A Statistical Approach to Monte Carlo Denoising

HIROYUKI SAKAI, Technische Universität Wien (TU Wien), Austria
CHRISTIAN FREUDE, Technische Universität Wien (TU Wien), Austria
THOMAS AUZINGER, Institute of Science and Technology Austria, Austria
DAVID HAHN, Technische Universität Wien (TU Wien), Austria
MICHAEL WIMMER, Technische Universität Wien (TU Wien), Austria

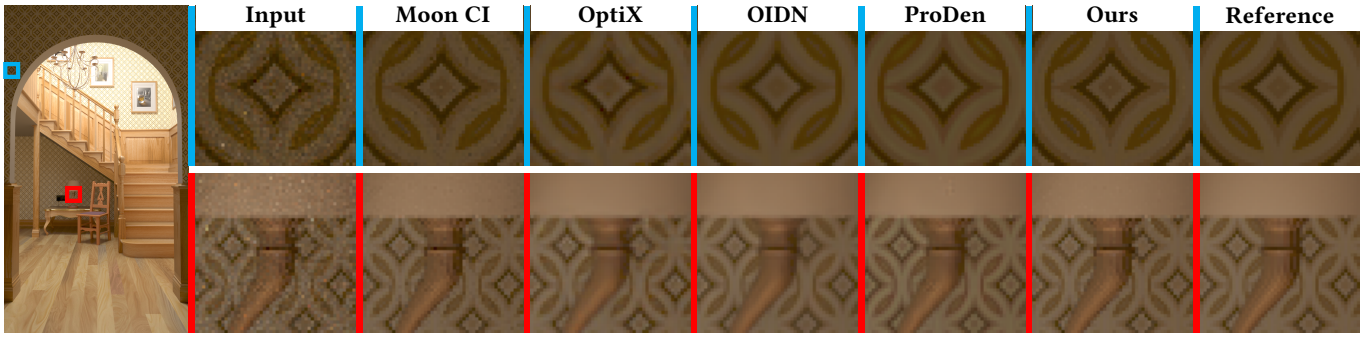

Fig. 1. Our statistical denoising method, using online estimates of sample statistics, achieves image quality comparable to current state-of-the-art methods, without any computation-heavy prior training. We compare our denoiser to the approach by Moon et al. [2013] ("Moon CI"), NVIDIA OptiX AI-Accelerated Denoiser ("OptiX"), Intel Open Image Denoise (OIDN), and progressive denoising [Firmino et al. 2022] ("ProDen"). These results have been generated using 256 samples per pixel (SPP) with the following denoising times; Moon CI: 35.3 ms, OptiX: 85.5 ms, OIDN: 19.5 ms, ProDen: 1834.9 ms, ours: 28.0 ms.

The stochastic nature of modern Monte Carlo (MC) rendering methods inevitably produces noise in rendered images for a practical number of samples per pixel. The problem of denoising these images has been widely studied, with most recent methods relying on data-driven, pretrained neural networks. In contrast, in this paper we propose a *statistical* approach to the denoising problem, treating each pixel as a random variable and reasoning about its distribution. Considering a pixel of the noisy rendered image, we formulate fast pair-wise statistical tests—based on online estimators—to decide which of the nearby pixels to exclude from the denoising filter. We show that for symmetric pixel weights and normally distributed samples, the classical Welch t-test is optimal in terms of mean squared error. We then show how to extend this result to handle non-normal distributions, using more recent confidence-interval formulations in combination with the Box-Cox transformation. Our results show that our statistical denoising approach matches the performance of state-of-the-art neural image denoising without having to resort to any computation-intensive pretraining. Furthermore, our approach easily generalizes to other quantities besides pixel intensity, which we demonstrate by showing additional applications to Russian roulette path termination and multiple importance sampling.

CCS Concepts: • **Computing methodologies** → **Ray tracing**.

Additional Key Words and Phrases: Monte Carlo rendering, path tracing, denoising, image filtering, statistics

SA Conference Papers '24, December 3–6, 2024, Tokyo, Japan

ACM ISBN 979-8-4007-1131-2/24/12
https://doi.org/10.1145/3680528.3687591

ACM Reference Format:
Hiroyuki Sakai, Christian Freude, Thomas Auzinger, David Hahn, and Michael Wimmer. 2024. A Statistical Approach to Monte Carlo Denoising. In *SIGGRAPH Asia 2024 Conference Papers (SA Conference Papers '24), December 3–6, 2024, Tokyo, Japan.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3680528.3687591

1 INTRODUCTION

Monte Carlo (MC) integration is ubiquitous in modern photorealistic rendering. The basic idea is to estimate an integral (pixel intensity) by a (weighted) average of random samples. However, compared to deterministic quadrature rules, MC integration is stochastic in nature and must therefore contend with uncertainty. This stochastic uncertainty manifests as variance—i.e., noise—affecting every sample taken during rendering.

The field of MC *denoising* originated from the idea of applying traditional image filters from the image-processing community to reduce noise in rendering. These filters essentially average multiple estimates (i.e., pixels) in image space, thereby decreasing variance. This approach can be highly effective in situations where the estimates are the same (or similar) in the limit, i.e., in regions of an image that show smooth color gradients. However, in cases of significant difference, such as along feature edges, filtering potentially introduces unwanted bias, often apparent as excessive blurring or color bleeding.

Denoising therefore must achieve a good bias–variance trade-off. In contrast to traditional image filtering, recent denoising approaches rely on neural networks to learn a mapping from a noisy input image, along with auxiliary information (in the form of so-called

G-buffers containing low-noise ground-truth scene information), to a low-error approximation of the desired image.

Departing from these pretrained machine learning approaches, in this paper, we propose a *general statistical framework* for denoising in MC rendering. Specifically, we establish a theoretical connection between minimizing mean squared error (MSE) for pair-wise symmetric weights and Welch's t-test for normally distributed samples. Using more recent results from the statistics literature, we then generalize this approach to reduce the assumptions on the distributions. As a practical implementation of this framework, we demonstrate an image-space denoising scheme, building upon the well-known joint bilateral filter. Our approach tracks online statistics of the per-pixel samples generated by a state-of-the-art MC renderer. These statistics describe the distribution of samples and allow us to draw statistically valid conclusions about which estimates among neighboring pixels should be combined in order to achieve a near-optimal trade-off between noise and bias. Our statistics-based filter effectively avoids blurring over legitimate features, including lighting effects (such as shadows or caustics) that are not present in the G-buffers (and would therefore be over-blurred by existing denoisers). All these statistics can be estimated online during rendering. Our filter runs on the GPU, taking around 30 ms for 1280×720 resolution images on commodity hardware.

Apart from image denoising, we demonstrate that our approach can generalize to other applications, such as Russian roulette (RR) path termination and multiple importance sampling (MIS). Our general approach has an important advantage relating to these generalizations: We do not rely on preexisting training data, therefore we can produce low-error estimates of any quantity of interest, not only the radiance produced by the renderer. For instance, we denoise the per-bounce incident radiance in our RR example or estimates for which sampling strategy outperforms the others (formalized as *win rates*) during MIS. For both quantities, obtaining sufficient training data for neural denoising approaches would be quite challenging.

In summary, we present the following main contributions:

- a general statistical framework that puts MC rendering into a statistical context, alongside a theoretical analysis on minimizing mean squared error in the setting of pair-wise testing,
- an image-space implementation of this framework that produces low-error estimates for any quantity of interest in the context of MC rendering,
- and applications of this denoising approach to standard image denoising, RR path termination, and MIS.

We provide additional materials (e.g., source code) for our paper at https://www.cg.tuwien.ac.at/StatMC.

2 RELATED WORK

The stochastic nature of Monte Carlo (MC) rendering introduces error in the form of variance. To reduce this error, one can either compute more samples or use some form of noise reduction, e.g., adaptive sampling or filtering [Huo and Yoon 2021; Zwicker et al. 2015]. A common approach is a posteriori noise reduction or *denoising*, which operates on the samples generated by MC rendering. Many methods work with (image-space) pixel estimates and range from classical denoising approaches [Overbeck et al. 2009; Xu and Pattanaik 2005] to more recent neural-network-based methods [Back et al. 2022; Firmino et al. 2022].

Classical approaches usually adapt and apply some form of filter while trying to find a balance between noise reduction and introduced bias (usually apparent in the form of excessive blurring). Many use image-filtering kernels, e.g., Gaussian [Rousselle et al. 2011], (joint) bilateral [Li et al. 2012; Liu et al. 2018b; Mara et al. 2017; Park et al. 2013; Rousselle et al. 2013; Sen and Darabi 2012; Xu and Pattanaik 2005; Zheng and Liu 2018], non-local means [Delbracio et al. 2014; Moon et al. 2013; Rousselle et al. 2012; Vicini et al. 2019], wavelet [Dammertz et al. 2010; Kalantari and Sen 2013; Overbeck et al. 2009; Schied et al. 2017, 2018], or non-local Bayes [Boughida and Boubekeur 2017] filters. Other methods build on diffusion [McCool 1999] or higher-order regression [Bauszat et al. 2011; Bitterli et al. 2016; Liu et al. 2018a; Moon et al. 2014, 2015, 2016; Yuan and Zheng 2017, 2018]. In addition to the noisy input image, many of the mentioned approaches incorporate additional auxiliary features like albedos and normals, as well as additional statistics like the corresponding variances. Among these, joint bilateral filtering with auxiliary features is most closely related to our denoising approach.

