Adaptive Sampling Optimization for Real-Time Path Tracing

Group member:

- Haochen Yang

Background and Motivation

Path tracing is a Monte Carlo rendering technique that simulates global illumination by randomly sampling light transport paths. While it produces physically accurate results, it suffers from **noise** due to the stochastic nature of Monte Carlo integration. Noise reduction typically requires a large number of samples per pixel (spp), but real-time applications (e.g., games, VR) impose strict computational budgets (e.g., 1,000–10,000 samples per frame).

Key Challenges:

1. **Variance Heterogeneity**: Noise is concentrated in high-variance regions (e.g., shadow boundaries, caustics, specular surfaces). Uniform sampling wastes resources on low-variance regions.
2. **Dynamic Scenes**: Variance distribution changes with camera/object motion, necessitating adaptive sampling strategies.

Objective: Design an optimization framework to dynamically allocate samples across pixels, minimizing overall noise under a fixed sample budget, while maintaining real-time performance.

Principles of Path Tracing and Monte Carlo Integration

1. **Rendering Equation**:
   The radiance $L(p, \omega_o)$ at point p in direction ω_o is computed as:

   $$L(p, \omega_o) = L_e(p, \omega_o) + \int_{\Omega} f_r(p, \omega_i, \omega_o) L(p, \omega_i) \cos\theta \, d\omega_i$$

   where f_r is the BRDF (Bidirectional Reflectance Distribution Function), L_e is emitted radiance, and Ω is the hemisphere of incoming directions.
2. **Monte Carlo Estimation**:
   The integral is approximated using N random samples:

   $$L(p) \approx \frac{1}{N} \sum_{i=1}^{N} \frac{f_r L_i \cos\theta}{p(\omega_i)}$$

   where $p(\omega_i)$ is the sampling PDF (Probability Density Function). The **variance** of this estimator is inversely proportional to N:

   $$Var(L) \propto \frac{\sigma_p^2}{N}$$

   Here, σ_p^2 depends on scene complexity (e.g., light paths, materials).
3. **Adaptive Sampling**:
   Instead of uniform sampling ($N_p = const$), allocate more samples to pixels with higher variance σ_p^2. This minimizes the **mean squared error (MSE)** of the image.

Mathematical Optimization Model

1. **Problem Definition**:
   Let the image resolution be $W \times H$, and the total sample budget per frame be N_{total}. For each pixel p, allocate N_p samples such that:

$$\sum_{p=1}^{W \times H} N_p \leq N_{total}, \quad N_p \geq 0.$$

2. **Objective Function**:
   Minimize the total noise (variance) while penalizing excessive sampling costs:

$$\min_{\{N_p\}} \sum_{p=1}^{W \times H} \left(\frac{\sigma_p^2}{N_p} + \lambda N_p \right)$$

   - **Term 1**: $\frac{\sigma_p^2}{N_p}$ represents the pixel's variance, inversely proportional to its sample count.
   - **Term 2**: λN_p penalizes large N_p to balance noise and computation.
   - λ is a hyperparameter controlling the trade-off (e.g., $\lambda = 0.1$ prioritizes noise reduction).

3. **Constraints**:
   - **Sample Budget**:

$$\sum_p N_p \leq N_{total}$$

   - **Non-negativity**:

$$N_p \geq 0 \quad \forall p$$

References

1. Veach, E., & Guibas, L. J. (1997). *Robust Monte Carlo Methods for Light Transport Simulation*. ACM SIGGRAPH.
2. Kajiya, J. T. (1986). *The Rendering Equation*. ACM SIGGRAPH.
3. Moon, B., et al. (2014). *Adaptive Rendering Based on Weighted Local Regression*. ACM Transactions on Graphics.
4. Pharr, M., et al. (2016). *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann. (Chapter 13: Monte Carlo Integration)