Optimal Multiple Importance Sampling for Bidirectional Path Tracing

Ronan CAILLEAU, Glenn KERBIRIOU, Paul-Élie PIPELIN and Corentin SALAÜN
École Supérieure d'Ingénieurs de Rennes (ESIR)

Petr VÉVODA*, Jaroslav KŘIVÁNEK*, Fabrice LAMARCHE†
**Chaos Czech* †*University of Rennes 1*

***Abstract*—Photorealistic image rendering consists in simulating light transport in 3D environments. It relies on estimating integrals with numerical methods such as Monte Carlo integration. Multiple Importance Sampling is a basic technique for improving the robustness of Monte Carlo estimators. A recent work in collaboration with the Chaos Czech research department has proposed an optimal solution for Multiple Importance Sampling that can significantly reduces the noise in rendered images. In this report, we present a way to integrate this new optimal method in a Bidirectional Path Tracer.**

***Index Terms*—Light Transport, Monte Carlo Integration, Optimal Multiple Importance Sampling, Rendering, Bidirectional Path Tracing**

I. INTRODUCTION

The creation of photorealistic images, almost impossible to distinguish from real photographs, is a complex task since it involves simulating the dispersion of light in the environment. Some algorithms have been proposed to simulate this process based on physical equations that describe how light propagates in scenes.

Simulation of light transport can be reduced to solving multi-dimensional integrals. Given the complexity of solving these integrals, state of the art methods generally use Monte Carlo estimators. Such estimators use random sampling of the integration domain to evaluate complex integrals. These estimators converge to the exact result with the number of samples taken. One of the key problems is improving the convergence rate of these estimators. This can be achieved by reducing the variance of the estimators. In image rendering, the variance of the estimators can be seen as the noise in the estimated image. Hence, reducing the variance of Monte Carlo estimators will result in less noise in the computed image and a greater visual quality.

One of the most used technique to improve these estimators is Multiple Importance Sampling (MIS). MIS is a method that combines multiple sampling techniques by weighting the samples given by each technique.

Recently, Kondapaneni et al. [1] have proposed an optimal solution for MIS, Optimal MIS, that provably minimizes the variance of MIS Monte Carlo estimators. Optimal MIS has shown great results on direct lighting evaluation, but has not yet been implemented in a full light transport algorithm such as Bidirectional Path Tracing.

Bidirectional Path Tracing [2] (BDPT) is a complex and robust rendering algorithm particularly suitable to render lighting effects such as caustics or mirror reflections. The rendering quality of this algorithm is largely due to the use of Multiple Importance Sampling.

The project has been proposed by the research department of Chaos Czech company and aims at implementing Optimal Multiple Importance Sampling [1] in a Bidirectional Path Tracing algorithm. The project follows the internship of Ronan Cailleau that took place in summer 2019.

II. PRESENTATION OF CHAOS CZECH AND THE PROJECT

Chaos Czech is a company specialized in 3D computer graphics software based in Czech Republic. The company, formerly named Render Legion, became part of Chaos Group in 2017. Today, the team is composed of about fifty people. Corona Renderer is the main software of the company. It can render perfectly photorealistic images, as well as biased images, but faster. Corona Renderer also handles a wide variety of materials and lighting techniques while showing good performance.

The philosophy of the company is to provide easy to use tools for artists and designers, allowing them to create great

imagery and animation. The software embeds lots of ready-made content, movement effects, physical simulation and is frequently updated with new features. Lots of improvements have been made on rendering quality thanks to the research department. Ronan Cailleau's internship, supervised by Jaroslav Křivánek and Petr Vévoda, took place in this philosophy of research and continuous improvement, and so does this project.

III. Organization of the project

During the project, we worked in direct communication with Chaos Czech. Our main interlocutor was Petr Vévoda, who greatly contributed to the original Optimal Multiple Importance Sampling publication. We organized meetings with him every two weeks to present our work and to expose the problems we were facing. He gave us some advice and helped us to schedule the different tasks related to the project.

In the team, we started by catching up on the Bidirectional Path Tracing algorithm and MIS. It was a key part of the project in order to understand the theory behind MIS and Bidirectional Path Tracing. During the project, we organized the tasks by importance and split them among the team members. During each meeting with Petr Vévoda we discussed about our results and the tasks to be prioritized.

As a research project, it was difficult to anticipate the problems and the time required to solve them. Moreover, in the beginning, Ronan Cailleau was the only one who had enough knowledge on the subject. We thus relied on his predictions and the ones of the project supervisors, Petr Vévoda, Jaroslav Křivánek and Fabrice Lamarche.

IV. Previous work

In the field of computer graphics, rendering is the process of automatically transforming 3D numerical models into images. This report focuses on physically based rendering which aims at accurately modeling the flow of light in the real world, and solving as correctly as possible the equation describing it.

The corner stone of physically based rendering is the physical equation of radiance equilibrium or *rendering equation* [3]. This fundamental equation (1) describes the evaluation of radiance leaving a surface given an incoming illumination.

$$L(x \to y) = L_e(x \to y) + \int_M f_s(z \to x \to y) G(x \leftrightarrow z) L(z \to x) dA(z) \quad (1)$$

With $L(x \to y)$ the radiance leaving the surface at the point x in the direction $x \to y$, $L_e(x \to y)$ the radiance at the point x in the direction $x \to y$ if the surface supporting x is a light source. $f_s(y \to x \to z)$ is the ratio of reflected, transmitted or absorbed radiance in x from the direction $x \to z$ to the direction $x \to y$, also called Bidirectional Scattering Distribution Function (BSDF). $G(x \leftrightarrow y) = V(x \leftrightarrow y)\frac{cos(\theta_i)cos(\theta_{i'})}{||x-y||^2}$ is the geometric term between x and y, $V(x \leftrightarrow y)$ stands for the mutual visibility between x and y, and θ_i and $\theta_{i'}$ are the polar angles at x and y with the direction $x \leftrightarrow y$. M is the set of all points of the scene.

