

AI5505: Photorealistic Rendering

Advanced Monte Carlo Techniques

Lecturer: Bochang Moon

Overview

- Study important techniques to improve efficiency of MC estimators
- An efficiency measure for estimators F
 - $e(F) = \frac{1}{V(F)T(F)}$
 - To improve efficiency, we need to reduce the variance and time
- Russian Roulette
- Splitting
- Importance Sampling

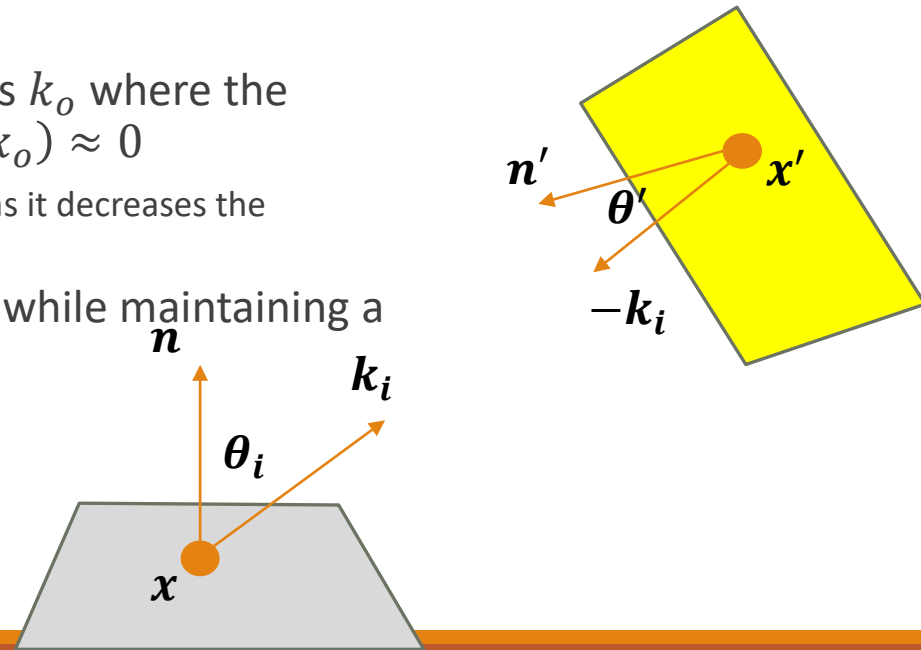
Russian Roulette

- e.g. direct lighting

- $$L_s(x, k_o) = \int_{all\ x'} \frac{\rho(k_i, k_o) L_e(x', -k_i) v(x, x') \cos\theta_i \cos\theta'}{\|x - x'\|^2} dA'$$
 - $$\approx \frac{1}{N} \sum_{i=1}^N \frac{\rho(k_i, k_o) L_e(x', -k_i) v(x, x') \cos\theta_i \cos\theta'}{p(x') \|x - x'\|^2}$$

- Problems

- Expensive to compute $v(x, x')$
- Let suppose that there are some directions k_o where the integrand's value is almost 0 due to $\rho(k_i, k_o) \approx 0$
 - In this case, evaluating $v(x, x')$ is not a good idea, as it decreases the efficiency
- Q. Can we somehow skip these directions while maintaining a correct answer?



Russian Roulette

- Given an estimator F , a new estimator F' with Russian Roulette can be given:
 - $F' = \begin{cases} \frac{F - qc}{1 - q} & \xi > q \\ c & \text{otherwise} \end{cases}$
 - q : termination probability
 - c is usually chosen as 0
- Consistency check
 - $E(F') = (1 - q) \left(\frac{E(F) - qc}{1 - q} \right) + qc = E(F)$
- Properties
 - It does not reduce variance, but improves efficiency by skipping unimportant parts

Russian Roulette

- Examples

- $L_s(k_o) \approx L_e(k_o) + \frac{\rho(k_i, k_o) L_f(k_i) \cos \theta_i d\sigma_i}{p(k_i)}$

- Problem: this is a recursive form, so ray depth can be infinite

- Can apply the Russian Roulette to path tracing so that the ray depth can be reduced

