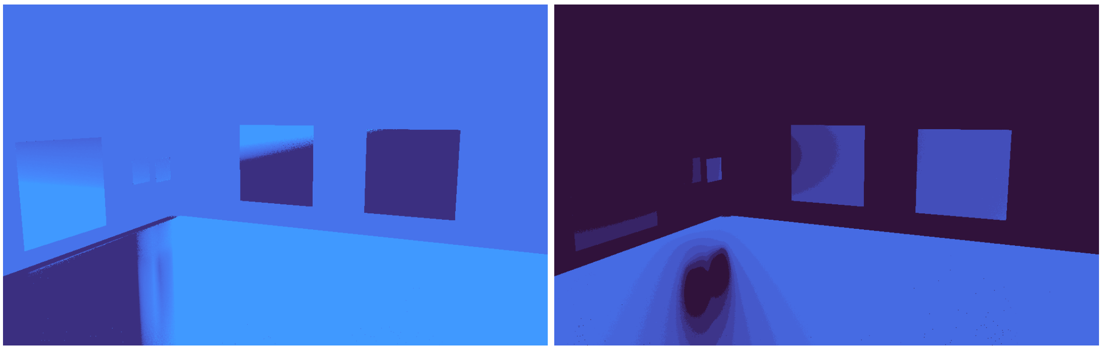
Internship report

Defensive sampling in Monte Carlo rendering

Hubert-Brierre Pierre

Supervised by
Iehl Jean Claude



Computer science department
University Claude Bernard Lyon 1

LIRIS

Lyon 1

ENS DE LYON

Contents

1 Abstract

1.1 English

Usually, to render an image using a Monte Carlo process, we cast many light rays in the scene using several algorithms to generate thoses rays. Some algorithms are very specific, working very well in certain complex situations, but being awful in more simple ones. Other algorithms are more general, being correct in every situation. The latter family of algorithms is called defensive sampling.

In this report we will talk about two defensive strategies. The first one smooths out the specular lobe to catch more incoming light. The second one estimates if there is more light on the left or right direction to cast more rays in that direction.

1.2 Français

Généralement, pour calculer une image en utilisant une méthode de Monte Carlo, on lance beaucoup de rayons dans la scène en utilisant divers algorithmes pour générer ces rayons. Certains algorithmes sont très spécifiques, fonctionnant exceptionnellement bien dans certaines situations complexes, mais étant plutôt mauvais dans des cas plus simples. D'autres algorithmes sont plus généreaux, ayant de bonnes performances dans la plupart des situations. Ces derniers sont appelés échantillonnage défensif

Je vais dans ce rapport aborder deux stratégies défensives. La première atténue le lobe spéculaire pour attraper plus de lumière. La deuxième estime la quantité de lumière venant de droite et de gauche afin de lancer plus de rayons dans la direction ayant plus de lumière.

2 Introduction

2.1 Life at laboratory

I did my internship at the LIRIS (Laboratoire d'InfoRmatique en Image et Systèmes d'information) with Jean Claude Iehl as my supervisor. I thank him for the time he gave me. We were seeing each other several hours per week, discussing my internship, my difficulties and my ideas as well as some related problems in the computer science field that were not directly linked to my research but were still cultivating to talk. He was very helpful to me because of his knowledge on the state-of-the-art, giving me papers to read, ideas to explore as well as pointing out when my ideas had already been explored (It was the case for the most part, it is very hard to have novel ideas in a field studied in depth for over 40 years).

The lab's life was very productive, I was in an open space with 8 other interns working on their own project. This gave me the opportunity to exchange with them and learn a few things about their methodologies and sometimes ask how they would solve my problems to open my eyes on some concepts that I might have missed. Especially with Pacome, another intern that worked on the same subject as me with another approach.
Furthermore, I could easily talk to everyone in the lab. This was particularly useful when I had the hint of an idea, I could just ask a specialist about it and he would tell me if it was worth investigating or not (It was mainly not worth my time).
We also had weekly meetings with the PhD students during which someone explained a scientific article. I have presented a meeting about severals ways to divide a sphere into parts of equal area. Finally, the proximity with other researchers gave me opportunities. I had the opportunity to assist two PhD defense and one HDR defense. All those presentations are important for our productivity. It is useless to search something alone. Just hearing someone else talking about their project gives you ideas and opens your mind (even if the presentation has nothing to share with your project, because in truth there is always a common point, the reasoning, the problem solving, the scientific method).
I was offered with 3 thesis subjects and I could ask for a 3 month internship (which I will start in October for my final year of studying at the ENS de Lyon)

The laboratory has to close during July. Everyone was working from home during this time. Hopefully this was not an issue for me as it was only during the last weeks of my internship. I could focus on writing my report and creating a clean repo for my code so that the work done by me can easily be accessed by anyone in the future.

2.2 Context

Image rendering is a major problem in today's industry. Creating physically accurate images quickly is crucial for several applications such as in :

- Animated movies : images in movies are very detailed, leading to months of rendering time. Even an improvement of 1% of rendering time can lead to days of computation saved.
- Video games : faster rendering for a video game allows for more details in the images and better visual effects.
- Advertising : Most online products sold are presented with a picture, usually rendered using a 3D model of the product and a physically accurate rendering engine.

2.3 Symbols and Notations

Here are the different notation used:

- $\mathbb{E}(X)$ is the expectancy of some random variable X
- $\mathbb{V}(X)$ is the variance of some random variable X
- For X a *c.r.v* means "For X a continuous random variable"
- For $(X_i)_{1\leq i\leq N}$ *i.i.d* means "For $(X_i)_{1\leq i\leq N}$ *c.r.v* independent and identically distributed"

2.4 Structure of the report

My report is structured in a way that makes the comprehension as easy as possible. Every proof is gathered at the end to allow the reader to focus on the idea explained and not on the mathematical details of it.

However, keep in mind that the mathematical part of my report is an essential part of my work. It is crucial for the development process, especially in a mathematical field such as mine. Having a formula and being able to reformulate, reduce, optimise is necessary to have a good comprehension on how things work and how to improve them. I spent days working on equations to progress in my thinking. The mere fact that there is an equation makes the algorithm possible. The fact that an equation is unsolvable is proof that I'm heading in a dead end.

Furthemore, without these equations, nobody could ever take back my work and improve it.

2.5 Methodology

My internship was divided into two parts.

- I started by studying the state-of-the-art. I implemented several papers to see how things work, to change parameters, and to try ideas. Overall the time spent doing that gave me a good grasp on the field as well as an intuition that will help me in my research of novelty. By doing that I understood what was really difficult in my internship, what was the problem and how did people try to solve it before me.
- The second part of my internship was more experimental, I had two main ideas to implement, test and tune to make a proof of concept. It is during this part that all the information gathered when reading papers really helped me, my intuition guided me in directions that were sometimes improving my work. It does not matter if I failed sometimes, it was to be expected, as long as I could find one thing that worked, every failed attempt was a hidden success.

3 Key Concepts

3.1 Ray casting

In computer graphics, to have an image physically accurate, we have no choice but to simulate the flow of light in the scene. We are doing that by casting rays through the scene to modelize photons moving from a light source to the camera. Except that we are usually casting them from the camera to a light source (which is equivalent because of the light reversibility principle) for performance issues.
The algorithm consists of casting a ray from the camera through a pixel and making it bounce in the scene as long as it does not hit a light source.

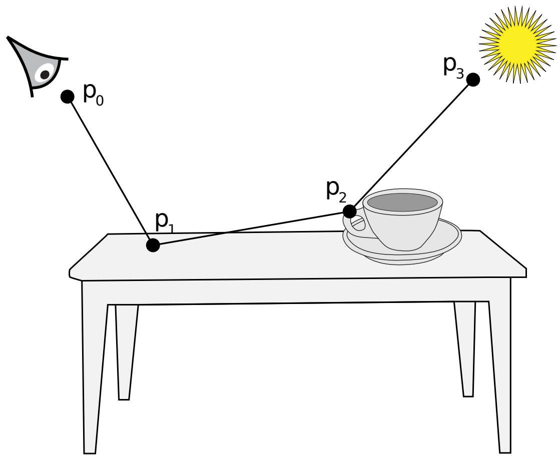


Figure 1: Illustration from PBRT book showing the path of a ray in the scene.

3.2 Light equation

The light transport is ruled by the light equation[5]. This equation describes how a light ray bounces when colliding with something.

$$L_o(p,\omega_0) = L_e(p,\omega_0) + \int_\Omega brdf(\omega,\omega_0)L_i(p,\omega)\cos(\omega\cdot n)d\omega$$

With:

- L_o the outputted light in a direction
- p the 3D point of intersection
- ω_0 the output direction
- Ω all possible direction (the upper hemisphere)

- ω a direction in the hemisphere
- $brdf(\omega, \omega_0)$ the light absorption of the material
- L_i the inputted light in a direction
- n the normal direction of the surface at the collision point
- L_e the emissive light in the outputted direction

It is important to see that the term L_i is obtained recursively using this integral at another intersection point q somewhere in the direction ω. This means that there are several recursive integrals to compute each time, making the problem really difficult to solve. This is why the objective of every monte carlo renderer is to compute this integral as efficiently as possible.

3.3 Monte Carlo for integral estimation and importance sampling

The Monte carlo integral estimation[9] is a Monte carlo algorithm that approximates the value of an integral using the following property:

For f a function to integrate over a set Ω, for X *c.r.v* over Ω with a probability density function p such that $f(x) \neq 0 \Rightarrow p(x) \neq 0$, then:

$$\int_{\Omega} f(x)dx = \mathbb{E}\left(\frac{f(X)}{p(X)}\right)$$

(Demonstration in annexe : 7.1)

This expectancy can be naively estimated by computing the average of N samples $(X_i)_{1 \leq i \leq N}$ *i.i.d* of probability density p:

$$\mathbb{E}\left(\frac{f(X)}{p(X)}\right) \approx \frac{1}{N}\sum_{i=1}^{N}\frac{f(X_i)}{p(X_i)}$$

By doing that, the squared error $\mathscr{E}^2$ of this method is:

$$\mathscr{E}^2 = \frac{1}{N}\mathbb{V}\left(\frac{f(X)}{p(X)}\right)$$

(Demonstration in annexe : 7.1)

Note : To have an error as low as possible, there are two options.

The first is to evaluate more samples (which can take some time because the error is proportional to the inverse square root of the number of samples).

The second is to decrease the variance of $\frac{f(X)}{p(X)}$ by choosing a "good" function p. The latter is called **Importance Sampling** in the literature.