A. Monte Carlo integration

In practice, the integral (1) has no closed form but can be estimated using numerical integration such as Monte Carlo integration. This method consists in sampling random values in the domain of the integral and evaluating the function for these values. A weighted sum of these evaluations is then used as an estimator of the integral. A Monte Carlo estimator $\langle F \rangle$ is given by the following formula:

$$\langle F \rangle = \frac{1}{N} \sum_{i=1}^{N} \frac{f(X_i)}{p(X_i)} \quad (2)$$

With f the function to integrate over the domain D, N the number of samples taken and $p(X_i)$ the probability density of producing the sample $X_i \in D$. It can easily be shown that such an estimator converges to the correct integral, i.e. $E[\langle F \rangle] = \int_D f(x)dx$ if $f(X) \neq 0 \Rightarrow p(X) \neq 0$ holds true.

However, a Monte Carlo estimator can show very high variance depending of the shape on the probability density function (pdf) used for sampling the domain. The more the pdf has the same shape of the absolute value of the integrand i.e. the more they are positively correlated, the lesser is the variance.

In the field of rendering, it is almost impossible to model a pdf that has the same shape as the function to integrate. This results in possibly high variance that leads to unwanted noise in the computed images. Thus, reducing noise is equivalent to reducing the variance of the estimator. Hence, one way to reduce variance is to focus sampling in the parts of the domain where the function to integrate has high absolute values. This process, called Importance Sampling, is the fist step to achieve variance reduction. Good sampling techniques will naturally lead to lower variance.

B. Multiple Importance Sampling and balance heuristic

Multiple Importance Sampling (MIS) is a method that aims at reducing variance and making more robust estimators by combining different sampling techniques. It is achieved by computing a weighted sum of the result of the different sampling techniques.

A MIS Monte Carlo estimator $\langle F \rangle^*$ is given by :

$$\langle F \rangle^* = \sum_{i=1}^{N} \sum_{j=1}^{n_i} w_i(X_{ij}) \frac{f(X_{ij})}{n_i p_i(X_{ij})} \quad (3)$$

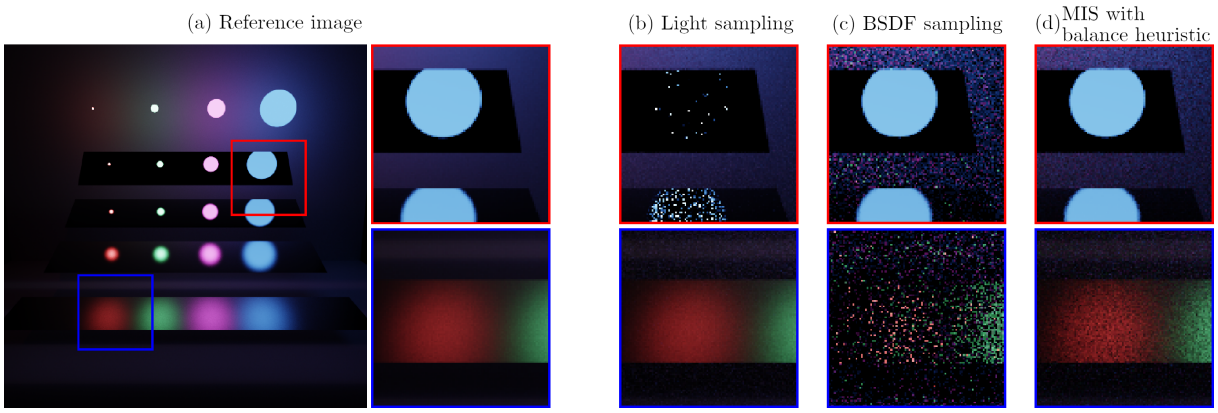

Figure 1. Particular regions of a scene (a) rendered in 64 samples per pixel using different sampling techniques : Light sampling (b), BSDF sampling (c) and Multiple Importance Sampling using BSDF and Light sampling combined with the balance heuristic (d). Notice the superior robustness of MIS.

With X_{ij} the j^{th} sample of the technique i, $w_i(X)$ a weighting function such that $f(X) \neq 0 \Rightarrow \sum_{i=1}^{N} w_i(X) = 1$ and $p_i(X) = 0 \Rightarrow w_i(X) = 0$. N is the number of sampling techniques, n_i the number of samples generated by the sampling technique i.

The variance of a Monte Carlo estimator using MIS, $V[\langle F \rangle^*]$, has the following form [4]:

$$V[\langle F \rangle^*] = \sum_{i=1}^{N} \int_D \frac{w_i(x)^2 f(x)^2}{n_i p_i(x)} dx - \sum_{i=1}^{N} \frac{1}{n_i} \langle w_i, f \rangle^2 \quad (4)$$

This equation is crucial in the goal of reducing the noise in rendered images. It is also the basis of the most common MIS heuristic: the balance heuristic [4]. This heuristic sets the weight of the samples from each techniques as follows:

$$w_i(X) = \frac{n_i p_i(X)}{\sum_{k=1}^{N} n_k p_k(X)} \quad (5)$$

Eric Veach has shown that the balance heuristic is nearly optimal. Indeed, the balance weights provided in formula (5) minimize the first term of the estimator variance given by the formula (4). In most of the cases, this minimization term is close to optimal but not in every situation.

