

Statistical Error Reduction for Monte Carlo Rendering

HIROYUKI SAKAI, Technische Universität Wien (TU Wien), Austria

CHRISTIAN FREUDE, Technische Universität Wien (TU Wien), Austria

MICHAEL WIMMER, Technische Universität Wien (TU Wien), Austria

DAVID HAHN, Technische Universität Wien (TU Wien), Austria

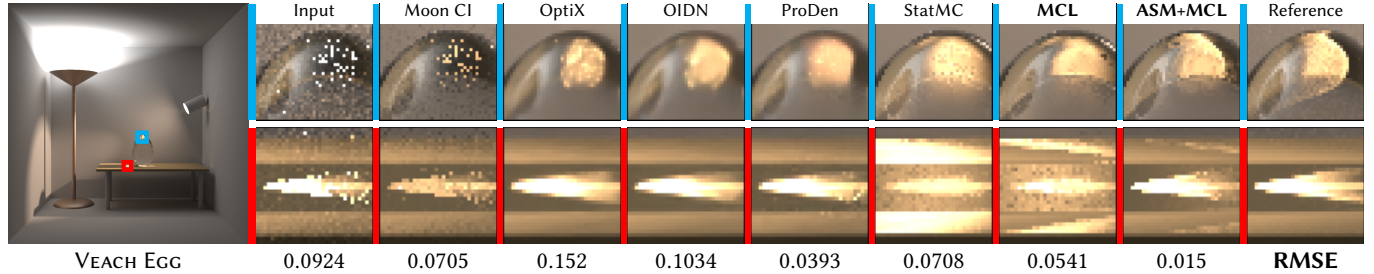


Fig. 1. Our statistical error-reduction framework combines multi-transform (‘M’) and cascaded (‘C’) denoising with linear explained-variance correction (‘L’) to denoise both radiance and variance estimates. The latter can be used for adaptive sampling (‘ASM’) to reduce image mean squared error (MSE) even further. We compare our denoiser (with and without adaptive sampling) to various state-of-the-art methods (see also Fig. 12 for details).

Denoising is an important post-processing step in physically based Monte Carlo (MC) rendering. While neural networks are widely used in practice, statistical analysis has recently become a viable alternative for denoising. In this paper, we present a general framework for statistics-based error reduction of both estimated radiance and variance. Specifically, we introduce a novel denoising approach for variance estimates, which can either improve variance-aware adaptive sampling or provide additional input for image denoising in a cascaded manner. Furthermore, we present multi-transform denoising: a general and efficient correction scheme for non-normal distributions, which typically occur in MC rendering. All these contributions combine to a robust denoising pipeline that does not require any pretraining and can run efficiently on current GPU hardware. Our results show distinct advantages over previous denoising methods, especially in the range of a few hundred samples per pixel, which is of high practical relevance. Finally, we demonstrate good convergence behavior as the number of samples increases, providing predictable results with low bias that are free of hallucinated neural artifacts. In summary, our statistics-based algorithms for adaptive sampling and denoising deliver fast, consistent, low-bias variance and radiance estimates.

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

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

Authors’ Contact Information: Hiroyuki Sakai, sakai@cg.tuwien.ac.at, Rendering and Modeling Group, Technische Universität Wien (TU Wien), Vienna, Austria; Christian Freude, freude@cg.tuwien.ac.at, Rendering and Modeling Group, Technische Universität Wien (TU Wien), Vienna, Austria; Michael Wimmer, wimmer@cg.tuwien.ac.at, Rendering and Modeling Group, Technische Universität Wien (TU Wien), Vienna, Austria; David Hahn, david.hahn@tuwien.ac.at, Rendering and Modeling Group, Technische Universität Wien (TU Wien), Vienna, Austria.



This work is licensed under a Creative Commons Attribution 4.0 International License. SA Conference Papers ’25, Hong Kong, Hong Kong
© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2137-3/2025/12
<https://doi.org/10.1145/3757377.3763995>

ACM Reference Format:

Hiroyuki Sakai, Christian Freude, Michael Wimmer, and David Hahn. 2025. Statistical Error Reduction for Monte Carlo Rendering. In *SIGGRAPH Asia 2025 Conference Papers (SA Conference Papers ’25)*, December 15–18, 2025, Hong Kong, Hong Kong. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3757377.3763995>

1 Introduction

Modern physically based rendering systems almost exclusively build on some variant of Monte Carlo (MC) integration. Their stochastic sampling process inherently produces some (random) noise in the resulting images. Denoising methods that post-process rendered images to reduce this noise have been extensively researched in the computer graphics literature. While most recent methods (e.g., [Firmino et al. 2022]) rely on neural networks to learn a map from noisy input images to low-noise outputs, previous work [Sakai et al. 2024] has shown that reasoning statistically about the sample distributions of nearby pixels can yield fast and efficient denoising without the overhead of prior training.

In this paper, we propose a general statistical framework for reducing error in MC rendering. Besides the classical problem of denoising image pixels (which from a statistical point of view are estimators of their population means), we also denoise estimates of the variance, which enables two novel applications: first, a statistically optimal adaptive sampling strategy that allocates batches of future samples to pixels based on (denoised) variance estimates; and second, a cascaded denoising pipeline that utilizes our denoised variances to improve the statistical tests employed to denoise the rendered image itself.

Sample distributions in MC rendering are generally quite different from the normal distribution, which is often assumed by statistical tests. The *Box–Cox transformation* can move these distributions closer to normality, but its effectiveness depends on a carefully selected transformation parameter in practice. We mitigate this issue

with a simple and memory-efficient approach that incorporates a set of different transformation parameters into our statistical denoising framework. Analyzing the sample distribution’s behavior under multiple transformations improves the validity of statistical tests and, therefore, the quality of our denoising method.

Finally, we note that when edges in an image cross the area of a pixel, variance estimates for such a pixel can become excessively large. This overestimated variance can in turn lead to overblurring during denoising, as a large variance estimate encourages the statistical tests to include more nearby pixels when filtering the image. We introduce a lightweight strategy that corrects for these issues by splitting the total variance into an explained and an unexplained component, and then linearly approximating the explained variance using the covariance between the sample values and the sample locations within each pixel. Our results show that this *linear explained-variance correction* clearly improves the denoising performance for these edge cases.

In summary, we present the following contributions in this paper:

- *multi-transform denoising*, building on the Box–Cox transformation and the Jarque–Bera statistic to quantify normality, which enables different regions of an image to utilize different transform parameters and thereby improves denoising quality for both mean and variance estimates;
- *linear explained-variance correction* to prevent overblurring of pixels containing multimodal sample distributions;
- and *variance denoising*, with applications to adaptive sampling and *cascaded denoising*, where we use corrected variance estimates to subsequently denoise the image itself.

We provide supplementary materials, such as the source code, at <https://www.cg.tuwien.ac.at/StatER>.

2 Related Work

The stochastic sampling in Monte Carlo (MC) rendering manifests itself as noise or error in the resulting images. A common approach to mitigate this noise is to use adaptive sampling or filtering [Huo and Yoon 2021; Zwicker et al. 2015]. In this work, we focus on a posteriori denoising of (image-space) pixel estimates.

Classical approaches usually use filters that are dynamically adapted in order to find a trade-off between reducing noise and introducing unwanted bias. Common choices are Gaussian [Rousselle et al. 2011], (joint) bilateral [Li et al. 2012; Liu et al. 2018; Mara et al. 2017; Park et al. 2013; Rousselle et al. 2013; Sen and Darabi 2012; Xu and Pattanaik 2005], 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. Alternatively, researchers have also used diffusion [McCool 1999], higher-order regression [Bauszat et al. 2011; Bitterli et al. 2016; Moon et al. 2014, 2015, 2016; Yuan and Zheng 2017, 2018], or statistical concepts, including histograms [Boughida and Boubekeur 2017; Delbracio et al. 2014; Park et al. 2013; Sen and Darabi 2012], confidence intervals [Moon et al. 2013, 2016], statistical testing [Sakai et al. 2024], and uncorrelated statistics [Back et al. 2023]. Other approaches combine multiple existing denoisers [Zheng et al. 2021]. To guide filtering, many approaches

include additional auxiliary features, such as albedos and normals, collectively referred to as “G-buffers”, or statistical estimates (e.g., sample variances).

