Volume xx (200y), Number z, pp. 1–10

Reweighting Firefly Samples for Improved Finite-Sample Monte Carlo Estimates

Tobias Zirr, Johannes Hanika, and Carsten Dachsbacher

Karlsruhe Institute of Technology

Abstract

Samples with high contribution but low probability density, often called fireflies, occur in all practical Monte Carlo estimators and are part of computing unbiased estimates. For finite-sample estimates, however, they can lead to excessive variance. Rejecting all samples classified as outliers, as suggested in previous work, leads to estimates that are too low and can cause undesirable artifacts. In this paper, we show how samples can be reweighted depending on their contribution and sampling frequency such that the finite-sample estimate gets closer to the correct expected value and the variance can be controlled. For this, we first derive a theory for how samples should ideally be reweighted and that this would require the probability density function of the optimal sampling strategy. As this PDF is generally unknown, we show how the discrepancy between the optimal and the actual sampling strategy can be estimated and used for reweighting in practice. We describe an efficient algorithm that allows for the necessary analysis of per-pixel sample distributions in the context of Monte Carlo Rendering without storing any individual samples, with only minimal changes to the rendering algorithm. It causes negligible runtime overhead, works in constant memory, and is well-suited for parallel and progressive rendering. The reweighting runs as a fast postprocess, can be controlled interactively, and our approach is non-destructive in that the unbiased result can be reconstructed at any time.

CCS Concepts

•***Computing methodologies*** → *Ray tracing;*

1. Introduction

Recent advancements in the simulation of light transport using Monte Carlo (MC) and Markov Chain Monte Carlo (MCMC) approaches have made photorealistic rendering a lot more efficient. However, depending on the difficulty of light transport, both suffer from high variance, leading to unpredictable results with limited render time. In MC methods, this shows in strong, uncorrelated noise. Our goal is to provide more stable image convergence with a focus on limiting the variance for a given sample budget.

High variance, that manifests in so-called fireflies if no effective estimators are available to sample some high-energy paths reliably, can easily prevent convergence of a rendering, even when other light transport phenomena may already appear converged. Such fireflies produce unpleasant bright spikes on the final result in still images, and distracting temporally unstable noise in animations.

In our paper, we propose a novel method of identifying and reweighting such problematic samples, in order to achieve a user-specified variance level for a given budget of samples per pixel. In essence, we achieve this by separating samples by their magnitudes into a small number of framebuffers rather than one. We demonstrate that carefully constructing and reweighting these framebuffers according to few simple formulas suffices to provide users with an accurate interactive control of the final variance level in post processing, in order to trade bias with variance.

While our work shares similarities with *density-based outlier rejection* [DWR10] who also identify fireflies based on sample (brightness) distributions, we provide a novel theory for more precise handling of problematic samples by reweighting. Moreover, we do not require complicated data structures or storage of individual sample information, but rather work in constant memory. This makes our approach simple to implement and also removes some failure cases of the original outlier rejection technique. Finally, our theory gives probabilities for the statistical assumptions, both in our work and previous work, to hold in practice.

2. Background and Related Work

Monte Carlo integration [MU49] estimates the value of a definite integral with samples $\mathbf{x}_i \in \Omega$ drawn from a probability density function (PDF), which, to reduce variance, is chosen as similar to the integrand as possible (*importance sampling*).

Path Integral Light transport methods compute a solution to the path integral that is defined for the *measurement contribution function* $f(\mathrm{X})$ [Vea98] over the product surface area measure dX:

$$I_j = \int_\Omega h_j(\mathrm{X}) f(\mathrm{X})\,\mathrm{dX}, \text{ with } \mathrm{dX} \equiv \mathrm{d}\mu(\mathrm{X}) = \prod_{i=0}^{k} \mathrm{d}A(\mathbf{x}_i).$$

The path space Ω is the space of all valid paths $\mathrm{X} = (\mathbf{x}_0, \ldots, \mathbf{x}_k)$ of all lengths $k \in [1, \infty)$ that connect a light source to the sensor via interactions at vertex positions $\mathbf{x}_i$; $h_j(\mathrm{X})$ is a pixel filtering function which is non-zero only for the support of the j-th pixel.

Monte Carlo Methods Path tracing [Kaj86] applies MC integration to estimate each I_j independently by sampling paths $\mathrm{X}_i \sim p_j(\mathrm{X}_i)$ starting from the camera where $p_j(\mathrm{X}_i)$ is a product of PDFs for BSDF, light, and lens sampling according to the

sampling strategy:

$$I_j \approx \frac{1}{N}\sum_{i=1}^{N}\frac{h_j(\mathrm{X}_i)f(\mathrm{X}_i)}{p_j(\mathrm{X}_i)}. \tag{1}$$

Other stochastic path generation methods start from the light sources [Arv86] or from both sides with bidirectional path tracing (BDPT) [VG94, LW93] with deterministic connections of subpaths. Similarly, many-lights methods [DKH*14] interpret vertices of light subpaths as virtual light sources illuminating the scene, i.e. connecting them to camera paths.

Local Path Sampling In MC-based rendering techniques, paths are usually sampled according to local PDFs (e.g. BSDF sampling) yielding path PDFs $p(\mathrm{X}) = p(\mathbf{x}_1)p(\mathbf{x}_2|\mathbf{x}_1)\cdots p(\mathbf{x}_k|\mathbf{x}_{k-1})$ which are not globally optimal. For example, for the last vertex (on the light source) the local PDF $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ differs for next event estimation and direct emitter hits.

Clamping and Reweighting Clamping MC samples to a fixed maximum (user-controlled or material-based [WKB12]) is the most simple strategy for controlling variance, particularly of VPLs [Kel97], and is available in most production software. It is well-known that clamping enforces bounded variance and convergence [KK06, Ion08]. [VGG15] reduce bias by adaptively clamping outliers to the expected value of fitted Pareto distributions. [JMD15] try to improve the robustness of MC estimates by computing n lower-sample estimates and taking statistical measures such as the median of the resulting per-pixel estimate distributions. DeCoro et al. [DWR10] construct a k-d tree of MC samples based on their image-space position and brightness, and only accept them as non-outliers once the sample density in each sample's neighborhood surpasses a certain threshold. While this is the work most closely related to ours, we avoid the runtime overhead of neighborhood searches in big dynamically updated sample data structures, and we analyze and handle problematic samples more precisely, preventing some failure cases of their binary classification method.

Image Filtering Detecting low-quality samples and processing them to yield a more pleasing image – typically considering sets of multiple samples – has been done in various ways, e.g. [SW00, SKBF12]. For an overview of more advanced noise filtering and reconstruction methods in combination with adaptive sampling, we refer the reader to [ZJL*15]. Although we also reduce noise in renderings, we do not introduce a new filtering or reconstruction method. In contrast, our contribution is a theory for identifying bad samples and a derivation of a factor for reweighting and reliability.

3. Variance Reduction by Reweighting

When using an unbiased Monte Carlo estimator F to sample light transport through a given pixel, the expected value $\mathbf{E}[F]$ of every sample is exactly the correct radiance for that pixel. However, depending on the quality of the estimator and the difficulty of lighting and geometry in the scene, the variance may be high. In particular, *firefly paths* – paths with high throughput but low probability density – often create bright spots in non-converged finite-sample results: the rare sampling of a path X is compensated by a large upscaling with $1/p(\mathrm{X})$.

Notation	Description
Ω	Space of light transport paths X
$\mathrm{X}, f(\mathrm{X})$	Light path X, and its measurement contribution
F	An estimator for the rendering equation, $\int_\Omega f\,\mathrm{dX}$
$N, \mathcal{X}^N, \mathrm{X}_i$	Sample count, and set of N samples $\mathrm{X}_i \in \Omega$
$\mathbf{E}[F], F^N$	Expected value, and N-sample estimate thereof
$\mathrm{X}_i, p(\mathrm{X}_i)$	Sampled path X_i, and its sampling PDF
S_i	Sample value $\mathrm{S}_i = \frac{f(\mathrm{X}_i)}{p(\mathrm{X}_i)}$ of sampled paths X_i
$p^*(\mathrm{X})$	Ideal PDF where $\frac{f(\mathrm{X})}{p^*(\mathrm{X})} = \mathbf{E}[F] = \text{const.}$
$r^*(\mathrm{X}_i)$	Ratio between ideal and actual PDF, $\frac{p^*(\mathrm{X}_i)}{p(\mathrm{X}_i)} = \frac{S_i}{\mathbf{E}[F]}$
$r^*(\mathrm{S}_i)$	Same, $r^*(\mathrm{S}_i) = \frac{S_i}{\mathbf{E}[F]} = r^*(\mathrm{X}_i)$, independent of X_i
$\mathcal{F}_\kappa$	Set of "firefly" paths X where $r^*(\mathrm{X}) \geq N/\kappa$
$w_\kappa(\mathrm{X})$	Reweighting function for paths $\mathrm{X} \in \mathcal{F}_\kappa$
$\mathcal{N}_\kappa$	$\Omega \setminus \mathcal{F}_\kappa$, i.e. set of paths X sampled better than $\mathcal{F}_\kappa$
$\mathbf{E}_\mathcal{S}[F]$	Conditional expected value on a subspace $\mathcal{S}$: $\mathbf{E}_\mathcal{S}[F] = \int_\mathcal{S} F\,p(\mathrm{X})\,\mathrm{dX} / \int_\mathcal{S} p(\mathrm{X})\,\mathrm{dX}$

Table 1: *Important notation used throughout the paper.*

In the following, we analyze the problem of firefly paths in *finite-sample* MC rendering. Particularly, we demonstrate that we can find weights $w(\mathrm{X})$ for reweighting badly sampled paths X that bring the resulting biased estimate closer to the correct value than an unbiased estimate (Sect. 3.3). For that, Sect. 3.1 introduces an appropriate criterion for badly sampled firefly paths. In Sect. 3.2, we show that a finite-sample estimate F^N from N samples is always greater than the expected value $\mathbf{E}[F]$ and thus incorrect, *if such firefly paths are among the samples*. For further motivation of a precise handling, we show that general rejection of *all* problematic paths, as e.g. done in [DWR10], trivially biases results below $\mathbf{E}[F]$. Table 1 gives an overview of the notation used in our paper.