An example of the impact of MIS in rendering is depicted in Figure 1. The figure compares two sampling techniques and their combinations using MIS with balance heuristic in two particular regions of the reference image. For direct illumination, the light sampling technique (sampling a point directly on a light source) is well suited for diffuse reflections but fails to render glossy reflections. For the BSDF sampling technique (sampling a scattering direction and see if there is a light in this direction), it is the opposite. MIS with balance heuristic offers a much more robust solution by trying to exploit the strengths of each technique.

C. Optimal Multiple Importance Sampling

Recent research has shown that the MIS balance heuristic can be far less optimal than what was previously thought in some cases. Eric Veach has shown that the balance heuristic was a good heuristic and also that minimizing the second term of the variance (4) could lead to slightly lower variance. Kondapaneni et al. [1] have shown that Eric Veach actually made the (common) hidden assumption that the weighting functions should be positive. This constraint is not necessary for a MIS estimator to be unbiased.

By fully minimizing the variance of the MIS estimator (4) and relieving the constraints of having only positive weights, the variance can actually be lowered much more than with the state of the art balance heuristic. Hence, the optimal weighting functions w_i^o are deduced by analytically minimizing the functional (4):

$$w_i^o(X) = \alpha_i \frac{p_i(X)}{f(X)} + \frac{n_i p_i(X)}{\sum_{j=1}^{N} n_j p_j(X)} \left(1 - \frac{\sum_{j=1}^{N} \alpha_j p_j(X)}{f(X)}\right) \quad (6)$$

With $\boldsymbol{\alpha}$ solving the linear system:

$$\mathbf{A}\boldsymbol{\alpha} = \mathbf{b}$$
with $\mathbf{A}$ the technique matrix and
$\mathbf{b}$ the contribution vector defined such as
$$a_{i,k} = \langle p_i, \frac{p_k}{\sum_{j=1}^{N} n_j p_j} \rangle \text{ and } b_j = \langle f, \frac{p_i}{\sum_{j=1}^{N} n_j p_j} \rangle$$
with $a_{i,k}$ the elements of the matrix $\mathbf{A}$ (7)

The main difference with the balance heuristic is that the integrand is used in the computation of the weights. This is due to the fact that the variance is minimized when the pdf and the integrand have a similar shape, which is not taken into account by the balance heuristic. Another important difference with the balance heuristic is the handling of zero contribution samples (i.e. $f(X) = 0$ but $p_i(X) \neq 0$). These

samples now have a contribution to the optimal result. Indeed, they have a non-null pdf of being sampled and hence will contribute to the estimation of the technique matrix. The consequence is that their pdf have to be correctly computed which is not necessarily the case in rendering engines to save rendering time.

Using this new estimator in rendering can lead to a significant noise reduction as shown in [1] and [5].

D. Bidirectional Path Tracing

Bidirectional Path Tracing [2], [6] is an algorithm used to solve the rendering equation (1). This particular algorithm relies on several sampling techniques, which are combined using MIS. Bidirectional Path Tracing evaluates the rendering equation over pixel p by generating paths the light could take connecting the camera and a light source. A path $\overline{x}$ of length n is a sequence of vertices $(x_1, ..., x_n)$ such as each vertex belongs to a surface. Also, the depth of a path is defined as $d = n - 2$. A path has to connect a camera and a light source to contribute to the rendered image. The image is generated by evaluating the radiance flux value I_p at pixel p of the image expressed as :

$$
\begin{aligned}
I_p = & \\
& \int_{M^2} L_e(x_1 \to x_0) G(x_0 \leftrightarrow x_1) W_p(x_0 \to x_1) dA(x_0) dA(x_1) \\
& + \int_{M^3} L_e(x_2 \to x_1) G(x_0 \leftrightarrow x_1) f_s(x_0 \to x_1 \to x_2) \\
& \quad G(x_1 \leftrightarrow x_2) W_p(x_1 \to x_2) dA(x_0) dA(x_1) dA(x_3) \\
& + ... \qquad (8)
\end{aligned}
$$

With $W_p(x \to x')$ the importance of the camera at pixel p between x and x' which can be seen as the opposite of the radiance, that creates a contribution in the camera sensor when meeting radiance. x_i is a point on the surfaces of the scene and $\leftrightarrow$ symbolizes the symmetry of the functions. This equation is equivalent to evaluating every paths of every length between cameras and lights. Notice that the equation (8) is a global extension of the rendering equation 1).

In practice, the paths connecting the lights and the camera are constructed by randomly generating two sub-paths, one starting from the camera and one from a light source. Vertices from the camera sub-path are labeled x_i and vertices from the light sub-path are labeled y_i. Then, every possible combination of connections between the two are computed as shown in Figure 2.

In the Figure 2, connecting the vertices x_2 and y_0 creates a path of length 4 (x_0, x_1, x_2, y_0). All ways to connect the sub-paths are the samples used to evaluate the equation (8) with Monte Carlo integration.