Neural-network-based approaches are currently among the state-of-the-art methods for MC denoising, such as Intel Open Image Denoise (OIDN) [Áfra 2025] and the NVIDIA OptiX AI-Accelerated Denoiser, which is based on the work of Chaitanya et al. [2017]. These approaches typically use a noisy image and G-buffers as inputs. They are pretrained on many input–reference pairs to map noisy renderings to a low-error approximation of the ground-truth image. Consequently, their performance is highly dependent on the training data. A notable exception was presented by Choi et al. [2024], who employ online learning that does not require pretraining. However, reconstructing features that are missing in the G-buffers (such as shading effects) or exhibit high variance is challenging and can lead to overblurring or so-called “hallucinations”. In addition, convergence is often not guaranteed but can be enforced using the approach by Firmino et al. [2022].

In this paper, we focus on non-neural techniques, motivated by the work of Sakai et al. [2024], who showed that a purely statistical approach can compete with current neural-network-based denoising methods. Their approach is, however, limited in some important ways. First, although they use a Box–Cox transformation to improve robustness for non-normal distributions, their approach relies on a single fixed global transformation parameter and therefore cannot adapt to different sample distributions in complex scenes. Furthermore, Sakai et al. [2024] denoise the mean relying on (noisy) variance estimates but do not leverage the potential of denoising the variance. We generalize their idea, adding variance denoising to build a general statistical denoising framework.

Adaptive Sampling methods aim to reduce noise by non-uniformly distributing samples to pixels, commonly according to some measure of variability: early methods use contrast [Hachisuka et al. 2008; Kalantari and Sen 2013; Mitchell 1987; Overbeck et al. 2009] or perceptual metrics [Bolin and Meyer 1998; Farrugia and Péroche 2004; Ramasubramanian et al. 1999], while more recent approaches use variance [Rousselle et al. 2012, 2013] or mean squared error (MSE) [Li et al. 2012; Moon et al. 2014; Rousselle et al. 2011]. Most recently, neural-network-based approaches have also been introduced for adaptive sampling [Bako et al. 2019; Firmino et al. 2023; Hasselgren et al. 2020; Huo et al. 2020; Kuznetsov et al. 2018; Liang et al. 2024; Salehi et al. 2022; Vogels et al. 2018; Xing and Chen 2020]. As with the filtering step, we focus on non-neural approaches for adaptive sampling in this work. Several methods derive sampling densities from estimated reconstruction errors of the used filters [Li et al. 2012; Overbeck et al. 2009; Rousselle et al. 2011, 2013], while some also apply basic smoothing operations [Kalantari and Sen 2013; Rousselle et al. 2012]. In contrast, we first reduce the error of the unbiased rendering using adaptive sampling before denoising the image. For this purpose, we derive sampling densities from our denoised variance estimates following the Neyman allocation [1934].

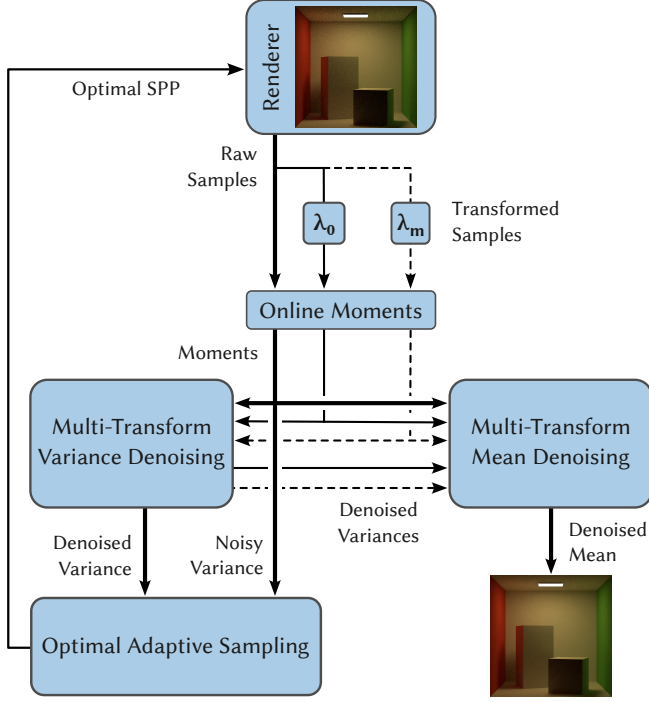


Fig. 2. Overview of our denoising framework, consisting of online moment computations for multi-transform variance and mean denoising, as well as an adaptive sampling strategy based on (denoised) variance estimates.

3 Overview

Here, we present a brief overview of our framework; in the following sections, we then summarize important concepts from the literature (§4) and describe our denoising approach in detail (§5). As illustrated in Fig. 2, for each pixel in a rendered image, we estimate statistical moments (related to the mean, variance, skewness, and kurtosis of the sample distribution), which are then used in statistical tests, formulated as membership functions that prevent filtering over pixel pairs whose distributions are significantly different. We employ the Box–Cox transformation to improve the normality of these distributions, and therefore the quality of these tests. One inherent problem with this approach is that the theoretically optimal transformation parameter may be different for every pair of pixels and must be known before transforming all samples within these pixels accordingly. While finding optimal parameters is clearly infeasible, we incorporate a fixed set of multiple transform parameters into our pipeline and select the best one in this set during denoising. “Best” may mean either closest to normality, as indicated by the Jarque–Bera statistic, Eq. (10), or maximizing Welch’s t -statistic; the former choice balances noise and bias, while the latter focuses on avoiding bias during filtering. We use Welch’s t -test for image denoising; extending our approach to variance denoising, we also need to test for differences in variances, building on Bonett’s test, Eq. (7). We show two applications for denoised variance estimates: (1) an adaptive sampling strategy (§5.3.1) following the statistical theory on stratified sampling, assigning samples to pixels according

to the Neyman allocation, Eq. (9), and (2) cascaded denoising (§5.3.2), where we employ denoised variances to improve image denoising. Finally, for image denoising, we also correct our variance estimates by subtracting a linear approximation of explained variance (§5.2). Throughout this work, we build on mean squared error (MSE) minimization because this metric (separable into bias and variance) is most accessible from a theoretical point of view. Extending our analysis to more elaborate metrics that better account for human perception is an interesting avenue for future work.

4 Background and Notation

In this section, we briefly summarize basic concepts from the statistics literature, as well as the core ideas of the statistical denoising approach by Sakai et al. [2024]. Given estimates of the per-pixel radiance $\hat{\theta}_i$ computed by MC integration of the rendering equation [Kajiya 1986; Veach 1997], they start from a general denoising filter:

$$\tilde{\theta}_j = \sum_i w_{ij} \hat{\theta}_i, \quad (1)$$

where the denoised estimator $\tilde{\theta}_j$ for pixel j is a convex combination of noisy estimators $\hat{\theta}_i$ of multiple (neighboring) pixels with weights w_{ij} such that all $w_{ij} \geq 0$ and $\sum_i w_{ij} = 1$. These weights comprise “a priori weights” ρ_{ij} , see also Eq. (21), defining a base filter that includes a priori information, such as G-buffers (normals or albedos), and the “statistical membership function” m_{ij} , which allows or disallows combination based on statistical testing:

$$w_{ij} = \rho_{ij} m_{ij} / (\sum_i \rho_{ij} m_{ij}). \quad (2)$$

As in Sakai et al. [2024], we require $m_{ij} = m_{ji}$ (symmetry, ensuring energy conservation), $\text{Var}(\hat{\theta}_j) \rightarrow 0 \implies m_{ij} = 0$ if $\|\theta_j - \theta_i\| > 0$ (convergence), and $m_{jj} = 1$ (identity), where θ_j is the ground-truth radiance of pixel j . To facilitate parallel execution and reduce memory usage, we only allow pair-wise evaluation of the membership function, i.e., m_{ij} only depends on statistics computed for pixel i and j (but must not use statistics of other pixels). Moreover, all these pixel statistics must support online calculation, i.e., they can be updated one sample at a time without storing individual samples.