Spectral Sample Values In our paper, we always assume scalar sample values $\mathrm{S}_i = f(\mathrm{X}_i)/p(\mathrm{X}_i)$. This can be interpreted as both luminance or one of many independent frequency bands.

3.1. Identifying Firefly Paths

Ideally, paths X would be sampled with an *ideal* probability density $p^*(\mathrm{X})$ proportional to $f(\mathrm{X})$, in contrast to the generally suboptimal PDFs $p(\mathrm{X})$ of actual estimators. In this case, each sample would yield the same zero-variance estimate $f(\mathrm{X})/p^*(\mathrm{X}) = \mathbf{E}[F]$. Let r^* be the ratio between an actual sample value $\mathrm{S} = f(\mathrm{X})/p(\mathrm{X})$, and the constant, ideal sample value $\mathbf{E}[F]$:

$$r^*(\mathrm{X}) = \frac{\mathrm{S}(\mathrm{X})}{\mathbf{E}[F]} = \frac{f(\mathrm{X})/p(\mathrm{X})}{f(\mathrm{X})/p^*(\mathrm{X})} = \frac{p^*(\mathrm{X})}{p(\mathrm{X})}. \tag{2}$$

Then, for a realizable estimator F with a suboptimal sampling strategy $p(\mathrm{X})$, the finite-sample estimate F^N after N samples X_i can be written as:

$$F^N = \frac{1}{N}\sum\frac{f(\mathrm{X}_i)}{p(\mathrm{X}_i)} = \frac{1}{N}\sum\frac{f(\mathrm{X}_i)}{\frac{p^*(\mathrm{X}_i)}{r^*(\mathrm{X}_i)}} = \mathbf{E}[F]\cdot\frac{1}{N}\sum r^*(\mathrm{X}_i). \tag{3}$$

Since $p(\mathrm{X})$ is suboptimal, $r^*(\mathrm{X})$ can get arbitrarily large for some badly sampled paths X. Therefore, compared to the correct result $\mathbf{E}[F]$ obtained with ideal sampling, such firefly paths might

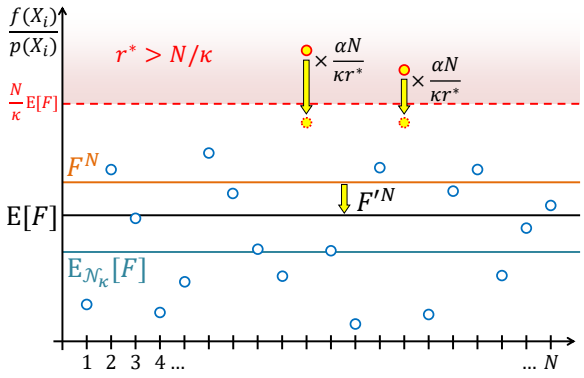

Figure 1: *N-sample MC integration of* $\mathbf{E}[F] = \int f(\mathrm{X})\,\mathrm{dX}$*: The result is either too high* ($F^N > \mathbf{E}[F]$) *if firefly paths are sampled, or too low if not* ($E_{\mathcal{N}_\kappa}[F] < \mathbf{E}[F]$)*. We identify firefly paths by their sampling quality (via* r^**) and show how to downscale their contributions (yellow arrows) to obtain a result* F'^N *closer to* $\mathbf{E}[F]$*.*

increase the result excessively. In particular, as soon as an individual sample S_i significantly exceeds $N \cdot \mathbf{E}[F]$ (or equivalently, $r^*(\mathrm{X}_i) >> N$), F^N can never come close to the correct result, even if all other samples are zero. Note that this does not contradict unbiasedness of the estimate, since the conditional expected value under the assumption that one such firefly path is sampled may differ from the unconditional expected value, which is the correct result. However, we will see that the assumption of sampling firefly paths is in fact reasonable in many cases. Moreover, since it is unusual for all other samples to be zero like in our example, it is obvious that even samples with slightly lower ratios r^* can still lead to high variance. Consequently, unlike binary outlier rejection [DWR10], we build our approach on a continuous criterion for firefly samples.

Firefly Sets We define the generalized sets of fireflies $\mathcal{F}_\kappa$ with respect to a threshold parameter κ, that include more but less problematic paths with increasing κ, is:

$$\mathcal{F}_\kappa = \left\{ \mathrm{X} \in \Omega \;\middle|\; r^*(\mathrm{X}) \geq \frac{N}{\kappa} \right\} \tag{4}$$

Complementary, we define the set of non-firefly paths with respect to κ as $\mathcal{N}_\kappa := \Omega \setminus \mathcal{F}_\kappa$. Fig. 1 illustrates $\mathcal{F}_\kappa$ as the area above the dashed red line. The particularly problematic set of paths in the introductory example with $r^*(\mathrm{X}_i) > N$, i.e. $\kappa = 1$, is denoted as $\mathcal{F}_1$.

3.2. Unbiased Finite-Sample Expected Values

In this section, we show that to be able to reconstruct the correct result from an unbiased *finite-sample* estimate F^N by down-weighting, we in fact need to assume that we sampled firefly paths from each nonempty set $\mathcal{F}_\kappa$. In order to analyze and handle fireflies, we have to refrain from the idealized analysis of expected values that is generally applied to unbiased estimators. Instead, we need to look at actual finite-sample estimates F^N computed from a set of N samples $\mathrm{X}_i \in \mathcal{X}^N \subset \Omega$. For the analysis in this subsection, we re-parameterize the set of fireflies to $\mathcal{F}_\varepsilon$ such that $r^*(\mathrm{X}) \geq N(1+\varepsilon)\ \forall \mathrm{X} \in \mathcal{F}_\varepsilon$ (equivalent to $\kappa = (1+\varepsilon)^{-1}$). Choosing any $\varepsilon > 0$, this allows us to show that unbiased estimates can potentially become arbitrarily large if $\mathcal{F}_\varepsilon \neq \emptyset$.

At least one firefly path We can see that whenever at least one firefly path is sampled, i.e. $\mathcal{X}^N \cap \mathcal{F}_\varepsilon \neq \emptyset$, the proper expected value $\mathbf{E}[F]$ is *never* included in the possible outcomes, and an overly bright pixel value has to occur:

$$F^N = \frac{1}{N} \sum_{\mathrm{X}_i \in \mathcal{X}^N} \mathrm{S}_i = \underbrace{\frac{1}{N} \sum_{\mathrm{X}_i \in \mathcal{N}_\varepsilon} S_i}_{\text{non-firefly paths}} + \underbrace{\frac{1}{N} \sum_{\mathrm{X}_i \in \mathcal{F}_\varepsilon} S_i}_{\text{firefly paths}}.$$

Even when ignoring the contribution of non-firefly paths, we see that F^N is greater than the expected value (with r^* from above):

$$\frac{1}{N} \sum_{\mathrm{X}_i \in \mathcal{F}_\varepsilon} \mathrm{S}_i \geq \frac{1}{N} \sum_{\mathrm{X}_i \in \mathcal{F}_\varepsilon} (1+\varepsilon) N \cdot \mathbf{E}[F] \geq (1+\varepsilon)\mathbf{E}[F], \tag{5}$$

thus $F^N \geq (1+\varepsilon)\mathbf{E}[F]$ is always greater than the correct result $\mathbf{E}[F]$ *under the assumption* that at least one firefly path is sampled.

