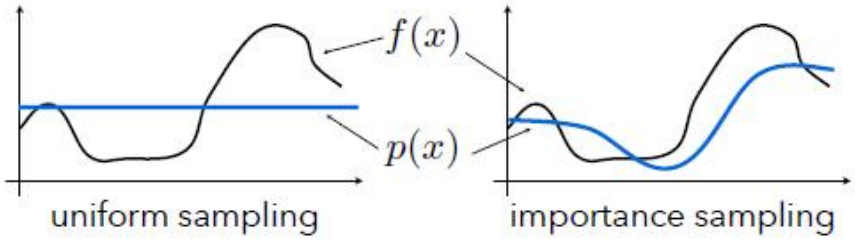
Three Variance Reduction Techniques Implementation

1. Importance Sampling (cosine-weighted)

Importance Sampling is a technique used to reduce the variance in Monte Carlo integration by sampling more frequently in regions that contribute more to the final result.

$$L_o\left(p,\omega_o\right) \approx \frac{1}{N}\sum_{i=1}^{N}\frac{L_i\left(p,\omega_i\right) f_r\left(p,\omega_i,\omega_o\right)\left(n\cdot\omega_i\right)}{p\left(\omega_i\right)}$$



$$f(\omega_i) = L_i\left(p,\omega_i\right) f_r\left(p,\omega_i,\omega_o\right)\left(n\cdot\omega_i\right)$$