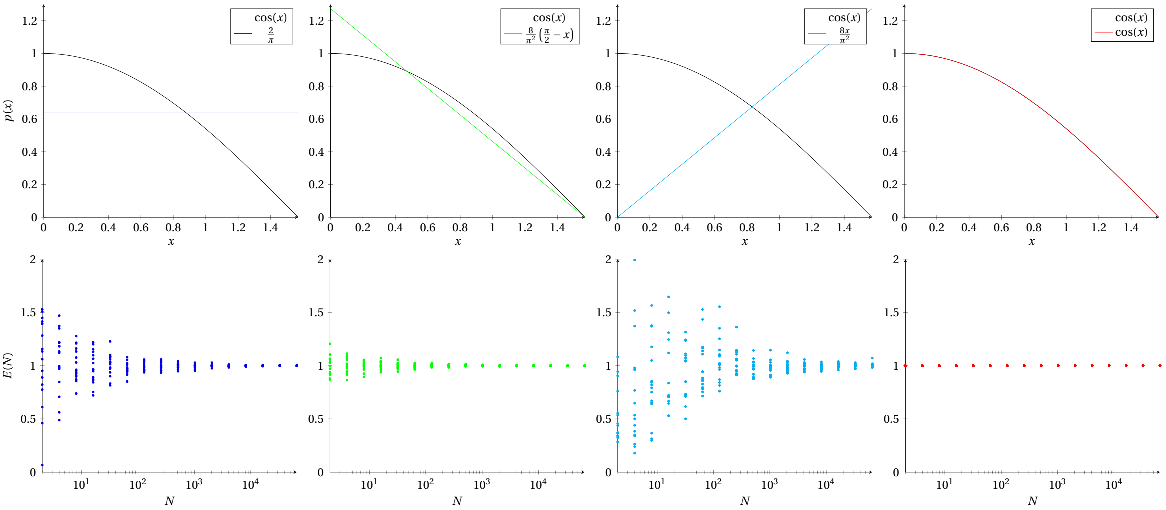


Figure 2:
Estimations of the integral $\int_0^{\frac{\pi}{2}} \cos(x)\,dx$ using the formula :
$E(N) = \frac{1}{N}\sum_{i=1}^{N} \frac{\cos(X_i)}{p(X_i)} \approx \int_0^{\frac{\pi}{2}} \cos(x)\,dx$ with three different probabilities density p (colored curves). Each dot represents the estimation of the integral using N different samples.
We can see that for a low number of samples, the dots are very sparse, meaning that approximations are not precise. We can also see that the more alike are the probability density and the integrand, the better are the approximations (note that the uniform distribution is not the worst sampling strategy here). Finally for the red curves, all dots are on the same line. The approximation is exact even when using only one sample. This is a special case where the probability density used is the integrand ($\cos(x)$), leading to a constant ratio $\frac{\cos(x)}{p(x)}$.

4 State-of-the-art

4.1 Sample strategies

As demonstrated before, the choice of the sampling strategy is crucial for the performance of the monte carlo integration. This is why researchers have been searching for strategies always more advanced to reduce the variance in the monte carlo integration.

One major problem in the light equation is the complexity of the integrand. A product of several independent terms. Because terms are independent, scientists have tried to efficiently sample those parts separately to increase the efficiency of the algorithm.

There exist two regular ways to measure the performance of a sampling strategy.

- The first is the *performance*. It measures the error after having cast a certain amount of light rays.
- The second is the *efficiency*. It measures the error after having spent a certain time computing.

The latter, in practice, is the best metric to optimise (people want nice images fast). However, this metric is hardware and implementation dependent. This is why, sometimes in the literature, scientists compare strategies by comparing their performances.

4.1.1 Uniform sampling

By default, we can use a uniform sampling in the hemisphere Ω. This is a really simple strategy, very inefficient in a complex scene. However, its simplicity makes it really quick to sample and very efficient in simple scenes.

4.1.2 BRDF sampling

A first optimisation is to sample proportionally to the $brdf$ term. This is usually easy because this term is analytical.
We can even go a bit further and sample a ray proportionally to the product of the $brdf$ and the cos.

4.1.3 Light sampling

Finally, the last usual sampling strategy used is to sample a direction toward the light sources directly. This is especially effective because in a lot of scenes, most of the light comes directly from the light sources. Meaning that the L_i term is strong in the direction of light sources. By avoiding casting rays in lightless regions, we can spare a lot of computation and time.

4.2 Multiple Importance Sampling

4.2.1 Theory

We have seen some sampling strategies, their strengths and weaknesses. In 1995, Veach found a way to combine thoses strategies and get the best of all worlds[14].

The idea is to sample each strategy and combine the sample.
In another way, for $p_1, \cdots, p_n$ different strategies, for $n_1, \cdots, n_n \in \mathbb{N}$, We give n_i sample to each strategy for a total of $N = \sum_{i=1}^{n} n_i$ samples.

The strength of this technique is that we can do a weighted average of all the samples to reduce the variance. Because some sampling strategies are more suited to sample certain configurations, we can increase the weight of samples created by thoses well suited strategies and decrease the weight of samples created by less interesting strategies.

There are usually two ways to compute the weight of a sample. For a sample $\omega \in \Omega$ sampled by the strategy i, we can give a weight $w_i(\omega)$ depending on $c_i = \frac{n_i}{N}$ to this sample:

- **The balance strategy :** $w_i(\omega) = \frac{c_i p_i(w)}{\sum_{j=1}^{n} c_j p_j(w)}$ Which is provably good[14].

- **The Power strategy :** For a coefficient β constant (usually $\beta = 2$): $w_i(\omega) = \frac{(c_i p_i(w))^\beta}{\sum_{j=1}^{n} (c_j p_j(w))^\beta}$ Which usually leads to smaller variances.

The new estimator is, for $(X_{i,j})_{1 \le j \le n_i}$ $i.i.d$ of density probability p_i:

$$\int_\Omega f(x)dx \approx \sum_{i=1}^{n} \frac{1}{n_i} \sum_{j=1}^{n_i} w_i(X_{i,j}) \frac{f(X_{i,j})}{p_i(X_{i,j})}$$

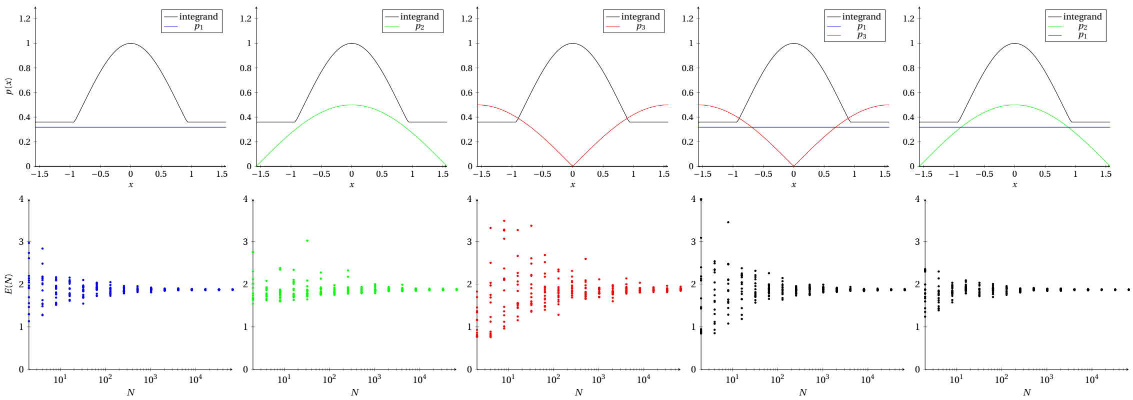


Figure 3:
Example for the balance strategy. Here we are trying to compute the integral of the black curve using Monte carlo integration. The blue, green and red plots are the result of the approximation using separately either the blue, green or red probability density. The black plot instead uses the balance strategy to combine those probability strategies. It's important to see that the balance strategy does not make miracles, if no primary sampling techniques are good to approximate the integrand, the MIS combination of them will have high variance (fourth plot). However, in some cases the achieved variance is lower than the variance of all primary samplers (fifth plot).

Of course other weights can work. (and some scientists are working with other weights[4, 10, 3]) as long as they respect two criterion:

- $p_i(\omega) = 0 \Rightarrow w_i(\omega) = 0$
- $f(\omega) \neq 0 \Rightarrow \sum_{i=1}^{n} w_i(\omega) = 1$

(Demonstration in annexe : 7.2)

Note here that, even if w_i are weights to compute the average of samples, they do not need to be positive. In fact Kondapaneni[6] has found the weights that minimize the variance of such estimators and they can be negative or greater than one. (However, computing those optimal weights is as hard as computing the light equation, it is not possible to use them in practice). Furthemore, it is important to see that the MIS strategy does not make miracles. If no primary estimator is well suited for the integrand, the MIS combination of them will not be suited either. This is why, during my internship, I focused my work on *defensive sampling*. My objective was not to have really good sampler, but to have samplers that are not so bad in the most possible configuration.

(Demonstration in annexe : 7.2)

4.2.2 Optimisation of n_i

From now on we will only consider the **Balance** heuristic. In this special case, the MIS estimator becomes:

$$\int_\Omega f(x)dx \approx \frac{1}{N}\sum_{i=1}^{n}\sum_{j=1}^{n_i}\frac{f(X_{i,j})}{p(C,X_{i,j})} = F$$

with $C = (c_i)_{1\leq i\leq n}$ and $p(C,X) = \sum_{i=1}^{n} c_i p_i(X)$
(Demonstration in annexe : 7.2)

Then, the variance of this estimator is:

$$\mathbb{V}(F) = \frac{1}{N}\sum_{i=1}^{n}\left(\int_\Omega \frac{c_i p_i(x) f^2(x)}{p^2(C,x)}\, dx - \frac{\mu_i^2}{c_i}\right)$$

With $\mu_i = \int_\Omega w_i(x) f(x)\, dx$
(Demonstration in annexe : 7.2)

The expression obviously depends on the coefficients c_i and can be minimized by a good choice of c_i (this corresponds to giving more samples to some strategies that might be more efficient in order to decrease the variance). Different methods exist to optimise those coefficient:

- The variance can be minimized through a gradient descent[11].
- We can use heuristics, easier to optimise to try and select goods coefficients[13, 12].
- I tried to set the coefficient proportionally to the inverse of the variance of each sampling strategy in order to give more samples to the strategy having the smallest variance. (this is still an heuristic, but way easier than the current state-of-the-art).

Usually in the literature we only consider two samplers to test a MIS estimator. This is why, from now on, we will only consider a Balance MIS with only two samplers (a **brdf** sampler and a **light** sampler). Because we have $n = 2$ and $c_1 + c_2 = 1$, we can parametrise the problem by $\alpha = c_1 = 1 - c_2$. A pixel with an α of 1 will only use the **brdf** sampler, and a pixel with an α of 0 will only use a **light** estimator. Furthemore, to avoid degenerate cases, usually people keep α bound in $[\tau, 1-\tau]$ with $\tau = 0.1$. In all my results, I will be using this guard with $\tau = 0.1$

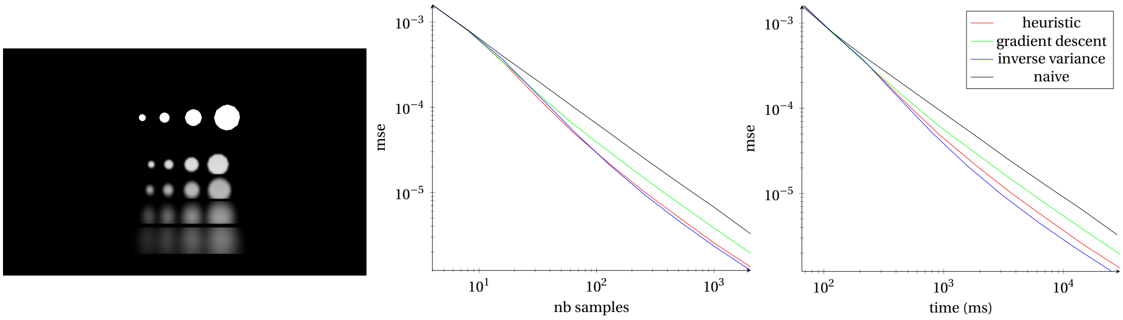


Figure 4:
The performance and efficiency of different estimators change only the number of samples allocated to each strategy. The scene in the left (made of four lights of different sizes and four iron blades of different roughnesses) was computed using a **brdf** sampler and a **light** sampler, combined using the balance MIS strategy. The gradient descent[13] and the heuristic[11] are state-of-the-art methods. The naive methods simply set $\alpha = 0.5$. The inverse variance method sets the coefficient proportional to the inverse of the variance of each sampler.