No firefly paths Conversely, if for any $\kappa \geq 1$ no firefly path from $\mathcal{F}_\kappa \supseteq \mathcal{F}_\varepsilon$ is sampled ($\mathcal{X}^N \cap \mathcal{F}_\varepsilon = \emptyset$) or all are rejected (as in outlier rejection [DWR10]), we cannot expect $\mathbf{E}[F]$ either and obtain a too low value (see Appendix A):

$$\mathbf{E}_{\mathcal{N}_\kappa}[F] = \frac{1}{N} \sum_{\mathrm{X}_i \in \mathcal{N}_\kappa} \frac{\int_{\mathcal{N}_\kappa} \mathrm{S}(\mathrm{X})\, p(\mathrm{X})\,\mathrm{dX}}{\int_{\mathcal{N}_\kappa} p(\mathrm{X})\,\mathrm{dX}} < \mathbf{E}[F], \tag{6}$$

where $\mathbf{E}_{\mathcal{S}}[X] = \frac{\int_{\mathcal{S}} X\, p(\mathrm{X})\,\mathrm{dX}}{\int_{\mathcal{S}} p(\mathrm{X})\,\mathrm{dX}}$ is the conditional expected value. Consequently, we want to avoid rejection of samples and minimize downweighting as much as possible. We will see that on average, less than κ samples from $\mathcal{F}_\kappa$ occur in each set of N samples $\mathcal{X}^N$.

3.3. Reweighting of Firefly Samples

Sect. 3.1 introduced a classification of paths X into firefly groups $\mathcal{F}_\kappa$ of different "severity", i.e. depending on κ. We now show that reweighting firefly paths $\mathrm{X} \in \mathcal{F}_\kappa$ with a weight $w_\kappa(\mathrm{X}) < 1$, taking into account the total sample count N, generally leads to more accurate results than the unbiased, unweighted result estimated with a suboptimal sampling density $p(\mathrm{X})$. The probability of sampling a path from one of these groups can be bounded as:

$$\int_{\mathcal{F}_\kappa} p(\mathrm{X})\,\mathrm{dX} \overset{(2,4)}{\leq} \int_{\mathcal{F}_\kappa} \frac{\kappa}{N} p^*(\mathrm{X})\,\mathrm{dX} \overset{\int_\Omega p^*=1}{<} \frac{\kappa}{N}, \tag{7}$$

that is, out of N samples we expect at most κ firefly paths from $\mathcal{F}_\kappa$ on average, and base our following analyses on this assumption. Of course, there is a non-zero probability of sampling more or less firefly paths, but our basic observations are unaffected by that.

To bring the result of our reweighted estimator F' as close to the expected result $\mathbf{E}[F]$ as possible, we use this assumption that we sample $\lceil\kappa\rceil$ fireflies, i.e. $|\mathcal{F}_\kappa \cap \mathcal{X}^N| = \lceil\kappa\rceil$: The N-sample conditional expected value $\mathbf{E}^N_\kappa[F']$ then is:

$$\mathbf{E}^N_\kappa[F'] = \frac{N-\kappa}{N} \mathbf{E}_{\mathcal{N}_\kappa}[\mathrm{S}_i] + \frac{\kappa}{N} \mathbf{E}_{\mathcal{F}_\kappa}\left[w_\kappa(\mathrm{S}_i)\mathrm{S}_i\right]. \tag{8}$$

We can show that for the weighting $w_\kappa(\mathrm{S}_i) = w_\kappa(\mathrm{X}_i) = \frac{\alpha N}{\kappa r^*(\mathrm{X}_i)}$, we can always find a global variable $\alpha < 1$ that brings this conditional expected value of the weighted estimator F' down to the exact $\mathbf{E}[F]$ (Fig. 1). Appendix B shows that α depends on the ideal PDF $p^*(\mathrm{X})$ and the set of firefly paths $\mathcal{F}_\kappa$, but not the individual path $\mathrm{X} \in \mathcal{F}_\kappa$.

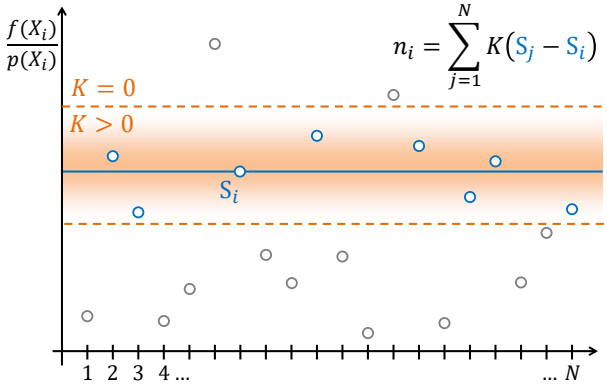

Figure 2: *Counting the number n_i of samples $S_j \in \mathcal{X}^N$ with values similar to S_i by evaluating a kernel function K centered around the value S_i ($K = 1$) and falling off to both sides (until $K = 0$).*

In practice, the ideal density $p^*(X)$ is obviously unknown, and while the exact $\alpha < 1$ if $\mathcal{F}_\kappa \neq \emptyset$ (Sect. 3.3), we conservatively set $\alpha = 1$, which keeps as much energy as possible. This brings the weighted estimate F'^N in-between the overly large unbiased estimate F^N and the correct value $\mathbf{E}[F]$.

3.4. Finite-Sample Expected Variances

We now analyze the effect of reweighting on the variance of the weighted estimator F'. For this we assume $|\mathcal{X}^N \cap \mathcal{F}_\kappa| = \lceil \kappa \rceil$ firefly samples as for the N-sample expected value $\mathbf{E}^N$ in Sect. 3.3. Analogously, we constrain the variance $\mathbf{V}_\mathcal{S}[X] = \mathbf{E}_\mathcal{S}[(X - \mathbf{E}[F])^2]$ to samples in subsets $\mathcal{S} = \mathcal{N}_\kappa$ and $\mathcal{S} = \mathcal{F}_\kappa$.

Unbiased Estimate The N-sample conditional variance $\mathbf{V}^N_\kappa[F]$ of the unbiased estimator F, assuming κ fireflies in N samples, is:

$$\begin{aligned}\mathbf{V}^N_\kappa[F] &= \frac{N-\kappa}{N^2}\mathbf{V}_{\mathcal{N}_\kappa}[S_i] + \frac{\kappa}{N^2}\mathbf{V}_{\mathcal{F}_\kappa}[S_i] \\ &> \frac{\kappa}{N^2}\mathbf{V}_{\mathcal{F}_\kappa}\left[\frac{N}{\kappa}\mathbf{E}[F]\right] = \frac{\kappa}{N^2}\left(\frac{N}{\kappa}\mathbf{E}[F] - \mathbf{E}[F]\right)^2 \\ &= \frac{\mathbf{E}[F]^2}{\kappa}\left(1 - \frac{2\kappa}{N} + \frac{\kappa^2}{N^2}\right).\end{aligned} \tag{9}$$

We observe that, as long as there are firefly paths ($\mathcal{F}_\kappa \neq \emptyset$), the variance of unbiased estimates is effectively worse than an arbitrarily large constant (depending on κ). This means that the image remains highly unstable and results will be far from converged. Note that this does not contradict convergence of Monte Carlo Integration, which is theoretically proven even if variance is unbounded.

Reweighted Estimate Reweighting provides a means to control the variance of the weighted estimator F' with fireflies:

$$\begin{aligned}\mathbf{V}^N\left[F'\right] &= \frac{N-\kappa}{N^2}\mathbf{V}^C_{\mathcal{N}_\kappa}[S_i] + \frac{\kappa}{N^2}\mathbf{V}^C_{\mathcal{F}_\kappa}\left[\frac{N}{\kappa}\frac{S_i}{r^*(X_i)}\right] \\ &< \frac{1}{N}\mathbf{V}^C_{\mathcal{N}_\kappa}[S_i] + \frac{1}{\kappa}\mathbf{E}[F]^2.\end{aligned} \tag{10}$$

The first term is a bounded variance divided by N, thus adhering to ideal MC convergence, the second is a constant divided by κ, which equals the variance of a κ-sample MC estimate with tightly bounded variance (i.e. no fireflies). Thus, reweighting, which is controlled by choosing κ, allows us to set a *constant* upper bound for the variance of the end result.

3.5. Summary of the Analysis

In this section, we showed that MC estimates can suffer from severe variance caused by bounded and unbounded firefly samples, the latter of which can cause unbiased finite-sample estimates to become arbitrarily large and thus prevent *practical* convergence.[†] We demonstrated that such samples are needed for a correct result and can be reweighted to improve the estimate. Finally, reweighting can be used to bound variance to a user-specified amount.

4. Variance Estimation by Sample Counting

In Sect. 3 we showed how a comparison of PDFs (computing r^*) can be used to indicate and reweight problematic firefly paths. Unfortunately, we generally have no way of knowing the ideal PDF p^*. In this section, we show two complementary ways of estimating r^* for reweighting the sample values S_i which we obtain from sampled paths X_i in a finite sample set $\mathcal{X}^N$ without knowledge of $p^*(X_i)$. Note that the individual path X_i that produced S_i is irrelevant and we use $r^*(X_i)$ and $r^*(S_i)$ interchangeably (see Eq. (2)).

