regions given n dof, more regions than that are likely to be infeasible.

- For each region r_i, determine the distance r_i to the rejected sample (in that region) nearest **c**.

The resulting collection samples gives an upper bound, along each of a sample of directions, of the distances from **c** to $\partial\mathcal{F}$. We define our estimate simply to be

$$\frac{1}{n}\sum r_i^n.$$

Henceforth, $f(\mathbf{c})$ will refer to this estimate, rather than the actual density.

Using the Density Function Using the density estimate, the learning phase of a planning algorithm could go as follows.

- Generate a fixed number of samples in C-space uniformly at random.

- Apply the Metropolis algorithm using the density estimate $f(\mathbf{c})$ to generate more samples.

- Attempt to link free samples to others nearby, building a graph.

Discussion With a good density function, a Metropolis-based sampling method might be a step forward in handling scenes with narrow passages. However, the proposed algorithm faces a number of difficulties. Evaluating the estimate $f(\mathbf{c})$ is likely to be time consuming, and, in a high-dof situation, the results will be quite noisy. Computing the tenth power of the distance, for instance, will be very sensitive to the inital values. The algorithm is suggested more as an indication of a possible research direction than a solution to the problem.

7 Conclusion

Motion planning has features that are reminiscent of light transport, but there are key differences. The Metropolis algorithm has been shown to help deal with difficulties in light transport that are similar to the "narrow passage" problem in motion planning. We have discussed one approach to applying the Metropolis algorithm to resolve these difficulties in motion planning.

References

[1] Nancy M. Amato and Yan Wu. A randomized roadmap method for path and manipulation planning. *IEEE Conference on Robotics and Automation*, 1996.

[2] J.F. Canny. *The Complexity of Robot Motion Planning*. ACM Doctoral Dissertation Award. MIT Press, 1988.

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

[4] M. Kalos and P. Whitlock. *Monte Carlo Methods*, volume I: Basics. John Wiley & Sons, 1986.

[5] L. Kavraki and J. C. Latombe. Randomized preprocessing of configuration space for fast path planning. *IEEE Conference on Robotics and Automation*, pages 2138–2145, 1994.

[6] Lydia E. Kavraki, Petr Švestka, Jean-Claude Latombe, and Mark H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, 1996.

[7] J.C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[8] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[9] M. H. Overmars and P. Švestka. A probabilistic learning approach to motion planning. In *Algorithmic Foundations of Robotics*. A. K. Peters, Wellesley, MA, 1995.

[10] Eric Veach and Leonidas J. Guibas. Metropolis light transport. In *Computer Graphics Annual Conference Series (SIGGRAPH)*, 1997.

[11] Steven A. Wilmarth, Nancy M. Amato, and Peter F. Stiller. Motion planning for a rigid body using random networks on the medial axis of the free space. *Proc. of the 15th Annual ACM Symposium on Computational Geometry (SoCG'99)*, 1999.