One can observe that all techniques are better than the naive method (which is to be expected). Furthemore, the more efficient method is not a state-of-the-art one but a simple one. This can be explained for several reasons. First, this method is very simple, which leads to very quick computation. Secondly, because of omitted details, the two state-of-the-art methods have to restart estimating quantities each time they update the coefficient α. A limitation that the other methods do not have. *During my internship I kept this in mind, trying to design sampling strategies that do not need to restart estimating things each time they update themselves.*

To use thoses techniques, one need to estimate certain quantities in every point of the scene and for every incoming direction. This is not achievable in practice. To palliate this, we usually compute the needed quantities on a per pixel basis. By doing that we assume that each ray passing through a pixel will intersect the scene at the same point and with the same incoming direction. This is not far from the truth before the first bounce. However this assumption does not hold after the first bounce. This is why, to test those methods, we restrict ourselves to the first bounce.

4.3 Data gathering

4.3.1 SD tree

There exists ways to still use those techniques when computing the full scene with bouncing rays.

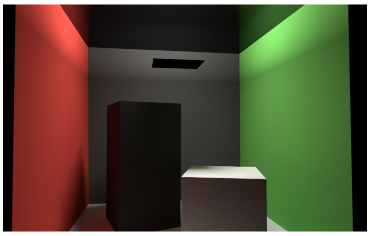
The first method is to restrict the use of thoses optimisation techniques to the first intersection and using a regular, not optimised MIS strategy for each other bounce[8].

The second is to use a spatio-directional tree (**SD** tree) [7]. This is a 5 dimensional tree that partitions the 3D scene cubes and the 2D hemisphere of each such cubes. If the cubes are small enough and the partition of each hemisphere is precise enough, the previous assumption holds and we can estimate our quantities to optimise the coefficient c_i. Be careful though because small partitions will have few rays to estimate quantities, leading to error in it and wrong choice of c_i

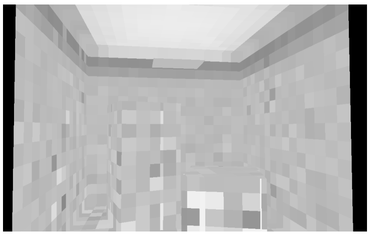
Keep in mind that those are not miracle solutions, the first solution only allows for one bounce optimisation. The second solution has a huge memory cost and cannot be as precise as wanted.

4.3.2 Example of use for SD tree

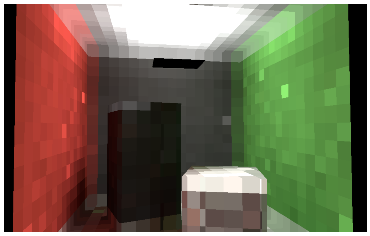
To better understand what is an SD tree, here is a simple application:
An obvious quantity that we can estimate using a SD tree is the incoming light. By doing that we can at each bounce either cast another ray to try to compute the incoming light or use the estimated incoming light stored in the SD tree (or an average of both). This really helps to compute scenes with indirect lighting. However, this method is *biased*, which means that it does not converge to the correct images, but offers noiseless images.



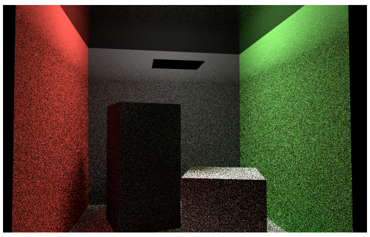
(a) The reference scene



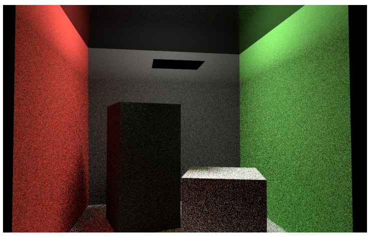
(b) The coefficient used for balance (in white $\alpha = 0$, in black $\alpha = 1$)



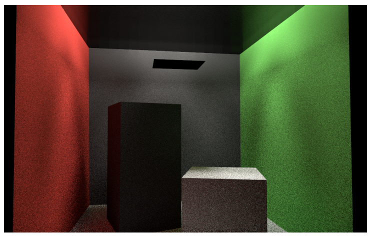
(c) The estimated incoming light in each cubes of the SD tree



(d) The scene rendered using the inverse variance heuristic



(e) The scene rendered using the inverse variance heuristic in the SD tree



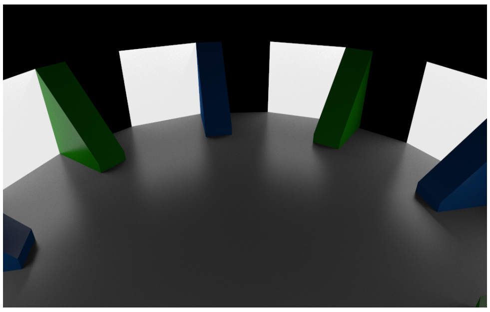
(f) The scene rendered by estimating the incoming light

Figure 5: A cornell box light by a strong indirect light toward a reflective ceiling rendered three different methods (each method with 512 rays per pixels)

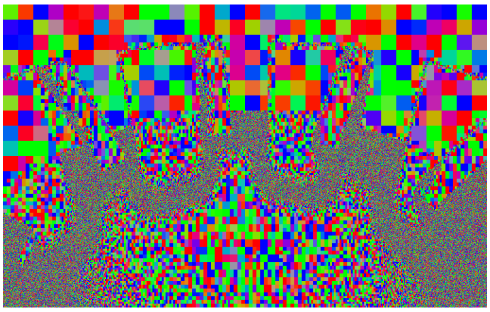
4.3.3 Pixel block

There is one final trick that we can use to speed up the computation and have better estimations, gather pixels by blocks. All previous techniques based on the estimation of something to adapt themself to the scene are greatly improved when gathering pixels together. Because all estimations suddenly gain 64 times more samples (for blocks of 8 by 8 pixels), increasing drastically the precision of the estimation.
However, this comes at the cost of having more spread out rays in the same estimators (like in part 4.3.1). To palliate this problem, one can simply split blocks once they reach a certain threshold (on the inside error[2], the number of ray cast, ...). This is the topic of this final method, that manages pixel blocks.

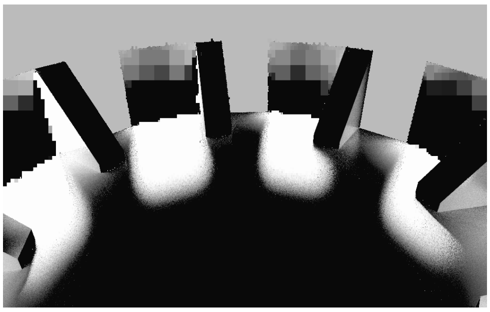
Another benefit from this method is that we can preemptively stop the computation of a block already noisefree[2] (some pixels are easier to compute than others) to focus on other blocks. This is called **Adaptive sampling** and it helps to have a good efficiency in our renderings.



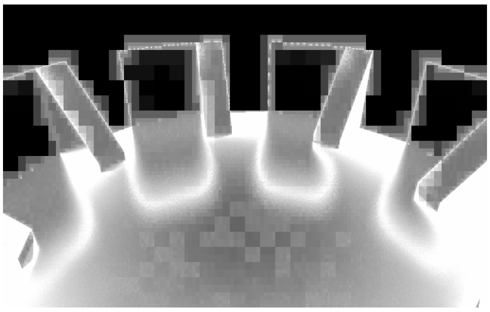
(a) The scene rendered using adaptive sampling



(b) The blocks at the end of computation (pixel in a same block share the same α)



(c) The α for each blocks



(d) The number of sample per pixel (black is 10 samples, white is 2048 samples)

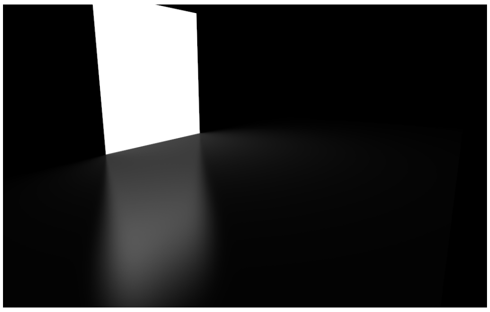
Figure 6: A scene rendered using adaptive sampling. Notice how the blocks with low variance are bigger and with less samples in it. It is indeed easier to compute them, so the algorithm stops processing them before having reduced them to the minimal size.

5 Our contributions

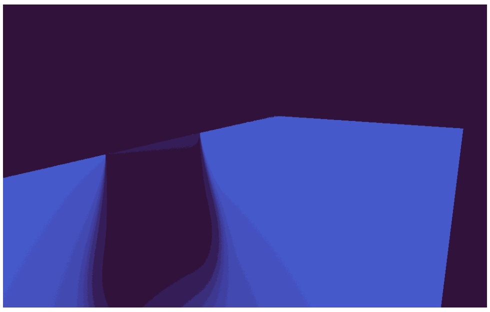
5.1 Large BRDF

5.1.1 Idea

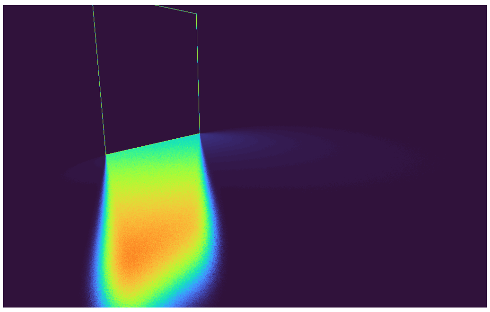
One of the two major ideas of this internship is based on the fact that, usually for reflective material (like metals or varnish), the brdf is a very narrow cone in the specular direction. Meaning that for those materials, when using a brdf sampler, the majority of rays will go in that direction. if there is no light in that direction, the estimator will suffer from a high variance. However, if the rays are nearly missing the light when using the brdf sampler, we can use the sampler of a brdf slightly less reflective material (a larger brdf) which will cast rays in a bigger cone in the specular direction and will hit the light more often, reducing the variance. (We are only changing the sampler associated to the material, not the material itself so the final colour will be the same).



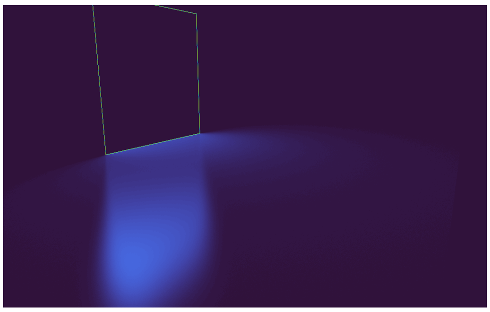
(a) The scene rendered



(b) The attenuation of reflectivity that minimizes the variance. (From 1 in black to 256 in blue)



(c) The variance of each pixel using the brdf sampler of a material 256 time less reflective



(d) The variance of each pixel using the classic brdf sampler.