In Sect. 4.1 we determine if a sample S_i is likely to be in a certain firefly set $\mathcal{F}_\kappa$ based on the number of samples $S_j \in \mathcal{X}^N$ with values similar to S_i (as illustrated in Fig. 2), resembling the approach by DeCoro et al. [DWR10]. In particular, we provide bounds on the reliability of such a classification. From that, Sect. 4.2 derives an estimation of $r^*(S_i)$ and corresponding reweighting based on sample counts. Sect. 4.3 describes a complementary approach to estimating $r^*(S_i)$ based on a comparison of S_i to a conservative estimate of $\mathbf{E}[F]$ that is continuously updated while reweighting. This additionally recovers some unproblematic darker samples and results in more uniform noise levels comparable to naïve sample value clamping, but without its excessive loss of brightness.

While this section might suggest that the storage of full sample sets $\mathcal{X}^N$ is required, Sect. 5 details an implementation with minimal runtime overhead and in constant memory bounds; in contrast to previous work [DWR10, Ion08] it does not require storing any individual sample values.

4.1. Firefly Classification by Sample Count

In this section, we will see that the likelihood that a sample S_i is not a firefly increases exponentially with the number n_i of samples $S_j \in \mathcal{X}^N$ with values similar to S_i, and that S_i most likely belongs to the set $\mathcal{N}_{n_i}$, i.e. $\kappa = n_i$. Intuitively, paths in $\mathcal{N}_{n_i} = \Omega \setminus \mathcal{F}_{n_i}$ no longer harm convergence starting at a total sample count of n_i. Fortunately, we can easily obtain this occurrence count n_i of similar samples from $\mathcal{X}^N$, e.g. similarly to DeCoro et al. [DWR10] by applying a kernel $K(S_j - S_i)$ centered around S_i, where $K(S_i - S_i) = 1$ falling off to both sides, yielding $n_i = \sum_j K(S_j - S_i)$ (compare Fig. 2). Following these observations, we find weights for all S_i by looking at the definition of $\mathcal{F}_{n_i}$ in Eq. (4) and setting $r^*(S_i)$ to its lower bound in the set criterion, i.e. $r^*(X_i) \approxeq N/n_i$.

[†] While theoretical convergence is guaranteed for all admissible $p(X) > 0$ where $f(X) > 0$, $p(X)$ may come arbitrarily close to 0 and $S(X)$ may thus be unbounded, trivially destroying any practically obtained mean value.

We now look at samples S_i and analyze into which sets they fall.

Bounded Fireflies Eq. (7) indicates that for any κ, we expect less than κ firefly samples $S_j \in \mathcal{F}_\kappa$ where $S_j > N/\kappa\,\mathbf{E}[F]$. Therefore, if we encounter $n_i \geq \kappa$ samples similar to S_i, there is a good chance from a maximum likelihood perspective that $S_i \notin \mathcal{F}_\kappa \subseteq \mathcal{F}_{n_i}$ (S_i is not a firefly with respect to κ), but $S_i \in \mathcal{N}_{n_i} \subseteq \mathcal{N}_\kappa$. This begs the question of how certain this assumption is: Since the samples S_j are independent, each sample has an equal probability less than n_i/N of belonging to $\mathcal{F}_{n_i}$ and we expect n_i's random variable N_i to be binomially distributed. Therefore, the chance that the sample S_i is actually a worse firefly than those in $\mathcal{F}_{n_i}$ can be assessed by bounding the binomial distribution function: If samples like S_i occur n_i times even though their paths are actually sampled less sufficiently than paths in $\mathcal{N}_{n_i}$ by at least a factor $\beta > 1$, and belong to $\mathcal{F}_{n_i/\beta}$, the probability that we find n_i or more such samples can be bounded by the Chernoff inequality (see Appendix C):

$$P(N_i \geq n_i \mid S_i \in \mathcal{F}_{n_i/\beta}) \leq \beta^{-n_i C(\beta)}, \tag{11}$$

where $C(\beta)$ is a constant greater than 0 for all $\beta > 1$. Consequently, the probability of misclassifying S_i into $\mathcal{N}_{n_i}$ vanishes with an increasing occurrence count n_i of sample values S_j similar to S_i. We can confirm the work by DeCoro et al. [DWR10] in that given a κ, an occurrence count of $n_i = \kappa$ is a good indicator that a corresponding sample S_i is well-behaved: It is likely that $S_i \leq 2\frac{N}{\kappa}\mathbf{E}[F]$ and thus does not destroy the estimate as discussed in Sect. 3.2: The probability for this is at least $1 - 1/2^\kappa$. In contrast to their work, we will only discard fireflies in $\mathcal{F}_1$ and find more optimal weights for the remaining problematic samples.

Unbounded Fireflies Also according to Eq. (7) we expect up to one sample in $\mathcal{F}_1$, i.e. one arbitrarily large sample S_i. For this sample we cannot estimate $r^*(S_i)$ as no similar sample exists and thus n_i is always one. Hence we cannot know if it belongs to $\mathcal{F}_1$ or even to any worse $\mathcal{F}_\kappa$ where $\kappa < 1$, and thus we cannot come up with a sensible reweighting and have to discard this sample. For the remaining samples, however, we expect smaller values $S_j \leq N\mathbf{E}[F]$.

4.2. Ideal PDF Ratio Estimation and Re-Weighting

Our first approach to estimating the ratio r^* between the actual PDF p and the ideal PDF p^* is based on these observations: We classify a sample S_i as $\in \mathcal{N}_{n_i}$ if we encounter n_i similar samples in $\mathcal{X}_{n_i}$. *Then, we ensure the necessary reweighting by determining the ideal ratio estimate as* $r^*(S_i) \approxeq N/n_i$ *from the set criterion of* $\mathcal{N}_{n_i}$*, since* S_i *still likely is in less problematic firefly sets* $\mathcal{F}_\kappa$ *for some* $\kappa > n_i$. As stated by Eq. 11, these assumptions are reasonable in a maximum-likelihood sense.

As noted previously, we must expect *one* potentially unbounded firefly sample that cannot be reweighted reliably. Also, in case of small occurrence count n_i, the reliability of the classification is low: If we discard all samples S_i where n_i does not exceed a minimum occurrence count $\kappa_{\min}$, Eq. (11) does not give a strong guarantee for successfully identifying and discarding all unbounded firefly samples for low $\kappa_{\min}$. In practice, we do not want to discard more than that *one* sample to retain smooth handling of firefly samples and thus image quality. Instead, we average the n_i of the immediate neighbor pixels *for that decision only*, which lowers the probability of misclassification *as if* $\kappa_{\min} = 9$ while keeping $\kappa_{\min} = 1$.

All other S_i can be reweighted by $w_c(S_i) := \frac{N}{\kappa}\frac{n_i - \kappa_{\min}}{N} = \frac{n_i - \kappa_{\min}}{\kappa}$, iff $n_i < \kappa + \kappa_{\min}$, as discussed in Sect. 3.3. We subtract $\kappa_{\min}$ to smoothly fade in samples as would be the case without any discarding. We can then choose a global parameter κ to control the variance in our image as analyzed in Sect. 3.4. We denote the corresponding ratio estimate based on n_i as $r_c^*(S_i) := \frac{N}{n_i - \kappa_{\min}}$.

4.3. Value-Based Lower Bounds

While sample occurrence counts provide a good means of detecting problematic samples, they are over-conservative, treating high- and low-energy samples alike: if only few similar values are found, they will be reweighted. Some of these still fit into the variance bounds of Sect. 3.4 due to their low value. In fact, images generated using naïve clamping of values to a global maximum that results in comparable variance bounds sometimes deviates less from the correct result in darker regions. However, such clamping completely breaks high dynamic ranges, leading to a severe overall loss of brightness in the image (see results Sect. 6.5 and Fig. 9).

Therefore, we would like to compute a less over-conservative sample weight $w(S_i)$ that behaves similarly to naïve value clamping for low values. In order to quantify low values, we additionally incrementally compute a lower bound of the expected value $\mathbf{E}_{\min}[F]$ from samples during reweighting. According to Eq. (2), this then lets us derive an *upper bound* $r_v^*(S_i) := S_i/\mathbf{E}_{\min}[F]$ for clamping $r^*(S_i)$, based on the sample value S_i rather than on n_i as in r_c^*.

To incrementally estimate $\mathbf{E}_{\min}[F]$, we consider our sample set $\mathcal{X}^N$ sorted by sample values in ascending order. We then assemble our reweighted estimate F'^N incrementally, denoting the incomplete estimate $F_i'^N$ including all samples up to i as:

$$F_1'^N = \frac{1}{N} w_c(S_1)\, S_1 = \frac{1}{N}\frac{N}{\kappa r_c^*(S_1)} S_1, \tag{12}$$

$$F_i'^N = F_{i-1}'^N + \frac{1}{N} w(S_i)\, S_i, \tag{13}$$

where $w(S_i) = \frac{N}{\kappa\, r^*(S_i)}$. This allows us to continuously update $\mathbf{E}_{\min}[F] = F_{i-1}'^N$ for computing $r_v^*(S_i)$ of every sample S_i. We can then safely use r_v^* as a conservative upper bound for setting $r^*(S_i) = \min\{r_c^*(S_i), r_v^*(S_i)\}$. Thus, we regain energy that would be kept by value clamping, while retaining the dynamic range of reliable high-energy transport.

