

# Path Tracing Renderer Using Monte Carlo Methods

Silas Maughan

August 9, 2024

## Abstract

This report presents a study on the implementation of a path tracing renderer using Monte Carlo methods to simulate realistic lighting in a 3D scene. Various sampling techniques and variance reduction methods are explored to enhance image quality and convergence speed. Experimental results demonstrate the effectiveness of these techniques in reducing noise and improving rendering efficiency. The report discusses the mathematical foundations, implementation details, and performance evaluations of different Monte Carlo sampling strategies and variance reduction techniques.

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Background and Related Work</b>	<b>3</b>
2.1	Monte Carlo Methods in Path Tracing . . . . .	3
2.2	Monte Carlo Integration . . . . .	3
2.3	Sampling Techniques . . . . .	4
2.4	Variance Reduction Techniques . . . . .	4
2.5	Related Work . . . . .	4
<b>3</b>	<b>Methodology</b>	<b>5</b>
3.1	Scene Setup . . . . .	5
3.2	Basic Ray Tracing Algorithm . . . . .	5

3.3	Monte Carlo Integration Framework . . . . .	5
3.4	Diffuse Renderers . . . . .	5
3.5	Sampling Techniques . . . . .	6
3.5.1	Uniform Sampling . . . . .	6
3.5.2	Importance Sampling . . . . .	6
3.5.3	Stratified Sampling . . . . .	7
3.6	Variance Reduction Techniques . . . . .	7
<b>4</b>	<b>Results and Analysis</b>	<b>8</b>
4.1	Current Progress . . . . .	8
4.2	Future Visualizations . . . . .	8
<b>5</b>	<b>Experiments and Results</b>	<b>8</b>
5.1	Image Quality and Convergence Analysis . . . . .	8
5.2	Noise Level and Performance Comparison . . . . .	9
5.3	Convergence Rate Study . . . . .	9
<b>6</b>	<b>Discussion</b>	<b>9</b>
6.1	Statistical Analysis . . . . .	9
6.2	Number of Samples and Image Accuracy . . . . .	9
<b>7</b>	<b>Conclusion</b>	<b>10</b>
<b>8</b>	<b>Future Work</b>	<b>10</b>
<b>9</b>	<b>References</b>	<b>11</b>

# 1 Introduction

Path tracing is a rendering technique used to create realistic images by simulating the way light interacts with objects in a scene. Unlike traditional ray tracing, which traces a single path of light from the eye to the light source, path tracing traces multiple light paths to account for complex interactions like reflection, refraction, and scattering. This report details the implementation of a path tracing renderer using Monte Carlo integration to approximate the rendering equation.

## 2 Background and Related Work

### 2.1 Monte Carlo Methods in Path Tracing

Monte Carlo integration is a statistical technique used to approximate the rendering equation by randomly sampling light paths and averaging their contributions to pixel color. The rendering equation, formulated by Kajiya in 1986, is given by:

$$L_o(\mathbf{x}, \omega_o) = L_e(\mathbf{x}, \omega_o) + \int_{\Omega} f_r(\mathbf{x}, \omega_i, \omega_o) L_i(\mathbf{x}, \omega_i) (\omega_i \cdot \mathbf{n}) d\omega_i \quad (1)$$

where:

- $L_o(\mathbf{x}, \omega_o)$  is the outgoing radiance at point  $\mathbf{x}$  in direction  $\omega_o$ .
- $L_e(\mathbf{x}, \omega_o)$  is the emitted radiance at point  $\mathbf{x}$  in direction  $\omega_o$ .
- $f_r(\mathbf{x}, \omega_i, \omega_o)$  is the bidirectional reflectance distribution function (BRDF) at point  $\mathbf{x}$  for incoming direction  $\omega_i$  and outgoing direction  $\omega_o$ .
- $L_i(\mathbf{x}, \omega_i)$  is the incoming radiance at point  $\mathbf{x}$  from direction  $\omega_i$ .
- $\omega_i \cdot \mathbf{n}$  is the cosine of the angle between the incoming direction  $\omega_i$  and the surface normal  $\mathbf{n}$ .
- $\Omega$  is the hemisphere above the point  $\mathbf{x}$ .

### 2.2 Monte Carlo Integration

Monte Carlo integration estimates an integral by averaging the function evaluated at randomly chosen points. The Monte Carlo estimator for the integral is:

$$\hat{I} = \frac{1}{N} \sum_{i=1}^N f(\mathbf{x}_i) \quad (2)$$

where  $\hat{I}$  is the estimated integral,  $N$  is the number of samples, and  $f(\mathbf{x}_i)$  is the integrand evaluated at the  $i$ -th sample point  $\mathbf{x}_i$ . The variance of the estimator decreases as the number of samples increases, making this method particularly suitable for high-dimensional integrals like the rendering equation.

## 2.3 Sampling Techniques

Different sampling strategies affect the efficiency and quality of the rendered images. Uniform sampling, importance sampling, and stratified sampling are common techniques used in Monte Carlo integration for path tracing.

