

# MIS Compensation: Optimizing Sampling Techniques in Multiple Importance Sampling

Seminar-Ausarbeitung von

Christian Navolskyi, B. Sc.

An der Fakultät für Informatik  
Institut für Visualisierung und Datenanalyse,  
Lehrstuhl für Computergrafik

November 19, 2019

# Contents

<b>1 Abstract</b>	<b>1</b>
<b>2 Introduction</b>	<b>2</b>
<b>3 Related Work</b>	<b>3</b>
<b>4 Multiple Importance Sampling</b>	<b>5</b>
4.1 Probability Basics . . . . .	5
4.2 Generating Samples after a specific Function . . . . .	6
4.3 Monte Carlo Integration . . . . .	6
4.3.1 Variance . . . . .	7
4.4 Importance Sampling . . . . .	7
4.5 Multiple Importance Sampling . . . . .	7
4.5.1 Balance Heuristic . . . . .	9
<b>5 Multiple Importance Sampling Compensation</b>	<b>11</b>
5.1 Optimality . . . . .	12
5.1.1 Bounds for the MIS-compensated solution . . . . .	12
<b>6 Application: Image Based Lighting</b>	<b>14</b>
6.1 Image Based Lighting . . . . .	14
6.2 MIS Compensation in Image Based Lighting . . . . .	14
6.2.1 Setup . . . . .	14
6.2.2 PDF solutions . . . . .	15
6.2.2.1 Normal-dependend solution . . . . .	15
6.2.2.2 Normal-independend solution . . . . .	15
6.2.3 Evaluation . . . . .	15
<b>7 Application: Path Guiding</b>	<b>16</b>
7.1 Path Guiding . . . . .	16
7.2 MIS Compensation in Path Guiding . . . . .	16
<b>8 Summary</b>	<b>17</b>
8.1 Future Work . . . . .	17
<b>Bibliography</b>	<b>18</b>

# 1. Abstract

Here is some sample text.

- (Write abstract (at the end)) ToDo
- (Write introduction. Structure as in the paper (overview, problem, others, solution proposal)) ToDo
- (Citation before the dot. In more than one sentence, name in the first sentence and reference in the last sentence.) ToDo
- (are all the citations correct?) ToDo
- (Does the date of access need to be in the references?) ToDo
- (Fig. 6 in the paper, how can multiple samples be drawn from the one-sample estimator?) ToDo

## 2. Introduction

Since Veach and Guibas introduced Multiple Importance Sampling (MIS) [VG95] it had a hugh impact in a variety of directions in computer graphics. Not only did it help to construct more advanced rendering algorithms like Vertex Connection and Merging [GKDS12], it also helped in calculating direct illumination from area light sources and environment maps [PJH18, Chapter 14.3] and subsurface scattering [KKCF13]. Besides being introduced in computer graphics MIS has also been used in other fields [HO14].

For the proof of MIS being almost optimal the assumption was made, that the distribution of samples among the techniques and the sampling densities are given and fixed. The only things left for fine tuning was the weighting function with the balance heuristic being a popular and widely adopted choice because of its simplicity and proven tight bounds when only using non-negative weights [Vea97, Theorem 9.2]. Recently researchers found that using not only non-negative weights can result in a truly optimal weighting function [IVG<sup>+</sup>19].

No previous work examined the effect of designing a sampling density itself for application in the MIS framework. Karlík et al. proposed a method to do exactly that and show that it can improve the MIS framework and reduce variance even further. They assumed multiple sampling techniques and pick one probability density function (pdf) of a technique and modify it in a way that it compensates for the averaging introduced by the balance heuristic. Their optimization can be applied beforehand and doesn't need adaptive updates as the work of Cappé et al. [CDG<sup>+</sup>08]. Others created product sampling methods [HEV<sup>+</sup>16] that try to match the integrand closer, but MIS compensation as, Karlík et al. call their method, on the other hand can even cause a pdf to be further away from the integrand but still reduce the variance [KŠV<sup>+</sup>19].

Karlík et al. show the effects of their MIS compensation with direct illumination using a HDR environment map and a scene with surfaces that have multiple different bidirectional reflectance distribution functions (BRDF). The standard approach with MIS would take samples proportional to the BRDF and the environment map, but they assume this approach to be too defensive and modify the pdf for the environment map in a way that the variance is reduced and therefore render times can be improved. Their modification is done by applying a formula to the tabulated pdf in a preprocessing step after which any renderer able to use HDR maps for MIS sampling can use the updated environment map for sampling without any further adjustment.

They also used their approach in path guiding [VHH<sup>+</sup>19] which learns a guiding density to improve the algorithm of [MGN17] by adjusting that density [KŠV<sup>+</sup>19].

### 3. Related Work

The MIS framework was introduced in [VG95] and combines multiple different sampling techniques with a weighting function to sample a more complex function better. In their work they proposed a few different weighting functions the most famous being the balance heuristic, but also the power and cutoff heuristic which perform better in certain scenarios where one technique matches the integrand well. Despite the balance heuristic being provably good Kondapaneni et al. [IVG<sup>+</sup>19] developed a weighting function that is truly optimal and uses negative weights to archive even better results.