- $$F' = \begin{cases} \frac{F - qc}{1 - q} & \xi > q \\ c & \text{otherwise} \end{cases}$$

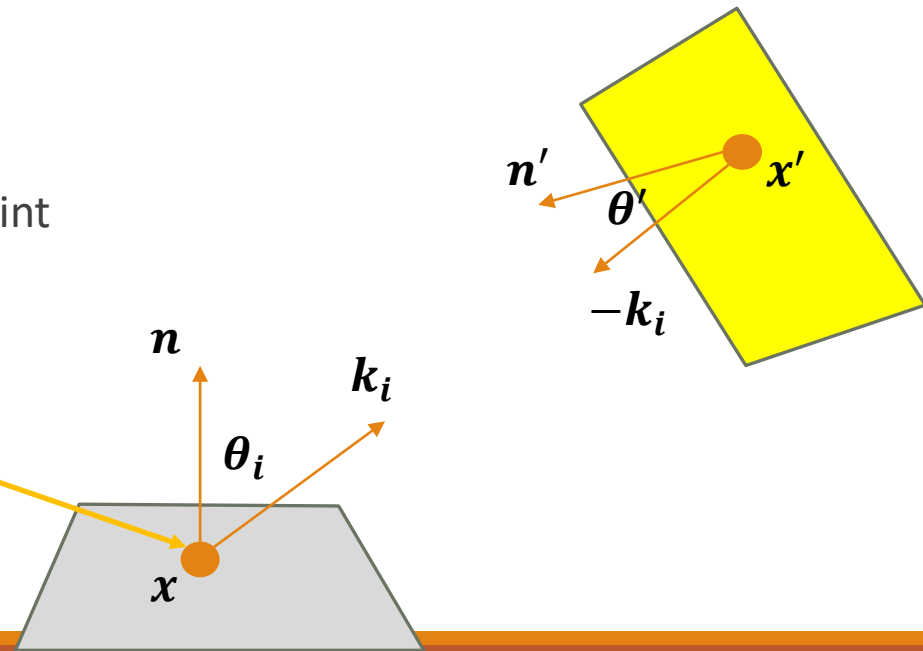
- Given a termination probability $q = 0.5$ and $c = 0$, we can terminate the reflection with the probability. In this case, we need to scale the radiance value with $\frac{1}{1 - q}$

Russian Roulette

- Bad examples
 - Apply this to the camera rays with a termination probability $q = 0.99$
 - In this case,
 - Only trace 1% of the camera rays
 - Most of pixels are black and a few pixels are very bright, although its expectation is still correct
 - The variance of the estimator will be much higher than the original estimator
- Efficiency-optimized Russian roulette
 - A technique that optimizes the parameter

Splitting

- Splitting is a technique to increase the number of samples for improving the efficiency
 - Allocate more rays to important dimensions
- E.g., direct lighting with a shortened version
 - $\int_A \int_S L_d(x, y, w) dx dy dw$
 - A: pixel area
 - S: light area
 - L_d : exitant radiance at the intersection point
 - (x, y): position on image

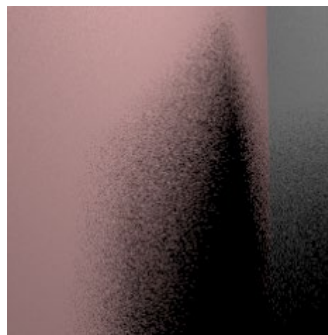


Splitting

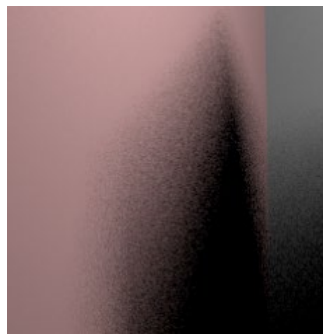
- direct lighting with a shortened version
 - $\int_A \int_S L_d(x, y, w) dx dy dw$
- A straightforward approach:
 - Generate N samples $(x_1, y_1, w_1), \dots (x_N, y_N, w_N)$
 - Evaluate $L_d(x_1, y_1, w_1), \dots, L_d(x_N, y_N, w_N)$
 - Need to generate N shadow rays
 - Average the radiance values
- Typically need a lot of samples given a large area light or many point lights
 - e.g., N = 100, 200 rays will be used (100 primary rays, 100 shadow rays)
 - Issue: 100 primary rays are often too much for a good antialiasing. Can we focus on