Figure 7: A simple scene rendered using different attenuations of reflectivity. As you can see, overall the best option is to not change the material reflectivity, however a closer look at the border of the light reflect shows that for some pixels, changing the reflectivity leads to a variance reduction.

Fortunately for us, the variance of a pixel is just an integral computation that we can estimate using the monte carlo integration:

For a brdf sampler (or any sampler on the hemisphere) p and the associated estimator $E_p(X) = \frac{f(X)}{p(X)}$ (with X a *c.r.v* of probability density p), for f the integrand of the light equation3.2 and $\mu = \int_\Omega f(\omega)\, d\omega$, we have:

$$\mathbb{V}(E_p) = \int_\Omega \left(\frac{f(\omega)}{p(\omega)} - \mu\right)^2 p(\omega) d\omega$$

(Demonstration in annexe : 7.2)

Which, by introducing another sampler on the hemisphere q can be written and estimated as:

$$\mathbb{V}(E_p) = \int_\Omega \left(\frac{f(\omega)}{p(\omega)} - \mu\right)^2 \frac{p(\omega)}{q(\omega)} q(\omega) d\omega$$

5.1.2 Variance estimation fail

From the observation that it is indeed useful for some pixels to use a larger brdf, I tried to parametrize the problem with respect to the reflectivity attenuation with no success.

I was using the Blinn-Phong material model[1]. When computing the variance of the estimator, I could not extract the reflectivity attenuation k.

The material BRDF is define as:

$$brdf(\omega, \omega_0) = k_d \frac{1}{2\pi} + (1 - k_d) \frac{n_s + 8}{8\pi} \cos(\theta_h)^{n_s}$$

with $k_d \in [0,1]$ the proportion of diffuse and specular reflectance of the material, $h = \frac{\omega + \omega_0}{|\omega + \omega_0|}$, n the normal of the surface and $\cos(\theta_h) = h \cdot n$ the cosine of the angle between them.

The associated sampler has a density probability:

$$p_k(\omega) = k_d \frac{\cos(\theta)}{\pi} + (1 - k_d) \frac{\frac{n_s}{k} + 1}{2\pi} \frac{\cos(\theta_h)^{\frac{n_s}{k}}}{4h \cdot \omega}$$

With $\cos(\theta) = \omega \cdot n$ and k the reflectivity attenuation.

The variance of this estimator can be expressed as:

$$\begin{aligned}
\mathbb{V}(E_{p_k}) &= \int_\Omega \left(\frac{f(\omega)}{p_k(\omega)} - \mathbb{E}(E_{p_k})\right)^2 \frac{p_k(\omega)}{p_{k'}(\omega)} p_{k'}(\omega) d\omega \\
&= \int_\Omega \frac{f^2(\omega)}{p_k(\omega)}\, d\omega - \int_\Omega 2 f(\omega) \mathbb{E}(E_{p_k})\, d\omega + \mathbb{E}(E_{p_k})^2 \int_\Omega p_k(\omega)\, d\omega
\end{aligned}$$

And by assuming $k_d = 0$, we have:

$$\int_\Omega \frac{f^2(\omega)}{p_k(\omega)}\, d\omega = \frac{2\pi}{\frac{n_s}{k}+1}\int_\Omega \frac{4h\cdot\omega f^2(\omega)}{\cos(\theta_h)^{\frac{n_s}{k}}}\, d\omega$$

(Demonstration in annexe : 7.2)

As you can see, the parameter k is still in the integral, which prevents any analytic optimisation of it.

5.1.3 Variance estimation success

Instead of finding some quantities to estimate in order to obtain the best k possible, I chose to restrict myself to a set of predefined value for this parameter and separately estimate the variance for each associated sampler using the formula:

$$\begin{aligned}\mathbb{V}(E_p) &= \int_\Omega \left(\frac{f(\omega)}{p(\omega)} - \mu\right)^2 \frac{p(\omega)}{q(\omega)}\, q(\omega) d\omega \\ &\approx \frac{1}{N}\sum_{i=1}^{N}\left(\frac{f(X_i)}{p(X_i)} - \mu\right)^2 \frac{p(X_i)}{q(X_i)}\end{aligned}$$

With $(X_i)_{1\leq i\leq N}$ *i.i.d* of probability density q.

This means that by casting only rays using one sampler, we can estimate the variance of another sampler with little additional cost. This is what I did to create an adaptive sampler. This sampler starts the computation by approximating the light as well as the variances for several samplers with different reflective coefficients using the classic brdf sampler. As the computation progresses and the estimation becomes more and more reliable, I can swap the sampler to the one having the smallest variance.

One strength of this method is that the variance estimation does not depend on the sampler used for the estimation:

For $X_1, \cdots, X_N$ N independent *c.r.v* of density probability $q_1, \cdots q_N$, we have:

$$\begin{aligned}\mathbb{V}(E_p) &= \int_\Omega \left(\frac{f(\omega)}{p(\omega)} - \mu\right)^2 p(\omega) d\omega \\ &\approx \frac{1}{N}\sum_{i=1}^{N}\left(\frac{f(X_i)}{p(X_i)} - \mu\right)^2 \frac{p(X_i)}{q_i(X_i)}\end{aligned}$$

(Demonstration in annexe : 7.1)

Which allows us to update the parameter k (i.e changing the sampler) without restarting the estimation.

It turns out that there is another way to estimate the variance of each sampler that has a better convergence rate:

$$\mathbb{V}(E_p) = \int_\Omega \frac{f^2(\omega)}{p(\omega)q(\omega)} q(\omega)d\omega - \mu^2$$

(Demonstration in annexe : 7.2)

Because I'm only interested in finding the sampler having the smallest variance, I can dismiss the μ^2 as both $\mathbb{V}(E_p)$ and $\mathbb{V}(E_p)+\mu^2$ have the same sampler minimizing them. I am then searching the sampler that minimize:

$$\begin{aligned}\mathbb{V}(E_p)+\mu^2 &= \int_\Omega \frac{f^2(\omega)}{p(\omega)q(\omega)} q(\omega)d\omega \\ &\approx \frac{1}{N}\sum_{i=1}^{N} \frac{f^2(X_i)}{p(X_i)q(X_i)}\end{aligned}$$

With $(X_i)_{1\leq i\leq N}$ *i.i.d* of probability density q.

This approximation is better because it is analytically more stable (we only add numbers, the other estimation has to compute the square of the difference between two estimations).

Of course we can update the parameter k without restarting the estimation:

$$\int_\Omega \frac{f^2(\omega)}{p(\omega)} d\omega \approx \frac{1}{N}\sum_{i=1}^{N} \frac{f^2(X_i)}{p(X_i)q_i(X_i)}$$

With $X_1,\cdots,X_N$ N independent *c.r.v* of density probability $q_1,\cdots q_N$. (Demonstration in annexe : 7.1)

5.2 Left-Right sampling

5.2.1 Idea

The second major idea of the internship is to cut the hemisphere in two parts (I will call them the left and right parts) and estimate which part has more light in order to send more rays in that part.

5.2.2 Basis

For that purpose, I created another sampler p_γ (called **left-right** sampler) that is parametrised by $\gamma\in[0,1]$ to give more samples in one side. ($\gamma=0$ means all samples are cast in the right part, $\gamma=1$ means all samples are cast in the left part). Considering an already existing sampler p, Ω_L (resp. Ω_R) the left (resp. right) part of Ω we have:

$$p_\gamma(\omega) = 2\gamma \mathbb{1}_{\omega \in \Omega_L} p(\omega) + 2(1-\gamma)\mathbb{1}_{\omega\in\Omega_R} p(\omega)$$

With $\mathbb{1}_t = \begin{cases} 1 & \text{if } t \text{ is true} \\ 0 & \text{otherwise} \end{cases}$

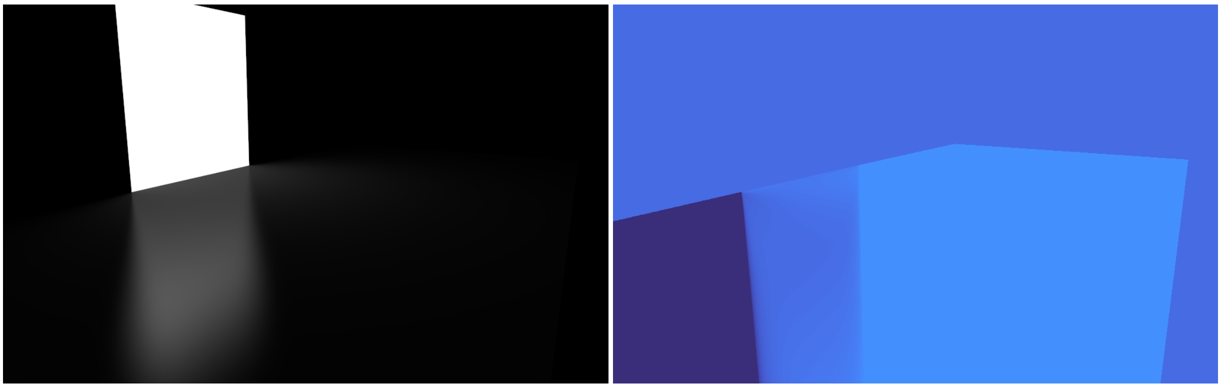
Note, this is a probability density under the assumption that $\int_{\Omega_L} p(\omega)d\omega = \int_{\Omega_R} p(\omega)d\omega = \frac{1}{2}$ (Demonstration in annexe : 7.3). This means that we must carefully choose p, Ω_L and Ω_R, this is a strong constraint as we will see later.

This new probability density has an associated variance:

$$\mathbb{V}_\gamma = \frac{V_L}{2\gamma} + \frac{V_R}{2(1-\gamma)} - \mu^2$$

With $V_L = \int_{\Omega_L} \frac{f^2(\omega)}{p(\omega)}\, d\omega$, $V_R = \int_{\Omega_R} \frac{f^2(\omega)}{p(\omega)}\, d\omega$ and $\mu = \int_\Omega f(\omega)d\omega$ (Demonstration in annexe : 7.3)

Which is minimal for $\gamma = \frac{\sqrt{V_L}}{\sqrt{V_L}+\sqrt{V_R}}$ (Demonstration in annexe : 7.3). Note that because V_L and V_R are independent from γ, wee can still estimate them even if we change the parameter γ (4.2.2) (this is critical to have good performance in the long run). (Demonstration in annexe : 7.1)



(a) The reference scene. (b) The γ coeficient (in black $\gamma = 0$, in blue $\gamma = 1$.)

Figure 8: Example of a scene with it's optimized γ coeficients.