In a similar direction to our statistics-based approach, Sen and Darabi [2012] presented a method called random parameter filtering (RPF), which uses histograms of sample vectors to calculate the mutual information between scene features and the random parameters of the MC process in order to adjust the weights of a joint bilateral filter and reduce the noise stemming from these random parameters. Even though their approach achieves good results at low sample counts, its computation becomes prohibitively expensive with an increasing number of samples. This downside was addressed by Park et al. [2013] by interpolating sparsely computed mutual information, which reduces but not fully eliminates the scaling issues for high sample counts and requires several seconds or multiple minutes to denoise a low-SPP image. In contrast, our approach is completely independent of the number of samples and requires only a few milliseconds to compute. A similar idea is to represent sample distributions as histograms [Boughida and Boubekeur 2017; Delbracio et al. 2014]. However, histograms require additional memory and their accuracy depends on the number and sizing of bins, which must be chosen a priori. In contrast, we only use a few statistical moments to compactly store information about sample distributions, which can also be efficiently updated with each new sample.

Li et al. [2012] surpass the denoising quality of RPF by using Stein's unbiased risk estimator (SURE) to select which one out of a discrete set of joint bilateral filters (of different spatial scales) minimizes the error. Similarly, Rousselle et al. [2013] use SURE to select among filters that differ in robustness to noise and sensitivity to image details. Following this general idea of "meta"-approaches, Zheng et al. [2021] combine multiple denoisers, whereas Firmino et al. [2022] apply denoising only when beneficial for convergence. In contrast, our approach does not have the overhead of computing multiple filters, or corresponding derivatives for error estimation, before the final reconstruction. We instead derive filter kernels directly from the pixel statistics and use statistical testing to ensure that only estimates with similar distributions are combined, leading

to a fast one-pass approach, which could also improve the results of meta-denoisers by acting as an additional base input.

Mara et al. [2017] presented a real-time approach using a pipeline consisting of several joint bilateral filter stages, and showed results comparable to non-local means (NLM) [Rousselle et al. 2012] and regression-based approaches [Bitterli et al. 2016; Moon et al. 2014]. Their method is specifically tailored for real time and builds on a custom renderer with special handling of direct and indirect illumination. Our approach uses a more straightforward pipeline, which can be easily integrated into any conventional MC renderer with little overhead while offering similar performance during filtering.

Similar to ours, some methods use confidence intervals [Moon et al. 2013, 2016; Sen and Darabi 2012] to exclude pixels from combination during filtering if their corresponding statistics differ significantly. These approaches use confidence intervals in specific subparts of their fairly complex pipelines, which are tailored specifically to MC image denoising. Similar statistical concepts were used by Back et al. [2023], who first applied the concept of uncorrelated statistics to denoising MC renderings. In contrast, we use statistical tests at the core of our efficient denoising pipeline, including higher-order central moments to relax the requirement of normally distributed samples implied by commonly used confidence intervals.

Neural-network-based approaches represent the current state-of-the-art in MC denoising, e.g., NVIDIA OptiX AI-Accelerated Denoiser (OAADN), which is based on Chaitanya et al.'s work [2017], or Intel Open Image Denoise (OIDN) [Áfra 2024]. Both approaches take a noisy image and so-called G-buffers (such as normals or albedos) as inputs. The neural networks then map those inputs to a low-error approximation of the ground truth. For this purpose, the networks have to be trained on numerous input–reference image pairs, so that an effective mapping can be established. Reconstructing image features that suffer from high variance and are not present in the G-buffers, such as lighting effects, is a difficult problem. In these cases, the networks must rely solely on the noisy image to differentiate between legitimate features and noise. Furthermore, the performance of such approaches hinges on the training and available data. Generally, there are no guarantees concerning their convergence behavior. For instance, OIDN [Áfra 2024] includes images at different degrees of convergence into the training to incorporate a notion of convergence. This stands in contrast to several non-neural methods that provide consistency [Back et al. 2023; Bitterli et al. 2016; Moon et al. 2013; Rousselle et al. 2013]. To alleviate this fundamental problem, Firmino et al. [2022] train an additional neural network that predicts per-pixel mixing factors to reduce the weight of the biased reconstruction in favor of the converging input. They use confidence intervals to ensure convergence by limiting the neural mixing weights such that the biased contribution vanishes as the variance decreases with increased sample counts.

Our approach, in contrast, does not require any training, as it directly uses the statistics gathered during rendering to infer relationships between pixels for filtering. Furthermore, we provide a theoretical guarantee concerning the consistency of our filter. We also show in our results that we can denoise quantities other than the radiance reaching the camera. In this paper, we always keep all filtering parameters constant as the number of samples per pixel (SPP) increases to demonstrate the convergence behavior of our method. In the future, automatic parameter tuning as done in neural denoising or integration of our method into meta-denoising strategies could further improve denoising performance.

3 BACKGROUND AND NOTATION

In this section, we summarize the statistical concepts and methods on which we build our general statistical framework in §4 and establish the connection to the common denoising problem in Monte Carlo rendering. As a starting point, we consider the well-known rendering integral [Kajiya 1986], where each pixel I_i in an image is formed as

$$I_i = \int_{\Omega_i} W_i(\omega) L'(\mathbf{x}, \omega)\, d\omega, \tag{1}$$

where W_i describes the sensor response and L' is the radiance incident on the i-th pixel, which covers the solid angle Ω_i from the camera location $\mathbf{x}$. This radiance must satisfy Kajiya's rendering equation throughout a virtual scene. Each pixel intensity is commonly approximated by Monte Carlo (MC) integration:

$$I_i \approx \frac{1}{n_i} \sum_{k=1}^{n_i} \frac{f_k}{p_k}, \tag{2}$$

where f_k refers to evaluations of the integrand in Eq. (1) for n_i randomly drawn samples from Ω_i with probability density p_k.

We now move on to a more abstract statistical interpretation of the rendering process outlined above. In particular, we interpret each pixel as a statistical estimator $\hat{\theta}_i(X_1, \ldots, X_{n_i})$ for the (unknown) ground-truth value θ_i (in this case I_i, although the same idea applies to other estimators), where we view each sample X_k as a random variable (i.e., a distribution from which the rendered sample f_k/p_k is drawn). We generally assume unbiased estimators, i.e., $\mathbb{E}[\hat{\theta}_i] = \theta_i$.

Using results from descriptive statistics, our core idea is to formulate statistical tests to decide which estimators are sufficiently similar so that they can be combined during denoising. In general, we employ the following descriptive statistics of the sample distribution: mean (μ), and the central moments of orders 2–4 ($M_l,\ l \in 2, 3, 4$), or their standardized variants, variance ($\sigma^2 = M_2$), skewness ($M_3/M_2^{3/2}$), and kurtosis (M_4/M_2^2) [Kenney and Keeping 1951]. The central moments are defined as

$$M_l = \frac{1}{n_i} \sum_k (X_k - \bar{X})^l, \tag{3}$$

where $\bar{X}$ is the sample mean. Crucially, all these statistics can be computed *online* [Meng 2015], building on the classical result by Welford [1962], i.e., by updating them one sample at a time during rendering without having to store all samples in memory. In our implementation, we use central moments up to the third order.

3.1 Denoising Estimators

In this work, we consider general denoising filters that construct the denoised estimator $\tilde{\theta}_j$ as a convex combination of (noisy) input estimators $\{\hat{\theta}_i\}$:

$$\tilde{\theta}_j = \sum_i w_{ij} \hat{\theta}_i, \tag{4}$$

where w_{ij} is the weight assigned to estimator $\hat{\theta}_i$; all weights must be non-negative, and $\sum_i w_{ij} = 1$. The key question now is: how to find

appropriate (sparse) weights w_{ij}? In our approach, we split these weights into two parts, Eq. (10): a base filter, such as the well-known joint bilateral filter, and a novel pair-wise statistical *membership function* m_{ij}, which decides which combinations of estimators are admissible.

Formally, the mean squared error (MSE) of the combined estimator $\tilde{\theta}_j$ can be decomposed into variance and bias

$$\mathrm{MSE}(\tilde{\theta}_j, \theta_j) = \mathbb{E}[(\tilde{\theta}_j - \theta_j)^2] = \mathrm{Var}(\tilde{\theta}_j) + \mathrm{Bias}(\tilde{\theta}_j, \theta_j)^2, \quad (5)$$

where θ_j refers to the *estimand*, i.e., the unknown ground-truth value for the j-th estimator. Note that even though we assume unbiased estimators $\hat{\theta}_j$, their filtered counterparts $\tilde{\theta}_j$ generally contain some bias. The variance and bias are given by

$$\mathrm{Var}(\tilde{\theta}_j) = \sum\nolimits_i w_{ij}^2 \, \mathrm{Var}(\hat{\theta}_i(n_i)), \text{ and} \quad (6)$$

$$\mathrm{Bias}(\tilde{\theta}_j, \theta_j) = \sum\nolimits_i w_{ij} \, \mathrm{Bias}(\hat{\theta}_i(n_i), \theta_j), \quad (7)$$

where $\mathrm{Bias}(\hat{\theta}_i, \theta_j) = \mathbb{E}(\hat{\theta}_i) - \theta_j$.