We need to let probability density function of sampling be proportional to the integrand function. In this case, I choose cosine-weighted term.

We can theta and phi as the formula below. (modified in *sampleDirection* function)

$$\theta = \cos^{-1}\sqrt{(1-\xi_1)}, \phi = 2\pi\xi_2$$

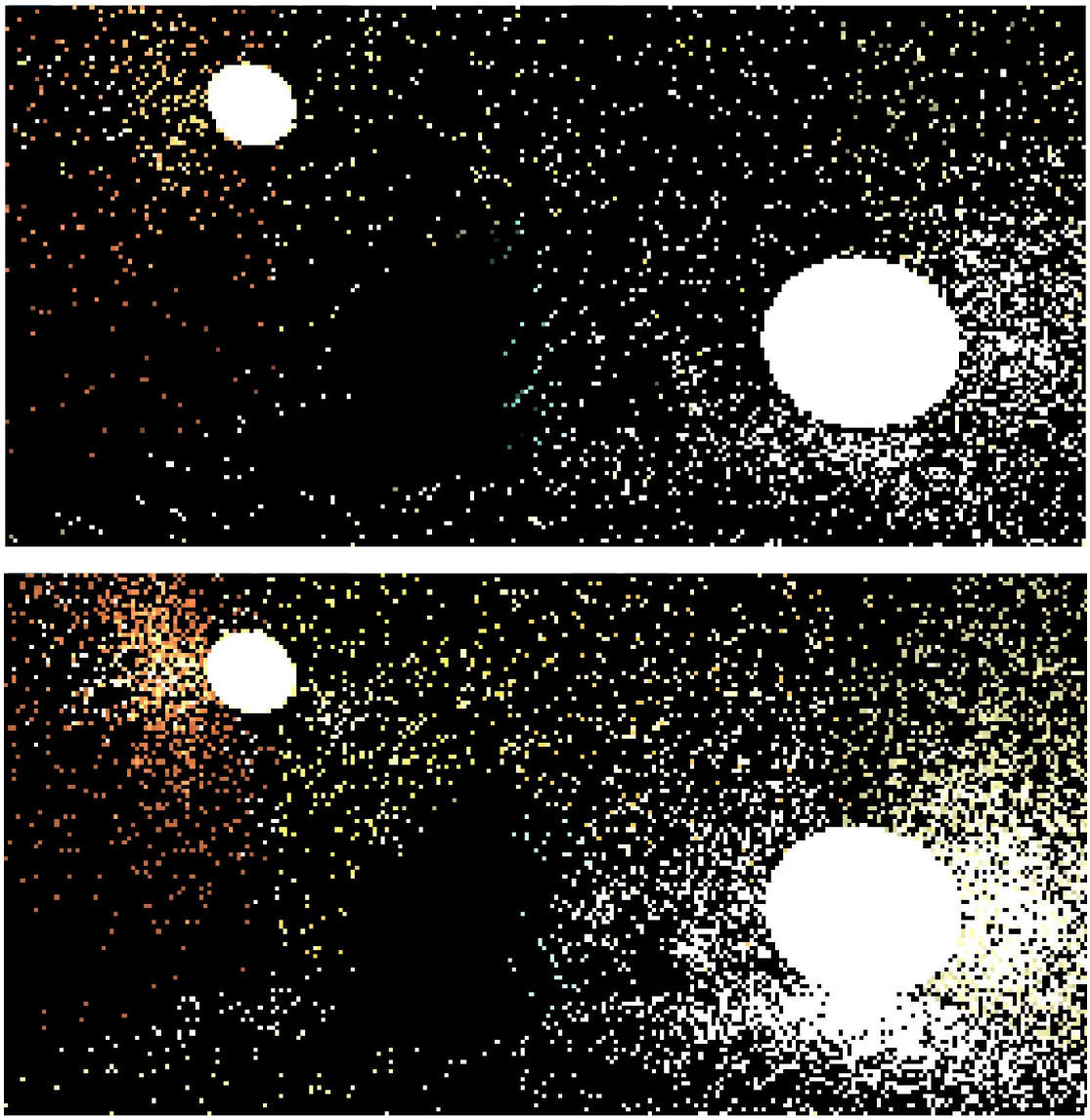
Converted spherical coordinates theta and phi into Euclidean coordinates 3D direction vector (x, y, z). Use *makeLocalFrame* function to convert coordinate system, ensuring that the sampled directions are aligned with the surface.