A lot of work has been put into optimizing the weighting functions, but Karlík et al. followed a different approach and worked on a technique to adjust one sampling density to reduce the variance.

Another direction of research in the field of MIS is to figure out how to distribute the samples over the different sampling techniques. Pajot et al. [PBPP11] proposed a method where they calculate a so called representativity based on common rendering information such as the BRDF, which is a measure of how well a technique matches the integrand. In a one-sample estimator this representativity can also be used to derive a probability for each sampling strategy.

Lu et al. [LPG13] created a method where no prior knowledge about the scene is needed to generate a sample distribution over two sampling strategies. They start by sampling with a small number of samples evenly distributed over the strategies and calculate a value to partition the total number of samples over the sampling techniques.

He and Owen [HO14] presented another method to get a partitioning of samples for more than two strategies by proving that the variance is jointly convex in the distribution of samples. They also proposed an improved method for convex optimization to find such a distribution.

Sbert et al. [SHSK16] worked on a modification of the balance heuristic that doesn't take the distribution of samples into account for calculating the weights. Their method works in two phases. The first phase uses 20% of the total samples equally distributed over all sampling methods to calculate the variance  $\sigma_i^2$ . After that the following stage is subdivided into eight substages each using 10% of the total number of samples and iteratively updating  $\sigma_i^2$  and calculating the distribution  $\alpha$  over the techniques for the next substage.

What all the above techniques have in common is that they all rely on initial samples to fine tune their succeeding execution. The proposed method of Karlík et al. on the other hand doesn't need any prior samples to be taken and therefore their performance impact should be much smaller [KŠV<sup>+</sup>19].

One widely used application of MIS is image based lighting with BSDF and HDR map sampling [PJH18]. When using a HDR environment map for direct illumination the pdf for that map is usually proportional to the brightness of the HDR map, Karlík et al. [KŠV<sup>+</sup>19] created a method that alters that sampling density to further decrease the variance. Agarwal et al. came up with the idea of stratifying the environment map and taking into account the visibility in the scene to create a method that can require up to two orders of magnitude less samples for the same quality [ARBJ03].

Another approach from Clarberg and Akenine-Möller uses product importance sampling and creating the BSDF on the fly during rendering. Their precomputation step uses quadtree-based multiplication while during rendering they build the approximation of the BSDF and evaluate the product in only one single tree traversal which makes it perform faster than other methods [CAM08].

The method of Karlík et al. does not change the sampling procedure itself and therefore can work without modifications of the environment map sampler [KŠV<sup>+</sup>19].

Herholz et al. worked on a method to create a pdf the resembles the whole product of the integral (BSDF and reflected light) we want to solve by training a Gaussian mixture model (GMM) for the individual factors with initial samples, which they then combine to draw samples during the integration [HEV<sup>+</sup>16].

Müller et al. also worked on path guiding, where they used an algorithm that iteratively learns the radiance distribution in the scene to get better samples from their distribution [MGN17].

The work of Karlík et al. builds on the algorithm proposed by Müller et al. to further improve the guiding density [KŠV<sup>+</sup>19].

## 4. Multiple Importance Sampling

In this chapter we will introduce all the theoretical background necessary to follow the upcoming chapters. We will start with the basic probability essentials followed by a section about drawing samples after an arbitrary function. Then we introduce the concept of Monte Carlo integration and have a short look at variance in that section too. After that importance sampling is explained and lastly multiple importance sampling will be introduced with the balance heuristic being the final section.

### 4.1 Probability Basics

For our use case we need to define random variables, probability distribution functions (pdf) and cumulative distribution functions (cdf). A random variable maps an event to a real number  $X : \Omega \rightarrow \mathbb{R}$ . In a discrete scenario our random variable corresponds to one exact event e.g. a dice throw showing three pips on the top. The probability for this can be expressed as  $P(X = 3) = \frac{1}{6}$  or as  $X(3) = \frac{1}{6}$  since the random variable  $X$  can take six different values which are all equally likely. To express the probability that an event is within a range of values we can use the cdf which is defined as

$$F(x) = P(X \leq x), x \in \mathbb{R}.$$

For the probability of a random variable in an interval  $[a, b)$  we can write

$$P(a \leq X < b) = F(b) - F(a) \text{ [Sch18].}$$

When we are talking about continuous events e.g. turning a wheel of fortune, every exact angle has a probability of nearing zero. But we can still express our probability for an angle interval with the cdf from above. In the continuous case we have a pdf that is used to calculate the cdf with

$$F(x) = \int_{-\infty}^x p(\tilde{x})d\tilde{x}.$$

The pdf has to be non-negative ( $\forall x : p(x) \geq 0$ ) and integrate to one ( $\int_{-\infty}^{\infty} p(x)dx = 1$ ) to be valid and “[...] describes the relative probability of a random variable taking on a particular value” [PJH18, Chapter 13.1].

