Monte Carlo Methods for Photon Scattering and Absorption

River Hansen

April 25, 2025

Abstract

This project explores the use of Monte Carlo methods for simulating photon scattering and absorption in a medium. Inspired by the Photon Propagation Code (PPC) used at IceCube, this model is a simplified recreation . The simulation involves an LED positioned at a fixed distance from a camera, and the resulting distribution of photons reaching the camera is analyzed. Using random sampling from exponential distributions, the distances to scattering and absorption events are determined, with rate parameters based on the scattering and absorption lengths. The Henyey-Greenstein phase function is employed to model scattering angles. The analysis varies several parameters, including scattering length, absorption length, camera distance, camera size, and the LED's initial angle, to investigate their effects on the photon distribution.

Methodology and Key Results

- **Random Sampling from Exponential Distribution:** Used inverse transform sampling to draw distances to scattering and absorption events from exponential distributions, such that the expected values match the scattering and absorption lengths.

$$f(x) = \lambda e^{-\lambda x}, \quad x > 0$$

$$E[X] = \frac{1}{\lambda}$$

$$x_s = -\frac{1}{\lambda_s}\ln(1-u_s), \quad x_a = -\frac{1}{\lambda_a}\ln(1-u_a)$$

$$u_s, u_a \sim \mathcal{U}(0,1), \quad \lambda_s = \frac{1}{\text{scattering length}}, \quad \lambda_a = \frac{1}{\text{absorption length}}$$

These equations were implemented in Python using `numpy` for random sampling and mathematical operations.

- **Scattering Angle from Henyey-Greenstein Phase Function:** The cosine of the scattering angle θ was sampled using the inverse transform method for the Henyey-Greenstein distribution.

$$\cos\theta = \begin{cases} 2\xi - 1, & \text{if } g = 0 \\ \frac{1+g^2-\left(\frac{1-g^2}{1-g+2g\xi}\right)^2}{2g}, & \text{if } g \neq 0 \end{cases}$$
$$\xi \sim \mathcal{U}(0,1)$$

  The variable g is the anisotropy parameter, where $g = 0$ corresponds to isotropic scattering. This equation was implemented in Python using `numpy`.

- **Analysis and Visualization:** Plotted the distribution using histograms and Kernel Density Estimation using `matplotlib` and `seaborn`.

- **Key Results:**
  - **Finding 1:** Increasing the scattering length concentrates more photons near the center, as photons travel further on average before scattering, reducing opportunities for lateral displacement.
  - **Finding 2:** Decreasing the absorption length results in more photons being concentrated near the center, as photons are absorbed sooner on average. This limits their ability to scatter widely before being removed from the system.
  - **Finding 3:** Scattering and absorption lengths have similar effects on photon distribution, as both parameters influence how concentrated photons are near the center. This overlap makes it difficult to reliably distinguish between scattering and absorption effects based solely on simulation data. The PPC used by IceCube's CamSim group has essentially the same issue and is something being actively researched.

Submitted Files

- `photon.py` – Defines the `photon` class, used to model individual photon behavior.
- `sim.py` – Executes the main simulation logic using the `photon` class.
- `report.pdf` – Comprehensive report detailing methodology, results, and analysis.
- `README.md` – Instructions for running the simulation and reproducing the results.

Use of Artificial Intelligence (AI)

This project used ChatGPT (OpenAI) to help format LaTeX and improve the clarity of the explanations.