5.2.3 Generalisation

We have just divided the hemisphere in 2 to decrease the variance, why not divide it in n part ? It is indeed possible: let $\gamma_1, \cdots, \gamma_n \in]0,1[$ with $\sum_{i=1}^n \gamma_i = 1$ and $\Omega_1, \cdots, \Omega_n$ a partition of Ω. Then we have a sampler p_{γ_i} defined by:

$$p_{\gamma_i}(\omega) = \sum_{i=1}^{n} n\gamma_i \mathbb{1}_{\omega\in\Omega_i} p(\omega)$$

This is a valid probability density under the strong condition that $\forall i \in [\![1, n]\!]$ $\int_{\Omega_i} p(\omega) d\omega = \frac{1}{n}$ (Demonstration in annexe : 7.3)

This new probability has an associated variance:

$$\mathbb{V}_{\gamma_i} = \sum_{i=1}^{n} \frac{V_i}{n\gamma_i} - \mu^2$$

With $V_i = \int_{\Omega_i} \frac{f^2(\omega)}{p(\omega)} d\omega$ and $\mu = \int_{\Omega} f(\omega) d\omega$ (Demonstration in annexe : 7.3)

This variance is minimal for $\gamma_i = \frac{\sqrt{V_i}}{\sum_{i=j}^{n} \sqrt{V_j}}$ (Demonstration in annexe : 7.3)

However, even if this is mathematically possible, there is some drawback at dividing too much the hemisphere:

- The more separate parts we have, the less samples we can cast in each different part, leading to poor estimation and poor results.
- The second drawback comes from the constraint $\forall i \in [\![1, n]\!], \int_{\Omega_i} p(\omega) d\omega = \frac{1}{n}$. This is not easy to achieve with an arbitrary number of parts n and any non trivial sampler p. We need to simplify the initial sampler p, leading to poor results.

5.2.4 Generalisation failed with the brdf sampler

Because the brdf sampler is symmetric with respect to the incident plane (a ray has the same probability of going left or right), the constraint $\int_{\Omega_L} p(\omega) d\omega = \int_{\Omega_R} p(\omega) d\omega = \frac{1}{2}$ is meet and we can use the brdf sampler as a primary sampler for the left-right sampler. However, I failed to generalize it.

Because Ω is the hemisphere, we can define it as:

$$\Omega = \left\{ \begin{pmatrix} \cos(\phi)\sin(\theta) \\ \sin(\phi)\sin(\theta) \\ \cos(\theta) \end{pmatrix} \text{ with } \theta \in [0, \frac{\pi}{2}], \phi \in [0, 2\pi] \right\}$$

With $0 = \theta_0 < \theta_1 < \cdots < \theta_n = \frac{\pi}{2}$ and $0 = \phi_0 < \phi_1 < \cdots < \phi_m = 2\pi$. We can then divide Ω into part $(\Omega_{i,j})_{\substack{0 \le i < n \\ 0 \le j < m}}$ defined by:

$$\Omega_{i,j} = \left\{ \begin{pmatrix} \cos(\phi)\sin(\theta) \\ \sin(\phi)\sin(\theta) \\ \cos(\theta) \end{pmatrix} \text{ with } \theta \in [\theta_i, \theta_{i+1}], \phi \in [\phi_j, \phi_{j+1}] \right\}$$

The objective is then to find such θ_i and ϕ_j such that $\int_{\Omega_{i,j}} p(\omega)\ d\omega = \frac{1}{nm}$. In the case of the brdf sampler, this constraint becomes:

$$\frac{1}{nm} = \int_{\Omega_{i,j}} p(\omega)\, d\omega = \int_{\theta_i}^{\theta_{i+1}} \frac{k_d}{\pi}(\phi_{j+1}-\phi_j)\frac{\sin(2\theta)}{2} + (1-k_d)\frac{n_s+1}{8\pi}\int_{\phi_j}^{\phi_{j+1}} \frac{\cos(\theta_h)^{n_s}}{h\cdot\omega}\sin(\theta) d\phi_h d\theta_h$$

(Demonstration in annexe : 7.3)

And there ends the computation, going further leads to no results, I am not able to compute the values of θ_i and ϕ_j for that equality to be true.

I choose to simplify the primary sampler. More precisely, I built a simple sampler for my special case such that the constraint $\int_{\Omega_{i,j}} p(\omega)\, d\omega = \frac{1}{nm}$ hold.

The idea is that I want to sample the vector h on a cosine lobe to have a good specular sampler. However, I tried it and totally discarding the diffuse term led to very high variance (because the specular term can be very small, leading to floatant errors). What I finally did was to divide Ω into part where the constraint $\int_{\Omega_{i,j}} \frac{n_s+1}{2\pi}\cos(\theta)^{n_s}\, dh = \frac{1}{nm}$ hold to set the different $\Omega_{i,j}$. Once the partition of Ω is done, the final sampler chooses uniformly one part (because the constraint gives equal probability to each part) and in that part, I draw a random vector h either from a cosine distribution or from a uniform distribution. The final sampler is:

$$p(h) = \sum_{i=0}^{n-1}\sum_{j=0}^{m-1} \mathbb{1}_{h\in\Omega_{i,j}}\left((1-k_d)\frac{n_s+1}{2\pi}\cos(\theta_h)^{n_s} + k_d\frac{1}{n2\pi(\cos(\theta_i)-\cos(\theta_{i+1}))}\right)$$

With $\cos(\theta_i) = \sqrt[n_s+1]{\frac{n-i}{n}}$ and $\phi_j = \frac{2\pi j}{m}$

Of course once the values for p, θ_i and ϕ_j are given, it's easy to check that everything works. The real challenge here was to find those objects. I do not pretend to have found the best partition of Ω or the best sampler to use, this can be a future work.

One can check that we have indeed:

$$\int_{\Omega_{i,j}} p(h)\, dh = \frac{1}{nm}$$

(Demonstration in annexe : 7.3)

And that p is a valid density probability:

- $\int_\Omega p(h)\, dh = 1$
- $\forall h \in \Omega, p(h) \geq 0$

(Demonstration in annexe : 7.3)

Finally, because $\omega = 2h\cdot\omega_0 h - \omega_0$, we have $\frac{d\omega}{dh} = 4h\cdot\omega_0$ and $p_\omega(\omega) = \frac{p(h)}{4(h\cdot\omega_0)}$, giving us the direction ω to sample in our graphic engine as well as it's probability.

6 Conclusion

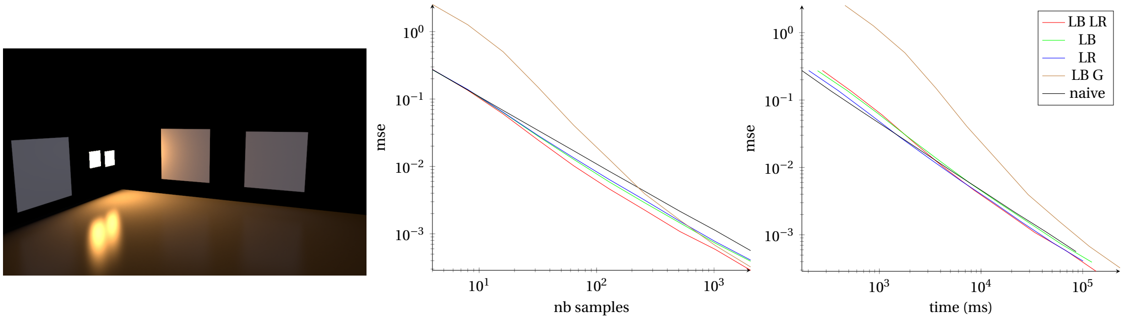


Figure 9: Efficiency and performance test done by rendering the scene in the left using pixel blocks of 8x8 and $\tau = 0.1$.
LB is the large brdf sampler, LR is the left-right sampler, LB LR is the combination of both, LB G is the generalized version of left-right combined with large brdf and naive is just the brdf sampler.

Here are the results of my test. We can see that in the long run every strategy is better than the naive one in terms of performance, however the generalized version of left-right is not efficient at all. Furthemore, this strategy is initially worse than the other because I had to simplify the primary sampler for the strategy to work (because of the constraint $\int_{\Omega_i} p(\omega)d\omega = \frac{1}{n}$). Overall, except for this strategy, the three others are better both in efficiency and in performance than the naive one. Special mention to the combination of the large brdf and the left right sampler how to achieve the best of both samplers and give the best performance.

However, those results are very scene dependent. Here are more test scene on wich the results are differents:

Overall, ours strategies are more efficient than the naive one (exept for certain scene, but always within a small margin). This make our strategies better defensive samplers than the naive one, specialy in scene where most of the light comme from a single direction.

To sum up, to render an image, we use probabilistic algorithms that have performances greatly dependent on the sampler used. There are several different samplers, all with their strengths and weaknesses and a way to combine thoses samplers to get the best of them. However, if no primary sampler is suited to the scene, the combination of them will not be suited either. This is why we need defensive sampling, a simple sampler design to be suited to most cases instead of being powerful in some special cases.

I designed two such samplers, the large brdf sampler and the left-right sampler, both are learning their environment to have a low variance on most possible configurations. Furthemore, we can merge thoses two samplers to further improve the performances and the use cases.