The overall goal of MC denoising in this context is to find weights that minimize the total MSE, or a similar error metric, i.e., achieve an optimal trade-off between variance (noise) and bias, across all denoised estimators (i.e., the whole image):

$$\{w_{ij}^*\} = \underset{\{w_{ij}\}}{\arg\min} \sum\nolimits_j \mathrm{MSE}(\tilde{\theta}_j, \theta_j). \quad (8)$$

However, this optimization problem cannot be addressed directly, as the ground-truth values are of course unknown, and we must therefore work with noisy estimates of variance and bias.

4 STATISTICAL FILTERING FRAMEWORK

In this section, we develop our statistical filtering framework. Considering an abstract set of estimators $\{\hat{\theta}_i\}$, we ask under which conditions combining a subset of these estimators improves mean squared error. Here, we approach the inherent noise-to-bias trade-off from a statistical perspective, formulating a *membership function* (m), fulfilling the requirements summarized in §4.1. We show how this approach relates to hypothesis testing in statistics, before describing the specifics of our image-space implementation in §5.

4.1 Problem Statement

Considering the general denoising problem introduced in §3, we formulate membership functions that decide which combinations of estimators are admissible during denoising, i.e., under which conditions the combination is likely to improve image quality. In particular, we require the following important properties:

(a) *Pair-wise* evaluation: for any pair of estimators $(\hat{\theta}_i, \hat{\theta}_j)$, m must be defined as a function of these estimators' statistics only, i.e., $m_{ij} = m(\mathcal{S}(\hat{\theta}_i), \mathcal{S}(\hat{\theta}_j))$, where $\mathcal{S}(\hat{\theta}_i)$ denotes descriptive sample statistics, such as $\mu_i, \sigma_i^2, M_{3,i}, M_{4,i}, \ldots$, of estimator $\hat{\theta}_i$.

(b) *Online* statistics: all these statistics $\mathcal{S}$ must be computable by an online algorithm, i.e., by updating their value one sample at a time during Monte Carlo (MC) rendering. Together with the pair-wise property, this requirement ensures that all components of our filtering pipeline can be implemented efficiently in terms of both parallel execution and memory usage.

(c) *Symmetry*: we require that $m_{ij} = m_{ji}$. In practice, symmetry of weights enforces energy preservation during filtering. Relaxing this requirement sometimes produces visually more pleasing results around bright outliers ("fireflies") at the cost of losing some overall brightness and possibly slightly higher mean squared error (MSE).

(d) *Convergence*: In order to guarantee convergence of the denoised result $\tilde{\theta}_i \to \theta_i \ \forall i$ with increasing sample size, we require that the membership function satisfies

$$\mathrm{Var}(\hat{\theta}_i) \to 0 \implies m_{ij} = 0 \text{ if } \|\theta_i - \theta_j\| > 0. \quad (9)$$

In other words, as the variance approaches zero, the membership function must exclude estimator j from the combination with estimator i if there is any difference in their estimands, which would introduce bias.

(e) *Identity*: An estimator does not introduce additional bias to itself; we therefore set $m_{ii} = 1$ by definition.

Finally, we can state our main problem as finding membership functions m_{ij} that satisfy properties (a–e) and deliver the best possible variance–bias trade-off within these constraints.

4.2 Our Approach

Here, we describe our general framework for formulating statistics-based membership functions respecting the aforementioned requirements. We then use these functions to address the denoising problem, Eq. (8), by defining the filter weights in Eq. (4), as follows:

$$\boxed{w_{ij} = \frac{\rho_{ij} \, m_{ij}}{\sum_i \rho_{ij} \, m_{ij}}.} \quad (10)$$

The first component of the weights, ρ_{ij}, provides the option to integrate existing filters based on a priori available information into our system. The main reasons for doing so are: (1) to limit the size of the filter kernel, thereby improving the runtime performance by limiting the amount of required membership-function evaluations; (2) to be able to build upon well-known existing approaches; and (3) to include additional low-noise a priori information available from the renderer (*G-buffers*, e.g., normals or albedos in image space). Note that while a priori information is often useful to preserve some features due to geometry or textures, it is inherently useless for other features that only manifest themselves through sampling (e.g., shadows or caustics). In this regard, our membership function and this a priori weight complement each other, incorporating both empirical and a priori information. In our implementation (§5), we set ρ_{ij} to represent a joint bilateral filter.

Existing filters are highly effective at reducing variance. The key task of the membership function m_{ij} is therefore to limit bias by excluding estimators from the filter whose estimands differ significantly. Following the requirements stated in §4.1, we now consider the MSE minimization problem, Eq. (8), for a pair of input estimators $\hat{\theta}_i, \hat{\theta}_j$. These estimators are combined with weight w into denoised estimators $\tilde{\theta}_i = w\hat{\theta}_i + (1-w)\hat{\theta}_j$ and $\tilde{\theta}_j = w\hat{\theta}_j + (1-w)\hat{\theta}_i$. Note that we enforce symmetry of the weighting here, $w_{ij} = w_{ji} = (1-w)$. Assuming unbiased input estimators, the bias of their combination is $\mathrm{Bias}(\tilde{\theta}_i, \theta_i) = \mathbb{E}(\tilde{\theta}_i - \theta_i) = (w-1)\theta_i + (1-w)\theta_j$, and analogously for $\mathrm{Bias}(\tilde{\theta}_j, \theta_j)$. Minimizing the sum of the error terms, Eq. (5), for

both denoised estimators then reads:

$$\begin{aligned} w^* = \arg\min_w \big(& w^2 \mathrm{Var}(\hat{\theta}_i) & + (1-w)^2 \mathrm{Var}(\hat{\theta}_j) & + \\ & \big((w-1)\theta_i & + (1-w)\theta_j \big)^2 & + \\ & w^2 \mathrm{Var}(\hat{\theta}_j) & + (1-w)^2 \mathrm{Var}(\hat{\theta}_i) & + \\ & \big((w-1)\theta_j & + (1-w)\theta_i \big)^2 & \big). \end{aligned} \tag{11}$$

For this objective, we can find the optimum by setting the derivative with respect to the weight to zero and solving for w^* (assuming at least some uncertainty, i.e., $\mathrm{Var}(\hat{\theta}_i) + \mathrm{Var}(\hat{\theta}_j) > 0$):

$$w^* = \frac{2(\theta_i - \theta_j)^2 + \mathrm{Var}(\hat{\theta}_i) + \mathrm{Var}(\hat{\theta}_j)}{2\left((\theta_i - \theta_j)^2 + \mathrm{Var}(\hat{\theta}_i) + \mathrm{Var}(\hat{\theta}_j)\right)}. \tag{12}$$

Note, however, that we do not know ground-truth values, or estimator variances, and therefore work with (noisy) estimates of these quantities instead, producing noisy results for w^*. Consequently, the trivial choice of setting $m_{ij} = 1 - w^*$ and $\rho_{ij} = 1$ does not yield satisfactory results. Our approach is therefore to use an existing smoothing filter for ρ_{ij} and enforce a binary membership function such that

$$m_{ij} = 1 \text{ if } (1 - w^*) > \gamma,\ 0 \text{ otherwise.} \tag{13}$$

Here, γ is a threshold that determines how discriminative the membership function is. In this way, we effectively use the pair-wise optimal weight as a test statistic (similar to those used in statistical hypothesis testing) and prevent the estimators from being filtered together if the introduced bias (i.e., the difference of their means) is too large relative to the sum of their variances. In our supplementary document, we show that this test is equivalent to Welch's t-test, $t < \gamma_w$ with $\gamma_w = \sqrt{1/(2\gamma) - 1}$. Our results use a critical value from Student's t-distribution of $\gamma_w = t_{1-\alpha/2,\nu}$, with the significance level $\alpha = 0.005$ and $\nu = n_i + n_j - 2$, i.e., the upper bound for the degrees of freedom approximated by the Welch–Satterthwaite equation [1946].

Welch's t-test is related to the confidence interval for the difference of two normally distributed means, see, for example, Eq. (18) in the paper by Curto [2023]. Consequently, we can relax the normality assumption by choosing a different confidence-interval formulation from the literature. Various formulations are available that consider not only the sample variance but also higher-order statistics, such as skewness, of the sample distribution. In our results, we find that the correction of the means proposed by Curto [2023], Eq. (20) there, yields good denoising behavior when combined with a Box-Cox transformation of the samples as detailed in §5.