Calculate sampling pdf as the formula below.

$$pdf(\theta,\phi) = \frac{cos\theta sin\theta}{\pi}$$

Finally, we can get a sample direction with probability based on importance sampling.

The below figure shows the result: the bottom image (with importance sampling) converge much further than the top image (no variance reduction, uniform random sampling). At the same time, there is no color shifts or brightness issues (no bias). Both images are rendered at resolution 256×128.



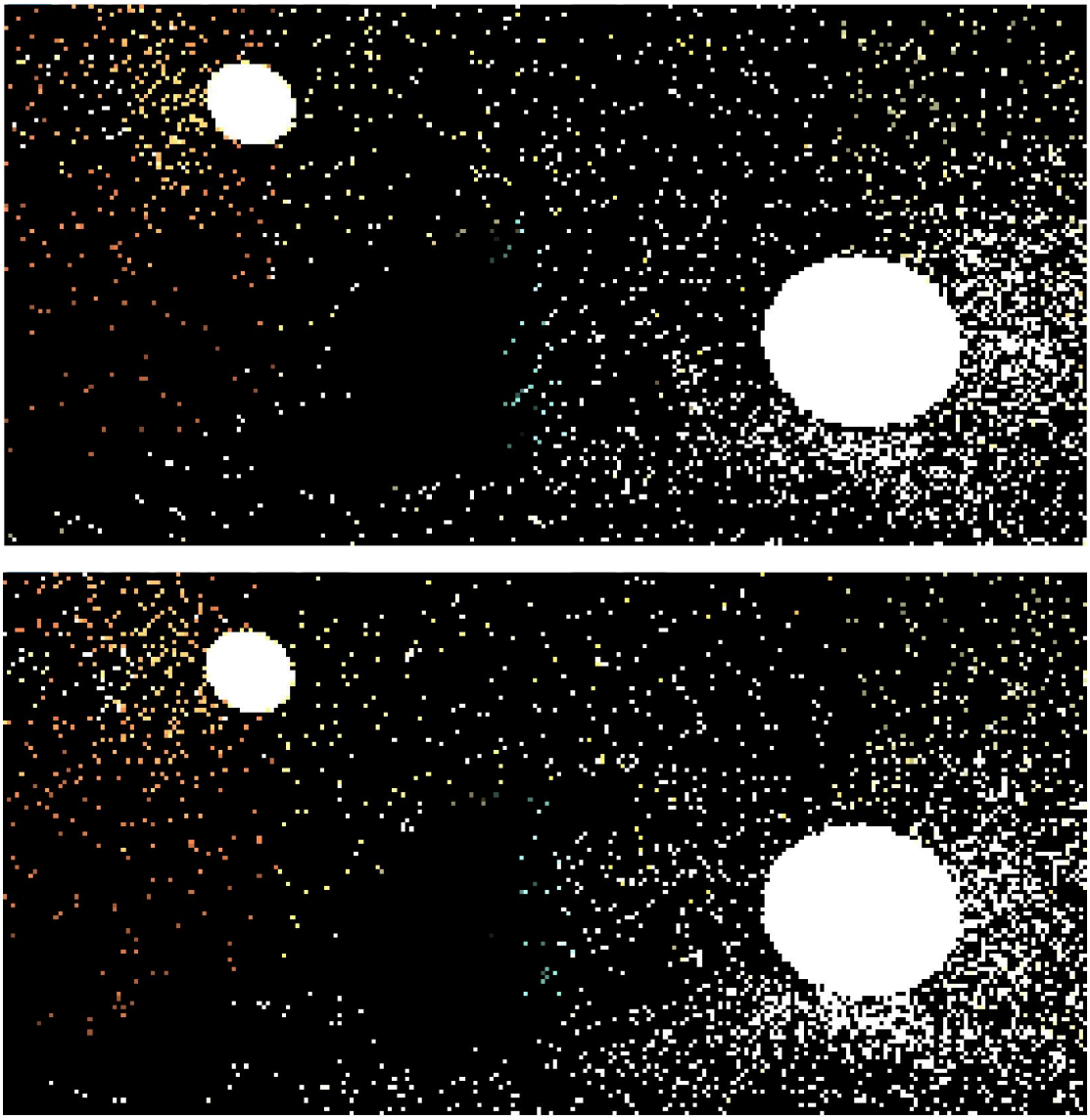
2. Quasi-Monte Carlo (Halton pattern)

Quasi-Monte Carlo (QMC) methods use low-discrepancy sequences (Halton sequence in this case) instead of random numbers to sample points in the integration space. Halton sequences have more regular patterns than purely random sequences, and they are designed to provide more uniform coverage of the integration space.

I implement a *haltonSequence* function to produce Halton sequence. After that, call it in *sample, sample2, sample3* functions with prime radical inverse base (2, 3, 5) instead of *uniformRandom()* to generate somewhat uniformly random sample coordinates ξ0 and ξ1

in Halton pattern. The rest part is same as the original.

The below figure shows the result: the bottom image (with Quasi-Monte Carlo sampling) seems no significant improvement compare to the top image (no variance reduction, uniform random sampling). At the same time, there is no color shifts or brightness issues (no bias). Both images are rendered at resolution 256×128.



3. Next-Event Estimation

The Next-Event Estimation (NEE) is used to sample light sources directly, estimating the contribution of each light source individually, which can be more efficient than relying solely on indirect illumination.

I implement s*ampleLight* function to sample an endpoint on light source sphere, calculate direction from current hit position and probability as the next sample. A

testVisibility function is also needed to check if the ray path is blocked by something else.

The below figure shows the result: the bottom image (with next-event estimation) converge much further than the top image (no variance reduction, uniform random sampling). At the same time, there is no color shifts or brightness issues (no bias). Both images are rendered at resolution 256×128.