In this statistical denoising framework, the goal is to choose weights w_{ij} , or more specifically, membership functions m_{ij} such that the image’s MSE is minimized:

$$\{w_{ij}^*\} = \arg \min_{\{w_{ij}\}} \sum_j \text{MSE}(\tilde{\theta}_j, \theta_j). \quad (3)$$

In this work, we focus on MSE because it can be decomposed into bias and variance terms. Sakai et al. [2024] have shown that, in theory, pair-wise MSE minimization is equivalent to using Welch’s t -test [1947] on the difference of the pixel means for m_{ij} :

$$m_{ij} = 1 \text{ if } |t| < t_{v,1-\alpha/2}, \quad m_{ij} = 0 \text{ otherwise}, \quad (4)$$

where Welch’s t -statistic $t = (\hat{\theta}_i - \hat{\theta}_j) / (\hat{\sigma}_i^2/n_i + \hat{\sigma}_j^2/n_j)^{1/2}$. Here $\hat{\sigma}_i^2 = \sum_k^{n_i} (\ell_{ik} - \hat{\theta}_i)^2 / (n_i - 1)$ is the sample variance, where ℓ_{ik} denotes the k -th sample value and n_i the number of samples for pixel i . We then compare this t -statistic against a quantile $t_{v,1-\alpha/2}$ of the Student t -distribution with degrees of freedom v and significance level α . Intuitively, the role of the test is to disallow combining pixels i and j if it detects a statistically significant difference $|t| \geq t_{v,1-\alpha/2}$.

One major drawback of this approach, which we address in the subsequent section, is that Welch’s t -test assumes normality and is highly susceptible to violations of this assumption. One common strategy is to apply the *Box–Cox transformation* to the (rendered) samples to move their distribution closer to normality, applying a power transformation to each sample value:

$$\ell_{ik}^{(\lambda)} = (\ell_{ik}^\lambda - 1)/\lambda \text{ if } \lambda \neq 0, \quad \ell_{ik}^{(\lambda)} = \ln(\ell_{ik}) \text{ if } \lambda = 0, \quad (5)$$

where the transformation parameter λ controls the effect of the transformation. A well-chosen λ can effectively compress or stretch sample values, thereby correcting skewness or kurtosis (heavy tails) and increasing normality. Thus, by considering statistics of Box–Cox-transformed samples, we can improve the validity of Welch’s t -test and consequently improve overall denoising quality.

4.1 Jarque–Bera Test for Normality

The effectiveness of the Box–Cox transformation in improving normality hinges on selecting an appropriate value for λ . In our work, we focus on quantifying deviations from normality to guide this selection. To be efficient in practice, however, this quantification must be computed without storing or processing individual samples, which means that many classical normality tests, including the well-known Kolmogorov–Smirnov test [Massey 1951], are unsuitable for our purpose. Conversely, the Jarque–Bera (JB) test relies only on estimates of skewness and kurtosis, which can be calculated online. The JB statistic—including the corrections by Urzua [1996]—is:

$$JB(\hat{\gamma}, \hat{\beta}, n) = \frac{\hat{\gamma}^2(n^2 + 4n + 3)}{6(n - 2)} + \frac{(\hat{\beta}(n + 1) - 3n + 3)^2(n^2 + 8n + 15)}{24n(n - 2)(n - 3)}, \quad (6)$$

where the sample skewness $\hat{\gamma} = \hat{\mu}_3/\hat{\mu}_2^{3/2}$ and kurtosis $\hat{\beta} = \hat{\mu}_4/\hat{\mu}_2^2$, using the l -th central moment $\hat{\mu}_l = \sum_{k=1}^n (\ell_k - \hat{\theta})^l/n$. Usually, the JB statistic is then compared to a quantile of the χ^2 -distribution. In this paper, we do not perform this test directly, but instead use the JB statistic to quantify normality of different distributions during multi-transform denoising.

4.2 Bonett’s Test for Equal Variances

Besides testing for equality (or difference) of means using Welch’s t -test, extending the statistical denoising concept to denoising variance estimates requires us to also test for equal variances in §5.3. While the t -test compares *absolute* differences, variances are best compared by considering their *relative* difference: we might ask, for example, whether values in one distribution are *twice* as spread out as those in another. Consequently, the statistics literature focuses on comparing *ratios* of variances, rather than differences, providing various classical statistical tests for equal variances [Bartlett 1937; Hartley 1950; Snedecor and Cochran 1989]. These classical tests are, however, also highly sensitive to non-normality. While the Box–Cox transformation improves normality, it cannot make any distribution perfectly normal. We therefore use a more recent test by Curto [2023], which is based on Bonett’s variance confidence interval for non-normal distributions [Bonett 2006]. This test involves constructing a confidence interval for the variance ratio

(i.e., $\exp(\hat{\sigma}_\Delta^2 \pm \hat{s}e_\Delta)$) and checking whether this interval contains 1:

$$\exp(\hat{\sigma}_\Delta^2 - \hat{s}e_\Delta) \leq 1 \leq \exp(\hat{\sigma}_\Delta^2 + \hat{s}e_\Delta), \text{ where} \quad (7)$$

$\hat{\sigma}_\Delta^2 = |\ln(c_i \hat{\sigma}_i^2) - \ln(c_j \hat{\sigma}_j^2)|$, $\hat{s}e_i = c_i((\hat{\beta}_i - (n_i - 3)/n_i)/(n_i - 1))^{1/2}$, $\hat{s}e_\Delta = z_{1-\alpha/2}(\hat{s}e_i^2 + \hat{s}e_j^2)^{1/2}$, $c_i = n_i/(n_i - z_{1-\alpha/2})$, and $z_{1-\alpha/2}$ is the $(1 - \alpha/2)$ -quantile of the normal distribution. The variances are considered equal if the inequalities in Eq. (7) hold and significantly different otherwise. We have compared various confidence intervals [Bonett 2006; Burch 2017; Curto 2023] for the variance in extensive simulation studies and found that Bonett’s test provided the most accurate and consistent results for our purposes.

4.3 Neyman Allocation

Finally, we consider the problem of dynamically allocating a sample budget to reduce error: a problem relevant for our adaptive sampling approach in §5.3.1. This problem is closely related to the statistical concept of stratified sampling: given a total sample budget n , the goal is to allocate a certain number n_j^* to each stratum j , where $n = \sum_j n_j^*$ and $n_j^* > 0$, such as to minimize the error of an estimate of the population mean *over all strata*:

$$n_j^* = \arg \min_{\{n_j\}} \sum_j \left(\frac{N_j}{N} \right)^2 \frac{\sigma_j^2}{n_j}, \text{ s. t. } \sum_j n_j^* = n, \quad (8)$$

where N_j is the size of stratum j and $N = \sum_j N_j$. Neyman [1934] has shown that the minimizer is

$$n_j^* = n N_j \sigma_j / (\sum_k N_k \sigma_k). \quad (9)$$

Having introduced the major building blocks, we develop our general statistics-based error-reduction framework in the following section, where all these components work together to tackle various issues related to noisy estimates in MC rendering.

5 Our Statistical Framework

Here, we present three complementary approaches that form the basis of our statistical framework: (§5.1) a statistically valid method to incorporate multiple sample transformations into the membership function, (§5.2) accounting for explained variance, and (§5.3) a novel membership function to denoise variance estimates. We provide an overview of our denoising framework in Fig. 2.

5.1 Multi-Transform Denoising

As discussed in §4, non-normal distributions can have a detrimental impact on the statistical tests we employ. This issue can be alleviated using the Box–Cox transformation, but the theoretically best transformation parameter may vary across the image. Previous work [Sakai et al. 2024] simply sets a single user-specified value for all pixels. Here, we generalize this approach by incorporating multiple Box–Cox parameter values for each pixel.

Note that valid pair-wise comparisons are only possible for statistics that stem from samples transformed with the same parameter λ . Ideally, one would determine the optimal parameter for each pixel pair, such as to maximize the validity of the statistical test. This approach, however, is infeasible in practice: the transformation must be applied to each sample individually and cannot be effectively applied to the resulting statistics later, which would require storing

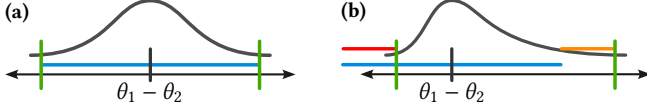


Fig. 3. The difference of normally distributed sample means (a) is again normally distributed. Here, Welch’s t -test (blue) is accurate, covering the actual percentiles (green). In the case of non-normality (b), the test becomes inaccurate in some areas: too permissive (red) or too discriminative (orange).