## 4.2 Generating Samples after a specific Function

Knowing how to sample after a specific distribution is essential for the next section 4.3, since this is our adjustment wheel to manipulate the quality of our rendering (for a given amount of time). Most of the time we don't need uniformly distributed samples, rather our samples should follow a specific characteristic of a material or a scene. For this we need to know how to generate samples that correspond to that characteristic.

Given a function  $f(x)$  from which we would like to draw samples from the first step is to make sure it fulfills the constraints for a pdf in an interval  $[a, b]$  we want to use. First we check that  $\forall x \in [a, b] : f(x) \geq 0$  and then calculate the integral with

$$F = \int_a^b f(x)dx. \quad (4.1)$$

**To Do**  $F$  can be used to normalize our goal function which yields us  $\tilde{f}(x) = \frac{f(x)}{F}$  as a valid pdf. The next step is to calculate our cdf as  $F(x) = \int_a^x \tilde{f}(t)dt$  which we will then invert to get  $F^{-1}(x)$ . Now we only need to draw an equally distributed random number (which most programming languages have a library for) in the interval  $[0, 1]$  which we call  $\xi$  and evaluate  $F^{-1}(\xi) = X$  to get  $X$  which is distributed proportional to  $f(.)$ . This method is called "inverse cdf" [Sch18]. (**Should I also cover 2D sampling?**)

**To Do**

(**Should I cover more sampling methods (rejection sampling e.g.)?**)

## 4.3 Monte Carlo Integration

Monte Carlo integration is a technique to approximate the integral of an arbitrary function  $f(x)$  by taking  $N$  random samples  $X_i$  from a pdf  $p(X_i)$ . As we know the expectation of a random variable is calculated by

$$E(X) = \int_{-\infty}^{\infty} xp(x)dx$$

or more generally

$$E(g(x)) = \int_{-\infty}^{\infty} g(x)p(x)dx.$$

For the next step we need the law of large numbers which states that

$$P\left[\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N X_i = E(X)\right] = 1$$

with  $X_i$  drawn from  $p(.)$  which means that the average of our samples will converge to the expected value [Vea97, Chapter 2.4.1]. From here we can formulate

$$\int_a^b g(x)p(x)dx \stackrel{\text{definition}}{=} E(g(x)) \stackrel{\text{law of large numbers}}{\approx} \frac{1}{N} \sum_{i=1}^N g(x_i)$$

and with  $g(x) = \frac{f(x)}{p(x)}$

$$\begin{aligned} \int_a^b \frac{f(x)}{p(x)} p(x)dx &\approx \frac{1}{N} \sum_{i=1}^N \frac{f(x_i)}{p(x_i)} \\ \Leftrightarrow \int_a^b f(x)dx &\approx \frac{1}{N} \sum_{i=1}^N \frac{f(x_i)}{p(x_i)} \end{aligned} \quad (4.2)$$

which is the Monte Carlo estimator [Sch18].

### 4.3.1 Variance

Generally the variance is described by

$$V(X) = \frac{1}{N} \sum_{i=1}^N (x_i - E(X))^2$$

for discrete random variables [Sch18] and

$$V(X) = \int_a^b (x - I)^2 p(x) dx$$

with  $I$  being the mean of the random variable  $X$  for continuous variables [wyz19]. When we look at the variance of our Monte Carlo integral we get

$$\begin{aligned} V(g(x)) &= \int_a^b (g(x) - F)^2 p(x) dx \\ &= \int_a^b \left( \frac{f(x)}{p(x)} - F \right)^2 p(x) dx \end{aligned} \tag{4.3}$$

with  $F$  being the integral from equation 4.1.

## 4.4 Importance Sampling

With the variance from equation 4.3 let's see how we can improve it. We can easily see that reducing the term  $\frac{f(x)}{p(x)} - F$  will finally reduce the variance the most since it will be exponentiated. Since  $F$  and  $f(\cdot)$  are fixed (that is the integral we want to calculate) only  $p(\cdot)$  is left for modification. When we choose

$$p(x) = \frac{f(x)}{\int_a^b f(x) dx}$$

we get

$$\begin{aligned} \frac{f(x)}{p(x)} - F &= \frac{f(x)}{\int_a^b f(x) dx} \int_a^b f(x) dx - F \\ &\stackrel{4.1}{=} \int_a^b f(x) dx - \int_a^b f(x) dx \\ &= 0 \end{aligned} \tag{4.4}$$

which is ideal as we get a variance of 0 and are done. The problem is that we don't know the integral at the start since that is what we want to calculate in the first place. When we look at our chosen pdf we see that  $p(\cdot)$  is proportional to  $f(\cdot)$  by a factor  $\frac{1}{\int_a^b f(x) dx} = \frac{1}{F} := c$ . So we should at least try to chose  $p(\cdot)$  somewhat proportional to  $f(\cdot)$ . The difficulty is that  $f(\cdot)$  can be arbitrary complex so we might only be able to approximate some part of it with one specific pdf.