Note that in contrast to naïve clamping, the resulting variance is still proportional to the individual pixel values. In cases where a minimum global variance is acceptable or desired in all image regions, $\mathbf{E}_{\min}[F]$ can simply be clamped to a minimum value by an artistic noise level parameter.

5. The Cascaded Sample Count Framebuffer

Previous outlier handling work relies on the storage of sample sets [Ion08], and complex sample-space tree data structures [DWR10]. This can incur significant storage and runtime overhead. In the following, we provide an efficient way of implementing the methods from Sect. 4.2 and Sect. 4.3 with only a small constant number of framebuffers that is logarithmic in the maximum sample luminance. The overhead to the renderer is negligible, since the only change will be splatting each sample to two

framebuffers rather than to one at a time, and evaluating a simple weighting formula based on sample brightness. We will see that this enables us to compute per-sample weights that closely match our reweighting theory in a simple postprocess, while we are still able to recover the unbiased result at any time. Note that all parameters κ, $\kappa_{\min}$, and $\mathbf{E}_{\min}[F]$ of our approach can thus be applied to the final renders in real time, enabling interactive tweaking of image variance and bias.

5.1. Split Sample Splatting

In order to implement the variance estimation described in Sect. 4 and to be able to reconstruct sample counts in post processing, we split the framebuffer of our renderer into a small constant number of framebuffers, a *cascade*, where each framebuffer B_j accumulates samples with brightnesses around the value b^j in the range $[b^{j-1}, b^{j+1}]$. The base b is a small constant that depends on the desired precision of variance control; in all our experiments we use $b = 8$. The ranges of all B_j overlap, and samples are split between buffers by a specific weighting scheme detailed in this section. The total number of required buffers then depends on the maximum sample brightness $\mathrm{S}_{\max}$:

$$j \leq \log_b \mathrm{S}_{\max}. \tag{14}$$

While $\mathrm{S}_{\max}$ is generally unknown, we assume (multiple) importance sampling to usually prevent sample brightnesses that exceed the brightest light source. In our experiments all cascades had empty B_j above that threshold. In cases where local importance sampling is also suboptimal (e.g. due to complex materials), more buffers or clamping are viable options. Note that the number of buffers only grows *logarithmically* with $\mathrm{S}_{\max}$.

Each sample S_i generated by the renderer is then splatted into the two buffers B_j and B_{j+1} in the cascade it falls into, where $b^j \leq \mathrm{S}_i$:

$$j = \lfloor \log_b \mathrm{S}_i \rfloor. \tag{15}$$

Before splatting S_i into B_j and B_{j+1} it is pre-multiplied by the weights α_j and α_{j+1}, respectively:

$$\alpha_j = \frac{b^j/\mathrm{S}_i - 1/b}{1 - 1/b}, \qquad \alpha_{j+1} = 1 - \alpha_j. \tag{16}$$

This hyperbolic weighting is basically an interpolation between buffers B_j and B_{j+1} which splits each sample between the two such that we can count the number of samples with similar brightnesses as described in the following, while the sum of all buffers B_j still yields the unbiased result. For $j < 0$, we simply splat S_i fully to B_0.

5.2. Per-Framebuffer Sample Counting in Post

As required for the estimation of $r^*(\mathrm{S}_i)$ in Sect. 4.2, the specific weights α_j allow us to estimate the number of times that samples of similar brightness have occurred and have therefore been accumulated in certain buffers B_j, even when no individual samples were stored. This is due to the fact that for any S_i, the *normalized* sum of its two destination buffers B_j and B_{j+1} (i.e., each divided by their maximum brightnesses $b^j \geq \alpha_j S_i$ and b^{j+1}) is:

$$\frac{\alpha_j \mathrm{S}_i}{b^j} + \frac{\alpha_{j+1}\mathrm{S}_i}{b^{j+1}} = \frac{b^j/\mathrm{S}_i - 1/b}{1 - 1/b}\frac{\mathrm{S}_i}{b^j} + \frac{1 - b^j/\mathrm{S}_i}{1 - 1/b}\frac{\mathrm{S}_i}{b^{j+1}} \tag{17}$$

$$= \frac{1 - \mathrm{S}_i/b^{j+1} + \mathrm{S}_i/b^{j+1} - 1/b}{1 - 1/b} = 1. \tag{18}$$

Let us first assume that all samples S_i in a consecutive pair of buffers B_j and B_{j+1} had the same value. Then, the normalized sum $NB_j/b^j + NB_{j+1}/b^{j+1}$ would simply compute the number of occurrences n_i. We need to multiply by N since our framebuffers are already normalized with respect to sample count. To handle the general case of samples with varying values, we use the sum

$$n_j = \frac{NB_{j-1}}{b^{j-1}} + \frac{NB_j}{b^j} + \frac{NB_{j+1}}{b^{j+1}}, \tag{19}$$

to count the number n_j of samples stored in B_j with values around b^j (weighted according to a kernel K like in Fig. 2).

5.3. Application to the Handling of Firefly Samples

In the following, we detail three ways of using our practical sample counting to apply the theory built by the previous sections.

5.3.1. Reweighting (Sect. 4.2 and 4.3 in Practice)

We reweight the samples accumulated in one buffer B_j at a time, as described in Sect. 4. In order to be able to keep track of a conservative estimate of the expected value as described in Sect. 4.3, we compose the buffers in ascending order of j, i.e. sample brightness. We use the n_j computed for B_j to estimate $r_c^*(\mathbf{B}_j)$ for all samples S_i accumulated in B_j, substituting n_i with n_j in the formula for $r_c^*(\mathrm{S}_i)$.

Value-Based Lower Bounds Eq. (19) implies that fully included sample values may be off by a factor of b, when B_{j-1} contains many samples, but B_j contains only few. In order to evenly match the resulting variance to the one caused by the value-based lower bounds from Sect. 4.3, we also use this maximum sample contribution b^j for computing $r_v^*(\mathrm{S}_i) = b^j/\mathbf{E}_{\min}[F]$, where we continuously update $\mathbf{E}_{\min}[F]$ with the weighted composition of all preceding framebuffers.

Outlier Rejection In order to reliably identify unbounded outliers that cannot be sensibly reweighted, we not only compute n_j based only on the individual pixel values in B_j, but additionally compute an average $\bar{n}_j$ over the n_j of the immediate pixel neighborhood, i.e. 3×3 pixel blocks. We use this average $\bar{n}_j$ as a rejection criterion with $\kappa_{\min} = 1$, and then use the individual n_j for reweighting. This gives us the reliability *as if* $\kappa_{\min} = 9$ as per Sect. 4.2, while minimizing correlation artifacts and retaining image quality.

5.3.2. Expected Value Estimation (Sect. 3.3 in Practice)

We also provide results that demonstrate the numerical accuracy of our theory, where we set $\kappa = 1$ in order to get as close to the expected value as possible, and compare the resulting image to a converged reference (Fig. 3b). We find that the reweighted image is indeed closer to the reference than the unbiased result, and any other reweighting with different values of κ. For these experiments, we adjust the value-based lower bounds for accuracy, rather than even variance, and use the value $r_v^*(\mathrm{S}_i) = b^j/\mathbf{E}_{\min}[F]$ from Sect. 4.3.

