

Monte Carlo and Importance Sampling

Phani Srikar

January 4, 2025

1 Introduction

In computer graphics, numerical integration is essential for solving equations like global illumination, ray tracing, and the rendering equation integrals, since it's efficient with discrete machine like computers over analytical methods. Efficient discrete sampling methods determine the accuracy and performance of these techniques.

In this article, we explore various sampling methods, starting from Monte Carlo methods to more sophisticated techniques like Importance Sampling and Quasi-Random sampling functions like Halton and Hammersley sequences.

By the end, you'll understand the trade-offs between different sampling techniques, how to use them. I will try to provide derivations, references and examples wherever I can.

2 Monte Carlo Sampling

Monte Carlo sampling uses statistical random sampling to evaluate integrals, by trying to reduce variance as much as possible. It is particularly useful in higher-dimension domains where we cannot estimate the PDF.

Example: Estimating the Integral of $f(x) = x^2$

Consider the integral of the function $f(x) = x^2$ over the interval $[0, 1]$, which we want to estimate using Monte Carlo sampling. The analytical solution to this is:

$$I = \int_0^1 x^2 dx = \left[\frac{x^3}{3} \right]_0^1 = \frac{1}{3} \approx 0.3333.$$

To estimate this integral using Monte Carlo sampling, we randomly sample N points x_1, x_2, \dots, x_N uniformly from the interval $[0, 1]$, and evaluate the function $f(x) = x^2$ at each of these points. The integral can be approximated by averaging the function values at these random points and then scaling by the length of the interval $b - a$, where $a = 0$ and $b = 1$ for our example.

The Monte Carlo estimate of the integral is:

$$I \approx \frac{b-a}{N} \sum_{i=1}^N f(x_i)$$

where:

- x_1, x_2, \dots, x_N are uniformly sampled points from the interval $[0, 1]$

- $f(x_i) = x_i^2$ is the value of the function at each point

For example, if we sample 5 random points $x_1 = 0.2, x_2 = 0.8, x_3 = 0.3, x_4 = 0.7, x_5 = 0.5$, we compute the function values:

$$f(x_1) = 0.2^2 = 0.04, \quad f(x_2) = 0.8^2 = 0.64, \quad f(x_3) = 0.3^2 = 0.09, \quad f(x_4) = 0.7^2 = 0.49, \quad f(x_5) = 0.5^2 = 0.25.$$

The Monte Carlo estimate of the integral is the average of these values:

$$I \approx \frac{(1-0)}{5}(0.04 + 0.64 + 0.09 + 0.49 + 0.25) = 0.302.$$

This is an approximation of the exact integral $\frac{1}{3}$. As N increases, the Monte Carlo estimate will converge to the exact value of the integral.

Cons: High variance

While Monte Carlo sampling is a powerful technique, it can still suffer from high variance due to random fluctuations in the sampled points. In this example, some regions of the interval might have more points clustered together, leading to inaccurate estimates in those regions. This is known as **clumping** of samples.

3. Importance Sampling

Importance sampling optimizes Monte Carlo by focusing samples in regions where the integrand contributes most to the result. Instead of uniform sampling, we sample from a distribution $p(x)$ that approximates the shape of the function $f(x)$. **Basically slapping Monte Carlo samples with a PDF (*probability distribution function*) reduces the variance and convergence rate.**

Adjusting the Integral When sampling from $p(x)$, the integral becomes:

$$I = \int \frac{f(x)}{p(x)} p(x) dx.$$

The Monte Carlo approximation is then:

$$I \approx \frac{1}{N} \sum_{i=1}^N \frac{f(x_i)}{p(x_i)},$$

where $x_i \sim p(x)$.

Example: Cosine-Weighted Hemisphere Sampling For diffuse shading, sampling directions proportional to $\cos(\theta)$ reduces variance. The PDF for cosine-weighted sampling is:

$$p(\omega_i) = \frac{\cos(\theta)}{\pi}.$$

The Monte Carlo estimate becomes:

$$L_o \approx \frac{1}{N} \sum_{i=1}^N \frac{L_i(\mathbf{x}, \omega_i) f_r(\omega_i, \omega_o) (\mathbf{n} \cdot \omega_i)}{p(\omega_i)}.$$

Substituting $p(\omega_i)$, the weight simplifies, and we achieve more efficient integration.