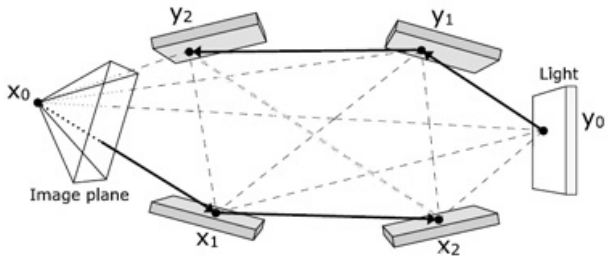

Figure 2. All possible paths that can be generated with 2 sub-paths of length 3. Image courtesy of Mikel Adamsen.

As a result, every combinations of vertices of same length contributes to the same integral. The different combinations of vertices of a length n are the samples used for the MIS.

In a BDPT, a sampling technique corresponds to a specific connection (s, t) between a light sub path of length s and camera sub path of length t, making a full path of length $s + t$. For example, a path of length 4 can be generated with 3 vertices from the light sub-path and 1 from the camera sub-path. It can also be created with 2 vertices from the light sub-path and 2 from the camera sub-path. We can evaluate the throughput and probability density of the sample in order to compute the Monte Carlo integration for the rendering equation (8).

V. Our solutions

This section develops the implementation of Optimal MIS in a BDPT engine named Physically Based Ray Tracing (PBRT) [7]. This renderer, based on [8], is one on the most commonly used in research projects. PBRT is an open source engine that handles many different rendering techniques, materials, lights and cameras. Our main contributions are the implementation of the optimal MIS solver, the correction of materials issues and some special case handling.

A. Optimal Multiple importance sampling implementation

The implementation of Optimal MIS in a BDPT is a non trivial task. It led us to modify the algorithm initially presented in [1] to make it simpler to apply and also to solve some of the special cases related to BDPT. Our new solver formulation is defined in Algorithm 1. We proved that this new version of the optimal MIS solver is equivalent to the previous one. In the algorithm, the balance weights of the j^{th} sample of the technique i with every technique are stored in the vector $\mathbf{W}_{i,j}$.

Algorithm 1 Optimal Multiple Importance Sampling solver

1: $\langle \mathbf{A} \rangle \leftarrow 0^{N \times N}, \langle \mathbf{b} \rangle \leftarrow 0^{N \times 1}$
2: **for** $iteration \leftarrow 0$ to $maxIteration - 1$ **do**
3: **for** $i \leftarrow 1$ to N **do**
4: $\{X_{i,j}\}_{j=1}^{n_i} \leftarrow$ draw n_i samples from technique p_i
5: **end for**
6: $\langle \mathbf{A} \rangle \leftarrow \langle \mathbf{A} \rangle + \sum_{i=1}^{N} \sum_{j=1}^{n_i} \mathbf{W}_{i,j} \mathbf{W}_{i,j}^T$
7: $\langle \mathbf{b} \rangle \leftarrow \langle \mathbf{b} \rangle + \sum_{i=1}^{N} \sum_{j=1}^{n_i} f(X_{i,j}) \mathbf{W}_{i,j} / \sum_k n_k p_k(X_{i,j})$
8: **end for**
9: $\langle \boldsymbol{\alpha} \rangle \leftarrow$ solve linear system $\langle \mathbf{A} \rangle \langle \boldsymbol{\alpha} \rangle = \langle \mathbf{b} \rangle$
10: **return** $\sum_{i=1}^{N} \langle \boldsymbol{\alpha} \rangle n_i$

In practice, a matrix and vector is associated to each pixel of the image and for each depth. For all samples generated at each pixel, the technique matrix $\mathbf{A}$ and the contribution vector $\mathbf{b}$ (section IV-C) are updated (lines 2-8).

Once all the samples have been generated, the linear system is solved to obtain the $\boldsymbol{\alpha}$ vector (line 9). Each coefficient of this vector corresponds to the contribution of one technique. Each contribution is then multiplied by the number of samples of the technique that produced it. The result returned by this solver is the amount of radiance captured by the pixel for all the generated paths.

The algorithm (1) is also called the direct estimator. It relies on the property that the sum of the vector $\boldsymbol{\alpha}$ is a biased estimator of the computed integral. Its bias has been measured to be in $O(n^{-1})$, while the variance of Monte Carlo estimators are in $O(n^{-\frac{1}{2}})$. So the bias could be visible for low samples count, but will be drowned under the variance at some point. Another algorithm, called the progressive estimator performs far worse than the direct one, but is unbiased. We also have a partial implementation of it, but it will not be discussed in this work.

Even if optimal weights are mathematically better than any other heuristic for equal sample count, in practice they are not always the best. Indeed, the computation time needed to update the solvers and solve the linear system makes the computation of optimal weights slower than the computation of balance weights. This is why we have added the possibility to choose which weighting method to use for each path depth. This way, it is possible to use only the optimal weight while they are faster than the balance heuristic.

B. Light tracing samples handling

Light tracing samples in Bidirectional Path Tracing are the light paths created with only one vertex in the camera sub-path (the camera vertex). An example that represents this case is shown in Figure 3.

The particularity of these samples is that the pixel intersected by the path cannot be predicted. Hence, these samples actually contribute to every pixel of the image with a zero contribution except for the pixel intersected by the path that will get the contribution. In the case of the balance heuristic, the zero contribution does not have to be taken into account for the whole image.