## 4.5 Multiple Importance Sampling

The rendering equation

$$L_o(x, \omega) = L_e(x, \omega) + \int_{\Omega^+} f_s(x, \omega, \omega_o) L_r(x, \omega_o) |n \cdot \omega_o|^+ d\omega_o \tag{4.5}$$

consists of the emitted light at point  $x$   $L_e(\cdot)$ , the bidirectional scattering distribution function (BSDF)  $f_s(\cdot)$ , the reflected light at point  $x$  in direction  $\omega$  expressed as  $L_r(\cdot)$  and

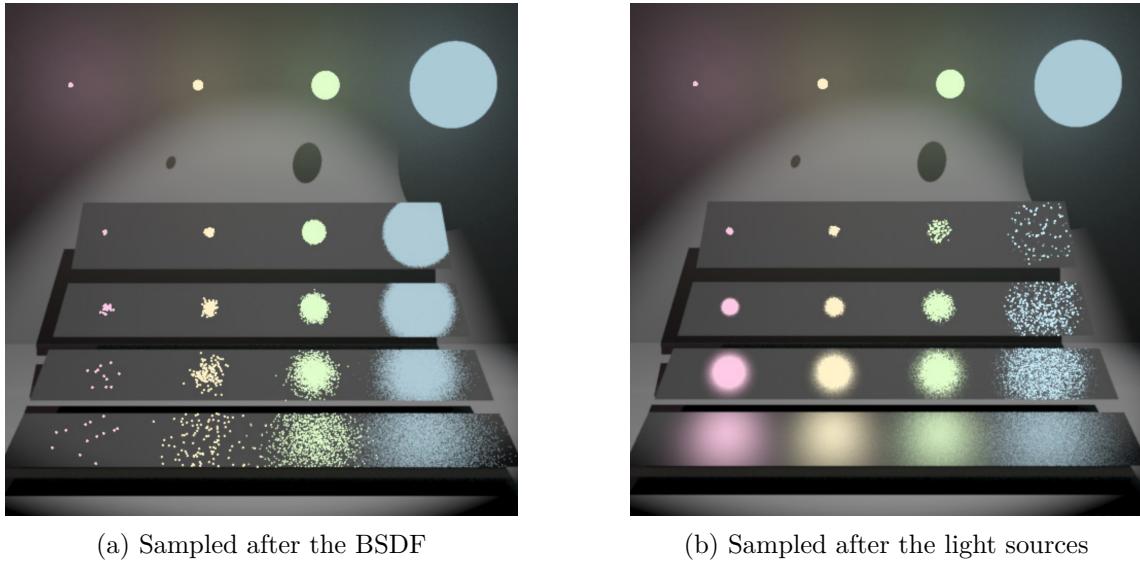


Figure 4.1: A scene with four light sources getting larger in diameter from left to right and four horizontal objects with BSDFs from diffuse at the bottom to more and more specular at the top.

- (a) This scene is sampled using a pdf that is proportional to the BSDF of the surface.
- (b) Here the scene was sampled with a pdf that generated samples on the light source. These graphics were taken from [Vea97, Figure 9.2].

the cosine to account for looking at an angle onto the surface  $|n \cdot \omega_o|^+$  where the  $^+$  means we are only taking values  $> 0$  so only directions pointing into the upper hemisphere. With this we can see that a single pdf that only is proportional to e.g., the BSDF leaves out the  $L_r(\cdot)$  term so it can't be optimal. Having multiple pdfs where each samples a different part of the rendering equation well a combination of them should create a good result. E. Veach created a good example of this problem in his thesis [Vea97] which can be seen in figure 4.1. He also introduced Multiple Importance Sampling to use a combination of pdfs to get better results than would be possible with only one pdf.

When sampling only the BSDF we see that in figure 4.1a the bottom left reflection is very noisy, because sampling a diffuse BSDF can give a wide range of directions and since the light source on the left is very small only very few directions will point directly on that light source. The more specular surfaces in that figure are well sampled which makes sense, because there are less possible directions to sample so the reflection will be well sampled.

On the other hand figure 4.1b shows the same scene but sampled after the light sources by sampling a random point on the light source and then evaluating the render equation to get the contribution. Here the opposite happened in the two corners we looked at before. The bottom left reflection is very well sampled since the direction to the light source is still valid (has some contribution in the BSDF) and therefore we can see the whole reflection. But now the top right reflection is noisy because the direction to a random point on the light source has no contribution in the BSDF.

If we could combine both approaches the result should cover all cases and show all reflections well. To do this Veach introduced Multiple Importance Sampling [Vea97, Chapter 9]. The updated estimator now looks like this:

$$F = \sum_{i=1}^k \frac{1}{n_i} \sum_{j=1}^{n_i} w_i(X_{i,j}) \frac{f(X_{i,j})}{p_i(X_{i,j})}. \quad (4.6)$$

Compared to the initial estimator from equation 4.2 we now have additional weight functions  $w_i(\cdot)$  and split up our samples over the  $k$  different sampling technique with  $n = \sum_{i=1}^k n_i$ . To keep the estimator unbiased the weighting function need to fulfill two constraints:

- $\sum_{i=1}^n w_i(x) = 1$  when  $f(x) \neq 0$
- $w_i(x) = 0$  when  $p_i(x) = 0$ .