As an orthogonal extension, we note that allowing asymmetric membership functions can produce visually more appealing results; while fewer constraints can also lead to improved MSE, the potential energy loss due to asymmetric weights may cause overall darker images and therefore slightly worse quantitative errors (Fig. 6g). Relaxing the symmetry assumption can be easily done by removing the second error term (last two lines) from Eq. (11). In this case, the optimal weight simplifies to

$$w^*_{\mathrm{asym}} = \frac{(\theta_i - \theta_j)^2 + \mathrm{Var}(\hat{\theta}_j)}{(\theta_i - \theta_j)^2 + \mathrm{Var}(\hat{\theta}_i) + \mathrm{Var}(\hat{\theta}_j)}. \tag{14}$$

Algorithm 1 Online statistics

Independently per pixel i ...
Initialize $\{n_i, \mu_i, \hat{M}_{2,i}, \hat{M}_{3,i}\} = 0$
for each MC sample x_k **do**
  $n_i \leftarrow n_i + 1$
  Box-Cox transform $x_k \rightarrow x'_k$, Eq. (15)
  Update pixel statistics with $\delta = x'_k - \mu_i$, [Meng 2015]:
    $\mu_i \leftarrow \mu_i + \delta/n_i$ (Welford)
    $\hat{M}_{2,i} \leftarrow \hat{M}_{2,i} + \delta(\delta - \delta/n_i)$
    $\hat{M}_{3,i} \leftarrow \hat{M}_{3,i} - 3(\delta/n_i)\hat{M}_{2,i} + \delta(\delta^2 - (\delta/n_i)^2)$
Central moments: $M_{l,i} = \hat{M}_{l,i}/n_i$
Variance: $\sigma_i^2 = \hat{M}_{2,i}/(n_i - 1)$ (Bessel-corrected)

Algorithm 2 Image-space filter weights w_{ij}

Input (per pixel): statistics $\{n, \mu, \sigma^2, M_2, M_3\}$ and G-buffers $\mathbf{p}$
Input (global): critical value γ_w
for each center pixel i in parallel **do**
  for each candidate pixel j within base-filter radius **do**
    Evaluate base filter $(\mathbf{p}_i, \mathbf{p}_j) \rightarrow \rho_{ij}$, Eq. (16)
    $\mathrm{Var}(\hat{\theta}_i) = \sigma_i^2/n_i, \mathrm{Var}(\hat{\theta}_j) = \sigma_j^2/n_j$
    Correct means to account for skewness, [Curto 2023]:
    $\theta_i \approx \mu_i + M_{3,i}/(6\sigma_i^2 n_i),\ \theta_j \approx \mu_j + M_{3,j}/(6\sigma_j^2 n_j)$
    Compute $w^*(\theta_i, \theta_j, \mathrm{Var}(\hat{\theta}_i), \mathrm{Var}(\hat{\theta}_j))$, Eq. (12) or (14)
    Compute t-statistic: $t = ((2(1 - w^*_{ij}))^{-1} - 1)^{1/2}$
    $m_{ij} = (t < \gamma_w)\,?\,1 : 0$
    $w_{ij} = \rho_{ij} m_{ij}$
  Normalize weights $\forall j : w_{ij} \leftarrow w_{ij}/\sum_i w_{ij}$

As before, we then replace the unknown ground-truth values with estimates based on the observed sample statistics, using either normality assumption or Curto's correction, to evaluate the membership function according to Eq. (13).

5 APPLICATION TO IMAGE-SPACE DENOISING

This section describes how to apply our general statistical framework to image-space denoising. In image space, the indices i and j each denote an individual pixel (not image-space coordinates). In particular, our implementation contains three major components. First, we track online statistics of the samples from the MC renderer, Alg. 1; this part is implemented as part of the renderer itself (we use pbrt-v3 [Pharr et al. 2016] for our results). Second, we select a joint bilateral filter to define the a priori weight ρ_{ij} in Eq. (10). Finally, we implement our statistical denoising pipeline, which takes the noisy image, sample statistics, and a priori weights as input to compute the filter weights, Alg. 2, and produce the final output image (we implement this part on the GPU using CUDA).

In MC rendering, we work with (weighted) radiance samples that cannot be negative, while a few paths may result in large contributions to a pixel's intensity. In statistical terms, the distribution of samples tends to be right-skewed. Given these observations, we employ the widely used Box-Cox transformation [Box and Cox 2018]

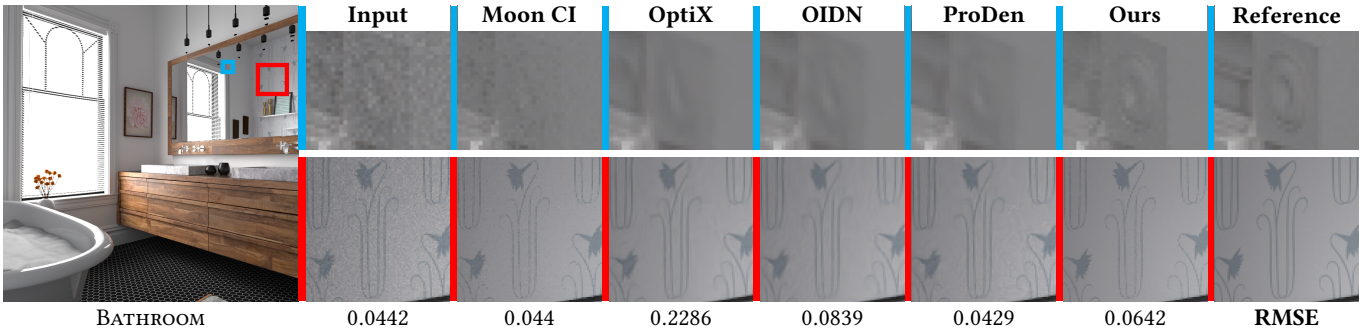

Fig. 2. Bathroom scene rendered with 8192 SPP. Denoising time for Moon CI: 39.7 ms, OptiX: 91.2 ms, OIDN: 20.2 ms, ProDen: 2027.5 ms, ours: 29.3 ms.

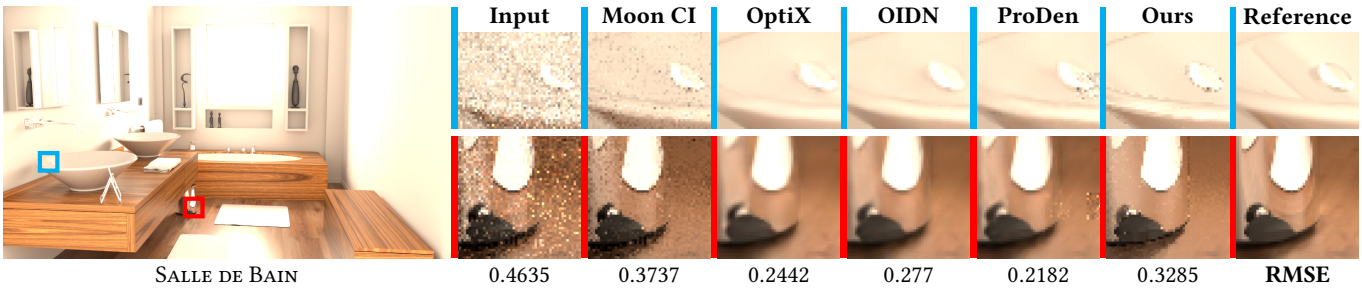

Fig. 3. Salle de Bain scene rendered with 32 SPP. Denoising time for Moon CI: 36.4 ms, OptiX: 141.6 ms, OIDN: 22.1 ms, ProDen: 1979.9 ms, ours: 28.5 ms. In the supplementary document, we provide results for this scene at a higher sample count of 512 SPP.

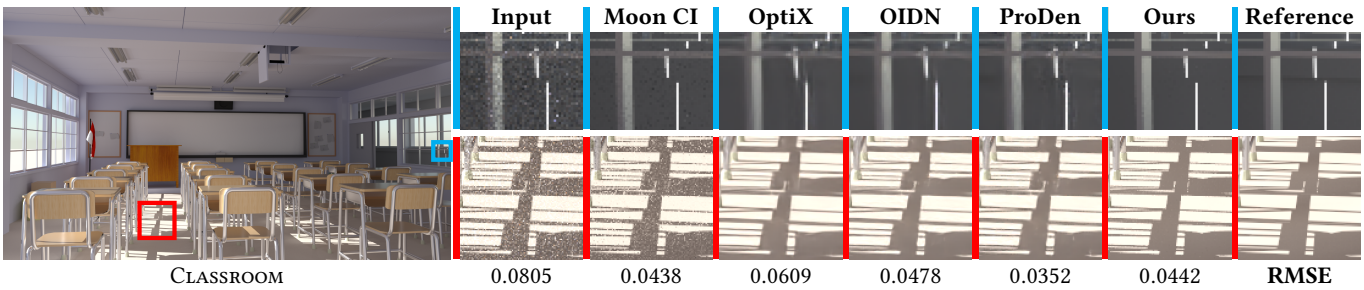

Fig. 4. Classroom scene rendered with 256 SPP. Denoising time for Moon CI: 35.6 ms, OptiX: 89.5 ms, OIDN: 20.8 ms, ProDen: 1840.5 ms, ours: 28.3 ms.

to "normalize" the rendered samples:

$$x'_k(\lambda) = \begin{cases} \log(x_k) & \text{if } \lambda = 0, \\ (x_k^\lambda - 1)/\lambda & \text{otherwise.} \end{cases} \tag{15}$$

We note that for samples generated by MC rendering, choosing $\lambda = 0$ is impractical, as many samples may be zero (e.g., paths that do not reach a light source before termination), where the log function is undefined. In our experiments, we find that $\lambda = 1/2$ yields good results in practice, as it effectively "compresses" high-valued outliers while avoiding excessive "stretching" of small values toward $-\infty$. For each transformed sample x'_k arriving from the renderer, we then update the online statistics following Alg. 1.