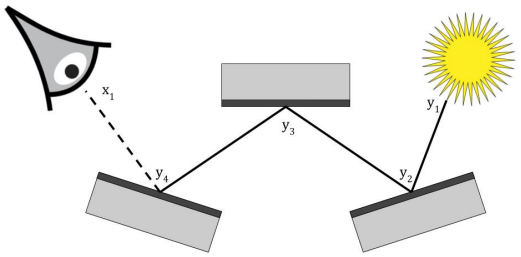

Figure 3. Schema of a light tracing sample

However, the method used to compute the optimal weights requires to update the technique matrix $\mathbf{A}$ associated with every pixel and to update the contribution vector for the intersected pixel. Updating all solvers for zero contributions would have an extremely high computational cost. For the pixel intersected by the path, all techniques have a non-zero pdf of producing this sample. Thus, the system will be updated normally. For all the other pixels, only the light tracing technique has been able to produce this sample. This gives a balance weight of 1 for the light tracing technique and 0 for all the others. Since the weights are the same regardless of the pixel, the update will be the same. So we only store for each pixel the number of light tracing samples that actually made it through the pixel. We can then use this number right before solving the system to account for all the zero contribution light tracing samples that have not been added to the matrix. This is much faster and requires almost no extra memory.

Another issue with the light tracer is its much higher number of samples than the other techniques. It was the motivation to design our own version of the direct estimator.

C. Zero contribution samples special case

Another special case is for the zero contribution samples. It is constituted of the samples that have a zero contribution to the estimator but a non-zero pdf of being sampled. These zero contribution samples are paths that do not transport radiance from the light to the camera. When using the balance weights, these cases were not taken into account since they had no impact on the weighted sum of the estimator equation (3). But now, they are necessary for the calculation of the Optimal weights.

There are different cases that result in zero contribution samples, some of them are depicted in Figure 4. The first example of the figure (top left) is the case of non visibility between the vertices when connecting the camera sub-path and the light sub-path. The second one (top right) shows the case where the path leaves the scene without making a connection between the 2 sub-paths. Then, the third one

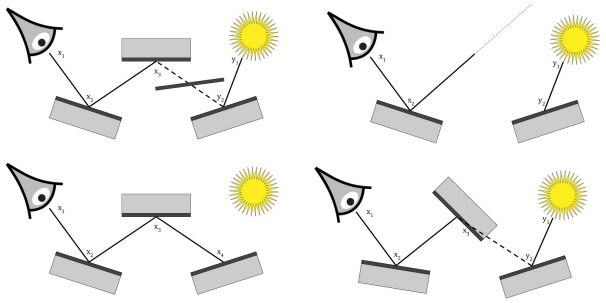

Figure 4. Examples of some zero contribution samples

(bottom left) shows the particular case where we only consider the path created by the camera sub-path and that does not end on a light source. Finally, the fourth case (bottom right) is when one of the BSDF (and equivalent) is zero (here, the BSDF on x_3 is zero since the material is not transparent).

Another case of zero contribution paths we call unsampled samples. It happens when a technique does not manage to produce a path. In this case, we assumed that a null special path was actually generated by the technique, and only the technique could produce it.

Finally, the choice to work with balance weights rather than sample pdf's in the optimal weight solver makes the handling of these particular contributions much simpler.

D. Materials correction

Some of the materials that were already implemented in PBRT have to be adapted to the new integrator.

Firstly, PBRT does not compute the pdf in some very specific cases for some transmitting materials. If the contribution is zero, the pdf is not necessarily computed to save some rendering time. This may lead to errors in the computation of the optimal weights, which would result in a bias in rendered images. This was specifically the case in specular transmitting materials.

Fixing this did not lead to less bias. Nevertheless, it improved the robustness of the engine which is primordial for an unbiased renderer.

Secondly, some microfact models in PBRT happen to sometimes compute a small negative pdf when sampling a direction. This situation is exceptional, happening about once per tens of millions of samples. However, it can lead to a crash of the engine or a huge bias due to the smallness and the negativeness of the pdf value ($\approx -10^{-6}$ with double precision). This problem is due to the normal distribution of the microfacet models used in PBRT. It happens in very rare conditions that the incident ray is not in the same hemisphere of the microfacet's normal which is calculated according to the distribution. This phenomenon leads the computation of a small negative cosinus, and then a small negative pdf. We decided to consider this case as a flaw of the normal distribution model and to correct it by retrieving the original normal of the primitive. This allows to keep the contribution and saves computing time while preventing from errors.

VI. Results

This section shows the various evaluation methods used to benchmark our implementation of Optimal MIS weights in a Bidirectional Path Tracer. It will with the classic balance weights in different scenarios. The metric used to compare images is the mean relative squared error (*MRSE*) that allows to highlight the relative variance compared to a reference. This metric is computed by dividing the squared difference between the image and the reference by the reference value. Compared to a classic mean squared error, the result is less dominated by the high values of radiance in some images.

A. Bidirectional Path Tracing

In order to successfully perform physically based rendering, the implementation must remain unbiased. It means the results should not contain any statistical error in the radiance estimate. An algorithm bias can be estimated by comparing its results with these from an algorithm proven to be unbiased as in the Figure 5. If the error of the image is not uniformly distributed, it is very likely that there is a bias. The more samples there are, the more reliable this indicator is, although it does not explicitly prove the absence of bias.