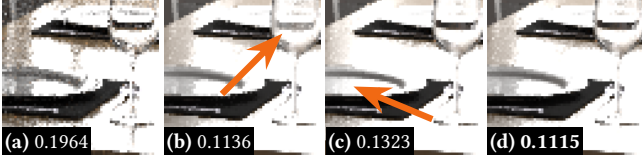


Fig. 4. **Multi-transform denoising:** Given a noisy image (a), applying one transformation may overblur some features, such as the shadow seen through the glass (b), while a different transformation may overblur others, like the shadow on the plate (c). Multi-transform denoising combines multiple transformations to prevent overblurring (d). Errors are reported as root mean square error (RMSE).

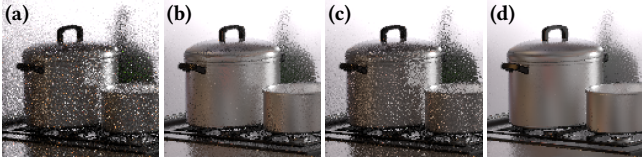


Fig. 5. **Multi-transform variance denoising:** Given noisy variance estimates (a), here for 64 samples per pixel (SPP), visualized as standard deviation, we show single transform variance denoising using Bonett’s test (b) and multi-transform variance denoising using the most discriminative transform (c). Choosing the best transform based on the Jarque–Bera statistic (d) provides the highest quality.

all samples. Beyond that, computing a good λ for every pair would be prohibitively expensive given the total number of pixel pairs.

We, instead, first select a set of parameter values $\{\lambda_m\}$, and transform each sample with each of these parameters independently during rendering. Consequently, we store only the corresponding set of statistical moments $\{\hat{\mu}_i^{(\lambda_m)}\}$ for each pixel. During denoising, we select the best parameter λ_m^* out of this set for each pixel pair. In principle, we want to bring the difference distribution for this pair closest to normality (compared to the other available λ_m values), which improves the validity of Welch’s t -test as illustrated in Fig. 3. In other words, we reduce the problem of determining the optimal Box–Cox parameter to finding the best set of moments among multiple differently-transformed sets per pixel pair.

In practice, we solve this problem in different ways, depending on the application: For denoising estimates of means (i.e., image denoising), we could evaluate the normality of the difference distribution via the Jarque–Bera statistic for the transformed moments. While maximizing normality aims to balance noise against bias, empirically, avoiding bias due to overblurring is often more effective. In our results, we therefore heuristically select λ_m^* such that the

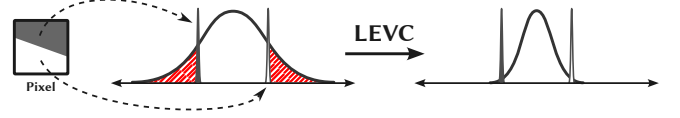


Fig. 6. Distinct bright and dark regions produce a multimodal sample distribution in a pixel (two narrow spikes in the plots). Estimating the variance based on moments approximates these spikes with a wide, unimodal distribution, thus overestimating the variance (red regions). Our linear explained-variance correction (LEVC) improves the estimate of the (unexplained) variance that is actually due to random sampling.

membership function is most discriminative, i.e., the transformation that maximizes Welch’s t -statistic. In practice this means that the membership function disallows combining pixels i and j if the test fails for *any* of the transformations in $\{\lambda_m\}$. We visually compare single and multiple transformations for denoising in Fig. 4.

For variance denoising, unfortunately, this simple heuristic does not yield satisfactory results; see Fig. 5. As detailed in §5.3, our membership function for variance denoising tests *ratios* of variance. In this case, we have to use a less discriminative strategy to reduce overall error, because variance (the mean of squared deviations) tends to be noisier than the mean. The most accurate approach would involve estimating (statistics of) the sampling distribution of the variance ratio for each pixel pair and quantify its difference to the F -distribution (i.e., the expected distribution under normality). However, to the best of our knowledge, there is no general method to obtain the required statistics about the variance-ratio distribution and no test to assess the deviation from the F -distribution. Noting that normality of the original distributions implies that their variance-ratio distribution follows the ideal F -distribution, we directly quantify deviation from normality for each pixel’s sample distribution. We then choose an appropriate transformation parameter λ_m^* for variance denoising based on the (adjusted) Jarque–Bera (JB) statistic, indicating the overall non-normality of the distributions for pixels i and j :

$$JB_{ij}^{(\lambda_m)} = JB(\hat{\gamma}_i^{(\lambda_m)}, \hat{\beta}_i^{(\lambda_m)}, n_i) + JB(\hat{\gamma}_j^{(\lambda_m)}, \hat{\beta}_j^{(\lambda_m)}, n_j), \quad (10)$$

where $\hat{\gamma}_i^{(\lambda_m)}$ and $\hat{\beta}_i^{(\lambda_m)}$ are skewness and kurtosis for pixel i transformed with λ_m and analogously for pixel j . Finally, for each pixel pair ij , we choose λ_m^* in $\{\lambda_m\}$ that minimizes $JB_{ij}^{(\lambda_m)}$.

5.2 Linear Explained-Variance Correction

When statistically denoising pixels where the integrand exhibits strong variations across the pixel’s area, excessively large variance estimates may lead to these pixels being combined with more neighbors than necessary, ultimately increasing bias. For instance, when a pixel covers part of a window (bright) and part of a window blind (dark), as illustrated in Fig. 6, the resulting variations between bright and dark samples lead to a high estimated variance. While in this example, G-buffers may suffice to avoid filtering across such an edge, more complex scene geometries, such as viewing the same situation through a mirror or hard shadows cast by directional lighting result in similar issues. We show the effect of our correction on denoised renderings of two scenes (with and without G-buffers) in Fig. 7.

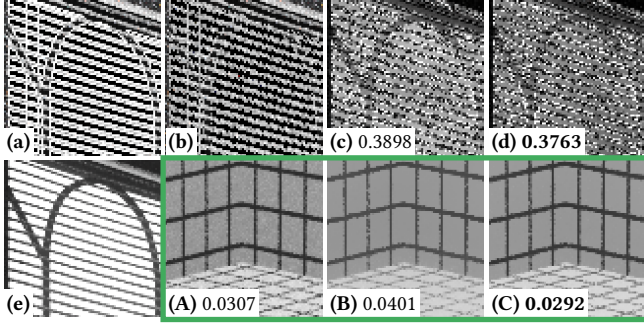


Fig. 7. **Linear explained-variance correction (LEVC)**: Total sample variance can lead to high variance estimates along feature edges (a; visualized as standard deviation), resulting in higher (squared) error in the denoised image due to overblurring (c). Our correction effectively accounts for explained variance (b) to prevent these issues, resulting in a lower error (d) of the final image (e). In another test scene, the input (A) has been denoised without LEVC and G-buffers (B), clearly showing overblurring of the grid lines. Employing our LEVC (C) mitigates this problem. Errors are reported as root mean square error (RMSE).

Here, we address this problem by decomposing the total variance of a pixel’s samples into *unexplained* and *explained* components, according to the *law of total variance* [Casella and Berger 2024]:

$$\begin{aligned} \text{Var}(L) &= \int_0^1 \text{Var}(L|X=x) dx + \int_0^1 (\mathbb{E}[L|X=x] - \mathbb{E}[L])^2 dx \\ &= \int_0^1 \sigma^2(x) dx + \int_0^1 (\ell(x) - \theta)^2 dx, \end{aligned} \quad (11)$$

where L denotes the sampled random variable, X the position within a pixel, and $\ell(x) = \mathbb{E}[L|X=x]$ describes the hypothetical means for a given position. The explained variance refers to variations of (hypothetical) ground-truth values $\ell(x)$ with the position x , i.e., variance that is explained by moving x within the pixel. We now look for an estimate of the explained variance.