As a base filter, we choose the joint bilateral filter [Eisemann and Durand 2004; Petschnigg et al. 2004] to define the a priori weights ρ_{ij} for any pair of pixels (i, j). These weights follow a Gaussian falloff with distance in combined image space and G-buffers:

$$\rho_{ij} = \exp(-\frac{1}{2}(\mathbf{p}_j - \mathbf{p}_i)^\mathsf{T} \Sigma^{-1} (\mathbf{p}_j - \mathbf{p}_i)), \tag{16}$$

where $\mathbf{p}_i$ denotes the a priori information for pixel i. We use image-space position, RGB albedo color, and surface normal for each pixel, i.e., $\mathbf{p}_i = (x_i, y_i, r_i, g_i, b_i, n_{ix}, n_{iy}, n_{iz})^\mathsf{T}$, in our results. Moreover, Σ denotes a covariance matrix that controls the rate of falloff for each dimension of $\mathbf{p}_i$. To enforce the sparsity of these weights, we limit the filter to a small neighborhood within a constant radius around each pixel in image space; our default covariance matrix is then $\Sigma = \mathrm{diag}(10, 10, 0.02, 0.02, 0.02, 0.1, 0.1, 0.1)$.

In practice, our statistical filter operates with three color channels per pixel in RGB color space and evaluates the membership function for each channel according to Alg. 2. We set the final weight w_{ij} to

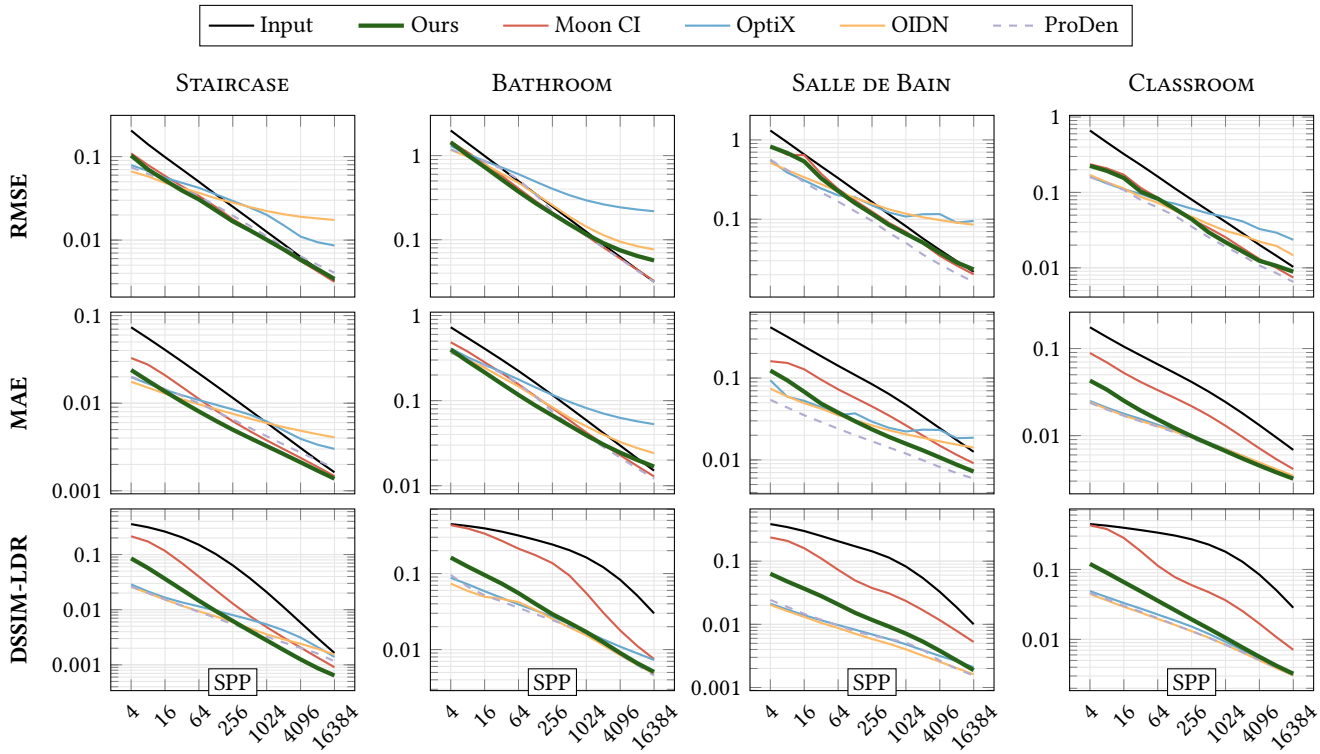

Fig. 5. Convergence plots for our denoising method, compared to the approach by Moon et al. [2013] ("Moon CI"), NVIDIA OptiX AI-Accelerated Denoiser ("OptiX"), Intel Open Image Denoise (OIDN), and progressive denoising [Firmino et al. 2022] ("ProDen"). Error metrics top-to-bottom: root mean square error (RMSE), mean absolute error (MAE), and structural dissimilarity (DSSIM) (evaluated after tone mapping to low-dynamic-range (LDR) images, because DSSIM requires a bound value range). Scenes left-to-right: Figs. 1–4.

zero unless all three channels pass the statistical test before applying the filter. In this way, we avoid color shifts that could occur if a channel is evaluated differently from the others.

In summary, our approach builds on three key insights: First, minimizing pair-wise MSE is closely related to Welch's t-test; this theoretical contribution forms the basis of our method. Second, all collected statistics are noisy estimates; enforcing binary membership functions prevents most of this noise from propagating downstream. Finally, Box-Cox transformation, as well as correcting the mean [Curto 2023; Johnson 1978], makes our method more robust to non-normality, effectively mitigating the influence of outliers ("fireflies").

The parameters of our method affect the results as follows: The threshold γ, Eq. (13), adjusts denoising strength. For symmetric weights, $\gamma = 0$ reverts to the base filter, $\gamma = 1/2$ and non-zero variance yields $m_{ij} = 0$ (when $i \neq j$), effectively disabling filtering. The covariance matrix in Eq. (16) specifies the extent of the filtering window, depending on distances in image space and differences in G-buffer values; higher values generally increase the number of membership evaluations and lead to smoother results. The Box-Cox parameter λ determines how samples are transformed: $\lambda = 1$ keeps their distribution shape unchanged, smaller values correct right-skewed distributions (reducing positive outliers), larger values correct left skewness.

6 RESULTS I — IMAGE DENOISING

In this section, we compare our statistical denoiser to state-of-the-art machine-learning-based denoisers (NVIDIA OptiX AI-Accelerated Denoiser and Intel Open Image Denoise), as well as a meta-denoiser [Firmino et al. 2022], on multiple well-known test scenes [Bitterli 2016] in Figs. 1–4. Figure 5 shows a quantitative convergence analysis for these examples. Note that without useful information from the G-buffers (e.g., objects seen in the mirror in Fig. 2), neural methods often fail to reconstruct sharp features. We also compare our membership formulation to the one-sided confidence-interval test proposed by Moon et al. [2013], using the 99.8% confidence level as proposed in their paper. Here, we compare the membership functions directly without applying their two-step denoising process.

We also analyze the effect of various components and parameters of our method in Fig. 6. Our statistical membership function greatly contributes to preserving the hard edges of shadows, which would be over-blurred by the base filter, as such lighting effects are not present in the G-buffers. Note how the G-buffers and our membership functions complement each other in reconstructing image features (Figs. 6b and c). The Box-Cox transformation further increases performance by "normalizing" the sample distributions.

All results were rendered and denoised on a desktop PC with an AMD Ryzen 9 5950X CPU and an NVIDIA RTX 3080 Ti GPU. Our

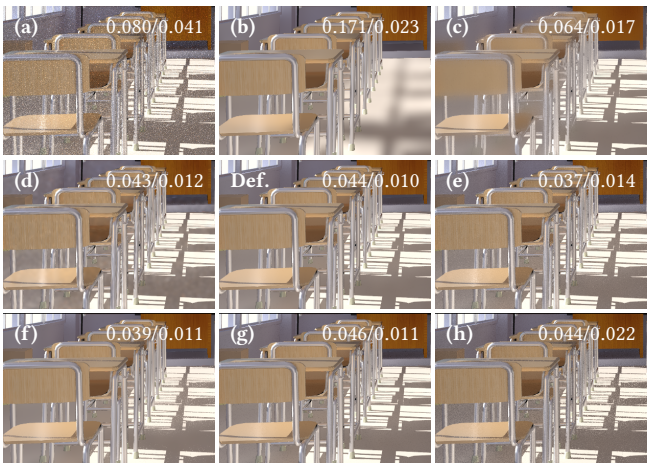

Fig. 6. Ablation study: we compare our default settings (center) to (a) the noisy input, (b) the joint bilateral base filter only (i.e., $m_{ij} = 1\ \forall i, j$), (c) a Gaussian base filter (i.e., no G-buffers, but using our membership functions), (d) using a smaller base filter (6-pixel radius; default is 20 pixels), (e) a more discerning significance level ($\alpha = 0.1$ instead of the default $\alpha = 0.005$), (f) disabling the Box-Cox transformation, (g) allowing asymmetric weights, Eq. (14), and (h) using the one-sided confidence-interval test by Moon et al. [2013]. Errors are given as RMSE/MAE.