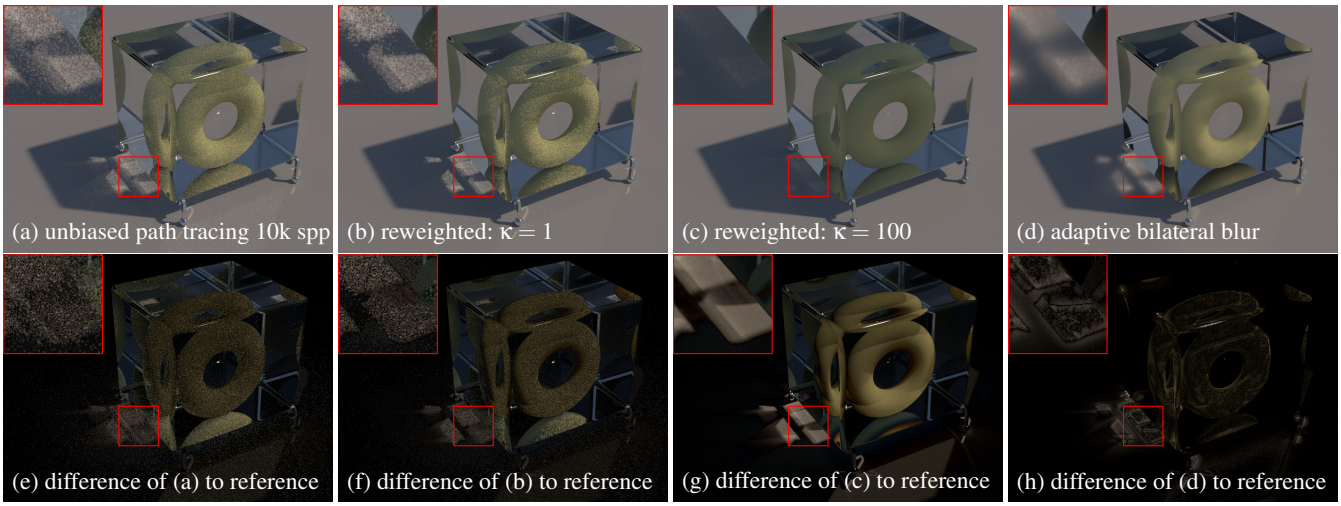

Figure 3: *Top row: a diffuse torus embedded in glass; bottom row: difference images to an 1.000.000 spp reference. (a) simple path tracing with next event estimation. (b) the same paths as in a), but reweighted to closely match the expected value as per Sect. 3.3 ($\kappa = 1$). Note how the difference is better than in the unbiased and the aggressively reweighted case. (c) reweighting to enforce bounded variance for a given sample budget, resulting in darker but noise-free images ($\kappa = 100$). (d) using the weights to compute an adaptive radius for a simple bilateral blur, guided by the reliable samples. This gives smooth and energy-conserving images that resemble those produced by photon mapping.*

5.3.3. Use in Image Filtering

By definition, r^* tells us the value a firefly sample *should ideally* take. This information can also be used for other purposes than simple downweighting. For instance, we can also spread the energy across the local neighborhood with an adaptive filter akin to [SW00]. For that, we determine the standard deviation of a Gaussian such that the peak value of the filter is exactly our reweighting factor; the spread of the filter thus automatically adapts to how well certain phenomena were sampled. Since the spread can be large and we want to avoid clearly visible bias by blurring over geometric edges, we employ a cross-bilateral filter, similar to virtual flash photography [MJL*13]. We retain energy by normalization using the masked filter weights. To demonstrate the idea, one result can be seen in Fig. 3d. Note that noise reduction filters are outside our core contribution, and we show this as a motivation for incorporating the information into more sophisticated filtering approaches.

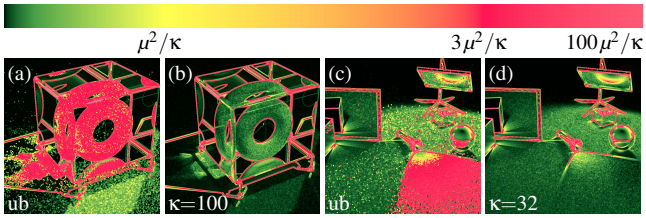

Figure 4: *False-color visualization of a naïve 5×5 pixel window variance estimate, for (a,b) Fig. 3 and (c,d) Fig. 8. In accordance with Section 3.4, reweighting scales the variance down to about $1\frac{\mu^2}{\kappa}$ of the windowed mean μ, such that it stays below $3\frac{\mu^2}{\kappa}$.*

6. Results and Discussion

We implemented our method in Mitsuba [Jak10]. All results were rendered using forward path sampling with next event estimation which easily cause a great variety of firefly samples in many scenes, giving us plenty of opportunity to demonstrate our technique. Note that our approach is not tied to any specific estimator. Other estimators like bi-directional path tracing also generate fireflies in many scenarios, which can be handled in the same way.

6.1. Validation of our Theory

In the following we validate the accuracy of our theoretical analyses, both visually and numerically. Fig. 3 shows the torus scene used in [JM12]. The scene features specular caustics of a small sun and clear sky light through a refractive glass cube. Caustic paths connecting the camera to the sun are therefore rarely sampled, resulting in severe fireflies initially, and strong noise afterwards, persisting for several hundred thousand samples per pixel.

Expected Value Fig. 3 shows results for several methods of reweighting: The unbiased result (a) differs more from the ground truth than the reweighted result (b) which brings the individual pixels closer to the expected value by setting $\kappa = 1$. Reweighting with $\kappa = 100$ (c) suppresses the noise of unconverged phenomena by further downweighting, while retaining the energy of well-sampled phenomena.

Expected Variance We also find that in accordance with our theory in Section 3.4, setting $\kappa = 100$ reduces caustic variance down to 1% of $\mathbf{E}[F]^2$: The corresponding false-color visualization of the reweighted result in Fig. 4(d) displays its variance compared to the unweighted result (c). The colors indicate that reweighting reliably shrinks variance into the range of 1% of the squared local mean.

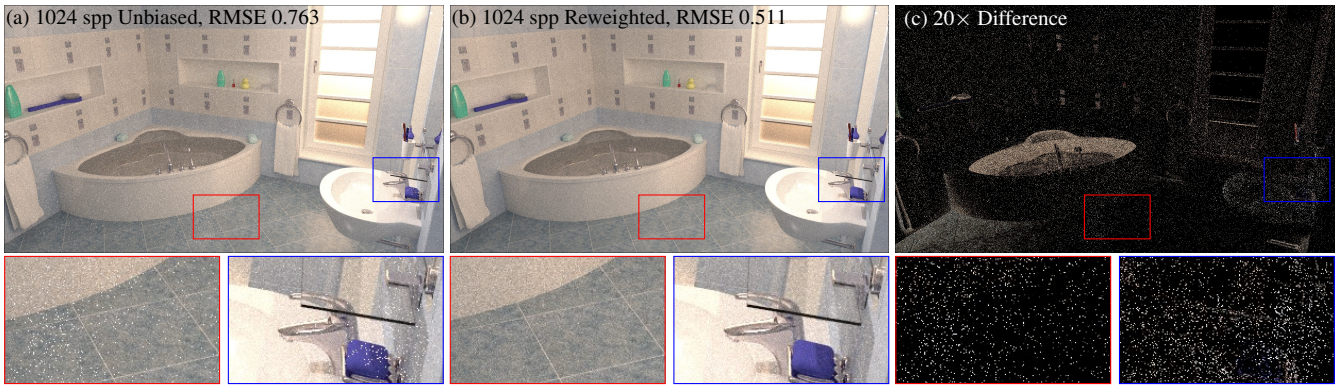

Figure 5: *A Bathroom rendered using path tracing: (a) 1024 samples per pixel (spp) unbiased rendering (b) equal-time 1024 sample rendering using our reweighting of "firefly" samples to bound variance, and (c) a difference image ×20. Given a fixed sample budget, we can determine a scaling factor by how much each high-variance path needs to be scaled down in order to yield results with a given variance. Particularly, that variance can be controlled interactively in a post processing step, and we show that our biased method can be used (Parameter $\kappa = 20$) to achieve lower root mean-squared error (RMSE) compared to unbiased results.*

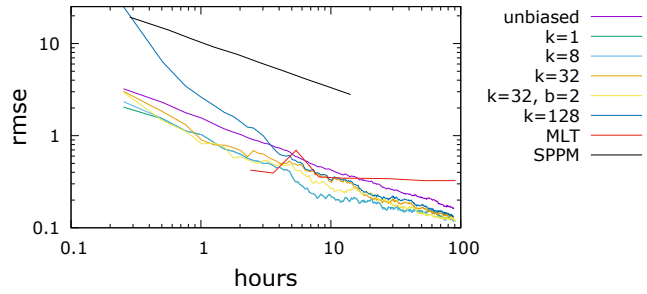

Figure 6: *Root mean squared error for different κ and cascade base $b = 2,8$ in the bathroom scene; for comparison we show Metropolis light transport (MLT) [Vea98] and stochastic progressive photon mapping (SPPM) [HJ09].*

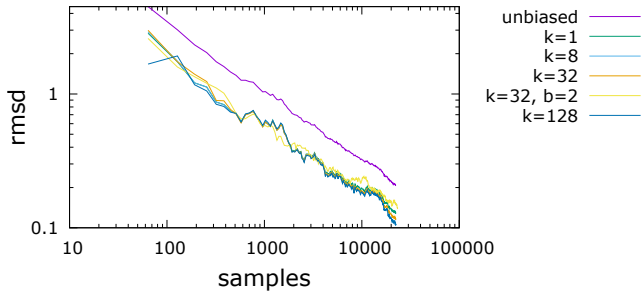

Figure 7: *Root mean squared difference using two different seeds for different κ, b in the bathroom scene. As expected, image stability is increased by greater values of κ.*

Handling of Fireflies Fig. 5 shows a realistic example of physically-based rendering with a lot of glossy, near-specular and specular light transport, and interior lighting with small light bulbs. As expected, fireflies caused by rarely sampled specular paths that would otherwise render the result useless are removed, while more glossy caustics are retained; the overall noise is reduced.