The first constraint makes sure that every point is neither reduced nor increased in its contribution which is important to keep the estimator unbiased. The second constraint states that a pdf that could not create this sample also must have no weight so it will be ignored. A concrete weighting function will be introduced in the next subsection.

#### 4.5.1 Balance Heuristic

A very popular weighting function also introduced by Veach is the balance heuristic [Vea97, Chapter 9.2.2]. The formula to calculate it is as follows:

$$w_i(x) = \frac{n_i p_i(x)}{\sum_{j=1}^k n_j p_j(x)}. \quad (4.7)$$

So the weight for a sample consists of the amount of samples  $n_i$  drawn with that technique, the probability  $p_i(x)$  for that sample divided by the sum of all techniques. **(show how it falls down to a simple standard monte carlo estimator when inserting into the formula (when more text is needed?))** ToDo



Figure 4.2: This image shows the same scene as in figure 4.1 but rendered using the balance heuristic. Taken from [Vea97, Figure 9.4].

Figure 4.2 show that with the balance heuristic the noisy areas we observed while only using a single pdf in figure 4.1 disappeared and all combinations of surface roughness and light source size are well represented. When we insert the weighting function from

equation 4.7 into equation 4.6 we get the following:

$$\begin{aligned}
F &= \sum_{i=1}^k \frac{1}{n_i} \sum_{j=1}^{n_i} \frac{n_i p_i(X_{i,j})}{\sum_{m=1}^k n_m p_m(X_{i,j})} \frac{f(X_{i,j})}{p_i(X_{i,j})} \\
&= \sum_{i=1}^k \sum_{j=1}^{n_i} \frac{f(X_{i,j})}{\sum_{m=1}^k n_m p_m(X_{i,j})} \\
&= \frac{1}{N} \sum_{i=1}^N \frac{f(X_i)}{\sum_{m=1}^k c_m p_m(X_i)} \\
&= \frac{1}{N} \sum_{i=1}^N \frac{f(X_i)}{p_{eff}(X_i)}.
\end{aligned}$$

The  $1/N$  term comes from  $n_m = c_m * N$ . When we compare the last line with the initial estimator from equation 4.2 we see that they are identical with  $p(x) = p_{eff}(x) = \sum_{m=1}^k c_m p_m(x)$ . With that we can conclude that using MIS with the balance heuristic corresponds to a normal MIS estimator drawing samples from a combined sampling distribution [Vea97, Chapter 9.2.2.1].

## 5. Multiple Importance Sampling Compensation

This chapter will cover the theory behind MIS compensation as proposed by Karlík et al. in [KŠV<sup>+</sup>19]. First the basic idea of MIS compensation will be explained after which the variance reduction will be proven.

As we can see in figure 5.1 when we use our pdfs regularly with the balance heuristic the resulting pdf is too defensive as the high values are undersampled and the low values are oversampled. The name for their techniques comes from the compensation for the averaging of the balance heuristic. One resulting pdf of their approach can be seen in figure 5.1d.

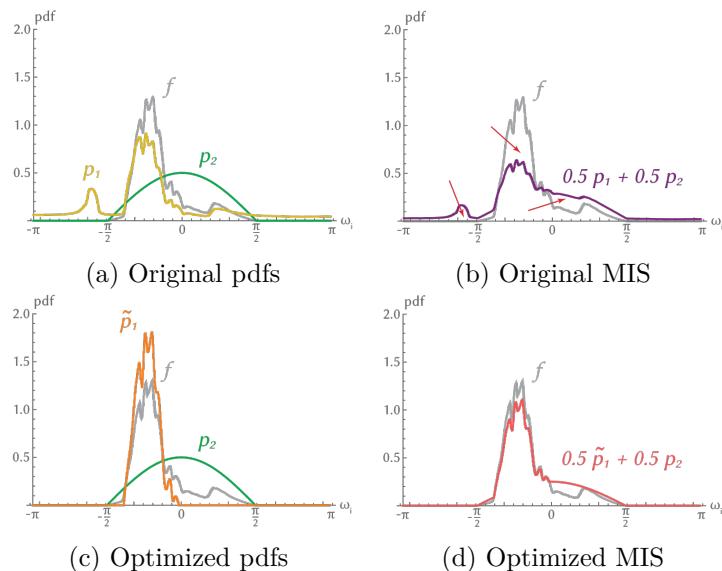


Figure 5.1: Comparison of the original pdfs without compensation (5.1a) and how the combined pdf looks like with the balance heuristic (5.1b) with the modified pdf (5.1c) and the balance heuristic using the optimized pdf (5.1d).  $f$  (gray) is the function we want to integrate. Figure 5.1b shows that the high values are undersampled and the low values are oversampled (see the red arrows). The optimized pdfs match the integrand much better when using MIS as seen in figure 5.1d. Figures taken from [KŠV<sup>+</sup>19, Figure 2].