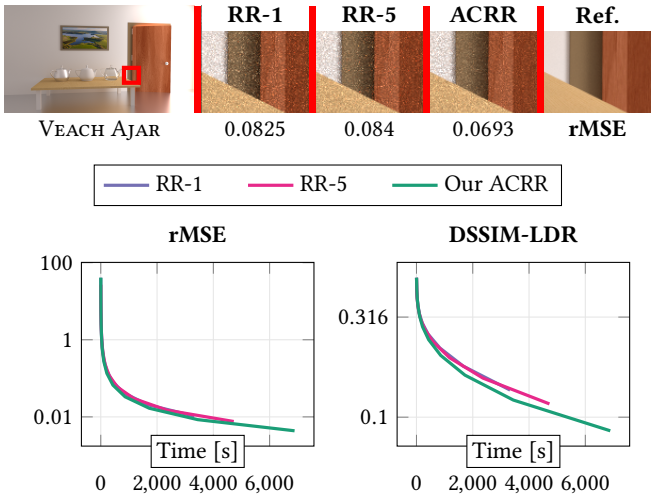

Fig. 7. Comparison between our approximate-contribution Russian roulette (ACRR) and classic throughput-based RR starting at the first bounce (RR-1) and at the fifth bounce (RR-5). Images show RR-1 at 2048 SPP (417.53 s), RR-5 at 1451 SPP (418.05 s) and ACRR at 987 SPP (417.00 s). Timings include GPU upload, denoising, and download. Here, we also show relative mean squared error (rMSE) [Rousselle et al. 2011], as it allows to gauge the sampling performance for individual pixels, regardless of their absolute value.

filtering runtimes are generally comparable to the duration of the inference step of Intel Open Image Denoise (OIDN) and substantially faster than progressive denoising [Firmino et al. 2022]. We set the base-filter radius to a relatively large size of 20 pixels (mostly for 1280×720 resolution images). Reducing the radius would further speed up our method, as the number of membership tests scales quadratically with the radius: while for 20 pixels, we require around 28 ms, for 6 pixels, we only require around 11 ms (Fig. 6d). Note that using the base filter alone would lead to noticeable over-blurring (Fig. 6b).

7 RESULTS II — FURTHER APPLICATIONS

The generality of our denoising framework allows us to go beyond simply denoising an image produced by MC rendering. In particular, we see great potential in leveraging the ability to quickly denoise arbitrary statistics of arbitrary quantities tracked during path tracing. This section presents two such applications: including *per-bounce approximate path contributions* in Russian roulette (RR), as well as an extension to multiple importance sampling (MIS) using *MIS strategy win rates*.

7.1 Approximate-Contribution Russian Roulette

Here, we extend classic throughput-based RR for path tracing. Instead of determining the path termination probability based on just the camera-centric throughput, we track and denoise the average incoming radiance for a particular bounce index per pixel. The termination probability at each bounce then considers the product of the path throughput and this average incoming radiance. Our goal is to improve termination probabilities for each path individually; thus, we divide the product by the average incoming radiance at the corresponding pixel of the camera.

Formally, the per-bounce incident radiance is defined as

$$L'_{j,k} = \int_{\mathbf{x}\in\Xi} p_{j,k}(\mathbf{x}) \int_{\omega_i\in\Omega} L_i(\mathbf{x}, \omega_i)\, \mathrm{d}\omega_i\, \mathrm{d}\mathbf{x}, \tag{17}$$

where Ξ is world space and $p_{j,k}(\mathbf{x})$ the probability of sampling a position $\mathbf{x}$ within that space for a given pixel j and bounce index k. In practice, we do not need to explicitly evaluate $p_{j,k}(\mathbf{x})$ as this probability is inherent to the path-tracing process. In fact, to estimate $L'_{j,k}$, we just have to average all incoming-radiance samples (over all paths) for a given pixel and bounce index, which can be easily integrated into common path-tracing code.

We then apply an iterative process during rendering: In each iteration, we track new incoming-radiance samples generated during path tracing, progressively refining our estimate of $L'_{j,k}$, Eq. (17), and the corresponding statistics required for denoising. At the end of each iteration, we denoise these estimates as described in §5. From the second iteration onward, where denoised estimates are available, we use modified path termination probabilities according to these radiance estimates.

Figure 7 shows improved image errors for our approximate-contribution version (ACRR) in an equal-time comparison over two variants of classic RR taking only path throughput into account. Classic RR often terminates only from a certain (e.g., fifth) bounce onward to avoid prematurely terminating high-contribution paths. Because of our improved path-contribution approximation, ACRR does not rely on this work-around. Rath et al. [2022] show that optimal weights for RR can be approximated using a spatio-directional data structure. While our method cannot achieve their sampling

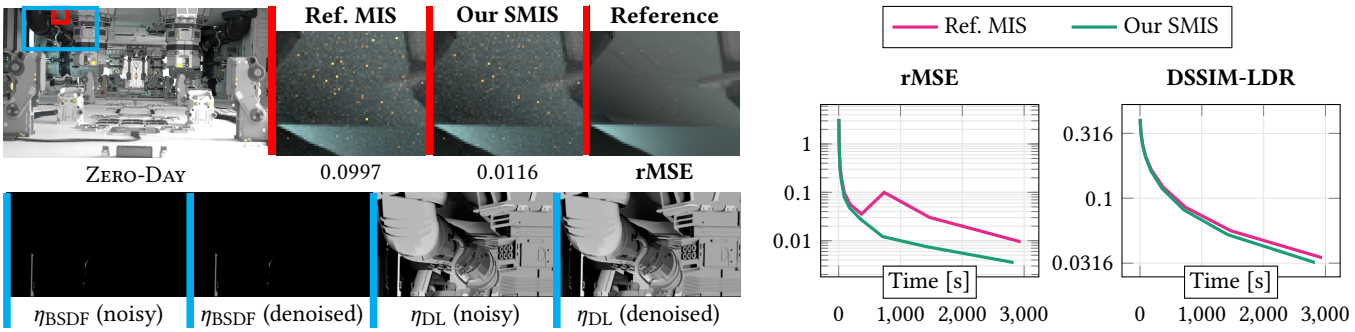

Fig. 8. Comparison of our selective multiple importance sampling (SMIS) to standard multiple importance sampling [Veach 1997] ("Ref. MIS") and visualization of first-bounce SMIS win rates for BSDF sampling (η_{BSDF}) and direct-light sampling (η_{DL}) before and after denoising. We use rMSE [Rousselle et al. 2011] to better assess the sampling performance for individual pixels. Images show Ref. MIS at 1024 SPP (734.80 s) and our SMIS at 1063 SPP (732.89 s).

efficiency, we illustrate how our denoising framework, based on regular image buffers, can be used as an alternative. We provide further details and results in the supplementary document.

7.2 Selective Multiple Importance Sampling

As our second example application, we present an extension to (multi-sample) multiple importance sampling (MIS) [Veach 1997; Veach and Guibas 1995]. The basic idea of MIS is to combine multiple sampling strategies by a weighted average, depending on the probability density function (PDF) of each strategy. The sampling strategies typically cover different features of an integrand, for instance, via bidirectional-scattering-distribution-function (BSDF) or direct-light sampling. However, in situations where one sampling strategy is clearly better than the other(s), MIS can introduce additional variance (noise), especially when using the balance heuristic. Veach [1997] also developed the cut-off, power, and maximum heuristics to mitigate this issue, generally moving large weights closer to one and small weights closer to zero. Choosing MIS weights and sample allocations is non-trivial and an active area of research [Grittmann et al. 2019; Kondapaneni et al. 2019; Szirmay-Kalos and Sbert 2022], and the effectiveness of a strategy depends on the given situation. We illustrate how our framework can be used to selectively perform MIS only in beneficial cases, independently of the chosen strategy.

For this, we extend MIS by identifying strategies that should be disabled in order to save computational effort and reduce noise. The primary metric for this decision is the *win rate*, $\eta_m = n^*_m/n_i$, of sampling strategy m, where a "win" (counted in n^*_m) is a non-zero sample whose PDF value of strategy m exceeds the PDF values of all other strategies. When determining this win rate—as in other MIS approaches—we evaluate all sampling strategies for each sample (not just the strategy that produced the sample), even if a strategy has already been disabled earlier. Similarly to the average radiance in the RR example described above, we estimate and denoise the win rates per pixel and per bounce. In contrast to radiance samples, we do not Box-Cox transform win rates. We again proceed iteratively, starting with all MIS strategies enabled, tracking and denoising win rates, and—from the second iteration onward—disabling inferior ($\eta_m < 10^{-3}$) sampling strategies (per pixel and per bounce). Finally, note that disabling a sampling strategy generally does not introduce bias, because two unbiased estimates (a MIS estimate and an estimate from a single sampling strategy) can be naturally combined without introducing bias. Figure 8 shows example results and equal-time convergence behavior for combining bidirectional scattering distribution function (BSDF) and direct-light sampling.

8 DISCUSSION AND CONCLUSION

We have presented a simple yet effective denoising method for Monte Carlo rendering. Using well-known image filters for variance reduction and statistical tests (membership functions) to prevent bias, we achieve state-of-the-art image quality. Our approach is entirely free of pretrained components, using descriptive statistics of the sample distributions instead that can be estimated online during rendering. The properties of our membership function guarantee convergence with increasing sample counts. At low sample counts, high variance dominates the image error. In this case, the membership function becomes less discriminating, thereby reducing that variance and accepting some bias. As rendering progresses and more samples are added, variance decreases and the bias becomes more relevant to the overall error. If bias is significant (relative to variance), we set the corresponding filter weight to zero, based on statistical tests, which essentially eliminates that source of bias. In the limit, as variance approaches zero, any bias is unacceptable and thus prevented by the membership function.