MSE The corresponding plot in Fig. 6 shows that also numerically, reweighting can bring images closer to the ground truth. Note how in accordance with theory, $\kappa = 1$ is closest. The graph also compares results using our common choice of framebuffer cascade base $b = 8$ to a finer separation base $b = 2$, which multiplies storage cost by three. Note that the quality is not substantially affected.

6.2. Temporal Coherence

As already seen in Fig. 4, higher values of κ reduce variance. This property is particularly desirable to guarantee temporal coherence when rendering animations. In order to numerically evaluate the improvement in stability, Fig. 7 plots the deviations of two identical frames when rendered with different random seeds. Reweighting significantly increases the robustness of unconverged results. In order to see how these numerical results also translate to visually more stable images, please refer to the supplemental video.

6.3. Storage Cost and Performance

We use a framebuffer cascade separation base of $b = 8$ for classifying samples into framebuffers in all our results except the MSE plots in Fig. 6, where we also show $b = 2$ for comparison. None of our $b = 8$-based results required more than six framebuffers in the cascade, which corresponds to a dynamic range of 32.768. An additional framebuffer would increase the dynamic range to 262.144. On an Intel i7-5820K-6-core CPU at 3.3 GHz, we could not reliably measure a performance overhead during rendering and found it negligible in the standard deviation of timings. The reweighting was done in real-time in OpenGL shaders on an NVIDIA Quadro M5000, taking less than 5 ms.

6.4. Effect on Complex Light Transport

Fig. 8 shows the flashlight scene featuring light transport of various difficulty levels. (a) The small light bulb inside the flashlight causes high variance for light reflected by the green specular reflector. (b) Reweighting removes this variance, while the direct illumination remains unaffected (no glass in front of the bulb). Note how the green caustic of the reflector is attenuated with increasing distance and variance. Another small light bulb in the black box on the right casts soft caustics through the glossy reflector suspended above.

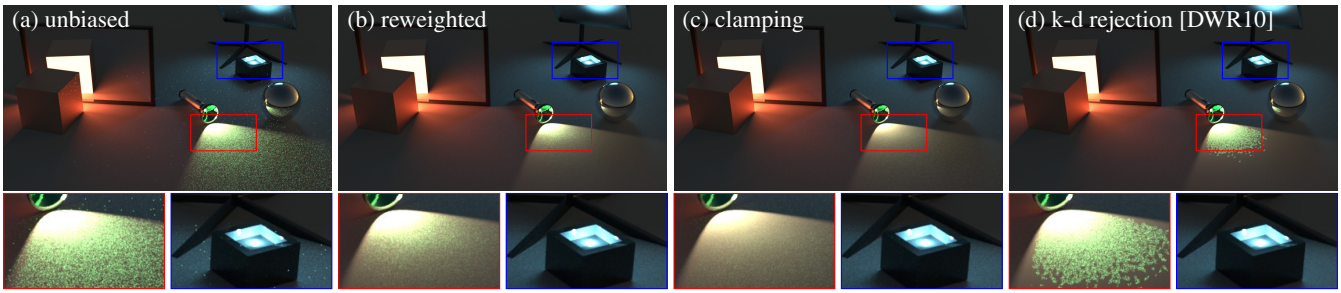

Figure 8: *A flashlight with a small light bulb reflecting off a green-tinted mirror. The scene also features two easier caustics: a specular reflection (red box) and a glossy bounce light (blue). (a) 10.000 unbiased forward path tracing samples. (b) Same paths reweighted to control variance ($\kappa = 32$), resulting in a darker but stable image. Notice how well-sampled phenomena such as the reflected area light and light reflected by the glossy diffuser are retained, while fireflies are removed and lamp caustics are attenuated with increasing variance. (c) Naïvly clamping sample values to 250. (d) Outlier rejection by DeCoro et al. [DWR10] using k-d trees. Their binary rejection criterion badly handles smooth changes in variance and their 6x6 pixel kernel introduces MCMC-like correlation artifacts.*

In contrast to specular caustics, these glossy caustics are sampled better by next event estimation and reweighting correctly retains them. Finally, a large area light is reflected by the mirror on the left. The resulting caustics are sufficiently sampled by forward path sampling, and reweighting correctly retains them, too.

6.5. Comparison to Related Techniques

Naïve value clamping as shown in Fig. 8(c) yields good results comparable to ours in dark regions, but naturally fails to retain the dynamic range in bright regions. Fig. 9 details this loss of brightness by clamping (a,c), which is preserved by our method (b,d).

A comparison to the outlier rejection algorithm proposed by DeCoro et al. [DWR10] is provided in Fig. 8(d). While variance is also reduced, their algorithm fails to adapt to the smooth change of variance in front of the reflector and introduces significant correlation artifacts, due to their naïve use of larger pixel neighborhoods.

6.6. Introduced Bias

Fig. 10 shows very difficult light transport, rendered with a simple path tracer with next event estimation. Our algorithm (correctly) identifies most of the samples as firefly paths and thus removes a lot of the energy, resulting in much darker images. Here, in a way pure reweighting methods are stretched to their natural limits: high variance everywhere forces them to strike a balance with large bias. Smaller κ remove less energy at the price of higher variance.

6.7. Additional Applications

Fig. 3(d) uses the weights computed for $\kappa=32$ to determine a corresponding bilateral Gaussian filter radius, as described in Sect. 5.3.3, instead of downweighting. This partially blurs the results, but conserves the overall energy – similar to those produced by consistent density estimation methods such as Photon Mapping.

7. Conclusions and Future Work

We showed how analyzing the core of the Monte Carlo method yields detailed knowledge about paths leading to high variance.

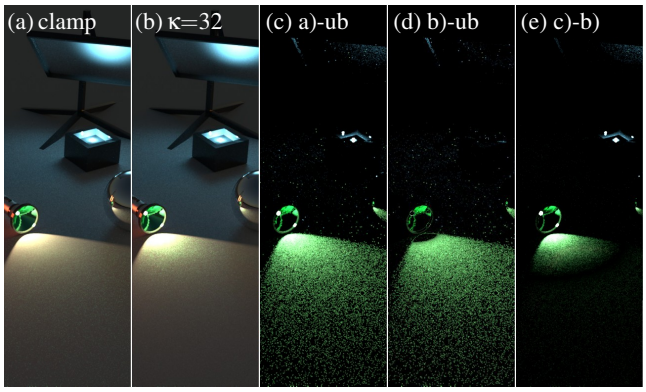

Figure 9: *Bias introduced by (a) naïve clamping compared to (b) reweighting, as in Fig. 8. Note the complete loss of dynamic range in the (c) difference of the naïvly clamped and the unbiased result. (d) Reweighting preserves the dynamic range. (e) The bias of clamping is emphasized by the difference to the reweighted result.*

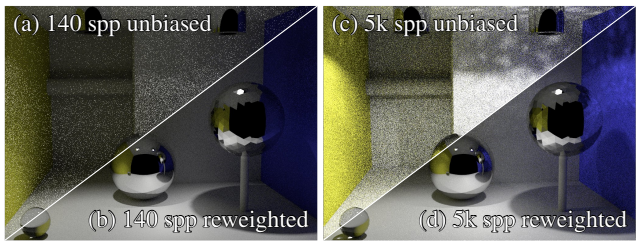

Figure 10: *(a) 140 forward path tracing samples, (b) reweighted with $\kappa=100$. (c) 5000 unbiased samples still require reweighting. (d) 5000 samples reweighted, $\kappa=100$. As a simple path tracer performs very badly in this scene, samples are downweighted a lot and much energy is lost until variance bounds are met.*

Even when using a set of estimators with MIS, such paths can occur if not all important phenomena can be sampled efficiently. By classifying samples into a cascade of firefly buffers, it is possible to blend together results with a desired variance at any point. We built a thorough theory that roots this reweighting in a comparison of actual path sampling probability densities to a hypothetical

ideal estimator. We showed that downscaling firefly contributions accordingly yields a biased but consistent estimator ($\kappa = 1$), which can converge to the real expected value faster than the unbiased version. Optionally, variance can be bounded more rigorously at the cost of bias ($\kappa > 1$). Additionally, we showed an application where the width of a simple reconstruction filter was estimated by our theory. We believe that this knowledge can be leveraged in the context of more sophisticated filters and adaptive sampling [ZJL*15].

Acknowledgements: The first author is funded by the Heidelberg Institute for Theoretical Studies, HITS gGmbH.