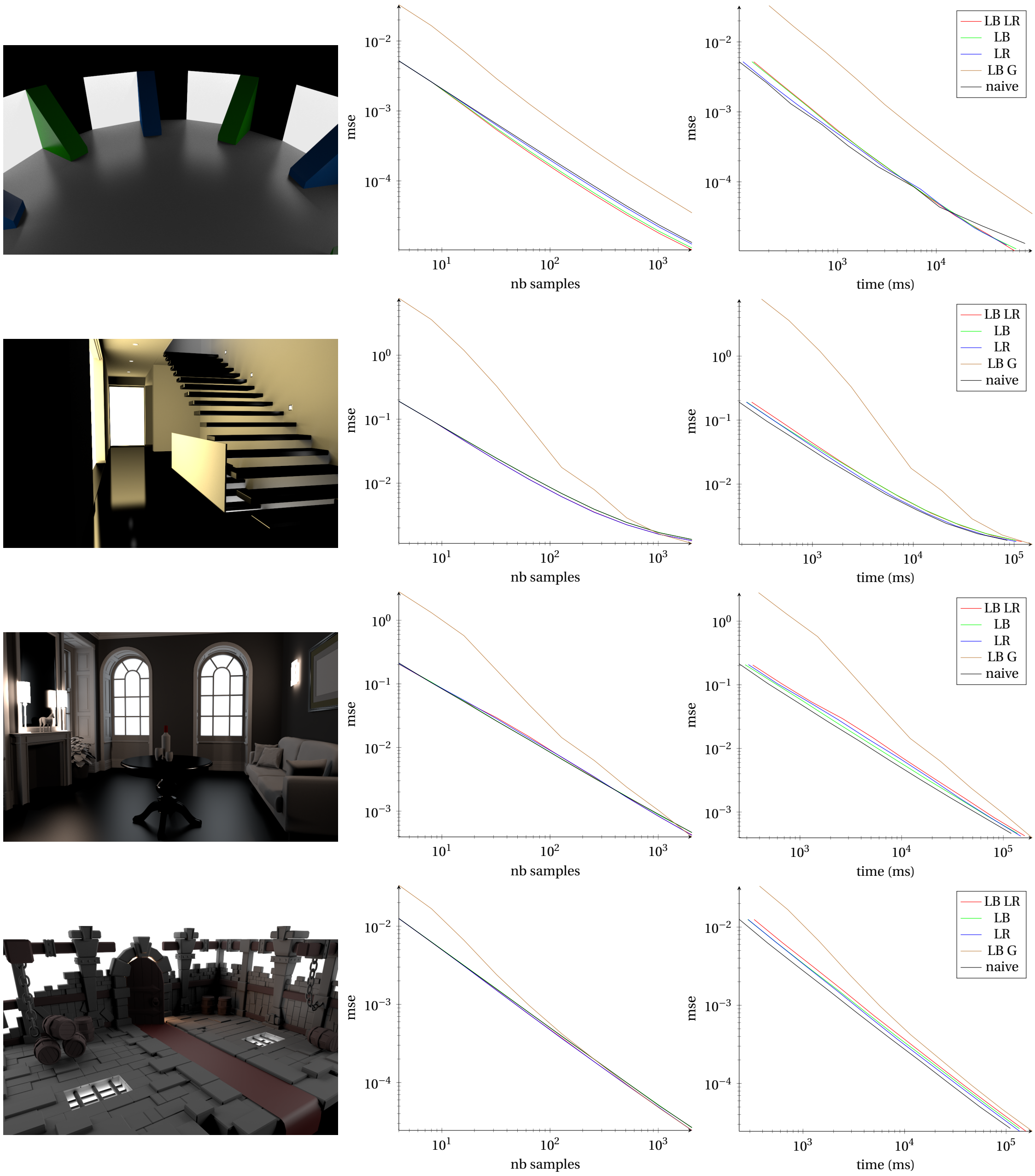


Figure 10: Efficiency and performance test done by rendering the scene in the left using pixel blocks of 8x8 and $\tau = 0.1$.
LB is the large brdf sampler, LR is the left-right sampler, LB LR is the combination of both, LB G is the generalized version of left-right combined with large brdf and naive is just the brdf sampler.

7 Proofs

7.1 Monte Carlo proof

Expectancy of monte carlo sampler

For f a function to integrate over a set Ω, for X *c.r.v* over Ω with a probability density function p such that $f(x) \neq 0 \Rightarrow p(x) \neq 0$, then:

$$\int_\Omega f(x)dx = \int_\Omega \frac{f(x)}{p(x)}\, p(x)dx = \mathbb{E}\left(\frac{f(X)}{p(X)}\right)$$

Error of monte carlo sampler

For X and $(X_i)_{1\le i\le N}$ *i.i.d* of density probability p. We have : $\mathbb{E}(X) \approx \frac{1}{N}\sum_{i=1}^{N} X_i$

The squared error $\mathcal{E}^2$ of this method is:

$$\begin{aligned}
\mathcal{E}^2 &= \mathbb{E}\left(\left(\mathbb{E}(X) - \frac{1}{N}\sum_{i=1}^{N} X_i\right)^2\right) \\
&= \mathbb{E}\left(\left(\mathbb{E}\left(\frac{1}{N}\sum_{i=1}^{N} X_i\right) - \frac{1}{N}\sum_{i=1}^{N} X_i\right)^2\right) \text{ because } X \text{ and } (X_i) \text{ are } \textit{i.i.d} \\
&= \mathbb{V}\left(\frac{1}{N}\sum_{i=1}^{N} X_i\right) \text{ by definition of the variance} \\
&= \frac{1}{N^2}\sum_{i=1}^{N} \mathbb{V}(X_i) \text{ because } (X_i) \text{ are } \textit{i.i.d} \\
&= \frac{1}{N^2}\sum_{i=1}^{N} \mathbb{V}(X) = \frac{1}{N}\mathbb{V}(X)
\end{aligned}$$

Monte carlo with several samplers

We can use several samplers in a monte carlo process: for f a function to integrate over a set Ω, $X_1, \cdots, X_N$ N independent *c.r.v* of density probability $q_1, \cdots q_N$, we have:

$$\int_\Omega f(\omega)\, d\omega = \frac{1}{N}\sum_{i=1}^{N}\int_\Omega \frac{f(\omega)}{q_i(\omega)}\, q_i(\omega)d\omega$$
$$\approx \frac{1}{N}\sum_{i=1}^{N}\frac{f(\omega)}{q_i(\omega)} \quad \text{monte carlo estimator with one sample}$$

7.2 MIS proofs

Balance and Power heuristics

To show that the balance and power heuristics are valid, we need to show that:

- $p_i(\omega) = 0 \Rightarrow w_i(\omega) = 0$
- $f(\omega) \neq 0 \Rightarrow \sum_{i=1}^{n} w_i(\omega) = 1$

Because the balance heuristic is a special case of the power heuristic (with $\beta = 1$), we only need to prove that the power heuristic is valid.

$$p_i(\omega) = 0 \Rightarrow w_i(\omega) = \frac{\left(c_i p_i(\omega)\right)^\beta}{\sum_{j=1}^{n}\left(c_j p_j(\omega)\right)^\beta} = 0$$

$$\sum_{i=1}^{n} w_i(\omega) = \sum_{i=1}^{n}\frac{\left(c_i p_i(\omega)\right)^\beta}{\sum_{j=1}^{n}\left(c_j p_j(\omega)\right)^\beta} = \frac{\sum_{i=1}^{n}\left(c_i p_i(\omega)\right)^\beta}{\sum_{j=1}^{n}\left(c_j p_j(\omega)\right)^\beta} = 1$$

MIS estimator

For $p_1, \cdots, p_n$ n probability density over Ω. For $n_1, \cdots, n_n \in \mathbb{N}$ the number of samples of each density and f a function over Ω. For X_i and $(X_{i,j})_{1\leq j\leq n_i}$ *i.i.d* of density probability p_i, the MIS estimator is:

$$\begin{aligned}
\int_\Omega f(x)\,dx &= \int_\Omega \sum_{i=1}^{n} w_i(x) f(x)\,dx \quad \text{because } \sum_{i=1}^{n} w_i(x) = 1 \\
&= \sum_{i=1}^{n} \int_\Omega w_i(x) f(x)\,dx \\
&= \sum_{i=1}^{n} \int_\Omega w_i(x) \frac{f(x)}{p_i(x)}\, p_i(x) dx \\
&= \sum_{i=1}^{n} \mathbb{E}\left(w_i(X_i) \frac{f(X_i)}{p_i(X_i)} \right) \\
&\approx \sum_{i=1}^{n} \frac{1}{n_i} \sum_{j=1}^{n_i} w_i(X_{i,j}) \frac{f(X_{i,j})}{p_i(X_{i,j})}
\end{aligned}$$

MIS estimator in the special case of the balance strategy

Assuming that $w_i(X) = \frac{c_i p_i(X)}{\sum_{j=1}^{n} c_j p_j(X)}$, the MIS estimator become:

$$\begin{aligned}
\sum_{i=1}^{n} \frac{1}{n_i} \sum_{j=1}^{n_i} w_i(X_{i,j}) \frac{f(X_{i,j})}{p_i(X_{i,j})} &= \sum_{i=1}^{n} \frac{1}{n_i} \sum_{j=1}^{n_i} \frac{c_i p_i(X_{i,j})}{\sum_{j=1}^{n} c_j p_j(X_{i,j})} \frac{f(X_{i,j})}{p_i(X_{i,j})} \\
&= \sum_{i=1}^{n} \frac{1}{c_i N} \sum_{j=1}^{n_i} c_i \frac{f(X_{i,j})}{p(C, X_{i,j})} \quad \text{because } n_i = c_i N \\
&= \frac{1}{N} \sum_{i=1}^{n} \sum_{j=1}^{n_i} \frac{f(X_{i,j})}{p(C, X_{i,j})}
\end{aligned}$$

Variance of the balance strategy

This estimator has the variance:

$$
\begin{aligned}
\mathbb{V}\left(\frac{1}{N}\sum_{i=1}^{n}\sum_{j=1}^{n_i}\frac{f(X_{i,j})}{p(C,X_{i,j})}\right) &= \frac{1}{N^2}\sum_{i=1}^{n}\sum_{j=1}^{n_i}\mathbb{V}\left(\frac{f(X_{i,j})}{p(C,X_{i,j})}\right) \quad \text{because the } (X_{i,j}) \text{ are } \textit{i.i.d} \\
&= \frac{1}{N^2}\sum_{i=1}^{n}\sum_{j=1}^{n_i}\mathbb{E}\left(\frac{f^2(X_{i,j})}{p^2(C,X_{i,j})}\right) - \mathbb{E}\left(\frac{f(X_{i,j})}{p(C,X_{i,j})}\right)^2 \\
&= \frac{1}{N^2}\sum_{i=1}^{n} n_i\left[\mathbb{E}\left(\frac{f^2(X_i)}{p^{(}C,X_i)}\right) - \mathbb{E}\left(\frac{f(X_i)}{p(C,X_i)}\right)^2\right] \quad \text{because } (X_{i,j})_{1\le j\le n_i} \text{ are } \textit{i.i.d} \\
&= \frac{1}{N^2}\sum_{i=1}^{n} c_i N\left[\int_\Omega \frac{f^2(x)}{p^2(C,x)}\ p_i(x)dx - \left(\int_\Omega \frac{f(x)}{p(C,x)}\ p_i(x)dx\right)^2\right] \\
&= \frac{1}{N}\sum_{i=1}^{n}\left[\int_\Omega \frac{f^2(x)}{p^2(C,x)}\ c_i p_i(x)dx - \frac{1}{c_i}\left(\int_\Omega \frac{f(x)}{p(C,x)}\ c_i p_i(x)dx\right)^2\right] \\
&= \frac{1}{N}\sum_{i=1}^{n}\int_\Omega \frac{f^2(x)}{p^2(C,x)}\ c_i p_i(x)dx - \frac{\mu_i^2}{c_i}
\end{aligned}
$$

With $\mu_i = \int_\Omega \frac{f(x)}{p(C,x)}\ c_i p_i(x)dx = \int_\Omega w_i(x) f(x)\ dx$

Large brdf proofs

Variance estimation

For a sampler p and the associated estimator $E_p(X) = \frac{f(X)}{p(X)}$ (with X a *c.r.v* of probability density p), for f the integrand of the light equation3.2 and $\mu = \int_\Omega f(\omega)\ d\omega$, the variance of E_p is:

$$
\begin{aligned}
\mathbb{V}(E_p) &= \int_\Omega \left(E_p(\omega) - \mathbb{E}(E_p)\right)^2\ p(\omega)d\omega \\
&= \int_\Omega \left(\frac{f(\omega)}{p(\omega)} - \int_\Omega \frac{f(\omega')}{p(\omega')}\ p(\omega')d\omega'\right)^2\ p(\omega)d\omega \\
&= \int_\Omega \left(\frac{f(\omega)}{p(\omega)} - \int_\Omega f(\omega')\ d\omega'\right)^2\ p(\omega)d\omega \\
&= \int_\Omega \left(\frac{f(\omega)}{p(\omega)} - \mu\right)^2\ p(\omega)d\omega
\end{aligned}
$$