We have focused on *binary* membership functions in this work. While we derive these functions from optimal pair-wise weights, recall that these weights are noisy estimates in practice. Introducing a threshold and restricting to binary memberships effectively prevents this noise from propagating through the pipeline. We leave investigating continuous membership functions for future work.

In contrast to neural-network-based approaches, our method does not require any computationally expensive training and does not risk adding "wrong" details that were present in the training data but should not be present in the output image.

Including our method as an additional input to existing meta-denoisers, such as progressive denoising [Firmino et al. 2022] or ensemble denoising [Zheng et al. 2021] could in turn deliver another step up in image quality for these methods. Furthermore, we also see

great potential for the applicability of our statistics-based approach to other types of estimators during MC rendering. In particular, variance estimates have been used for adaptive sampling [Rousselle et al. 2012], MIS [Grittmann et al. 2019], or path guiding [Rath et al. 2020] in the past. Efficiently denoising variance estimates with our framework could yield improved performance of these methods in the future.

Another promising avenue for future work is the extension of our method to the temporal domain: Conceptually, estimates at different points in time of an animation can be treated equivalently to estimates at different image-space positions. Including such estimates could improve denoising performance and temporal coherence for animation denoising.

ACKNOWLEDGMENTS

We would like to thank Lukas Lipp for fruitful discussions, Károly Zsolnai-Fehér and Jaroslav Křivánek for valuable contributions to early versions of this work, and Bernhard Kerbl for help with our CUDA implementation. Moreover, we would like to thank the creators of the scenes we have used: Wig42 for "Wooden Staircase" (Fig. 1), "Grey and White Room" (Fig. S6), and "Modern Living Room" (Fig. S8); nacimus for "Bathroom" (Fig. 3, S5); NovaZeeke for "Japanese Classroom" (Fig. 4, 6); Beeple for "Zero-Day" (Fig. 8); Jay-Artist for "White Room" (Fig. S7); Mareck for "Contemporary Bathroom" (Fig. 2); Christian Freude for "Glass Caustics" (Fig. S10); and Benedikt Bitterli for "Veach Ajar" (Fig. 7, S2), "Veach MIS" (Fig. S4), and "Fur Ball" (Fig. S11). This work has received funding from the Vienna Science and Technology Fund (WWTF) project ICT22-028 ("Toward Optimal Path Guiding for Photorealistic Rendering") and the Austrian Science Fund (FWF) project F 77 (SFB "Advanced Computational Design").

REFERENCES

Attila T. Áfra. 2024. Intel® Open Image Denoise. https://www.openimagedenoise.org.

Jonghee Back, Binh-Son Hua, Toshiya Hachisuka, and Bochang Moon. 2022. Self-Supervised Post-Correction for Monte Carlo Denoising. In *SIGGRAPH '22: Special Interest Group on Computer Graphics and Interactive Techniques Conference, Vancouver, BC, Canada, August 7 - 11, 2022*, Munkhtsetseg Nandigjav, Niloy J. Mitra, and Aaron Hertzmann (Eds.). ACM, 18:1–18:8. https://doi.org/10.1145/3528233.3530730

Jonghee Back, Binh-Son Hua, Toshiya Hachisuka, and Bochang Moon. 2023. Input-Dependent Uncorrelated Weighting for Monte Carlo Denoising. In *SIGGRAPH Asia 2023 Conference Papers, SA 2023, Sydney, NSW, Australia, December 12-15, 2023*, June Kim, Ming C. Lin, and Bernd Bickel (Eds.). ACM, 9:1–9:10. https://doi.org/10.1145/3610548.3618177

Pablo Bauszat, Martin Eisemann, and Marcus A. Magnor. 2011. Guided Image Filtering for Interactive High-quality Global Illumination. *Comput. Graph. Forum* 30, 4 (2011), 1361–1368. https://doi.org/10.1111/J.1467-8659.2011.01996.X

Benedikt Bitterli. 2016. Rendering resources. https://benedikt-bitterli.me/resources/.

Benedikt Bitterli, Fabrice Rousselle, Bochang Moon, José Antonio Iglesias Guitián, David Adler, Kenny Mitchell, Wojciech Jarosz, and Jan Novák. 2016. Nonlinearly Weighted First-order Regression for Denoising Monte Carlo Renderings. *Comput. Graph. Forum* 35, 4 (2016), 107–117. https://doi.org/10.1111/CGF.12954

Malik Boughida and Tamy Boubekeur. 2017. Bayesian Collaborative Denoising for Monte Carlo Rendering. *Comput. Graph. Forum* 36, 4 (2017), 137–153. https://doi.org/10.1111/CGF.13231

G. E. P. Box and D. R. Cox. 2018. An Analysis of Transformations. *Journal of the Royal Statistical Society: Series B (Methodological)* 26, 2 (12 2018), 211–243. https://doi.org/10.1111/j.2517-6161.1964.tb00553.x arXiv:https://academic.oup.com/jrsssb/article-pdf/26/2/211/49099371/jrsssb_26_2_211.pdf

Chakravarty R. Alla Chaitanya, Anton S. Kaplanyan, Christoph Schied, Marco Salvi, Aaron E. Lefohn, Derek Nowrouzezahrai, and Timo Aila. 2017. Interactive reconstruction of Monte Carlo image sequences using a recurrent denoising autoencoder. *ACM Trans. Graph.* 36, 4 (2017), 98:1–98:12. https://doi.org/10.1145/3072959.3073601

José Dias Curto. 2023. Confidence intervals for means and variances of nonnormal distributions. *Communications in Statistics - Simulation and Computation* 52, 9 (2023), 4414–4430. https://doi.org/10.1080/03610918.2021.1963448

Holger Dammertz, Daniel Sewtz, Johannes Hanika, and Hendrik P. A. Lensch. 2010. Edge-avoiding À-Trous wavelet transform for fast global illumination filtering. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Conference on High Performance Graphics 2010, Saarbrücken, Germany, June 25-27, 2010*, Justin Hensley, Philipp Slusallek, David K. McAllister, and Christiaan P. Gribble (Eds.). Eurographics Association, 67–75. https://doi.org/10.2312/EGGH/HPG10/067-075

Mauricio Delbracio, Pablo Musé, Antoni Buades, Julien Chauvier, Nicholas Phelps, and Jean-Michel Morel. 2014. Boosting monte carlo rendering by ray histogram fusion. *ACM Trans. Graph.* 33, 1 (2014), 8:1–8:15. https://doi.org/10.1145/2532708

Elmar Eisemann and Frédo Durand. 2004. Flash photography enhancement via intrinsic relighting. *ACM Trans. Graph.* 23, 3 (2004), 673–678. https://doi.org/10.1145/1015706.1015778

Arthur Firmino, Jeppe Revall Frisvad, and Henrik Wann Jensen. 2022. Progressive Denoising of Monte Carlo Rendered Images. *Comput. Graph. Forum* 41, 2 (2022), 1–11. https://doi.org/10.1111/CGF.14454

Pascal Grittmann, Iliyan Georgiev, Philipp Slusallek, and Jaroslav Křivánek. 2019. Variance-aware multiple importance sampling. *ACM Trans. Graph.* 38, 6 (2019), 152:1–152:9. https://doi.org/10.1145/3355089.3356515

Yuchi Huo and Sung-Eui Yoon. 2021. A survey on deep learning-based Monte Carlo denoising. *Comput. Vis. Media* 7, 2 (2021), 169–185. https://doi.org/10.1007/S41095-021-0209-9

Norman J. Johnson. 1978. Modified t Tests and Confidence Intervals for Asymmetrical Populations. *J. Amer. Statist. Assoc.* 73, 363 (1978), 536–544. http://www.jstor.org/stable/2286597

James T. Kajiya. 1986. The rendering equation. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1986, Dallas, Texas, USA, August 18-22, 1986*, David C. Evans and Russell J. Athay (Eds.). ACM, 143–150. https://doi.org/10.1145/15922.15902

Nima Khademi Kalantari and Pradeep Sen. 2013. Removing the Noise in Monte Carlo Rendering with General Image Denoising Algorithms. *Comput. Graph. Forum* 32, 2 (2013), 93–102. https://doi.org/10.1111/CGF.12029

J. F. Kenney and E. S. Keeping. 1951. *Mathematics Of Statistics: Part Two* (2nd ed.). D. Van Nostrand Company, Inc., Princeton, New Jersey. https://archive.org/details/in.ernet.dli.2015.223161/

Ivo Kondapaneni, Petr Vévoda, Pascal Grittmann, Tomás Skrivan, Philipp Slusallek, and Jaroslav Křivánek. 2019. Optimal multiple importance sampling. *ACM Trans. Graph.* 38, 4 (2019), 37:1–37:14. https://doi.org/10.1145/3306346.3323009

Tzu-Mao Li, Yu-Ting Wu, and Yung-Yu Chuang. 2012. SURE-based optimization for adaptive sampling and reconstruction. *ACM Trans. Graph.* 31, 6 (2012), 194:1–194:9. https://doi.org/10.1145/2366145.2366213

Yu Liu, Changwen Zheng, and Hongliang Yuan. 2018a. Denoising Monte Carlo Renderings based on a Robust High-order Function. In *Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2018) - Volume 1: GRAPP, Funchal, Madeira, Portugal, January 27-29, 2018*, Dominique Bechmann, Ana Paula Cláudio, and José Braz (Eds.). SciTePress, 288–294. https://doi.org/10.5220/0006650602880294