**Uniform Sampling:** Samples are taken uniformly across the domain. While simple to implement, this method can result in high variance and slow convergence.

**Importance Sampling:** Samples are drawn from a distribution that closely matches the integrand, reducing variance and improving convergence.

**Stratified Sampling:** The domain is divided into strata, and samples are taken from each stratum. This technique reduces variance by ensuring a more even coverage of the domain.

## 2.4 Variance Reduction Techniques

Variance reduction techniques aim to minimize the noise in the rendered image without significantly increasing the number of samples. These techniques include antithetic variates, control variates, and Russian roulette termination.

**Antithetic Variates:** Pairs of negatively correlated samples are used to reduce variance.

**Control Variates:** Known functions with calculable expected values are used to adjust the estimator, reducing variance.

**Russian Roulette Termination:** Paths are terminated probabilistically to balance the computational cost and variance reduction.

## 2.5 Related Work

Path tracing and Monte Carlo methods have been extensively studied in computer graphics. Key contributions include Kajiya's rendering equation, Veach's work on bidirectional path tracing, and various techniques for variance reduction and importance sampling.

## 3 Methodology

### 3.1 Scene Setup

A basic 3D scene is created with simple geometric objects, a light source, and a camera. The scene includes objects like spheres and planes, and a point light source to illuminate the scene. The camera is positioned to capture the entire scene.

### 3.2 Basic Ray Tracing Algorithm

A basic ray tracing algorithm is implemented to handle the initial light path calculations. Rays are traced from the camera through each pixel and into the scene. The algorithm calculates intersections with objects, reflects or refracts rays as necessary, and accumulates the color contributions along the path.

### 3.3 Monte Carlo Integration Framework

The Monte Carlo integration framework is developed for the path tracing renderer. Light paths are traced stochastically, and their contributions are averaged to approximate the solution to the rendering equation. Specifically, the Monte Carlo estimator for the integral can be represented as:

$$\hat{I} = \frac{1}{N} \sum_{i=1}^N f(\mathbf{x}_i) \quad (3)$$

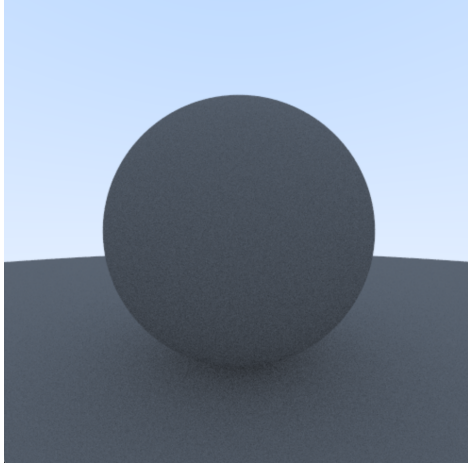
where  $\hat{I}$  is the estimated integral,  $N$  is the number of samples, and  $f(\mathbf{x}_i)$  is the integrand evaluated at the  $i$ -th sample point  $\mathbf{x}_i$ .

### 3.4 Diffuse Renderers

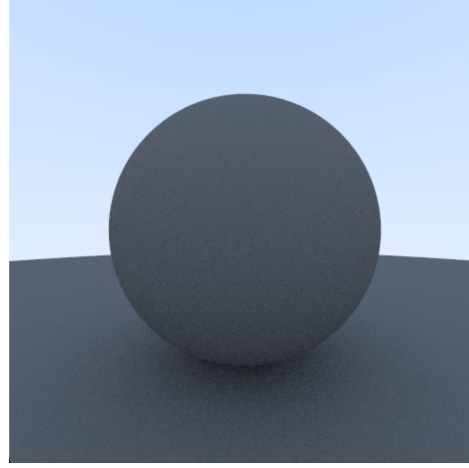
Two diffuse rendering techniques have been implemented so far:

**Completely Diffuse:** In this technique, a random vector is generated from the hit point, resulting in completely uniform scattering rays.

**Lambertian Diffuse:** Here, a random vector from within a unit sphere at the hit point is added to the reflected normal, causing the rays to tend toward the normal direction.



(a) Rendered image using Uniform Diffuse Renderer



(b) Rendered image using Lambertian Diffuse Renderer

Figure 1: Comparison of rendering techniques

## 3.5 Sampling Techniques

### 3.5.1 Uniform Sampling

Random samples are generated uniformly across the hemisphere above each intersection point. The contributions of these samples are averaged to estimate the pixel color.

### 3.5.2 Importance Sampling

Importance sampling is implemented by generating samples according to the distribution of the BRDF at each intersection point. This focuses sampling efforts on the most significant light paths, reducing variance. The estimator in importance sampling is given by:

$$\hat{I} = \frac{1}{N} \sum_{i=1}^N \frac{f(\mathbf{x}_i)}{p(\mathbf{x}_i)} \quad (4)$$

where  $p(\mathbf{x}_i)$  is the probability density function used for sampling.