From a given set of samplers  $T$  they pick one sampler  $t \in T$  and call its pdf  $p_t$  the free pdf that will be used for compensation so that the combined pdf reduces the variance when used with the balance heuristic.

In an optimal solution the combined pdf  $p_{eff}(x) = f(x)/F$  would lead to zero variance as shown in equation 4.4. For the purposes of MIS compensation the combined pdf can also be written as  $p_{eff}(x) = q(x) + c_t p_t(x)$  where  $q(x) = \sum_{i \in T \setminus \{t\}} c_i p_i(x)$  and  $p_t$  is the free pdf with  $c_t$  being its fraction of the total samples. We can reorder to get a formula for

$$p_t(x) = \frac{f(x)}{c_t F} - \frac{q(x)}{c_t}. \quad (5.1)$$

This however does not guarantee that  $p_t$  is a valid pdf, for this they clamped and normalized it and got this formula:

$$\tilde{p}_t(x) = \frac{1}{b} \max\{0, p_t(x)\} \quad (5.2)$$

with a normalization factor of  $b = \int_X \max\{0, p_t(x)\} dx$ .

Their compensated pdf fills the gap between the other samplers and the target function  $f(x)$  since it effectively samples only the parts that the other samplers missed in regard to  $f(x)$ . Whenever  $f(x) > 0$  also  $p_{eff}(x) > 0$  has to hold true for it to be unbiased. When we assume  $q(x) = 0$  then  $p_t(x) > 0$ , because of its definition 5.1. If  $q(x) > 0$  then  $p_t(x)$  could become 0, but then still  $p_{eff}(x) > 0$  would be the case. So the combined pdf is valid but it doesn't guarantee to reduce variance, because of the max operator and the re-normalization.

## 5.1 Optimality

In this section we want to find the truly optimal pdf  $p_t^*(x)$  for our free pdf. First we look at the variance which can be written as  $E(x^2) - E(x)^2$  that equals in our case  $J(p) - F^2$  with  $J(p) = \int_X \frac{f(x)^2}{q(x) + c_t p(x)}$ . The optimal compensating pdf would then be  $p_t^*(x) = \arg \min_p J(p)$ .

To make sure  $p_t^*(x)$  is a valid pdf we also set these two constraints

$$p_t^*(x) > 0, \text{ and } \int_X p_t^*(x) dx = 1$$

In the original paper [KŠV<sup>+</sup>19, Appendix A] they used the Karush-Kuhn-Tucker constraints to derive the optimal solution as

$$p_t^\pm(x) = \frac{f(x)}{\sqrt{c_t \lambda}} - \frac{q(x)}{c_t}, \quad p_t^*(x) = \max\{0, p_t^\pm(x)\}.$$

$\lambda$  is the Lagrange multiplier and ensures normalization.

Because of the max operator there is no analytical formulation for the solution and an iterative calculation is impractical in practice, but if the MIS-compensated solution from equation 5.2 is close-enough to the optimal one it can be used instead. The next section will show that the compensated solution is indeed often similar to the optimal one.

### 5.1.1 Bounds for the MIS-compensated solution

To show the similarity of the optimal solution and the MIS-compensated one they derived the following bound for  $\lambda$ :

$$c_t F^2 \leq \lambda \leq \frac{1}{c_t} F^2.$$

For a more detailed explanation of how they got the bounds please refer to [KŠV<sup>+</sup>19, Appendix B]. From there we can then see that if  $p_t^\pm(x) > 0$  then  $\lambda = c_t F^2$  and with that

$$p_t^*(x) = \tilde{p}_t(x).$$

In case they are not the same they checked how badly the compensated pdf could influence variance compared to the optimal one and the worst case increase in variance is  $\frac{1}{c_t}$  as they proved in [KŠV<sup>+</sup>19, Appendix C]. They also executed some tests to see how bad the compensated pdf would really become and got a factor of 1.6 times the variance at most.

They showed that the compensated pdf is often equal to the optimal one and in cases where it is not it's still a good solution, since it can be easily obtained and doesn't introduce too much variance.

(is it okay to refer to the original paper for the derivation of the optimal solution (appendix proves). Yes it is, the idea should be clear here and the reference also to the paper.) ToDo

(show how F will be approximated by  $\lambda$ . maybe no approach to approximation is shown) ToDo

(Explain their work. Formulas should be clearly explained.) ToDo

**ToDo**

## 6. Application: Image Based Lighting

In this chapter we start by describing image based lighting shortly. Following that the approach of Karlík et al. is explained to use in image based lighting. ([Explain the contents of this chapter \(at the end\)](#))

### 6.1 Image Based Lighting

Creating realistic lighting for a scene can be very complex and time consuming. To avoid modelling a lighting environment a commonly used method is to create an image of a real world environment and using that for the direct illumination of a scene. This approach is called image based lighting and was introduced back in 1998 by Debevec in [Deb98].