Yu Liu, Changwen Zheng, Quan Zheng, and Hongliang Yuan. 2018b. Removing Monte Carlo noise using a Sobel operator and a guided image filter. *Vis. Comput.* 34, 4 (2018), 589–601. https://doi.org/10.1007/S00371-017-1363-Z

Michael Mara, Morgan McGuire, Benedikt Bitterli, and Wojciech Jarosz. 2017. An efficient denoising algorithm for global illumination. In *Proceedings of High Performance Graphics, HPG 2017, Los Angeles, CA, USA, July 28 - 30, 2017*. ACM, 3:1–3:7. https://doi.org/10.1145/3105762.3105774

Michael D. McCool. 1999. Anisotropic diffusion for Monte Carlo noise reduction. *ACM Trans. Graph.* 18, 2 (1999), 171–194. https://doi.org/10.1145/318009.318015

Xiangrui Meng. 2015. Simpler Online Updates for Arbitrary-Order Central Moments. arXiv:1510.04923 [stat.CO] https://arxiv.org/abs/1510.04923

Bochang Moon, Nathan Carr, and Sung-Eui Yoon. 2014. Adaptive Rendering Based on Weighted Local Regression. *ACM Trans. Graph.* 33, 5 (2014), 170:1–170:14. https://doi.org/10.1145/2641762

Bochang Moon, José Antonio Iglesias Guitián, Sung-Eui Yoon, and Kenny Mitchell. 2015. Adaptive rendering with linear predictions. *ACM Trans. Graph.* 34, 4 (2015), 121:1–121:11. https://doi.org/10.1145/2766992

Bochang Moon, Jong Yun Jun, JongHyeob Lee, Kunho Kim, Toshiya Hachisuka, and Sung-Eui Yoon. 2013. Robust Image Denoising Using a Virtual Flash Image for Monte Carlo Ray Tracing. *Comput. Graph. Forum* 32, 1 (2013), 139–151. https://doi.org/10.1111/CGF.12004

Bochang Moon, Steven McDonagh, Kenny Mitchell, and Markus H. Gross. 2016. Adaptive polynomial rendering. *ACM Trans. Graph.* 35, 4 (2016), 40:1–40:10. https://doi.org/10.1145/2897824.2925936

Ryan S. Overbeck, Craig Donner, and Ravi Ramamoorthi. 2009. Adaptive wavelet rendering. *ACM Trans. Graph.* 28, 5 (2009), 140. https://doi.org/10.1145/1618452.1618486

Hyosub Park, Bochang Moon, Soomin Kim, and Sung-Eui Yoon. 2013. P-RPF: Pixel-Based Random Parameter Filtering for Monte Carlo Rendering. In *2013 International*

Conference on Computer-Aided Design and Computer Graphics, CAD/Graphics 2013, Guangzhou, China, November 16-18, 2013. IEEE, 123–130. https://doi.org/10.1109/CADGRAPHICS.2013.24

Georg Petschnigg, Richard Szeliski, Maneesh Agrawala, Michael F. Cohen, Hugues Hoppe, and Kentaro Toyama. 2004. Digital photography with flash and no-flash image pairs. *ACM Trans. Graph.* 23, 3 (2004), 664–672. https://doi.org/10.1145/1015706.1015777

Matt Pharr, Wenzel Jakob, and Greg Humphreys. 2016. *Physically Based Rendering: From Theory to Implementation (3rd ed.)* (3rd ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. 1266 pages.

Alexander Rath, Pascal Grittmann, Sebastian Herholz, Petr Vévoda, Philipp Slusallek, and Jaroslav Krivánek. 2020. Variance-aware path guiding. *ACM Trans. Graph.* 39, 4 (2020), 151. https://doi.org/10.1145/3386569.3392441

Alexander Rath, Pascal Grittmann, Sebastian Herholz, Philippe Weier, and Philipp Slusallek. 2022. EARS: efficiency-aware russian roulette and splitting. *ACM Trans. Graph.* 41, 4 (2022), 81:1–81:14. https://doi.org/10.1145/3528223.3530168

Fabrice Rousselle, Claude Knaus, and Matthias Zwicker. 2011. Adaptive sampling and reconstruction using greedy error minimization. *ACM Trans. Graph.* 30, 6 (2011), 159. https://doi.org/10.1145/2070781.2024193

Fabrice Rousselle, Claude Knaus, and Matthias Zwicker. 2012. Adaptive rendering with non-local means filtering. *ACM Trans. Graph.* 31, 6 (2012), 195:1–195:11. https://doi.org/10.1145/2366145.2366214

Fabrice Rousselle, Marco Manzi, and Matthias Zwicker. 2013. Robust Denoising using Feature and Color Information. *Comput. Graph. Forum* 32, 7 (2013), 121–130. https://doi.org/10.1111/CGF.12219

F. E. Satterthwaite. 1946. An Approximate Distribution of Estimates of Variance Components. *Biometrics Bulletin* 2, 6 (1946), 110–114. http://www.jstor.org/stable/3002019

Christoph Schied, Anton Kaplanyan, Chris Wyman, Anjul Patney, Chakravarty R. Alla Chaitanya, John Burgess, Shiqiu Liu, Carsten Dachsbacher, Aaron E. Lefohn, and Marco Salvi. 2017. Spatiotemporal variance-guided filtering: real-time reconstruction for path-traced global illumination. In *Proceedings of High Performance Graphics, HPG 2017, Los Angeles, CA, USA, July 28 - 30, 2017*. ACM, 2:1–2:12. https://doi.org/10.1145/3105762.3105770

Christoph Schied, Christoph Peters, and Carsten Dachsbacher. 2018. Gradient Estimation for Real-time Adaptive Temporal Filtering. *Proc. ACM Comput. Graph. Interact. Tech.* 1, 2 (2018), 24:1–24:16. https://doi.org/10.1145/3233301

Pradeep Sen and Soheil Darabi. 2012. On filtering the noise from the random parameters in Monte Carlo rendering. *ACM Trans. Graph.* 31, 3 (2012), 18:1–18:15. https://doi.org/10.1145/2167076.2167083

László Szirmay-Kalos and Mateu Sbert. 2022. Robust Sample Budget Allocation for MIS. In *43rd Annual Conference of the European Association for Computer Graphics, Eurographics 2022 - Short Papers, Reims, France, April 25-29, 2022*, Nuria Pelechano and David Vanderhaeghe (Eds.). Eurographics Association, 17–20. https://doi.org/10.2312/EGS.20221022

Eric Veach. 1997. *Robust Monte Carlo methods for light transport simulation*. Ph. D. Dissertation. Stanford University, USA. https://searchworks.stanford.edu/view/3911108

Eric Veach and Leonidas J. Guibas. 1995. Optimally combining sampling techniques for Monte Carlo rendering. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1995, Los Angeles, CA, USA, August 6-11, 1995*. ACM, 419–428. https://dl.acm.org/citation.cfm?id=218498

Delio Vicini, David Adler, Jan Novák, Fabrice Rousselle, and Brent Burley. 2019. Denoising Deep Monte Carlo Renderings. *Comput. Graph. Forum* 38, 1 (2019), 316–327. https://doi.org/10.1111/CGF.13533

B. P. Welford. 1962. Note on a Method for Calculating Corrected Sums of Squares and Products. *Technometrics* 4, 3 (1962), 419–420. https://doi.org/10.1080/00401706.1962.10490022

Ruifeng Xu and Sumanta N. Pattanaik. 2005. A Novel Monte Carlo Noise Reduction Operator. *IEEE Computer Graphics and Applications* 25, 2 (2005), 31–35. https://doi.org/10.1109/MCG.2005.31

Hongliang Yuan and Changwen Zheng. 2017. Adaptive rendering based on a weighted mixed-order estimator. *Vis. Comput.* 33, 6-8 (2017), 695–704. https://doi.org/10.1007/S00371-017-1381-X

Hongliang Yuan and Changwen Zheng. 2018. Adaptive rendering based on robust principal component analysis. *Vis. Comput.* 34, 4 (2018), 551–562. https://doi.org/10.1007/S00371-017-1360-2

Changwen Zheng and Yu Liu. 2018. Removing Monte Carlo Noise with Compressed Sensing and Feature Information. In *Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2018) - Volume 1: GRAPP, Funchal, Madeira, Portugal, January 27-29, 2018*, Dominique Bechmann, Ana Paula Cláudio, and José Braz (Eds.). SciTePress, 145–153. https://doi.org/10.5220/0006671601450153

Shaokun Zheng, Fengshi Zheng, Kun Xu, and Ling-Qi Yan. 2021. Ensemble denoising for Monte Carlo renderings. *ACM Trans. Graph.* 40, 6 (2021), 274:1–274:17. https://doi.org/10.1145/3478513.3480510

Matthias Zwicker, Wojciech Jarosz, Jaakko Lehtinen, Bochang Moon, Ravi Ramamoorthi, Fabrice Rousselle, Pradeep Sen, Cyril Soler, and Sung-Eui Yoon. 2015. Recent Advances in Adaptive Sampling and Reconstruction for Monte Carlo Rendering. *Comput. Graph. Forum* 34, 2 (2015), 667–681. https://doi.org/10.1111/CGF.12592