

Path Tracing Renderer Using Monte Carlo Methods

Your Name

July 21, 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.

Contents

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

2.2 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.

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:

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

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 .

3.4 Sampling Techniques

3.4.1 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.

3.4.2 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:

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

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

3.4.3 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:

$$\text{Var}(\hat{I}) = \frac{1}{N} \sum_{i=1}^N \text{Var}(f(\mathbf{x}_i) - \mathbb{E}[f(\mathbf{x}_i)]) \quad (4)$$

3.5 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:

$$\hat{I} = \frac{1}{2N} \sum_{i=1}^N (f(\mathbf{X}_i) + f(\mathbf{X}'_i)) \quad (5)$$

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:

$$\hat{I} = \frac{1}{N} \sum_{i=1}^N (f(\mathbf{x}_i) + c(h(\mathbf{x}_i) - \mathbb{E}[h(\mathbf{x})])) \quad (6)$$

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:

$$\hat{I} = \frac{1}{N} \sum_{i=1}^N \frac{f(\mathbf{x}_i)}{1-p} \quad (7)$$

4 Experiments and Results

4.1 Image Quality and Convergence Analysis

The impact of each sampling method on image quality and convergence is visualized and analyzed. Uniform sampling, importance sampling, and stratified sampling are compared in terms of the noise level and convergence speed. Figures ??, ??, and ?? show the rendered images using each technique, respectively.

4.2 Noise Level and Performance Comparison

Noise levels and performance improvements with different variance reduction techniques are measured and compared. Figures ??, ??, and ?? demonstrate the noise reduction achieved by each variance reduction technique.

4.3 Convergence Rate Study

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

5 Discussion

5.1 Statistical Analysis

Statistical analysis is used to demonstrate the effects of different sampling techniques and variance reduction methods on image quality. The variance

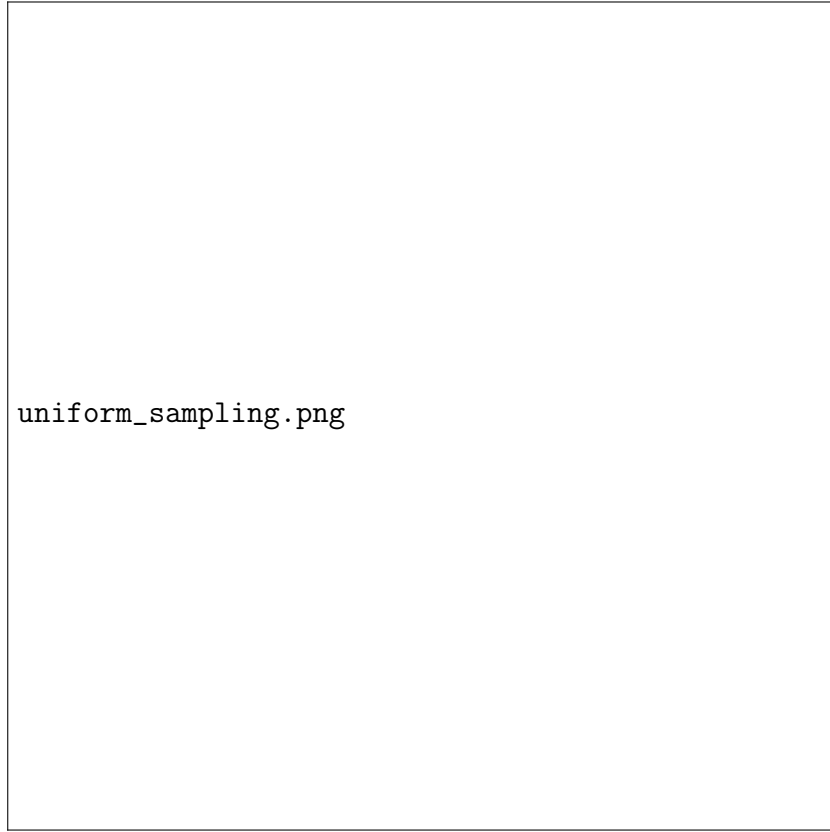


Figure 1: Rendered image using Uniform Sampling

and bias of each method are discussed, and the effectiveness of each technique in reducing noise and improving convergence is evaluated.

5.2 Number of Samples and Image Accuracy

The analysis shows 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 quality, and rendering time are discussed.



Figure 2: Rendered image using Importance Sampling

6 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.