Large brdf analytical failure

In the case where the sampler is $p_k(\omega) = k_d \frac{\cos(\theta)}{\pi} + (1-k_d)\frac{\frac{n_s}{k}+1}{2\pi}\frac{\cos(\theta_h)^{\frac{n_s}{k}}}{4h\cdot\omega}$, he have:

$$\begin{aligned}
\mathbb{V}(E_{p_k}) &= \int_\Omega \left(\frac{f(\omega)}{p_k(\omega)} - \mu\right)^2 \frac{p_k(\omega)}{p_{k'}(\omega)}\, p_{k'}(\omega) d\omega \\
&= \int_\Omega \frac{f^2(\omega)}{p_k^2(\omega)} \frac{p_k(\omega)}{p_{k'}(\omega)}\, p_{k'}(\omega) d\omega - \int_\Omega 2\frac{f(\omega)}{p_k(\omega)}\mu \frac{p_k(\omega)}{p_{k'}(\omega)}\, p_{k'}(\omega) d\omega + \int_\Omega \mu^2 \frac{p_k(\omega)}{p_{k'}(\omega)}\, p_{k'}(\omega) d\omega \\
&= \int_\Omega \frac{f^2(\omega)}{p_k(\omega)}\, d\omega - \int_\Omega 2f(\omega)\mu\, d\omega + \int_\Omega \mu^2 p_k(\omega)\, d\omega
\end{aligned}$$

With:

$$\int_\Omega \frac{f^2(\omega)}{p_k(\omega)}\, d\omega = \int_\Omega \frac{f^2(\omega)}{k_d \frac{\cos(\theta)}{\pi} + (1-k_d)\frac{\frac{n_s}{k}+1}{2\pi}\frac{\cos(\theta_h)^{\frac{n_s}{k}}}{4h\cdot\omega}}\, d\omega$$

And by assuming $k_d = 0$ to only consider the specular part of the brdf, we get:

$$\begin{aligned}
\int_\Omega \frac{f^2(\omega)}{p_k(\omega)}\, d\omega &= \int_\Omega \frac{f^2(\omega)}{\frac{\frac{n_s}{k}+1}{2\pi}\frac{\cos(\theta_h)^{\frac{n_s}{k}}}{4h\cdot\omega}}\, d\omega \\
&= \frac{2\pi}{\frac{n_s}{k}+1}\int_\Omega \frac{4h\cdot\omega f^2(\omega)}{\cos(\theta_h)^{\frac{n_s}{k}}}\, d\omega
\end{aligned}$$

Large brdf stable variance estimate

Finally, after thinking twice about the equation, one can see that the variance can also be written as:

$$
\begin{aligned}
\mathbb{V}(E_p) &= \int_\Omega \left(\frac{f(\omega)}{p(\omega)} - \mu\right)^2 p(\omega) d\omega \\
&= \int_\Omega \left(\frac{f(\omega)}{p(\omega)} - \mu\right)^2 \frac{p(\omega)}{q(\omega)} q(\omega) d\omega \\
&= \int_\Omega \frac{f^2(\omega)}{p^2(\omega)} \frac{p(\omega)}{q(\omega)} q(\omega) d\omega - \int_\Omega 2\frac{f(\omega)}{p(\omega)}\mu \frac{p(\omega)}{q(\omega)} q(\omega) d\omega + \int_\Omega \mu^2 \frac{p(\omega)}{q(\omega)} q(\omega) d\omega \\
&= \int_\Omega \frac{f^2(\omega)}{p(\omega)q(\omega)} q(\omega) d\omega - 2\mu \int_\Omega f(\omega)\, d\omega + \mu^2 \int_\Omega p(\omega) d\omega \\
&= \int_\Omega \frac{f^2(\omega)}{p(\omega)q(\omega)} q(\omega) d\omega - 2\mu^2 + \mu^2 \\
&= \int_\Omega \frac{f^2(\omega)}{p(\omega)q(\omega)} q(\omega) d\omega - \mu^2
\end{aligned}
$$

7.3 Left-Right sampler proofs

Left-right sampler

Let's consider the general cases with $p_{\gamma_i}(\omega) = \sum_{i=1}^n n\gamma_i \mathbb{1}_{\omega\in\Omega_i} p(\omega)$ (the left-right sampler is just the special case with $n = 2$). Then assuming $\forall i \in [\![1, n]\!], \int_{\Omega_i} p(\omega) d\omega = \frac{1}{n}$:

$$
\begin{aligned}
\int_\Omega p_{\gamma_i}(\omega)\, d\omega &= \sum_{i=1}^n \int_{\Omega_i} p_{\gamma_i}(\omega)\, d\omega \quad \text{because } (\Omega_i) \text{ is a partition of } \Omega \\
&= \sum_{i=1}^n \int_{\Omega_i} n\gamma_i p(\omega)\, d\omega \quad \text{on } \Omega_i,\ p_{\gamma_i} = n\gamma_i p \\
&= \sum_{i=1}^n n\gamma_i \frac{1}{n} \quad \text{because } \int_{\Omega_i} p(\omega) d\omega = \frac{1}{n} \\
&= \sum_{i=1}^n \gamma_i = 1
\end{aligned}
$$

Furthemore, if the condition $\int_{\Omega_i} p(\omega)\, d\omega = \frac{1}{n}$ does not hold, there is no optimisation possible. In that case, there exists k such that $\int_{\Omega_k} p(\omega)\, d\omega \neq \frac{1}{n}$.

$$
\int_\Omega p_{\gamma_i}(\omega)\, d\omega = \sum_{i=1}^n n\gamma_i \int_{\Omega_i} p(\omega)\, d\omega
$$

By making $\gamma_k \to 1$ (and $\forall i \neq k, \gamma_i \to 0$ to keep the constraint $\sum_{i=1}^n \gamma_i = 1$) we get:

$$\sum_{i=1}^{n} n\gamma_i \int_{\Omega_i} p(\omega)\ d\omega \underset{\gamma_k \to 1}{\longrightarrow} n \int_{\Omega_k} p(\omega)\ d\omega \neq 1$$

Which is not possible because $\int_\Omega p_{\gamma_i}(\omega)\ d\omega$ must be equal to 1.

Variance left-right sampler

Let's consider the general cases with $p_{\gamma_i}(\omega) = \sum_{i=1}^{n} n\gamma_i \mathbb{1}_{\omega\in\Omega_i} p(\omega)$ (the left-right sampler is just the special case with $n = 2$). In that case, the variance can be written as:

$$\begin{aligned}
\mathbb{V}_{\gamma_i} &= \int_\Omega \left(\frac{f(\omega)}{p_{\gamma_i}(\omega)} - \mu\right)^2\ p_{\gamma_i}(\omega) d\omega \\
&= \int_\Omega \frac{f^2(\omega)}{p_{\gamma_i}^2(\omega)} - 2\frac{f(\omega)}{p_{\gamma_i}}\mu + \mu^2\ p_{\gamma_i}(\omega) d\omega \\
&= \int_\Omega \frac{f^2(\omega)}{p_{\gamma_i}^2(\omega)}\ p_{\gamma_i}(\omega) d\omega - 2\mu \int_\Omega \frac{f(\omega)}{p_{\gamma_i}}\ p_{\gamma_i}(\omega) d\omega + \mu^2 \\
&= \int_\Omega \frac{f^2(\omega)}{p_{\gamma_i}(\omega)}\ d\omega - 2\mu^2 + \mu^2 \quad \text{because } \mu = \int_\Omega f(\omega) d\omega \\
&= \sum_{i=1}^{n} \int_{\Omega_i} \frac{f^2(\omega)}{n\gamma_i p(\omega)}\ d\omega - \mu^2 \quad \text{on } \Omega_i,\ p_{\gamma_i} = n\gamma_i p
\end{aligned}$$

Left-right sampler best γ_i

Let's consider the general cases with $p_{\gamma_i}(\omega) = \sum_{i=1}^{n} n\gamma_i \mathbb{1}_{\omega\in\Omega_i} p(\omega)$ (the left-right sampler is just the special case with $n = 2$). Let's use the Lagrange multiplier on $\mathbb{V}_{\gamma_i}$ with the constraint that $\sum_{i=1}^{n} \gamma_i = 1$:

$$\mathcal{L}_{\mathbb{V}_{\gamma_i}}(\gamma_1, \cdots, \gamma_n, \lambda) = \sum_{i=1}^{n} \frac{V_i}{n\gamma_i} - \mu^2 + \lambda\left(1 - \sum_{i=1}^{n} \gamma_i\right)$$

Let $(\gamma_1^*, \cdots, \gamma_n^*, \lambda^*)$ be a stationary point of the lagrangian function. Then:

1. $0 = \frac{\partial \mathcal{L}_{\mathbb{V}_{\gamma_i}}}{\partial \gamma_i} = \frac{-V_i}{n{\gamma_i^*}^2} - \lambda^*$
2. $0 = \frac{\partial \mathcal{L}_{\mathbb{V}_{\gamma_i}}}{\partial \lambda} = 1 - \sum_{i=1}^{n} \gamma_i^*$

From 1 We have : $0 = \frac{-V_i}{n{\gamma_i^*}^2} - \lambda^* \Rightarrow \gamma_i^* = \sqrt{\frac{-V_i}{n\lambda^*}}$

From 2 We have : $0 = 1 - \sum_{i=1}^{n} \gamma_i^* = 1 - \sum_{i=1}^{n} \sqrt{\frac{-V_i}{n\lambda^*}} \Rightarrow \lambda^* = \frac{-1}{n}\left(\sum_{i=1}^{n} \sqrt{V_i}\right)^2$

Finally, we have : $\gamma_i^* = \sqrt{\frac{-V_i}{n\lambda^*}} = \sqrt{\frac{-V_i}{n\frac{-1}{n}\left(\sum_{j=1}^n \sqrt{V_j}\right)^2}} = \frac{\sqrt{V_i}}{\sum_{j=1}^n \sqrt{V_j}}$

We have found a stationary point. Let's check that this is a minima:

1. $\frac{\partial^2 \mathscr{L}_{\mathbb{V}_{\gamma_i}}}{\partial \gamma_i^2} = \frac{2V_i}{n{\gamma_i^*}^3} \geq 0$
2. $\frac{\partial^2 \mathscr{L}_{\mathbb{V}_{\gamma_i}}}{\partial \lambda^2} = 0$

Finally:

- if $\exists \omega \in \Omega$ such that $f(\omega) \neq 0$ then $\exists i, w \in \Omega_i$ and $V_i > 0 \Rightarrow \frac{\partial^2 \mathscr{L}_{\mathbb{V}_{\gamma_i}}}{\partial \gamma_i^2} > 0$ and we have a minima.
- if $\nexists \omega \in \Omega$ such that $f(\omega) \neq 0$ then the solution of the light equation is 0 and any sampler will give the exact value with only one sample.