4. Quasi-Random Sequences: Halton and Hammersley

Quasi-random (aka low discrepancy) sequences are deterministic sequences that aim to uniformly cover a given domain. They are particularly useful in numerical integration and rendering tasks where efficient sampling is required. This section will derive and explain the relationship between the radical inverse function, Van der Corput sequence, Hammersley sequence, and Halton sequence. We start by defining the assumptions of a prime base.

2.1 Prime Base and Arithmetic Progression

Let p be a prime number. Any non-negative integer k can be expressed in the base p as:

$$k = a_0 + a_1p + a_2p^2 + \cdots + a_mp^m,$$

where $0 \leq a_i < p$ are the digits of k in base p . The summation above represents an arithmetic progression where each term contributes to the positional representation of k .

To normalize this representation into the unit interval $[0, 1)$, we divide k by increasing powers of p , effectively creating a fractional expansion:

$$\phi_p(k) = \frac{a_0}{p} + \frac{a_1}{p^2} + \frac{a_2}{p^3} + \cdots + \frac{a_m}{p^{m+1}}.$$

This function $\phi_p(k)$ is known as the **radical inverse function** in base p .

2.2 Radical Inverse Function

The radical inverse function $\phi_p(k)$ essentially mirrors the digits of k in base p and places them after the decimal point. For a given base p :

$$\phi_p(k) = \sum_{i=0}^m \frac{a_i}{p^{i+1}},$$

where a_i are the coefficients (digits) in the base- p expansion of k .

Example (Base 2): For $k = 5$, its binary representation is 101_2 :

$$\phi_2(5) = \frac{1}{2^1} + \frac{0}{2^2} + \frac{1}{2^3} = 0.625.$$

The radical inverse function is the core building block for constructing quasi-random sequences such as the Van der Corput, Hammersley, and Halton sequences.

2.3 Van der Corput Sequence

The **Van der Corput sequence** is a one-dimensional low-discrepancy sequence constructed directly from the radical inverse function. For base $p = 2$:

$$x_k = \phi_p(k), \quad k = 0, 1, 2, \dots$$

Properties: - One-dimensional. - Uniformly distributed in $[0, 1)$. - Useful for basic low-discrepancy sampling in one dimension.

Example (Base 2): For $k = 0, 1, 2, 3, 4, \dots$:

$$x_k = \{0, 0.5, 0.25, 0.75, 0.125, \dots\}.$$

2.4 Hammersley Sequence

The **Hammersley sequence** generalizes the Van der Corput sequence into multiple dimensions. It combines the radical inverse function for the first $d - 1$ dimensions with a uniform scaling for the last dimension. For a sequence of N samples in d -dimensions:

$$\mathbf{x}_k = \left(\phi_{p_1}(k), \phi_{p_2}(k), \dots, \phi_{p_{d-1}}(k), \frac{k}{N} \right),$$

where p_1, p_2, \dots, p_{d-1} are distinct prime bases.

Properties: - Multi-dimensional. - First $d - 1$ dimensions use radical inverse functions. - The last dimension is uniformly scaled by $\frac{k}{N}$.

Example (2D Hammersley Sequence, Bases 2 and 3): For $N = 5$, the sequence is:

$$\mathbf{x}_k = \left(\phi_2(k), \frac{k}{5} \right), \quad k = 0, 1, 2, 3, 4.$$

2.5 Halton Sequence

The **Halton sequence** is a further generalization of the Van der Corput sequence to higher dimensions, using distinct prime bases for each dimension. For a d -dimensional sequence:

$$\mathbf{x}_k = (\phi_{p_1}(k), \phi_{p_2}(k), \dots, \phi_{p_d}(k)),$$

where p_1, p_2, \dots, p_d are the first d prime numbers.

Properties: - Multi-dimensional. - Uses radical inverse functions for all dimensions. - Efficient for low-dimensional sampling.

Difference Between Hammersley and Halton: - Hammersley uses a uniform scaling for the last dimension ($\frac{k}{N}$), while Halton uses the radical inverse function for all dimensions. - Halton sequences are more suitable for cases where the number of samples N is not predetermined.