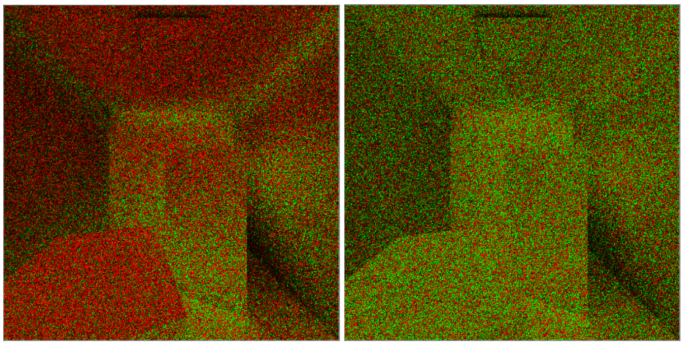

Figure 5. Visualization of the error distribution between a biased image and the reference (left) and between our implementation's image and the reference (right). The red and green colors represent a positive and a negative error respectively. Notice the red-dominant areas with the biased image

We compared results of the implementation of Optimal MIS with the ones produced by PBRT's Path Tracer and Bidirectional Path Tracer with balance heuristic on several scenes. It appears that the error is uniformly distributed over the images as the number of samples increases, indicating the absence of bias in our implementation of Optimal MIS in PBRT's Bidirectional Path Tracer.

B. Comparison to balance weights

As seen in IV-C, the optimal MIS weights analytically minimize the variance of the Monte Carlo estimators which should reduce the noise in rendered images. More importantly, the results obtained with the optimal MIS weights must be at least as good as these obtained using the balance heuristic for an equal number of samples.

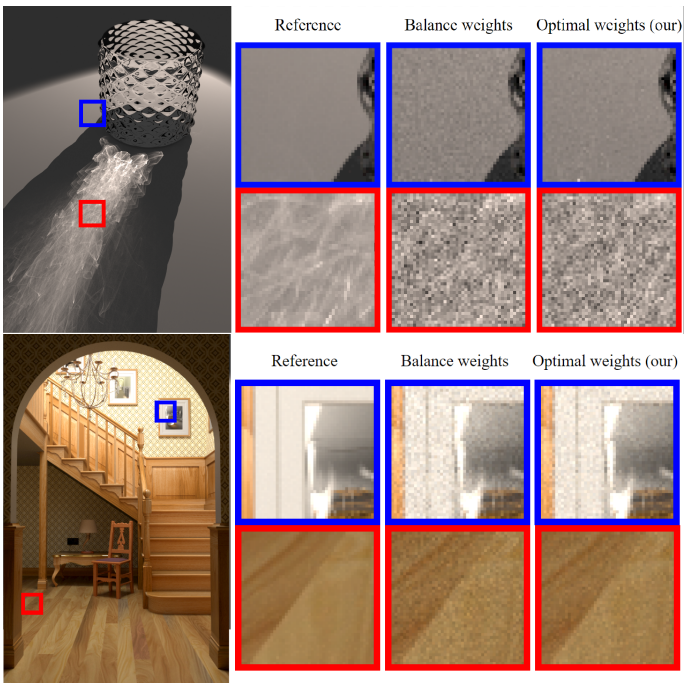

Figure 6. Comparison of BDPT with balance heuristic and Optimal MIS for a same number of samples on scenes GLASS (top) and STAIRCASE (bottom). The noise is significantly reduced for direct illumination on diffuse surfaces with optimal weights compared to balance weights.

Figure 6 compares the new optimal weights with the state of the art balance heuristic on different scenes. These images have been generated with the same number of samples. Zooming allows to see the noise in the images on small regions.

In both images, the noise reduction is clearly visible on diffuse surfaces in direct lighting. For the same number of samples, the optimal weights allow a large reduction in the variance of the estimators. This noise reduction is particularly important in areas where many techniques feature a high probability density of producing a sample but are not very good, but one technique is actually better than the others.

However, when only one technique is good and features a high probability density of producing a sample, there are no great improvements over the balance heuristic, meaning the balance heuristic is already almost optimal in this situation. This is the case of the caustic created in GLASS scene (top image, red zoom), since the light tracing technique is the only one able to render it. Here, the noise reduction between the balance heuristic and Optimal MIS is barely visible.

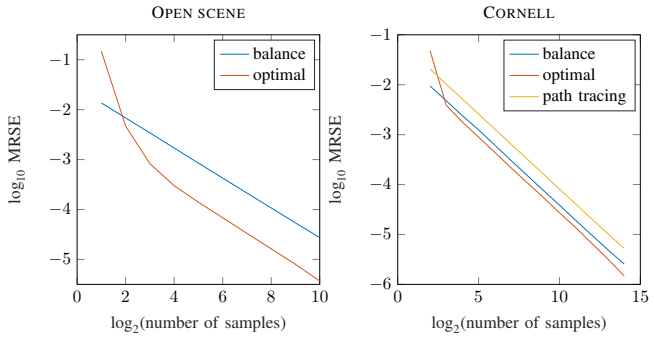

Figure 7. MRSE of balance and optimal weights versus the number of samples.

Figure 7 shows the evolution of the MRSE of the balance weights and the optimal MIS weights versus the number of samples. In this example, for every number of samples, the optimal weights lead to better results than the balance weights in terms of MRSE. In the worst cases, Optimal MIS is equivalent to the balance heuristic. There is an exception: if the number of samples is too low (usually under 4 from our observations), then some systems are estimated with too much noise, resulting in unstable resolution and a salt and pepper noise, leading to extra variance. Also, the bias is clearly visible in the computed images at these low number of samples. This is why in the previous curves, the MRSE of the optimal result is higher than the balance at the beginning.

C. Efficiency of the optimal weights

Optimal MIS has been proven better than balance heuristic for a same number of samples, but it is also more expensive in terms of computing time and memory consumption. When evaluating optimal weights, the size of the largest matrices of the linear system is proportional to the square of the maximum path length. Thus, the longer the paths are, the more expensive and time-consuming it is to store and update the matrices.