Splitting

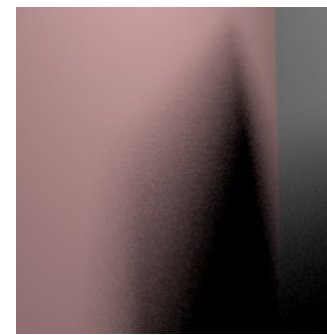
- Typically need a lot of samples given a large area light or many point lights
 - e.g., $N = 100$, 200 rays will be used (100 primary rays, 100 shadow rays)
 - Issue: 100 primary rays are often too many for a good antialiasing.
 - Splitting
 - $\frac{1}{N} \frac{1}{M} \sum_{i=1}^N \sum_{j=1}^M \frac{L(x_i, y_i, w_{i,j})}{p(x_i, y_i) p(w_{i,j})}$
 - Are able to use 5 image samples and take 20 light samples per image sample. Total ray # = $5 + 5 \times 20 = 105$
 - Still use 100 shadow rays for high-quality soft shadows, but the total number of rays is reduced



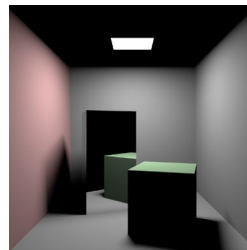
4 shadow rays /
pixel



16 shadow rays /
pixel



64 shadow rays /
pixel



Importance Sampling

- A variance reduction technique for Monte Carlo estimators
 - Allocate more samples to the important region where the integrand's value is high
- A Monte Carlo estimator:
 - $F_N = \frac{1}{N} \sum_{i=1}^N \frac{f(X_i)}{p(X_i)}$
- e.g., $L_s(k_o) \approx L_e(k_o) + \frac{\rho(k_i, k_o) L_f(k_i) \cos \theta_i d\sigma_i}{p(k_i)}$
 - An intuition is that if we take more samples in terms of k_i that makes $\cos \theta_i$ high, we can reduce the variance of the estimator

Importance Sampling

- e.g., Evaluate an integral $\int f(x)dx$
- Note that we can choose an arbitrary pdf, $p(x)$
- What if we choose $p(x) \propto f(x)$ or $p(x) = cf(x)$
 - $c = \frac{1}{\int f(x)dx}$ a constant for normalization
 - $F_N = \frac{1}{N} \sum_{i=1}^N \frac{f(X_i)}{p(X_i)} = \frac{1}{N} \sum_{i=1}^N \frac{1}{c}$
 - $V(F_N) = 0$
- Notes
 - In practice, we cannot choose $p(x)$ in this way, but it provides some intuition
 - If we choose $p(x)$ similarly compared to the shape of $f(x)$, we are able to reduce variance

Importance Sampling

- A counterexample:
 - Evaluate an integral $\int f(x)dx$
 - $p(x) = \begin{cases} 99.01 & x \in [0,0.01) \\ 0.01 & x \in [0.01,1] \end{cases}$
 - $f(x) = \begin{cases} 0.01 & x \in [0,0.01) \\ 1.01 & x \in [0.01,1] \end{cases}$
 - $\int f(x)dx = 1$
 - Most of samples will be taken from $[0,0.01)$, and $\frac{f(X_i)}{p(X_i)} \approx 0.0001$ which is far from 1
 - In this case, the variance will increase a lot
- Note
 - In practice, it is easy to apply an important sampling to the rendering, by considering only some terms
 - Taking account for all terms is ideal but this can be very challenging

More Topics?

- Evaluate an integral $\int f(x)g(x)dx$
- If we have each important sampling scheme for the functions $f(x)$ and $g(x)$, how can we combine the techniques?
- Multiple important sampling addresses this issue and this will be discussed later