2.6 Summary

- The **radical inverse function** forms the foundation for all these sequences. - The **Van der Corput sequence** is a 1D low-discrepancy sequence based on the radical inverse. - The **Hammersley sequence** extends this to d -dimensions by combining radical inverse functions with uniform scaling. - The **Halton sequence** generalizes Van der Corput to d -dimensions using radical inverse functions in distinct prime bases.

While Monte Carlo and Importance Sampling improve accuracy, they rely on pseudo-random sequences, which can still lead to uneven sample distribution. ****Quasi-random sequences****, like Halton and Hammersley, ensure low-discrepancy distributions, providing better coverage of the domain.

Halton Sequence The Halton sequence generates points in multi-dimensional space using coprime bases (e.g., 2, 3, 5 for dimensions 1, 2, and 3). For dimension d :

$$H(n) = \left(\phi_2(n), \phi_3(n), \dots, \phi_p(n) \right),$$

where $\phi_b(n)$ is the radical inverse function in base b .

Hammersley Sequence The Hammersley sequence is a variant of Halton optimized for 2D integration. For N samples, it generates points:

$$H(n) = \left(\frac{n}{N}, \phi_b(n) \right),$$

where $\phi_b(n)$ is the radical inverse function.

Example in Graphics Halton and Hammersley sequences are precomputed for sampling 2D domains like: - Pixel sampling in anti-aliasing. - Environment map lighting.

For higher dimensions, higher prime bases ensure uniformity.

2.7 Quasi-Random Sequences

Quasi-random sequences, like Halton or Hammersley, are deterministic sequences designed to reduce variance by evenly distributing samples over a domain. These sequences exhibit low-discrepancy, meaning they minimize gaps and clustering compared to purely random sampling.

2.8 Low-Discrepancy vs High-Discrepancy Sampling

The discrepancy of a set of samples measures how evenly they cover the domain:

- **Low-Discrepancy Sampling:** Ensures a uniform distribution, leading to more efficient integration in low-dimensional problems. Examples: Halton, Hammersley, Sobol.
- **High-Discrepancy Sampling:** Includes purely random or stratified sampling. Often requires more samples to converge but is flexible for higher-dimensional problems.

2.9 Examples in Graphics

- **Random Sampling:** Pixel anti-aliasing by jittering sample positions within a pixel.
- **Monte Carlo Sampling:** Path tracing for light transport, where light paths are randomly sampled through the scene.
- **Low-Discrepancy Sampling:** Environment map sampling using Halton sequences for better distribution of light samples.

3 Sampling Functions and Names

Some commonly used sampling functions include:

- **Halton Sequence:** Generates low-discrepancy points for multi-dimensional sampling.
- **Hammersley Sequence:** Optimized for 2D sequences, often used for sampling a hemisphere.
- **Sobol Sequence:** Suitable for higher-dimensional problems, commonly used in global illumination.
- **Stratified Sampling:** Divides the domain into strata to reduce variance in Monte Carlo methods.
- **Blue Noise Sampling:** Ensures even spacing between samples, often used for image sampling and point distributions.

5. Optimizing Quasi-Random Sequences for Graphics

Precomputing Sequences In 2D rendering (e.g., sampling a hemisphere), precomputing Halton or Hammersley sequences ensures faster runtime sampling. Precomputing reduces overhead and avoids generating sequences on-the-fly.

Higher Bases for Higher Dimensions As the number of dimensions increases, using higher bases in Halton sequences avoids correlation artifacts and ensures even distribution across dimensions.

Practical Considerations: - Use Halton for multi-dimensional sampling (e.g., BRDF importance sampling). - Use Hammersley for 2D domains like image-based sampling or anti-aliasing.

Conclusion

Efficient sampling is crucial for rendering. Monte Carlo and Importance Sampling provide flexible ways to integrate complex functions, while quasi-random sequences like Halton and Hammersley reduce variance and improve convergence. By combining these methods and optimizing them for specific use cases, graphics programmers can achieve high-quality results with fewer samples.

Use Monte-Carlo when you can define the PDF, if that's possible use importance sampling, if you want in-built PDF use quasi-random for low-discrepancy random sequences...but this can cause issues with high-frequency areas so stick to using PDFs then.