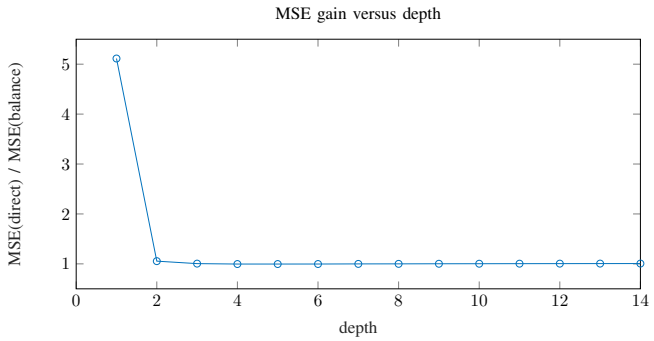

Figure 8. Gain of MSE versus path depth in the CORNELL scene.

We can see in figure 8 that the gain of variance immediately drops after depth 1 (path length 3). It is still noticeable at depth 2, and not noticeable after.

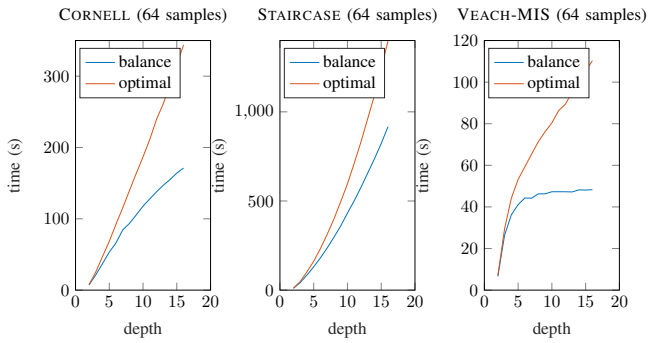

Figure 9. Comparison of the rendering time of the same scenes between balance heuristic and Optimal MIS. CORNELL scene is a closed box with the front wall open, some paths are not traced up to their maximum depth. The staircase is completely closed, so all path are completely traced. VEACH-MIS scene is widely open, so almost no paths are traced after depth 5.

Computing the optimal weights necessarily takes more time than the balance heuristic as shown in Figure 9. The extra time for computing the optimal weights could be used to compute more samples with the balance heuristic, and the later could end up with a lower variance.

The extra rendering time can vary greatly depending on the openness of the scene. The overall cost of computing Optimal MIS is acceptable for small depths. For long paths though, it is a waste of time to compute the optimal weights and it is better to use the regular balance heuristic.

The memory consumption also has to be addressed. We have to store a technique matrix and a contribution vector for each pixel and for each path depth. Storing a linear system of n techniques requires $\frac{n(n+1)}{2}$ floats for the technique matrix by exploiting the symmetry of the matrix and $3n$ floats for the contribution vector as we store a different vector for each color channel. Since there is a linear system for each path depth, the memory consumption increases with the cube of the maximum path depth as illustrated in Figure 10.

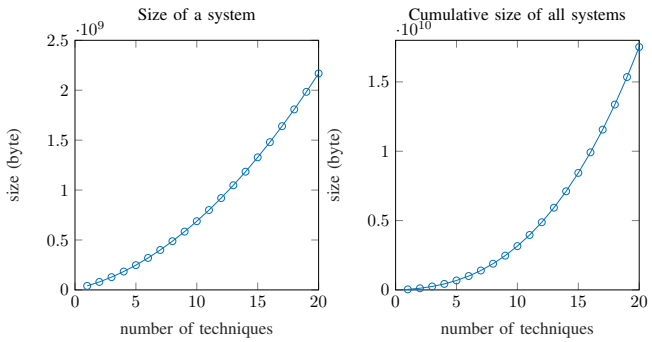

Figure 10. Theoretical size and theoretical cumulative size of storing the linear systems for each depth.

The overall memory cost of storing all the linear systems is another strong arguments in favor of not using the optimal weights for long paths. Without taking into account that some memory is also required to store the scene and other data structures for the ray tracing.

A solution is that with the implementation proposed, it is possible to switch from the optimal weights to the regular balance weights for any path length. As optimal weights are more efficient for short path lengths, we propose to switch back to balance weights for the paths of length 6 or more. In this manner, the engine benefits from the variance reduction of the optimal weights for the short paths and the computation performance of the balance weights for longer ones. We believe there is still room for improvement in the technical performance of our implementation.

D. Discussion and future work

The quality of the sampling techniques has a huge impact on the variance reduction, especially with Optimal MIS. Using other tools that lead to better techniques such as path guiding [9] that learns the best way to generate samples for each scene and scene specialized sampling techniques with pre-calculated sampling techniques. Also, Optimal MIS can be used in other algorithms using MIS like Vertex Connection and Merging [10] or Multiplexed Metropolis Light Transport [11].

A new sub optimal MIS heuristic has been proposed at the end of 2019. This new heuristic is the Variance aware MIS [5] that shows better result than the balance heuristic for a same computation time. This new heuristic allows less variance reduction than optimal MIS on simple cases but is faster. In a renderer such as a Bidirectional Path Tracer, it could be interesting to compare the performance of this two approaches and combine them in order to keep their strengths.

We will make an article from this project describing in detail the solution proposed for light tracing and zero contribution sample handling. We will also write a PBRT-style chapter detailing how to integrate and how works the optimal MIS in complex rendering engine.