Unfortunately, to directly evaluate the explained variance, we would need to store information on a sub-pixel level, i.e., either the average or the variance of L for a particular position x within a pixel. For a practical solution that can be computed online, we first write the explained variance as

$$\int_0^1 (\ell(x) - \theta)^2 dx = \int_0^1 |\ell(x) - \theta| |\ell(x) - \theta| dx. \quad (12)$$

Note that, if we were to simply evaluate either integral directly using sample values and sample mean, the result would estimate the total variance. It turns out that we can provide a linear approximation of the explained variance using the covariance $\text{Cov}(X, \mathbb{E}[L|X])$. The main motivation to do so is that the covariance can be estimated online, does not require storing information on a sub-pixel level, and naturally quantifies the dependence of $\ell(x)$ on x to first order. Specifically, we now approximate the first factor by a linear function with (unknown) incline c :

$$\int_0^1 |\ell(x) - \theta| |\ell(x) - \theta| dx \approx \int_0^1 c|x - 1/2| |\ell(x) - \theta| dx. \quad (13)$$

Noting that the covariance is

$$\text{Cov}(X, \mathbb{E}[L|X]) = \int_0^1 (x - 1/2) (\ell(x) - \theta) dx, \quad (14)$$

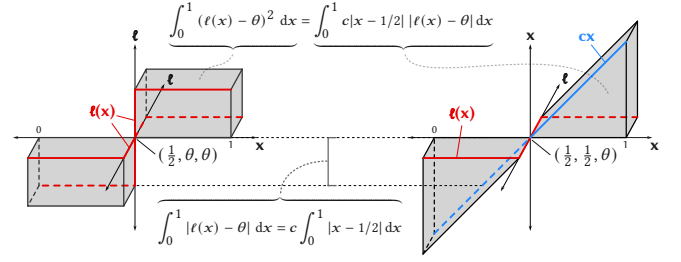


Fig. 8. Both explained variance (left) and scaled covariance (right) can be interpreted as volumes (gray). For our linear explained-variance correction, we choose the coefficient c such that our approximation (right) has equal mean absolute deviation (MAD). In this simple example, the approximation is exact; in practice, $\ell(x)$ is unknown, but we can estimate its MAD online.

we further approximate the explained variance by

$$\int_0^1 c|x - 1/2| |\ell(x) - \theta| dx \approx c |\text{Cov}(X, \mathbb{E}[L|X])|, \quad (15)$$

and finally set c to the ratio of the mean absolute deviations (MADs):

$$c = \int_0^1 |\ell(x) - \theta| dx / \int_0^1 |x - 1/2| dx = 4 \int_0^1 |\ell(x) - \theta| dx. \quad (16)$$

In summary, we linearly approximate the explained variance by scaling the covariance to match the MAD observed in the sample distribution. Fig. 8 illustrates a simple example where the explained variance can be estimated exactly. Our approximation relies only on covariance and MAD; both quantities can be estimated online during rendering. So far, we have focused on a single coordinate x . While covariances along orthogonal axes are naturally separable, the MAD is not. We choose to estimate the MAD for the entire pixel (integrating over the 2D domain of the pixel) rather than computing separate conditional MADs per axis. This approximation is exact when sample values depend on only one axis and gradually loses accuracy as the dependence between axes increases. To summarize, we improve the variance estimate $\hat{\sigma}^2$ by subtracting the linear explained-variance correction, which we approximate using the covariances in x and y direction, while the scaling factor c is four times the MAD over the entire pixel area:

$$\hat{\sigma}_{\text{LEVC}}^2 = \hat{\sigma}^2 - c (|\text{Cov}(X, \mathbb{E}[L|X])| + |\text{Cov}(Y, \mathbb{E}[L|Y])|), \quad (17)$$

where X and Y represent pixel coordinates in the respective dimensions. In Alg. 2 (suppl.), we provide pseudocode for this correction.

5.3 Variance Denoising

One of the key advantages of our statistics-based denoising framework is that it easily generalizes beyond denoising pixel colors (i.e., sample means). Here, we show how to apply the same framework to denoising variance estimates (Fig. 5), along with example applications for adaptive sampling and for further improving subsequent denoising of the mean. Analogous to mean denoising, we compute a convex combination of noisy variance estimates $\hat{\sigma}_i^2$ per pixel i to determine the denoised result $\tilde{\sigma}_j^2 = \sum_i w_{ij} \hat{\sigma}_i^2$.

Following Eq. (2), the weights w_{ij} are composed of the same a priori weight ρ_{ij} as for mean denoising (incorporating distance in image, normal, and albedo space) and a novel membership function

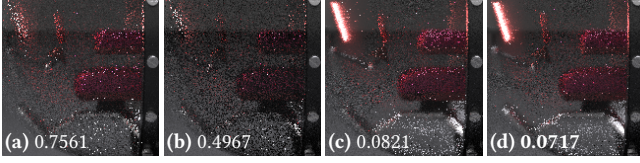


Fig. 9. **Optimal adaptive sampling:** Uniform sampling (a) often fails to adequately sample high-noise areas. Using a noisy variance estimate for adaptive sampling (b) can exacerbate the problem when variance is underestimated. While our denoised variance improves sample allocation (c), we generally obtain the best error reduction using the maximum of both noisy and denoised variance during adaptive sampling (d). Errors are reported as root mean square error (RMSE).

for the variance. Thus, the key question is how to determine an appropriate membership function m_{ij} for variance denoising. As discussed in §4.2, variance estimates should be tested for relative scale. Consequently, we use Bonett’s test, Eq. (7), evaluating the confidence interval for the variance ratio, described by $(\hat{\sigma}_\Delta^2, \hat{s}_\Delta)$, for each pixel pair ij . Noting that both $\hat{\sigma}_\Delta^2 \geq 0$ and $\hat{s}_\Delta \geq 0$, we can simplify the test by first taking the logarithm and rearranging terms to define our membership function for variance denoising:

$$m_{ij} = 1 \text{ if } \hat{\sigma}_\Delta^2 \leq \hat{s}_\Delta, \quad m_{ij} = 0 \text{ otherwise.} \quad (18)$$

5.3.1 Application I: Optimal Adaptive Sampling. One natural application is to use denoised variance estimates for adaptive sampling, i.e., allocating a number of future samples out of a given budget to each pixel in order to reduce the overall image error. If we measure this error as the total MSE (which is regularly used as an image quality metric), Neyman’s allocation provides a theoretically *optimal* solution. This result follows from minimizing image MSE, defined as $\widehat{\text{MSE}}_{\text{img}} = m^{-1} \sum_j (\hat{\theta}_j - \theta_j)^2$, where m is the number of pixels in the image and θ_j is the ground-truth pixel color (which is currently estimated by $\hat{\theta}_j$). Assuming an unbiased renderer, i.e., $E[\hat{\theta}_j] = \theta_j$, $\widehat{\text{MSE}}_{\text{img}}$ is an estimator of the average variance per pixel:

$$E[\widehat{\text{MSE}}_{\text{img}}] = m^{-1} \sum_j E[(\hat{\theta}_j - \theta_j)^2] = m^{-1} \sum_j (\sigma_j^2 / n_j). \quad (19)$$

Note that minimizing the right-hand side of Eq. (19) is equivalent to the stratified sampling problem in Eq. (8) when all strata have unit size. Therefore, Neyman’s result, Eq. (9), provides the optimal sample allocation for a given sample budget $n = \sum_j n_j^*$:

$$n_j^* = n \sigma_j / \sum_k \sigma_k. \quad (20)$$

In practice, however, we do not know the true pixel variance σ_j^2 . Therefore, we first start rendering with an equal number of samples per pixel, then estimate and denoise the pixel variances, and finally use these variance estimates, $\hat{\sigma}_j^2$, to adaptively allocate the next batch of samples according to Eq. (20). This process then repeats for each subsequent batch, iteratively increasing the total sample budget n . Monte Carlo (MC) rendering sometimes generates large outliers (fireflies) for the pixel estimates $\hat{\theta}_j$, resulting in a corresponding high variance estimate $\hat{\sigma}_j^2$. In these cases, using the denoised variance $\tilde{\sigma}_j^2$ fails to allocate enough samples to these outliers. Conversely, in cases where the variance $\hat{\sigma}_j^2$ is heavily underestimated, a pixel might not receive any additional samples. Consequently, to mitigate both

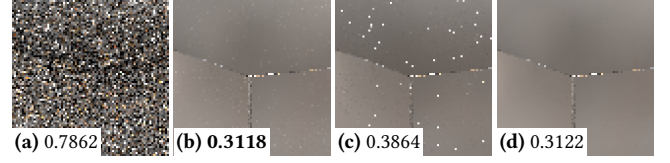


Fig. 10. **Cascaded denoising:** A noisy input image (a), here at 4 SPP, can be denoised using noisy variance estimates (b) and denoised variance estimates (c). The former leaves some residual noise while the latter tends to “single out” outliers, both due to local underestimations of variance. Using the maximum of the noisy and denoised variances often produces the most visually pleasing result (d). Errors are reported as RMSE.

of these issues, we always assign at least four samples to each pixel and allocate the remaining sample budget proportional to modified deviations $\tilde{\sigma}_j = \max(\hat{\sigma}_j, \tilde{\sigma}_j)$. We demonstrate the effect of our adaptive sampling strategy in Fig. 9 and describe how to account for the resulting varying sample sizes during denoising and further details in our supplementary document.