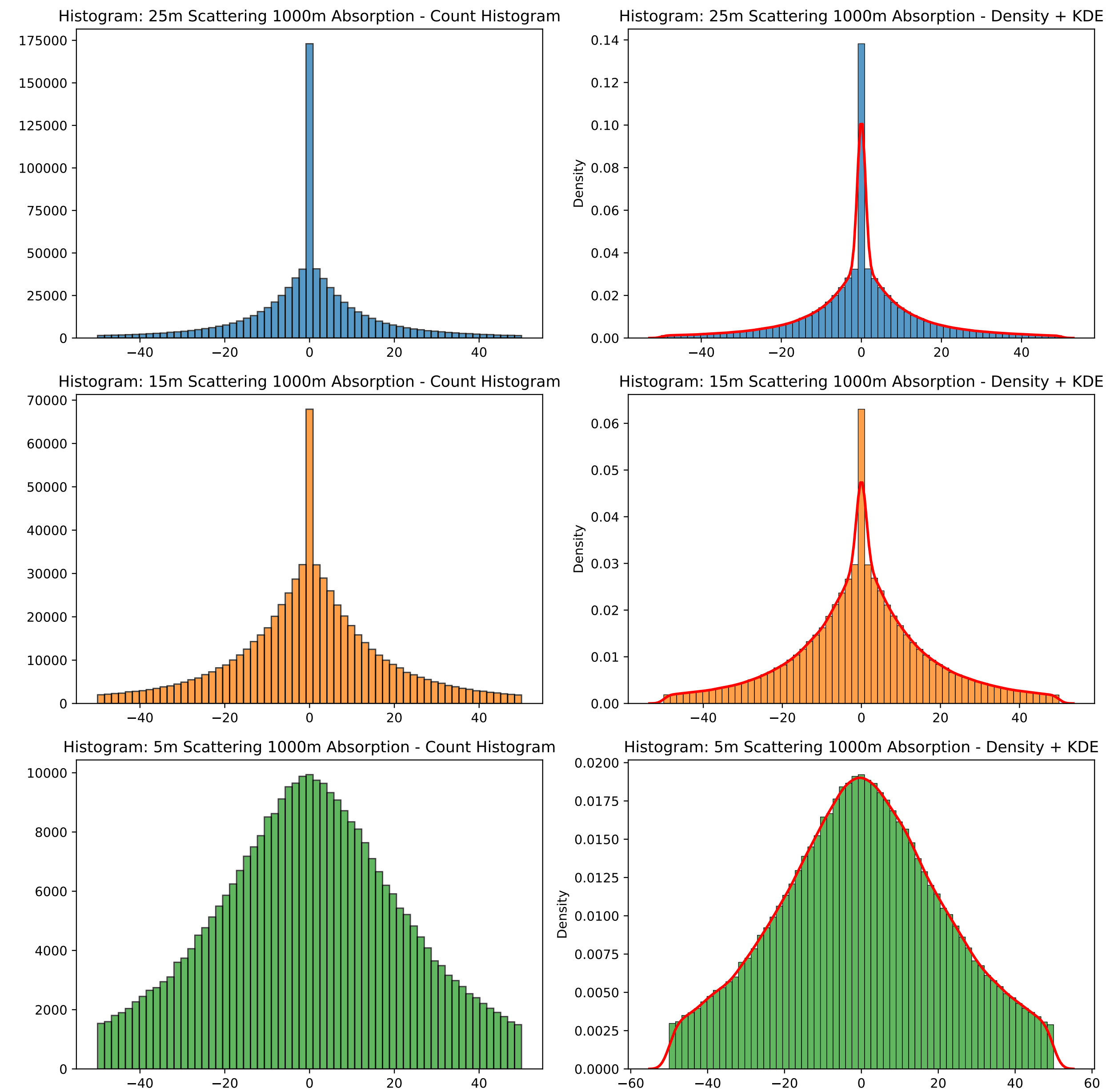

Figure 1: Varied scattering length with constant absorption length of 1000m. Simulated 10^6 photons oriented directly at the camera from 50m. Camera size was 50m, and $g = 0.85$ for forward scattering.