VII. CONCLUSION

We managed to implement Optimal Multiple Importance Sampling in a Bidirectional Path Tracing rendering engine. To achieve this, we pointed out specific problems regarding Optimal MIS integration, and proposed effective solutions. Beyond that, we have proposed a practical way to calculate the optimal weights directly from the state of the art balance weights. Then, we were able to benchmark the improvements brought by this new approach compared to the state of the art balance heuristic. Finally, we brought to light the performance limitations of our implementation and proposed a number of

recommendations related to its use.

Doing so, we achieved the main objective of the project and our work provides a methodological basis for the integration of Optimal MIS in other contexts. In addition to the technical success, this project allowed us to develop important skills in photo-realistic rendering. This is a field of research that we are particularly interested in, and we will be able to re-use the knowledge we have acquired.

ACKNOWLEDGEMENTS

We thank Petr Vévoda for his time and help during the project. His knowledge of rendering and Optimal Multiple Importance Sampling was a great help.

We are grateful to Fabrice Lamarche who lead the project at the University of Rennes. He always took time for us when we needed him and helped us at writing this article and our poster with a very precise and commented rereading.

We would like to thank Kadi Bouatouch who gave us some courses on light transport simulation to help us at the beginning of the project. He also gave us some advice for the writing of this report.

Our last thanks goes to Benedikt Bitterli [12] for the beautiful PBRT test scenes. We used some of them in our research and to present our results.

TRIBUTE TO JAROSLAV KŘIVÁNEK

We sadly learned that December the 1st, Jaroslav Křivánek passed away in an accident. He was a great source of inspiration and the instigator of this project.

He was a corner stone in the Chaos Czech company and the computer graphics group at the Charles University of Prague.

We dedicate this work to him.

REFERENCES

[1] I. Kondapaneni, P. Vévoda, P. Grittmann, T. Skřivan, P. Slusallek, and J. Křivánek, "Optimal multiple importance sampling," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, p. 37, 2019.

[2] E. P. Lafortune and Y. D. Willems, "Rendering participating media with bidirectional path tracing," in *Rendering techniques' 96*. Springer, 1996, pp. 91–100.

[3] J. T. Kajiya, "The rendering equation," in *ACM SIGGRAPH computer graphics*, vol. 20, no. 4. ACM, 1986, pp. 143–150.

[4] E. Veach, *Robust Monte Carlo methods for light transport simulation*. Stanford University PhD thesis, 1997, vol. 1610.

[5] P. Grittmann, I. Georgiev, P. Slusallek, and J. Křivánek, "Variance-aware multiple importance sampling," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 6, pp. 1–9, 2019.

[6] E. Veach and L. Guibas, "Bidirectional estimators for light transport," in *Photorealistic Rendering Techniques*. Springer, 1995, pp. 145–167.

[7] M. Pharr, W. Jakob, and G. Humphreys. (2016) Git.

[8] ——, *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2016.

[9] 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

[10] 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

[11] T. Hachisuka, A. S. Kaplanyan, and C. Dachsbacher, "Multiplexed metropolis light transport," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, pp. 1–10, 2014.

[12] B. Bitterli, "Rendering resources," 2016, https://benedikt-bitterli.me/resources/.

APPENDIX

APPENDIX A
EQUIVALENCE OF OUR DIRECT ESTIMATOR

The technique matrix $\boldsymbol{A}$ and contribution vector $\boldsymbol{b}$ are defined by:

$$a_{i,k} = \int_D \frac{p_i p_k}{\sum_j n_j p_j} \tag{9}$$

$$b_i = \int_D \frac{f p_i}{\sum_j n_j p_j} \tag{10}$$

Then the vector $\boldsymbol{\alpha}$ that solves the system is an estimator of F.

$$\boldsymbol{A}\boldsymbol{\alpha} = \boldsymbol{b} \tag{11}$$

Let's define an alternative version of the system:

$$\boldsymbol{A}^*\boldsymbol{\alpha}^* = \boldsymbol{b}^* \tag{12}$$

with:

$$a^*_{i,k} = \int_D \frac{n_i p_i n_k p_k}{\sum_j n_j p_j} \tag{13}$$

$$b^*_i = \int_D \frac{f n_i p_i}{\sum_j n_j p_j} \tag{14}$$

We will now prove that the alternative system is equivalent to the previous one.

Let's define the $\boldsymbol{N}$ the sample count matrix:

$$\boldsymbol{N} = \begin{pmatrix} n_1 & & 0 \\ & n_2 & \\ & \ddots & \\ 0 & & n_N \end{pmatrix} \tag{15}$$

Then, the system (11) is equivalent to:

$$\boldsymbol{N}\boldsymbol{A}\boldsymbol{N}\boldsymbol{N}^{-1}\boldsymbol{\alpha} = \boldsymbol{N}\boldsymbol{b} \tag{16}$$

Then, we can see that:

$$\boldsymbol{A}^* = \boldsymbol{N}\boldsymbol{A}\boldsymbol{N} \tag{17}$$

$$\boldsymbol{b}^* = \boldsymbol{N}\boldsymbol{b} \tag{18}$$

$$\boldsymbol{\alpha}^* = \boldsymbol{N}^{-1}\boldsymbol{\alpha} \tag{19}$$

So by solving the alternative system, we can get the $\boldsymbol{\alpha}^*$ vector, and then deduce $\boldsymbol{\alpha} = \boldsymbol{N}\boldsymbol{\alpha}$. Then:

$$\langle F^o \rangle = \sum_i \alpha_i = \sum_i \alpha^*_i n_i \tag{20}$$

Which is what is implemented in our algorithm 1.