Left right generalized sampler failed implementation

In the case where $p(\omega) = k_d \frac{\cos(\theta)}{\pi} + (1-k_d)\frac{n_s+1}{2\pi}\frac{\cos(\theta_h)^{n_s}}{4h\cdot\omega}$, we can try to find a partition of Ω fulfilling the constraint $\int_{\Omega_{i,j}} p(\omega)\ d\omega = \frac{1}{nm}$:

$$
\begin{aligned}
\frac{1}{nm} &= \int_{\Omega_{i,j}} p(\omega)\ d\omega \\
&= \int_{\Omega_{i,j}} k_d \frac{\cos(\theta)}{\pi} + (1-k_d)\frac{n_s+1}{2\pi}\frac{\cos(\theta_h)^{n_s}}{4h\cdot\omega}\ d\omega \\
&= \int_{\theta_i}^{\theta_{i+1}}\int_{\phi_j}^{\phi_{j+1}} \left(k_d \frac{\cos(\theta)}{\pi} + (1-k_d)\frac{n_s+1}{2\pi}\frac{\cos(\theta_h)^{n_s}}{4h\cdot\omega}\right) \sin(\theta) d\phi d\theta \\
&= \int_{\theta_i}^{\theta_{i+1}} \frac{k_d}{\pi}\int_{\phi_j}^{\phi_{j+1}} \cos(\theta)\sin(\theta)\ d\phi_h + (1-k_d)\frac{n_s+1}{8\pi}\int_{\phi_j}^{\phi_{j+1}} \frac{\cos(\theta_h)^{n_s}}{h\cdot\omega} \sin(\theta) d\phi d\theta \\
&= \int_{\theta_i}^{\theta_{i+1}} \frac{k_d}{\pi}(\phi_{j+1}-\phi_j)\cos(\theta)\sin(\theta) + (1-k_d)\frac{n_s+1}{8\pi}\int_{\phi_j}^{\phi_{j+1}} \frac{\cos(\theta_h)^{n_s}}{h\cdot\omega} \sin(\theta) d\phi d\theta \\
&= \int_{\theta_i}^{\theta_{i+1}} \frac{k_d}{\pi}(\phi_{j+1}-\phi_j)\frac{\sin(2\theta)}{2} + (1-k_d)\frac{n_s+1}{8\pi}\int_{\phi_j}^{\phi_{j+1}} \frac{\cos(\theta_h)^{n_s}}{h\cdot\omega} \sin(\theta) d\phi d\theta
\end{aligned}
$$

Left right generalized sampler implementation

Assuming that $p(h) = \sum_{i=0}^{n-1}\sum_{j=0}^{m-1} \mathbb{1}_{h\in\Omega_{i,j}}\left((1-k_d)\frac{n_s+1}{2\pi}\cos(\theta_h)^{n_s} + k_d\frac{1}{n2\pi(\cos(\theta_i)-\cos(\theta_{i+1}))}\right)$, with $\cos(\theta_i) = \sqrt[n_s+1]{\frac{n-i}{n}}$ and $\phi_j = \frac{2\pi j}{m}$, we can show that $\int_{\Omega_{i,j}} p(\omega)\ d\omega = \frac{1}{nm}$:

$$\begin{aligned}
\int_{\Omega_{i,j}} p(h)\, dh &= \int_{\Omega_{i,j}} \sum_{i=0}^{n-1}\sum_{j=0}^{m-1} \mathbb{1}_{h\in\Omega_{i,j}} \left((1-k_d)\frac{n_s+1}{2\pi}\cos(\theta_h)^{n_s} + k_d\frac{1}{n2\pi(\cos(\theta_i)-\cos(\theta_{i+1}))}\right) dh \\
&= \int_{\Omega_{i,j}} (1-k_d)\frac{n_s+1}{2\pi}\cos(\theta_h)^{n_s} + k_d\frac{1}{n2\pi(\cos(\theta_i)-\cos(\theta_{i+1}))}\, dh \\
&= (1-k_d)\int_{\Omega_{i,j}} \frac{n_s+1}{2\pi}\cos(\theta_h)^{n_s}\, dh + k_d\int_{\Omega_{i,j}} \frac{1}{n2\pi(\cos(\theta_i)-\cos(\theta_{i+1}))}\, dh
\end{aligned}$$

Assuming that $\int_{\Omega_{i,j}} \frac{n_s+1}{2\pi}\cos(\theta_h)^{n_s}\, dh = \frac{1}{nm}$ and $\int_{\Omega_{i,j}} \frac{1}{n2\pi(\cos(\theta_i)-\cos(\theta_{i+1}))}\, dh = \frac{1}{nm}$, we have:

$$\begin{aligned}
\int_{\Omega_{i,j}} p(h)\, dh &= (1-k_d)\int_{\Omega_{i,j}} \frac{n_s+1}{2\pi}\cos(\theta_h)^{n_s}\, dh + k_d\int_{\Omega_{i,j}} \frac{1}{n2\pi(\cos(\theta_i)-\cos(\theta_{i+1}))}\, dh \\
&= (1-k_d)\frac{1}{nm} + k_d\frac{1}{nm} \\
&= \frac{1}{nm}
\end{aligned}$$

The assumption is correct:

$$\begin{aligned}
\int_{\Omega_{i,j}} \frac{1}{n2\pi(\cos(\theta_i)-\cos(\theta_{i+1}))}\, dh &= \int_{\theta_i}^{\theta_{i+1}}\int_{\phi_j}^{\phi_{j+1}} \frac{1}{n2\pi(\cos(\theta_i)-\cos(\theta_{i+1}))}\sin(\theta)d\phi d\theta \\
&= \int_{\theta_i}^{\theta_{i+1}} (\phi_{j+1}-\phi_j)\frac{1}{n2\pi(\cos(\theta_i)-\cos(\theta_{i+1}))}\sin(\theta)d\theta \\
&= (\phi_{j+1}-\phi_j)\frac{\int_{\theta_i}^{\theta_{i+1}}\sin(\theta)d\theta}{n2\pi(\cos(\theta_i)-\cos(\theta_{i+1}))} \\
&= (\phi_{j+1}-\phi_j)\frac{[-\cos(\theta)]_{\theta_i}^{\theta_{i+1}}}{n2\pi(\cos(\theta_i)-\cos(\theta_{i+1}))} \\
&= (\phi_{j+1}-\phi_j)\frac{\cos(\theta_i)-\cos(\theta_{i+1})}{n2\pi(\cos(\theta_i)-\cos(\theta_{i+1}))} \\
&= \left(\frac{2\pi(j+1)}{m}-\frac{2\pi j}{m}\right)\frac{1}{n2\pi} \quad \text{because } \phi_i = \frac{2\pi i}{m} \\
&= \frac{1}{nm}
\end{aligned}$$

And:

$$
\begin{aligned}
\int_{\Omega_{i,j}} \frac{n_s+1}{2\pi}\cos(\theta_h)^{n_s}\, dh &= \int_{\theta_i}^{\theta_{i+1}}\int_{\phi_j}^{\phi_{j+1}} \frac{n_s+1}{2\pi}\cos(\theta_h)^{n_s}\ \sin(\theta_h)d\phi_h d\theta_h \\
&= \frac{n_s+1}{2\pi}\int_{\theta_i}^{\theta_{i+1}} (\phi_{j+1}-\phi_j)\cos(\theta_h)^{n_s}\ \sin(\theta_h)d\theta_h \\
&= \frac{n_s+1}{2\pi}(\phi_{j+1}-\phi_j)\left[\frac{-\cos(\theta_h)^{n_s+1}}{n_s+1}\right]_{\theta_i}^{\theta_{i+1}} \\
&= \frac{1}{2\pi}(\phi_{j+1}-\phi_j)\left(\cos(\theta_i)^{n_s+1}-\cos(\theta_{i+1})^{n_s+1}\right)
\end{aligned}
$$

Because we have $\phi_i = \frac{2\pi i}{m}$ and $\cos(\theta_i) = \sqrt[n_s+1]{\frac{n-i}{n}}$:

$$
\begin{aligned}
\frac{1}{2\pi}(\phi_{j+1}-\phi_j)\left(\cos(\theta_i)^{n_s+1}-\cos(\theta_{i+1})^{n_s+1}\right) &= \frac{1}{2\pi}\left(\frac{2\pi(j+1)}{m}-\frac{2\pi j}{m}\right)\left(\sqrt[n_s+1]{\frac{n-i}{n}}^{\,n_s+1} - \sqrt[n_s+1]{\frac{n-i-1}{n}}^{\,n_s+1}\right) \\
&= \frac{1}{2\pi}\frac{2\pi}{m}\left(\frac{n-i}{n}-\frac{n-i-1}{n}\right) \\
&= \frac{1}{nm}
\end{aligned}
$$

Left right generalized is a valid sampler

p is a valid density probability:

- $\forall h \in \Omega, p(h) = \sum_{i=0}^{n-1}\sum_{j=0}^{m-1} \mathbb{1}_{h\in\Omega_{i,j}}\left((1-k_d)\frac{n_s+1}{2\pi}\cos(\theta_h)^{n_s} + k_d\frac{1}{n2\pi(\cos(\theta_i)-\cos(\theta_{i+1}))}\right) \geq 0$
- $\int_\Omega p(h)\, dh = \sum_{i=0}^{n-1}\sum_{j=0}^{m-1}\int_{\Omega_{i,j}} p(h)\, dh = \sum_{i=0}^{n-1}\sum_{j=0}^{m-1}\frac{1}{nm} = 1$

References

[1] James F Blinn. Models of light reflection for computer synthesized pictures. In *Proceedings of the 4th annual conference on Computer graphics and interactive techniques*, pages 192–198, 1977.

[2] Holger Dammertz, Johannes Hanika, Alexander Keller, and Hendrik Lensch. A hierarchical automatic stopping condition for monte carlo global illumination. 2010.

[3] Pascal Grittmann, Iliyan Georgiev, Philipp Slusallek, and Jaroslav Křivánek. Variance-aware multiple importance sampling. *ACM Transactions on Graphics (TOG)*, 38(6):1–9, 2019.

[4] Pascal Grittmann, Ömercan Yazici, Iliyan Georgiev, and Philipp Slusallek. Efficiency-aware multiple importance sampling for bidirectional rendering algorithms. *ACM Transactions on Graphics (TOG)*, 41(4):1–12, 2022.

[5] James T Kajiya. The rendering equation. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 143–150, 1986.

[6] Ivo Kondapaneni, Petr Vévoda, Pascal Grittmann, Tomáš Skřivan, Philipp Slusallek, and Jaroslav Křivánek. Optimal multiple importance sampling. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019.

[7] Thomas Müller, Markus Gross, and Jan Novák. Practical path guiding for efficient light-transport simulation. In *Computer Graphics Forum*, volume 36, pages 91–100. Wiley Online Library, 2017.

[8] Yaobin Ouyang, Shiqiu Liu, Markus Kettunen, Matt Pharr, and Jacopo Pantaleoni. Restir gi: Path resampling for real-time path tracing. In *Computer Graphics Forum*, volume 40, pages 17–29. Wiley Online Library, 2021.

[9] Art B. Owen. *Monte Carlo theory, methods and examples.* 2013.

[10] Mateu Sbert and Víctor Elvira. Generalizing the balance heuristic estimator in multiple importance sampling. *Entropy*, 24(2):191, 2022.

[11] Mateu Sbert, Vlastimil Havran, and László Szirmay-Kalos. Optimal deterministic mixture sampling. *Eurographics (Short Papers)*, 73:76, 2019.

[12] Mateu Sbert and László Szirmay-Kalos. A linear heuristic for multiple importance sampling. *EURASIP Journal on Advances in Signal Processing*, 2023(1):31, 2023.

[13] László Szirmay-Kalos and Mateu Sbert. Robust sample budget allocation for mis. In *Proceedings of the Eurographics*, 2022.

[14] Eric Veach and Leonidas J Guibas. Optimally combining sampling techniques for monte carlo rendering. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 419–428, 1995.