### 3.5.3 Stratified Sampling

Stratified sampling is implemented by dividing the hemisphere into strata and sampling within each stratum. This ensures a more even coverage of the hemisphere and reduces variance compared to uniform sampling. The variance of the estimator in stratified sampling can be expressed as:

$$\text{Var}(\hat{I}) = \frac{1}{N} \sum_{i=1}^N \text{Var}(f(\mathbf{x}_i) - \mathbb{E}[f(\mathbf{x}_i)]) \quad (5)$$

## 3.6 Variance Reduction Techniques

Variance reduction techniques such as antithetic variates, control variates, and Russian roulette termination are introduced to enhance image quality.

**Antithetic Variates:** Pairs of negatively correlated samples are used to reduce variance. If  $\mathbf{X}$  and  $\mathbf{X}'$  are antithetic pairs, then the estimator becomes:

$$\hat{I} = \frac{1}{2N} \sum_{i=1}^N (f(\mathbf{X}_i) + f(\mathbf{X}'_i)) \quad (6)$$

**Control Variates:** Known functions with calculable expected values are used to adjust the estimator, reducing variance. If  $h(\mathbf{x})$  is a control variate with known expected value  $\mathbb{E}[h(\mathbf{x})]$ , the estimator is:

$$\hat{I} = \frac{1}{N} \sum_{i=1}^N (f(\mathbf{x}_i) + c(h(\mathbf{x}_i) - \mathbb{E}[h(\mathbf{x})])) \quad (7)$$

**Russian Roulette Termination:** Paths are terminated probabilistically to balance the computational cost and variance reduction. The probability of termination is denoted by  $p$ , and the estimator becomes:

$$\hat{I} = \frac{1}{N} \sum_{i=1}^N \frac{f(\mathbf{x}_i)}{1-p} \quad (8)$$

## 4 Results and Analysis

### 4.1 Current Progress

The following components have been completed:

- Scene setup with simple objects (spheres) and a camera.
- Basic ray tracing algorithm.
- Implementation of Monte Carlo integration framework.
- Implementation of random sampling for light paths.
- Calculation of pixel colors based on average contributions from sampled paths.
- Completely diffuse renderer.
- Lambertian diffuse renderer.

### 4.2 Future Visualizations

Figures 1a and 1b show the rendered images using the completely diffuse renderer and the Lambertian diffuse renderer, respectively. Future visualizations will include results from uniform sampling, importance sampling, and stratified sampling techniques.

## 5 Experiments and Results

### 5.1 Image Quality and Convergence Analysis

The impact of each sampling method on image quality and convergence will be visualized and analyzed. Uniform sampling, importance sampling, and stratified sampling will be compared in terms of the noise level and convergence speed.

Figure 2: Rendered image using Uniform Sampling



Figure 3: Rendered image using Importance Sampling

Figure 4: Rendered image using Stratified Sampling

## 5.2 Noise Level and Performance Comparison

Noise levels and performance improvements with different variance reduction techniques will be measured and compared. Figures 5, 6, and 7 will demonstrate the noise reduction achieved by each variance reduction technique.

Figure 5: Rendered image using Antithetic Variates

## 5.3 Convergence Rate Study

Experiments will be conducted to study the convergence rate of the path tracer. The number of samples per pixel will be varied, and the resulting image quality will be evaluated to understand the relationship between the number of samples and image accuracy. Figure 8 will show the convergence rate for different sampling techniques.

# 6 Discussion

## 6.1 Statistical Analysis

Statistical analysis will be used to demonstrate the effects of different sampling techniques and variance reduction methods on image quality. The variance and bias of each method will be discussed, and the effectiveness of each technique in reducing noise and improving convergence will be evaluated.

## 6.2 Number of Samples and Image Accuracy

The analysis will show how the number of samples affects image accuracy and convergence. More samples generally lead to higher accuracy but at a higher computational cost. The trade-offs between sample count, image

Figure 6: Rendered image using Control Variates

Figure 7: Rendered image using Russian Roulette Termination

quality, and rendering time will be discussed. The efficiency of various variance reduction techniques will be evaluated to determine their practicality in different rendering scenarios.

## 7 Conclusion

The study demonstrates that Monte Carlo methods are effective for path tracing and realistic image synthesis. Importance sampling and stratified sampling significantly improve image quality and convergence speed compared to uniform sampling. Variance reduction techniques further enhance the rendering efficiency by reducing noise. Future work could explore more advanced sampling strategies and real-time rendering optimizations. The implementation of these techniques in a practical renderer highlights the importance of statistical methods in achieving high-quality, efficient rendering solutions.

## 8 Future Work

The following tasks are planned for further development:

- Implementation and comparison of uniform sampling, importance sampling, and stratified sampling.
- Visualization and analysis of the impact of each sampling method on image quality and convergence.
- Addition of variance reduction techniques to the path tracer.
- Measurement and comparison of noise levels and performance improvements.
- Conducting experiments to study the convergence rate of the path tracer.

Figure 8: Convergence rate for different sampling techniques

- Using statistical analysis to demonstrate how the number of samples affects image accuracy.

## 9 References