5.3.2 Application II: Cascaded Denoising. Welch’s t -test (employed to denoise pixel colors) relies on estimates of the pixel variances. This consideration suggests another application for our variance denoising, which we call *cascaded denoising*: instead of using noisy variance estimates directly, we first denoise them and use the resulting smoothed variances to evaluate the membership function during image denoising. As before, we must again take care of fireflies: denoising (i.e., reducing) their large variance estimates would make Welch’s test more likely to exclude their neighbors from filtering, leaving the fireflies in the denoised image, which is clearly undesirable. Consequently, we use the modified variance estimates $\tilde{\sigma}_j^2$ in practice. Replacing the “raw” variance estimates with our modified, denoised variances generally improves denoising overall, although potentially introducing a small additional bias. Consequently, our cascaded denoising tends to provide visually more pleasing results (Fig. 10) while performing slightly worse in numerical comparisons. In order to apply cascaded denoising in conjunction with our multi-transform denoising (for pixel means), as described in §5.1, we denoise the variances for each transformation separately.

6 From Theory to Practice

Our statistical framework is implemented in pbrt-v3 [Pharr et al. 2016] as an iterative renderer that delivers a specified number of samples per pixel (SPP) in each iteration (average SPP in case of adaptive sampling). While rendering in this version of pbrt runs on the CPU—where we also compute the online statistics summarized in Alg. 1 in our supplementary document—all denoising operations are implemented on the GPU. We use the algorithm by Meng [2015] to compute the first four central moments and the algorithm by Pébay [2008] for the covariances, with a slight modification: as we already know the expected value of both in-pixel coordinates, $E[x] = 1/2$, we can use it directly instead of an estimate, thereby improving the covariance estimate. For the MAD, we use a biased, but consistent, estimator based on Welford’s [1962] algorithm.

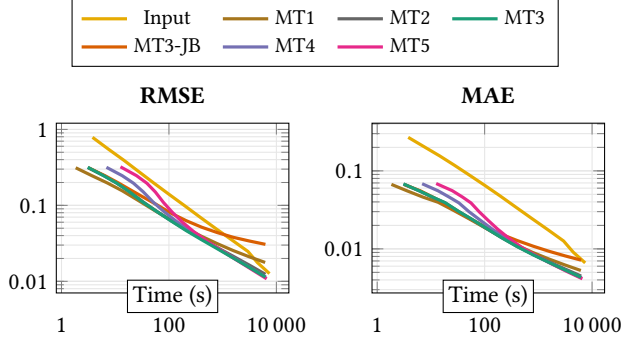


Fig. 11. Plots illustrating the trade-off associated with the used number of transforms for denoising (MCL). Here, we have used the KITCHEN scene (Fig. S10) and the following transformation sets—MT1: {0.5}, MT2: {0.5, 0.75}, **MT3** (default): {0.25, 1}, MT4: {0.25, 0.5, 0.75, 1}, MT5: {0.01, 0.25, 0.5, 0.75, 1, 3}, as well as MT3 using the Jarque–Bera statistic instead of the most-discriminative heuristic (“MT3-JB”). In the supplementary document, we provide these plots for the case of adaptive sampling.

For all our results, we use a joint bilateral base filter, with image-space positions (x, y) , albedo (r, g, b) , and (world-space) surface normals (n_x, n_y, n_z) as a priori information (G-buffers). Following the notation in Eq. (2), we set $\mathbf{p}_i = (x_i, y_i, r_i, g_i, b_i, n_{ix}, n_{iy}, n_{iz})^\top$ and use a covariance matrix $\Sigma = \text{diag}(10, 10, 0.02, 0.02, 0.02, 0.1, 0.1, 0.1)$ as well as a filter radius of 20 pixels. Our base filter, therefore, applies the a priori weights

$$\rho_{ij} = \exp \left(-0.5 (\mathbf{p}_j - \mathbf{p}_i)^\top \Sigma^{-1} (\mathbf{p}_j - \mathbf{p}_i) \right). \quad (21)$$

Multi-transform variance denoising involves potentially applying linear explained-variance correction and denoising the variance for all transformations $\{\lambda_m\}$ if cascaded denoising is used. We then perform mean denoising based on these variance estimates (potentially corrected, denoised, or both). In Eq. (4), we use $v = n_i + n_j - 2$, the upper bound of the Welch–Satterthwaite equation, as in prior work [Sakai et al. 2024]. If optimal adaptive sampling is enabled, we calculate the optimal SPPs for the next iteration, which then continues with rendering and collecting statistics. For details, please also refer to the pseudocode listings in our supplementary document.

7 Results

In our statistical error-reduction framework, all components can be combined into a unified statistical denoising pipeline. We compare our framework with two classical methods: the one-sided confidence interval by Moon et al. [2013] (“Moon CI”) and the statistical denoising technique by Sakai et al. [2024] (“StatMC”), as well as three neural denoising methods: the latest versions of Intel’s Open Image Denoise (OIDN) [Áfra 2025], NVIDIA’s OptiX AI-Accelerated Denoiser [Chaitanya et al. 2017], and the progressive denoising approach by Firmino et al. [2022] (“ProDen”). We indicate our components by ‘M’: multi-transform denoising (§5.1), ‘C’: cascaded denoising (§5.3.2), and ‘L’: linear explained-variance correction (§5.2), as well as our adaptive sampling strategy (§5.3.1) either with (“ASM”) or without (“ASN”) multi-transform denoising

of variance. For instance, we refer to the combination of all components as “ASM+MCL”. Please also refer to our supplementary document for more fine-grained comparisons, demonstrating the performance of individual components of our framework. We set $\alpha = 0.005$ for Welch’s t -test and Bonett’s test and apply two transformations for multi-transform denoising: $\lambda_1 = 0.25$ and $\lambda_2 = 1$. In Fig. 11, we illustrate the motivation behind this choice: while the improvement from one transformation (“MT1”) to two (“MT2” and “MT3”) is notable, more than two transformations (“MT4” and “MT5”) yield diminishing returns, because the gains are outweighed by the overhead of computing and utilizing the extra statistics. For Moon et al.’s [2013] method, we use $\alpha = 0.002$ as recommended in their paper. When using adaptive sampling, we start with an initial (uniformly distributed) batch of 32 SPP, as starting too early increases the chances of pixels “getting stuck” with a low number of samples due to early underestimations of variance. We then adaptively allocate batches of 32 additional (average) SPP in each iteration.

We conducted rendering and denoising on three compute servers, each equipped with an AMD EPYC 7413 CPU and an NVIDIA A40 GPU. Compared to rendering, denoising—even with multi-transform denoising of both variance and mean—requires only a fraction of the total runtime. Please note that our denoising time is largely independent of the number of rendered samples. For example, in the result shown in Fig. 12, Moon et al.’s method took 37.4 ms, OptiX 114.3 ms, OIDN 22 ms, ProDen 4129.7 ms, StatMC 25.2 ms, our MCL 77.3 ms, and our ASM+MCL 157.1 ms. We also evaluated denoising times on a commodity NVIDIA RTX 3080 Ti GPU, where all methods are only about 9.4 % slower than on the A40 GPU. In Table S1, we provide denoising times for our main results.

As our framework builds on the theory of minimizing MSE, we observe the largest improvements in our results for this metric. In our comparisons, we further report mean absolute error (MAE) and structural dissimilarity (DSSIM) (based on tone-mapped low-dynamic-range (LDR) images). In Figs. 1, 12–14, we observe that neural denoisers often overblur intricate features, whereas our method typically preserves sharper detail, particularly in scenes containing hair. Optimal adaptive sampling (ASM) provides a significant improvement in terms of MSE, evident in our plots (Fig. 15). Here, we highlight a distinct advantage of our statistical denoiser: because it already considers the sample distributions of pixels, including potentially varying sample sizes, it can synergistically improve upon the error reduction afforded by adaptive sampling with minimal overblurring. Neural denoisers, on the other hand, generally have less knowledge of the sample distributions and therefore often fail to improve upon the adaptive sampling result. Additionally, they are typically trained assuming a uniform number of samples per rendered image, and accommodating varying sample sizes (resulting from adaptive sampling) would further complicate the already complex training process for these denoisers. We illustrate this situation in the equal-time comparison given in Fig. 16 and further explore it in our supplementary document, which includes a variety of comparisons and additional ablation studies, such as extended multi-transform results and an example using the (blue-noise) Z sampler by Ahmed and Wonka [2020].