Appendix A: Unbiased Finite-Sample Expected Value

We derive the expected value in case that no firefly path is sampled. In Eq. 20 we plug in $\mathbf{E}[F] = f/p^*$ and use $\int_{\mathcal{N}} \mathrm{dX} = \int_{\Omega} \mathrm{dX} - \int_{\mathcal{F}} \mathrm{dX}$:

$$\mathbf{E}_{\mathcal{N}}[F] = \frac{1}{N} \sum_{\mathrm{X}_i \in \mathcal{N}} \frac{\int_{\mathcal{N}} \frac{f(\mathrm{X})}{p(\mathrm{X})} p(\mathrm{X})\,\mathrm{dX}}{\int_{\mathcal{N}} p(\mathrm{X})\,\mathrm{dX}} \tag{20}$$

$$= \frac{\int_{\Omega} f(\mathrm{X})\,\mathrm{dX} - \int_{\mathcal{F}} \mathbf{E}[F] p^*(\mathrm{X})\,\mathrm{dX}}{\int_{\mathcal{N}} p(\mathrm{X})\,\mathrm{dX}} = \frac{\mathbf{E}[F](1 - \int_{\mathcal{F}} p^*(\mathrm{X})\,\mathrm{dX})}{\int_{\mathcal{N}} p(\mathrm{X})\,\mathrm{dX}}$$

$$< \mathbf{E}[F] \frac{1 - N/\kappa \int_{\mathcal{F}} p(\mathrm{X})\,\mathrm{dX}}{\int_{\mathcal{N}} p(\mathrm{X})\,\mathrm{dX}} \tag{21}$$

$$= \mathbf{E}[F] \frac{1 + N/\kappa\,(-1 + (1 - \int_{\mathcal{F}} p(\mathrm{X})\,\mathrm{dX}))}{\int_{\mathcal{N}} p(\mathrm{X})\,\mathrm{dX}} \tag{22}$$

$$\leq \mathbf{E}[F]\Big(-N/\kappa + 1 + N/\kappa\Big) = \mathbf{E}[F]. \tag{23}$$

Appendix B: Optimal Reweighting

In order to determine the optimal α, we re-arrange Eq. (8) for the conditional expected value $\mathbf{E}^N_\kappa[F']$ of the weighted estimator F':

$$\mathbf{E}^N_\kappa[F'] = \frac{N-\kappa}{N} \mathbf{E}_{\mathcal{N}_\kappa}[\mathrm{S}_i] + \frac{\kappa}{N} \mathbf{E}_{\mathcal{F}_\kappa}\left[\frac{\alpha N}{\kappa r^*(\mathrm{X}_i)} \mathrm{S}_i\right]$$

$$= \mathbf{E}[F]\left(\frac{N-\kappa}{N} \frac{\int_{\mathcal{N}_\kappa} p^*(\mathrm{X})\,\mathrm{dX}}{\int_{\mathcal{N}_\kappa} p(\mathrm{X})\,\mathrm{dX}} + \alpha\right) \stackrel{!}{=} \mathbf{E}[F] \tag{24}$$

$$\Leftrightarrow \alpha = 1 - \frac{N-\kappa}{N} \frac{\int_{\mathcal{N}_\kappa} p^*(\mathrm{X})\,\mathrm{dX}}{\int_{\mathcal{N}_\kappa} p(\mathrm{X})\,\mathrm{dX}} \geq 1 - \int_{\mathcal{N}_\kappa} p^*(\mathrm{X})\,\mathrm{dX}. \tag{25}$$

Therefore, $\alpha \geq$ (ideal probability mass of all firefly paths in $\mathcal{F}_\kappa$).

Appendix C: Variance Estimation Probability Bound

Eq. 11 gives an upper bound for the binomially-distributed probability of misclassifying samples, based on [AG89]:

$$F\left(N - n_i, N, 1 - \frac{n_i}{\beta N}\right) \leq \exp\left(-N D\left(\frac{n_i}{N} \parallel \frac{n_i}{\beta N}\right)\right), \tag{26}$$

$$D(a \parallel p) = a \log \frac{a}{p} + (1-a) \log \frac{1-a}{1-p}. \tag{27}$$

Since $\left(\frac{N-n_i/\beta}{N-n_i}\right)^{N-n_i} = \left(1 + \frac{n_i - n_i/\beta}{N-n_i}\right)^{N-n_i} \leq \exp\left(n_i - \frac{n_i}{\beta}\right)$, we get:

$$\exp\left(-n_i \log \beta - (N-n_i) \log \frac{N-n_i}{N-n_i/\beta}\right) \tag{28}$$

$$\leq \exp\left(-n_i \log \beta + n_i - \frac{n_i}{\beta}\right) \leq \beta^{-n_i C(\beta)}. \tag{29}$$

The constant $C(\beta) = 1 + \frac{1/\beta - 1}{\log \beta}$ is greater than 0 for all $\beta > 1$ and monotonously increasing towards infinity. E.g., $C(\beta = 2) \approxeq 0.2787$.

References

[AG89] ARRATIA R., GORDON L.: Tutorial on large deviations for the binomial distribution. *Bulletin of Mathematical Biology 51*, 1 (1989), 125–131. 10

[Arv86] ARVO J.: Backward ray tracing. In *Developments in Ray Tracing, ACM SIGGRAPH Course Notes* (1986), pp. 259–263. 2

[DKH*14] DACHSBACHER C., KRIVÁNEK J., HASAN M., ARBREE A., WALTER B., NOVÁK J.: Scalable realistic rendering with many-light methods. *Computer Graphics Forum 33*, 1 (2014), 88–104. 2

[DWR10] DECORO C., WEYRICH T., RUSINKIEWICZ S.: Density-based outlier rejection in Monte Carlo rendering. *Computer Graphics Forum (Proc. Pacific Graphics) 29*, 7 (Sept. 2010). 1, 2, 3, 4, 5, 9

[HJ09] HACHISUKA T., JENSEN H. W.: Stochastic progressive photon mapping. *ACM Transactions on Graphics (Proc. SIGGRAPH) 28*, 5 (Dec. 2009), 141:1–141:8. 8

[Ion08] IONIDES E. L.: Truncated importance sampling. *Jrnl. of Computational and Graphical Statistics 17*, 2 (2008), 295–311. 2, 4, 5

[Jak10] JAKOB W.: Mitsuba Renderer, 2010. Source available at http://www.mitsuba-renderer.org. 7

[JM12] JAKOB W., MARSCHNER S.: Manifold exploration: a Markov chain Monte Carlo technique for rendering scenes with difficult specular transport. *ACM Transactions on Graphics (Proc. SIGGRAPH) 31*, 4 (2012), 58:1–58:13. 7

[JMD15] JUNG J. W., MEYER G., DELONG R.: Robust statistical pixel estimation. *Comput. Graph. Forum 34*, 2 (May 2015), 585–596. 2

[Kaj86] KAJIYA J. T.: The rendering equation. *Computer Graphics (Proc. of SIGGRAPH)* (1986), 143–150. 1

[Kel97] KELLER A.: Instant radiosity. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1997), SIGGRAPH '97, ACM Press/Addison-Wesley Publishing Co., pp. 49–56. 2

[KK06] KOLLIG T., KELLER A.: *Illumination in the Presence of Weak Singularities*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 245–257. 2

[LW93] LAFORTUNE E. P., WILLEMS Y. D.: Bi-directional path tracing. In *Proc. of Conference on Computational Graphics and Visualization Techniques* (1993), pp. 145–153. 2

[MJL*13] MOON B., JUN J. Y., LEE J., KIM K., HACHISUKA T., YOON S.-E.: Robust image denoising using a virtual flash image for Monte Carlo ray tracing. *Computer Graphics Forum 32*, 1 (2013). 7

[MU49] METROPOLIS N., ULAM S.: The Monte Carlo method. *Journal of the American Statistical Association 44*, 247 (1949), 335–341. 1

[SKBF12] SCHWENK K., KUIJPER A., BEHR J., FELLNER D.: Practical noise reduction for progr. stochastic ray tracing with perceptual control. *IEEE Computer Graphics and Applications 32*, 6 (2012), 46–55. 2

[SW00] SUYKENS F., WILLEMS Y.: Adaptive filtering for progressive monte carlo image rendering. In *Proc. Intl. Conference in Central Europe on Computer Graphics)* (Plzen, Czech Republic, February 2000), pp. 220–227. 2, 7

[Vea98] VEACH E.: *Robust Monte Carlo methods for light transport simulation*. PhD thesis, Stanford University, 1998. AAI9837162. 1, 8

[VG94] VEACH E., GUIBAS L.: Bidirectional estimators for light transport. In *Proc. Eurographics Workshop on Rendering* (1994), pp. 147–162. 2

[VGG15] VEHTARI A., GELMAN A., GABRY J.: Pareto Smoothed Importance Sampling. *ArXiv e-prints* (July 2015). 2

[WKB12] WALTER B., KHUNGURN P., BALA K.: Bidirectional lightcuts. *ACM Trans. Graph. 31*, 4 (July 2012), 59:1–59:11. 2

[ZJL*15] ZWICKER M., JAROSZ W., LEHTINEN J., MOON B., RAMAMOORTHI R., ROUSSELLE F., SEN P., SOLER C., YOON S. E.: Recent advances in adaptive sampling and reconstruction for Monte Carlo rendering. *Computer Graphics Forum 34*, 2 (2015), 667–681. 2, 10