Computer Graphics - Assignment 3

Bassam Adnan

2023121003

Scene	10	100	1000
Big	16279	164874	1481039
Med	15515	157063	1427693
Small	16645	160663	1434966
Many	37177	339155	3458495

Uniform hemisphere sampling

Scene	10	100	1000
Big	16272	219554	1563294
Med	16804	164032	1450730
Small	15912	164081	1741247
Many	37653	343860	3458945

Weighted - Cosine sampling

Scene	10	100	1000
Big	18062	178670	1450492
Med	14986	149229	1476749
Small	14266	142566	1376272
Many	29159	149229	2963076

Importance Sampling

Given above are the time taken for each scene to render (in ms) for each sampling strategy for values of *spp* (samples per pixel) as $10, 100, 1000$

Q1. Why can't we render point and directional lights with uniform hemisphere sampling or cosine weighted sampling?

This majorly stems from the fact that if we try to sample a direction in case of the mentioned sampling strategies, following the algorithm we built for Monte Carlo, we would have to shoot a ray towards that direction from the point that the camera ray intersected the scene with, say p, (after obviously converting it into the context of the scene) we check for its intersection with any light in the scene, in case of point lights which don't have any dimensions in terms of length/width/size, its impossible to check if this sampled ray hit the point light. Infact, the probability of sampling a ray towards the direction of the point light is ≈ 0. Similar argument follows for direction light, since they only fall from a certain direction, and can be approximated as point lights , just far away , checking for light intersection becomes tough. Due to this reason we do not sample a ray from p but instead choose pre-determined ray directions to shoot from p (say, in the direction of the point light for example). This ensures we are only looking at relevant rays at all iterations.

Its also worth noting this is possible in area lights since an they exist in space with dimensions and hence sampling a direction from p such that it hits the light area is much more than 0.

Q2.Why does the noise increase for the same number of samples in the case of uniform hemisphere and cosine weighted sampling as the size of the area light decreases?

The Monte Carlo rendering (MC) system estimates integration values by tracking light (sampling) in a multidimensional space. Although images can be rendered by a small number of samplings, the estimation results relative to the real value have a large error; these errors in the rendering results are a large number of noise because the variance of Monte Carlo estimation method decreases linearly according to the number of sampling increase; so in order to get a more accurate estimation, results often need a lot of sampling.
The Variance of a Monte Carlo estimate F_n is given by

$$\sigma^2[F_N] = \sigma^2[\frac{1}{N}\sum_{i=1}^{N}\frac{f(X_i)}{pdf(X_i)}]$$

$$= \frac{1}{N}\sum_{i=1}^{N}\sigma^2[\frac{f(X_i)}{pdf(X_i)}]$$

$$\frac{1}{N}\sigma^2[Y]$$

$$\sigma[F_N] = \frac{1}{\sqrt{N}}\sigma[Y]$$

Where Y represents the specific evaluation at a point Y_i where Y_i is the substituted term. All terms carry usual meanings, we have N samples and can see that the standard deviation converges with the rate $O(\sqrt{N})$ However if we do end up fixing a particular N we would have to look at the term $\frac{f(X_i)}{pdf(X_i)}$. In particular, the *pdf* term is of importance here.
In case of uniform hemisphere sampling, this term boils down to

$$\frac{1}{2\pi}$$

In case of cosine sampling, given the angle between the light and p, θ this term becomes

$$\frac{\cos\theta}{\pi}$$

and the numerator term $f(X_i)$ depends on if the point sampled does end up hitting the light, intuitively, this term will be 0 if the direction sampled with the *pdf* is checked for intersection with smaller area lights then they will be less likely to intersect and hence increase noise per pixel. It is also worth noting that since we do sample on directions more likely to hit in case of cosine sampling we could reduce noise by as much as 30% since the function f is more likely to intersect with the sample.

0.1 Citations

Xiwen Chen, Jianfei Shen, ”Monte Carlo Noise Reduction Algorithm Based on Deep Neural Network in Efficient Indoor Scene Rendering System”, Advances in Multimedia, vol. 2022, Article ID 9169772, 9 pages, 2022. https://doi.org/10.1155/2022/9169772

Wojciech Jarosz. Efficient Monte Carlo Methods for Light Transport in Scattering Media. Ph.D. dissertation, UC San Diego, September 2008.

CHAPTER 15, On the Importance of Sampling, Matt Pharr, NVIDIA