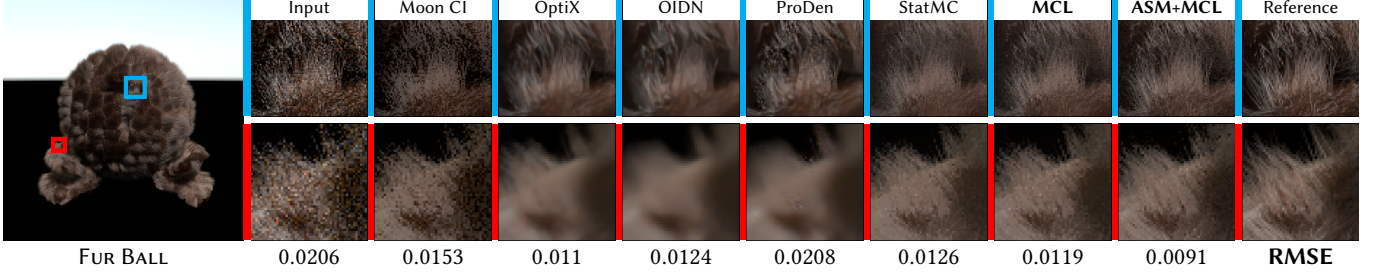


Fig. 12. We compare our full denoising pipeline (“MCL”) (with and without adaptive sampling “ASM” based on multi-transform denoised variance) to the approach by Moon et al. [2013] (“Moon CI”), NVIDIA OptiX AI-Accelerated Denoiser (“OptiX”), Intel Open Image Denoise (“OIDN”), progressive denoising [Firmino et al. 2022] (“ProDen”), and the statistical denoiser by Sakai et al. [2024] (“StatMC”). The sample budget is 64 SPP (average for ASM+MCL).

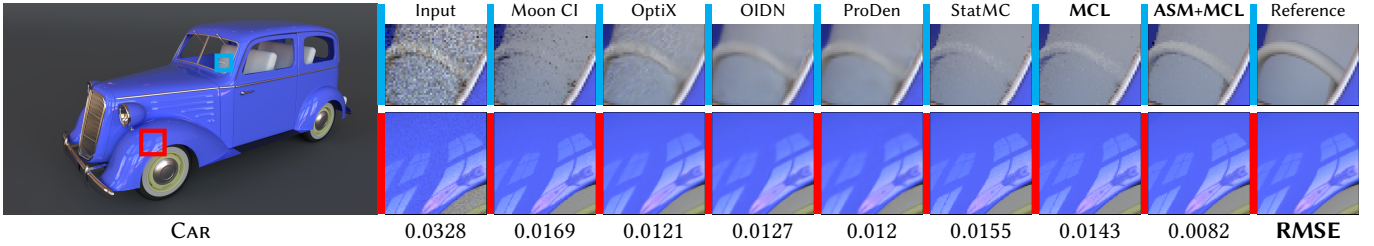


Fig. 13. CAR rendered with 128 SPP (average for ASM+MCL).

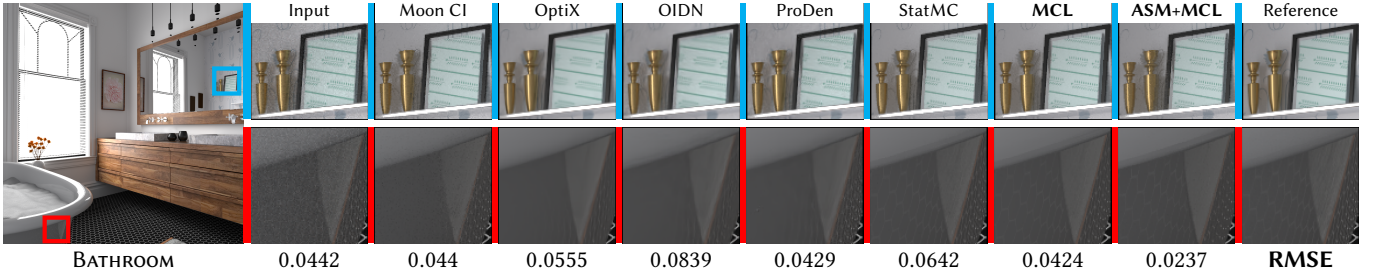


Fig. 14. BATHROOM rendered with 8192 SPP (average for ASM+MCL).

8 Discussion and Conclusion

In summary, we have presented a general statistical framework for Monte Carlo (MC) denoising that improves upon the convergence behavior of previous denoising methods and introduces a novel denoising scheme for variance estimates. Our variance denoising enables two key applications: adaptive sampling, building on the theoretically optimal Neyman allocation during rendering, and cascaded denoising, where improved variance estimates increase the performance of significance tests employed to denoise the final image. Finally, our linear explained-variance correction (LEVC) further improves upon our variance estimates and therefore our denoising performance. All our novel components combine into an efficient denoising pipeline that runs on the GPU.

While we present a denoising system that integrates several novel components, each component could, in principle, benefit other methods as well, opening multiple avenues for future work. For example, our multi-transform membership functions could be incorporated

as a kernel mask in other filters. Similarly, methods that use variance estimates as indicators of sampling noise (e.g., [Grittmann et al. 2019]) could benefit from our linear explained-variance correction (LEVC) or variance denoising. More broadly, we view statistical transformations as a general tool for tackling the difficult problem of outliers in MC rendering. Our multi-transform approach extends previous work [Sakai et al. 2024] to a wider variety of distribution shapes. In this work, we build upon the theory of mean squared error (MSE) minimization. While we observe good denoising quality for other error metrics in our results, minimizing MSE does not always lead to visually pleasing results, especially at very low sample counts—a regime where neural denoisers tend to perform well. In the future, adapting statistical tests to different (e.g., perceptual) error metrics promises further improvement in the visual quality provided by our framework. The versatility of our pipeline, in principle, also opens up avenues for future work on temporal denoising for animations.