To get the lighting of a scene in the real world you could place a mirror sphere in the middle of your scene and the a photo of that sphere ideally from far away so the camera itself doesn't take up a large part of the scene. This method is called sphere mapping. During rendering you place your scene in the center of that textured sphere and if a shadow ray doesn't hit any object of the scene you read out the texel on the sphere that corresponds to the direction of the ray.

A different method is to take six pictures of the scene from the top, bottom, left, right, front and back and then use the same evaluation method as with the sphere with the direction of the shadow ray direction to get the texel from the environment map. For more information on how to use environment maps/image based lighting consider [env19].

### 6.2 MIS Compensation in Image Based Lighting

Here we will discuss the application example from Karlík et al. in [KŠV<sup>+</sup>19, Section 6-7].

#### 6.2.1 Setup

They calculate the unoccluded direct illumination with the following rendering equation:

$$L_{dir}(x, \omega_o) = \int_{H(n)} L_I(\omega_i) f_r(x, \omega_i, \omega_o) |n \cdot \omega_i|_+ d\omega_i.$$

$x$  is the surface position and  $\omega_o$  the outgoing view direction. The integration domain  $H(n)$  is the upper hemisphere centered around the normal  $n$ . The HDR environment map is used through  $L_I(\omega_i)$ ,  $f_r(x, \omega_i, \omega_o)$  corresponds to the BRDF at position  $x$  with incoming

direction  $\omega_i$  and outgoing direction  $\omega_o$ .  $|n \cdot \omega_i|_+$  is the positive cosine of the angle between the normal and the incoming direction i.e.  $\max\{0, |n \cdot \omega_i|\}$ .

In a standard Monte Carlo renderer two pdfs would be used, one proportional to the BRDF-cosine product and another one proportional to the environment map. Those two would then be combined in MIS with the balance heuristic. Generally the BRDF-cosine product pdf is given analytically and depends on the position, the outgoing and the incoming direction whereas the environment map pdf is mostly tabulated and only depends on the incoming direction. Since the second one is easier to modify, that one was used as the free pdf in their work.

### 6.2.2 PDF solutions

general sampling density is obtained by inserting render eq from 6.2.1 into equation 5.2 and get ... Solution depends on out direction and position and tabulation would be impractical for that so more applicable approximate solutions are presented

#### 6.2.2.1 Normal-dependend solution

For this they assumed a diffuse BRDF with albedo  $\rho \equiv \frac{1}{\pi}$ . depends on normal direction but can be easily precomputed for a number of directions. is practical but uses much memory and might need changes in the rendering implementation.

#### 6.2.2.2 Normal-independend solution

Therefore this method gets rid of the normal dependency by averaging over all normal directions (for more detailed explanation see [KŠV<sup>+</sup>19, Appendix D]). "Formula" corresponds to simple subtraction from the tabulated HDR map pdf and re-normalization

### 6.2.3 Evaluation

(Explain how their approach improves ibl variance)

**ToDo**

(If space left show results)

**ToDo**

# 7. Application: Path Guiding

In this chapter we will give a short overview of what path guiding is. After that we will look at how Kerlík et al. [KŠV<sup>+</sup>19] applied MIS compensation on path guiding to further improve render quality.

## 7.1 Path Guiding

The idea behind importance sampling is to let the sampling density be proportional to the integrand to get a small as possible variance in the result or image in our use case. Normally we can use the BRDF and the direct illumination to create two sampling distributions, but one important part is the indirect illumination which is not known beforehand and therefore can't be used easily. That's exactly what path guiding is for. It learns the incident radiance and builds a distribution based on this. This distribution is then used to create a new image from scratch to create a better distribution until the final iterations yields the image after combining all iterations weighted by their inverse variance [VHH<sup>+</sup>19]. During rendering the distribution can be normally used in MIS.

## 7.2 MIS Compensation in Path Guiding

**ToDo**

(Explain how their approach improves path guiding)

## 8. Summary

(Write a short summary of what they achieved)

ToDo

### 8.1 Future Work

(Mention what could be done next based on this work)