7 References



Figure 3: Rendered image using Stratified Sampling



Figure 4: Rendered image using Antithetic Variates

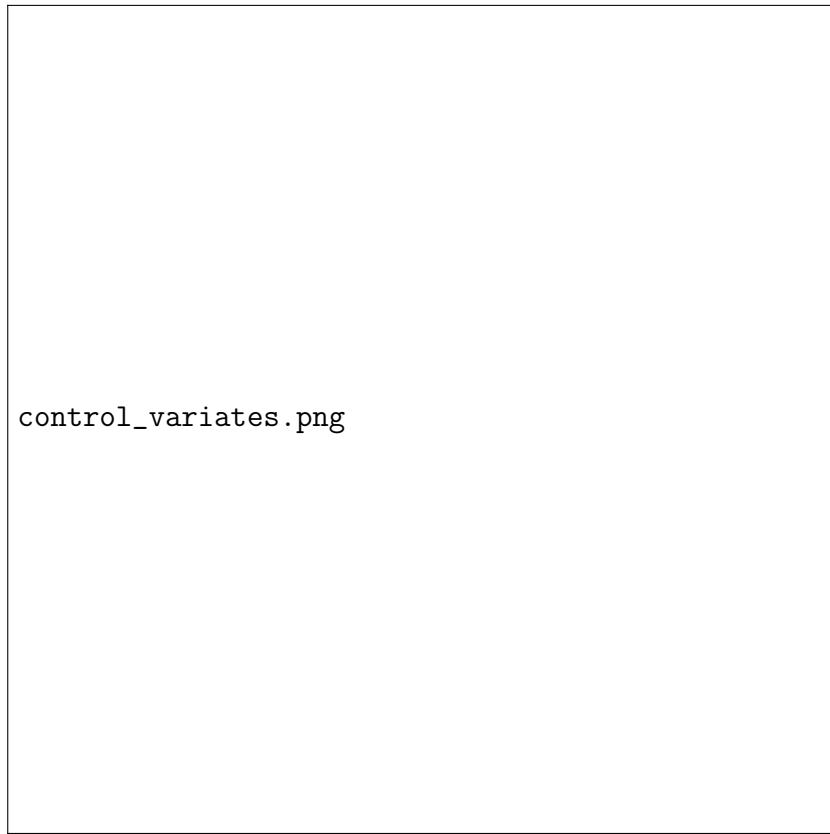


Figure 5: Rendered image using Control Variates

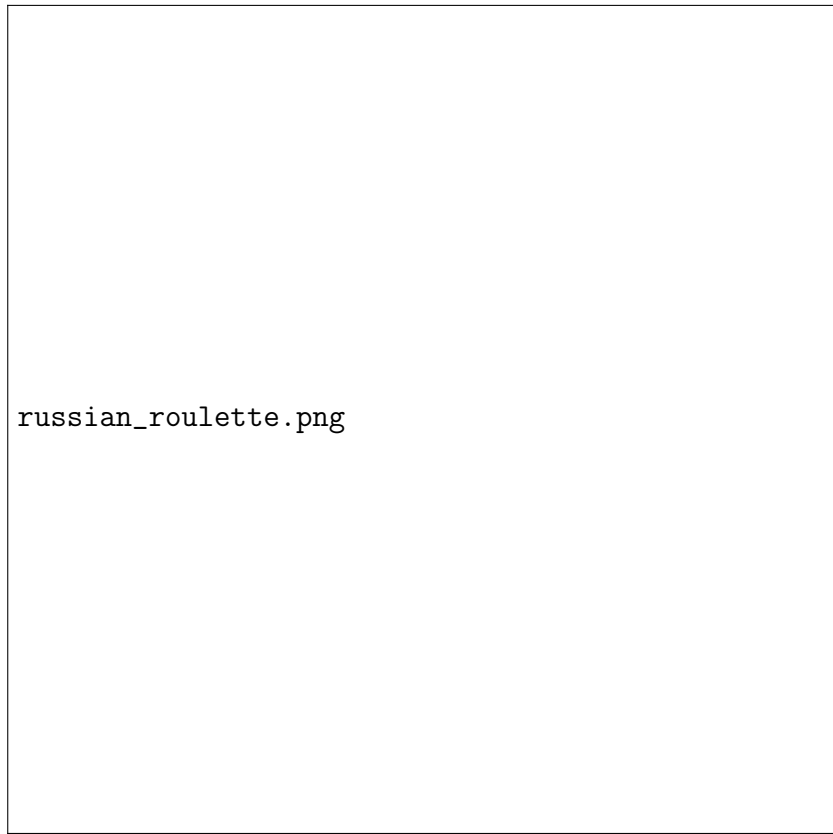
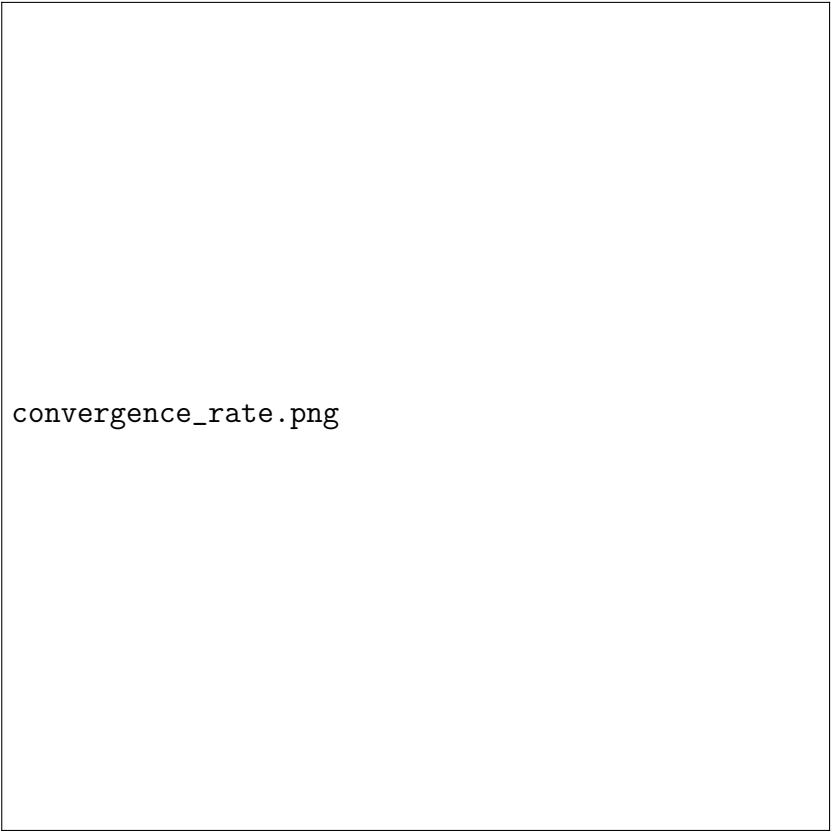


Figure 6: Rendered image using Russian Roulette Termination



convergence_rate.png

Figure 7: Convergence rate for different sampling techniques