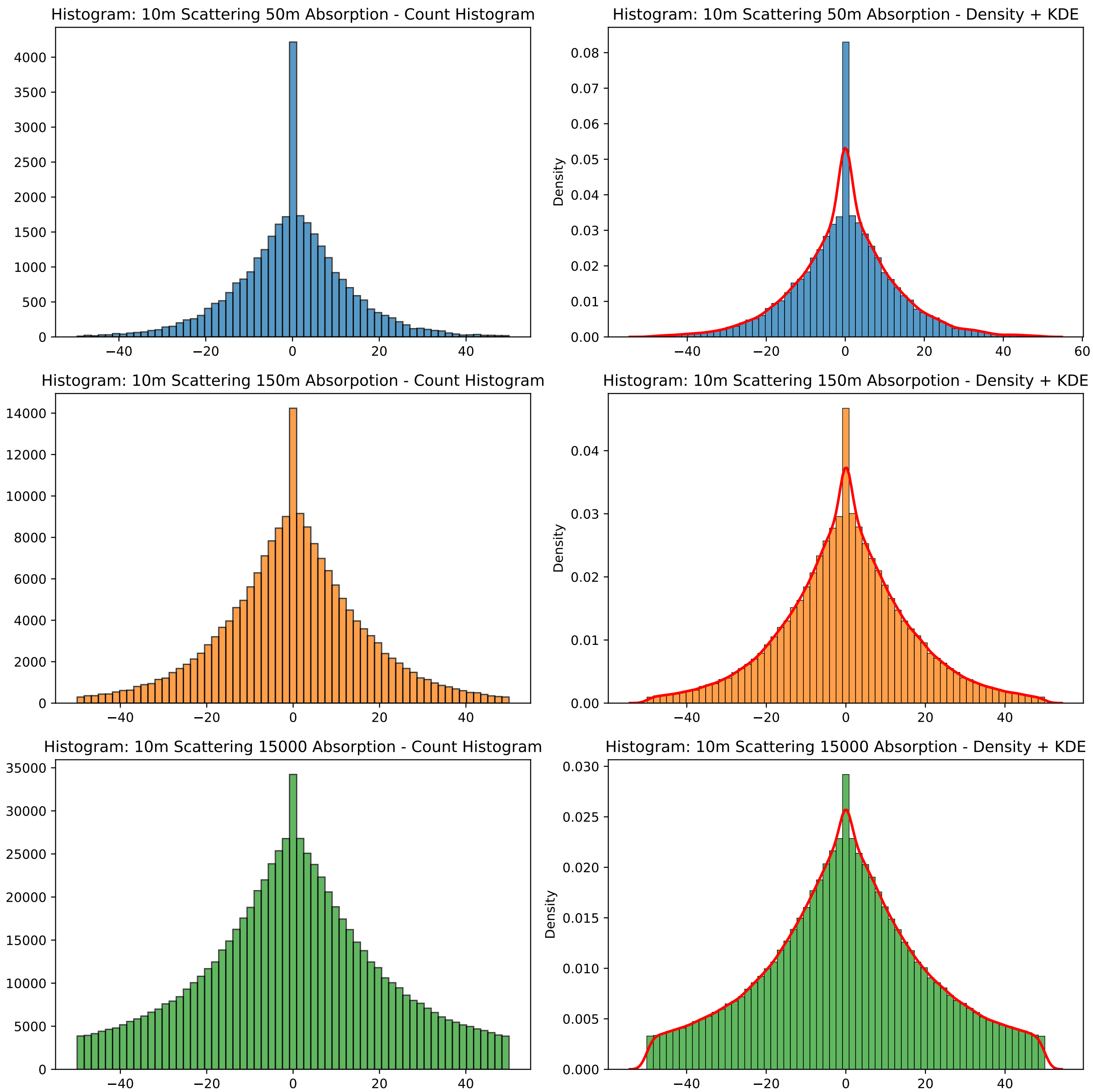

Figure 2: Varied absorption length with constant scattering length of 10m. Simulated 10^6 photons oriented directly at the camera from 50m. Camera size was 50m, and $g = 0.85$ for forward scattering.