ToDo

# Bibliography

- [ARBJ03] S. Agarwal, R. Ramamoorthi, S. Belongie, and H. W. Jensen, “Structured importance sampling of environment maps,” 2003.
- [CAM08] P. Clarberg and T. Akenine-Möller, “Practical product importance sampling for direct illumination,” *Comput. Graph. Forum*, vol. 27, pp. 681–690, 04 2008.
- [CDG<sup>+</sup>08] O. Cappé, R. Douc, A. Guillin, J.-M. Marin, and C. P. Robert, “Adaptive importance sampling in general mixture classes,” *Statistics and Computing*, vol. 18, no. 4, pp. 447–459, Dec 2008. [Online]. Available: <https://doi.org/10.1007/s11222-008-9059-x>
- [Deb98] P. Debevec, “Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography,” ser. SIGGRAPH 98, 1998. [Online]. Available: <http://www.pauldebevec.com/Research/IBL/>
- [env19] “Chapter 7. environment mapping techniques,” November 2019. [Online]. Available: [http://developer.download.nvidia.com/CgTutorial/cg\\_tutorial\\_chapter07.html](http://developer.download.nvidia.com/CgTutorial/cg_tutorial_chapter07.html)
- [GKDS12] I. Georgiev, J. Křivánek, T. Davidovič, and P. Slusallek, “Light transport simulation with vertex connection and merging,” *ACM Trans. Graph.*, vol. 31, no. 6, pp. 192:1–192:10, Nov. 2012. [Online]. Available: <http://doi.acm.org/10.1145/2366145.2366211>
- [HEV<sup>+</sup>16] S. Herholz, O. Elek, J. Vorba, H. Lensch, and J. Krivanek, “Product importance sampling for light transport path guiding,” *Computer Graphics Forum*, vol. 35, 06 2016.
- [HO14] H. Y. He and A. B. Owen, “Optimal mixture weights in multiple importance sampling,” 2014.
- [IVG<sup>+</sup>19] K. Ivo, P. Vévoda, P. Grittman, T. Skřivan, P. Slusallek, and J. Křivánek, “Optimal multiple importance sampling,” *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2019)*, vol. 38, no. 4, pp. 37:1–37:14, Jul. 2019.
- [KKCF13] A. King, C. Kulla, A. Conty, and M. Fajardo, “Bssrdf importance sampling,” in *ACM SIGGRAPH 2013 Talks*, ser. SIGGRAPH ’13. New York, NY, USA: ACM, 2013, pp. 48:1–48:1. [Online]. Available: <http://doi.acm.org/10.1145/2504459.2504520>
- [KŠV<sup>+</sup>19] O. Karlík, M. Šik, P. Vévoda, T. Skřivan, and J. Křivánek, “Mis compensation: Optimizing sampling techniques in multiple importance sampling,” *ACM Trans. Graph. (SIGGRAPH Asia 2019)*, vol. 38, no. 6, 2019.
- [LPG13] H. Lu, R. Pacanowski, and X. Granier, “Second-order approximation for variance reduction in multiple importance sampling,” *Computer Graphics Forum*, vol. 32, 10 2013.

- [MGN17] T. Müller, M. Gross, and J. Novák, “Practical path guiding for efficient light-transport simulation,” *Computer Graphics Forum*, vol. 36, no. 4, pp. 91–100, Jun. 2017. [Online]. Available: <http://dx.doi.org/10.1111/cgf.13227>
- [PBPP11] A. Pajot, L. Barthe, M. Paulin, and P. Poulin, “Representativity for robust and adaptive multiple importance sampling,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 8, pp. 1108–1121, Aug 2011.
- [PJH18] M. Pharr, W. Jakob, and G. Humphreys, *Physically Based Rendering: From Theory To Implementation*, 3rd ed. MK, Morgan Kaufmann, 2018. [Online]. Available: <http://www.pbr-book.org/>
- [Sch18] J. Schudeiske, “Fotorealistische bildsynthese,” 2018, course slides from lecture at the KIT.
- [SHSK16] M. Sbert, V. Havran, and L. Szirmay-Kalos, “Variance analysis of multi-sample and one-sample multiple importance sampling,” *Computer Graphics Forum*, vol. 35, no. 7, pp. 451–460, 2016. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13042>
- [Vea97] E. Veach, “Robust monte carlo methods for light transport simulation,” Ph.D. dissertation, Stanford University, December 1997.
- [VG95] E. Veach and L. J. Guibas, “Optimally combining sampling techniques for monte carlo rendering,” *SIGGRAPH 95 Proceedings*, pp. 419–428, August 1995.
- [VHH<sup>+</sup>19] J. Vorba, J. Hanika, S. Herholz, T. Müller, J. Křivánek, and A. Keller, “Path guiding in production,” in *ACM SIGGRAPH 2019 Courses*, ser. SIGGRAPH ’19. New York, NY, USA: ACM, 2019, pp. 18:1–18:77. [Online]. Available: <http://doi.acm.org/10.1145/3305366.3328091>
- [wyz19] “Variance and standard deviation of a random variable,” November 2019. [Online]. Available: [https://www.wyzant.com/resources/lessons/math/statistics\\_and\\_probability/expected\\_value/variance](https://www.wyzant.com/resources/lessons/math/statistics_and_probability/expected_value/variance)



# **Erklärung**

Ich versichere, dass ich die Arbeit selbstständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, die wörtlich oder inhaltlich übernommenen Stellen als solche kenntlich gemacht und die Satzung des KIT zur Sicherung guter wissenschaftlicher Praxis in der jeweils gültigen Fassung beachtet habe. Die Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt und von dieser als Teil einer Prüfungsleistung angenommen.

Karlsruhe, den November 19, 2019

(Christian Navolskyi, B. Sc.)