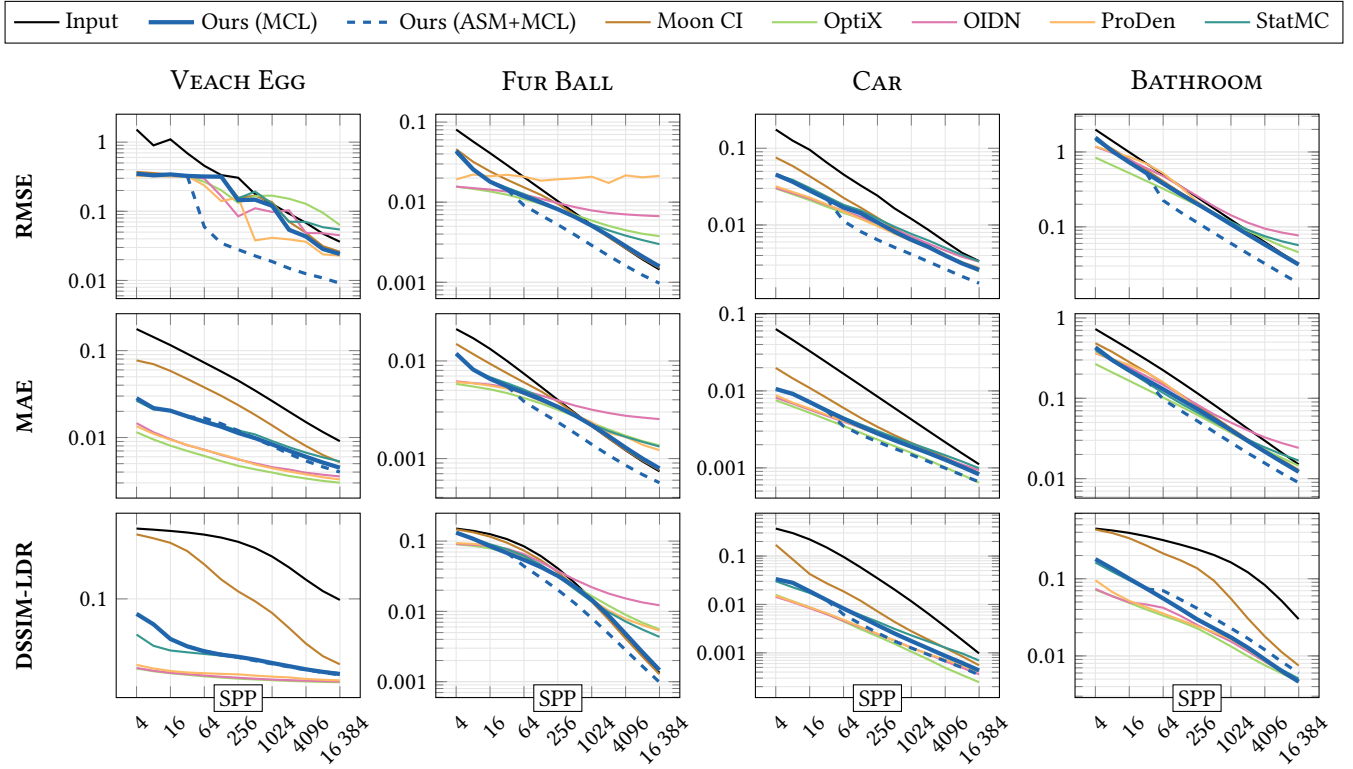


Fig. 15. Error plots comparing our denoising “MCL” (and adaptive sampling “ASM+MCL”) to the approach by Moon et al. [2013] (“Moon CI”), NVIDIA OptiX AI-Accelerated Denoiser (“OptiX”), Intel Open Image Denoise (“OIDN”), progressive denoising [Firmiro et al. 2022] (“ProDen”), and the statistical denoiser by Sakai et al. [2024] (“StatMC”). As explained in §7, for adaptive sampling we start with 32 SPP to avoid allocating too few samples due to significant underestimations of variance.

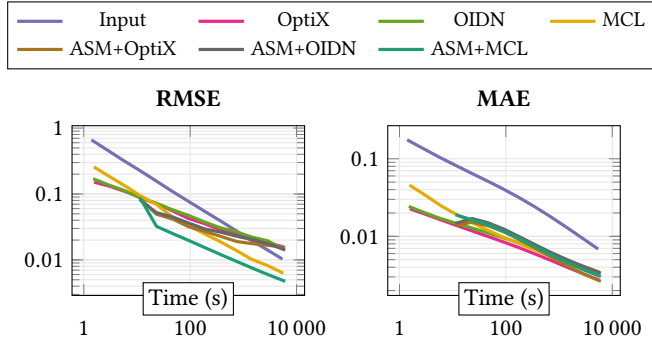


Fig. 16. Comparison between our denoising and adaptive sampling methods and neural-network denoisers using either uniformly sampled or our adaptively sampled images as inputs for the CLASSROOM scene, Fig. S9.

Acknowledgments

We thank Thomas Auzinger for providing LaTeX plugins, José Dias Curto for support with confidence intervals, and Markus Schütz for assistance with the CUDA implementation. We also thank the creators of the scenes we used: Benedikt Bitterli for “Veach, Bidir

Room” (Figs. 1, S12), “Cornell Box” (Fig. 2), and “Fur Ball” (Fig. 12); Jay-Artist for “Country Kitchen” (Figs. 4, 5, 10, S2, S10, S16); Marek for “Contemporary Bathroom” (Figs. 7, 14, S13); thecali for “4060.b Spaceship” (Fig. 9); piopis for “Old Vintage Car” (Fig. 13); Cem Yuksel for “Straight Hair” (Fig. S4) and “Curly Hair” (Fig. S5); UP3D for “Little Lamp” (Fig. S6); axel for “Glass of Water” (Fig. S7); MrChimp2313 for “Victorian Style House” (Fig. S8); NovaAshbell for “Japanese Classroom” (Fig. S9); and Bepple for “Zero-Day” (Fig. S11). Statistical simulation studies were conducted using the Austrian Scientific Computing (ASC) infrastructure. This work has been funded by the Vienna Science and Technology Fund (WWTF) [Grant ID: 1047379/ICT22028]. This research was funded in whole or in part by the Austrian Science Fund (FWF) [10.55776/F77]. For open-access purposes, the author has applied a CC BY public copyright license to any author-accepted manuscript version arising from this submission. The authors acknowledge TU Wien Bibliothek for financial support through its Open Access Funding Programme.

References

- Attila T. Áfra. 2025. Intel® Open Image Denoise. <https://www.openimagedenoise.org>.
- Abdalla G. M. Ahmed and Peter Wonka. 2020. Screen-space blue-noise diffusion of Monte Carlo sampling error via hierarchical ordering of pixels. *ACM Trans. Graph.* 39, 6 (2020), 244:1–244:15. doi:10.1145/3414685.3417881

- 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. doi:10.1145/3610548.3618177
- Steve Bako, Mark Meyer, Tony DeRose, and Pradeep Sen. 2019. Offline Deep Importance Sampling for Monte Carlo Path Tracing. *Comput. Graph. Forum* 38, 7 (2019), 527–542. doi:10.1111/CGF.13858
- M. S. Bartlett. 1937. Properties of Sufficiency and Statistical Tests. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences* 160, 901 (1937), 268–282. <http://www.jstor.org/stable/96803>
- 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. doi:10.1111/J.1467-8659.2011.01996.X
- 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. doi:10.1111/CGF.12954
- Mark R. Bolin and Gary W. Meyer. 1998. A Perceptually Based Adaptive Sampling Algorithm. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1998, Orlando, FL, USA, July 19-24, 1998*, Steve Cunningham, Walt Bransford, and Michael F. Cohen (Eds.). ACM, 299–309. doi:10.1145/280814.280924
- Douglas G. Bonett. 2006. Approximate confidence interval for standard deviation of nonnormal distributions. *Comput. Stat. Data Anal.* 50, 3 (2006), 775–782. doi:10.1016/j.csda.2004.10.003
- Malik Boughida and Tamy Boubekeur. 2017. Bayesian Collaborative Denoising for Monte Carlo Rendering. *Comput. Graph. Forum* 36, 4 (2017), 137–153. doi:10.1111/CGF.13231
- Brent D. Burch. 2017. Distribution-dependent and distribution-free confidence intervals for the variance. *Statistical Methods & Applications* 26, 4 (2017), 629–648. doi:10.1007/s10260-017-0385-z
- George Casella and Roger L. Berger. 2024. *Statistical Inference* (2nd ed.). Chapman & Hall/CRC, New York. doi:10.1201/9781003456285
- 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. doi:10.1145/3072959.3073601
- Hajin Choi, Seokpyo Hong, Inwoo Ha, Nahyup Kang, and Bochang Moon. 2024. Online Neural Denoising with Cross-Regression for Interactive Rendering. *ACM Trans. Graph.* 43, 6 (2024), 221:1–221:12. doi:10.1145/3687938
- José Dias Curto. 2023. Confidence intervals for means and variances of nonnormal distributions. *Commun. Stat. Simul. Comput.* 52, 9 (2023), 4414–4430. doi:10.1080/03610918.2021.1963448
- Holger Dammert, Daniel Sewtz, Johannes Hanika, and Hendrik P. A. Lensch. 2010. Edge-avoiding Å-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. doi: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. doi:10.1145/2532708
- Jean-Philippe Farrugia and Bernard Péroche. 2004. A Progressive Rendering Algorithm Using an Adaptive Perceptually Based Image Metric. *Comput. Graph. Forum* 23, 3 (2004), 605–614. doi:10.1111/J.1467-8659.2004.00792.X
- 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. doi:10.1111/CGF.14454
- Arthur Firmino, Jeppe Revall Frisvad, and Henrik Wann Jensen. 2023. Denoising-Aware Adaptive Sampling for Monte Carlo Ray Tracing. In *ACM SIGGRAPH 2023 Conference Proceedings, SIGGRAPH 2023, Los Angeles, CA, USA, August 6-10, 2023*, Erik Brunvand, Alla Sheffer, and Michael Wimmer (Eds.). ACM, 32:1–32:11. doi:10.1145/3588432.3591537
- Pascal Grittmann, Iliyan Georgiev, Philipp Slusallek, and Jaroslav Krivánek. 2019. Variance-aware multiple importance sampling. *ACM Trans. Graph.* 38, 6 (2019), 152:1–152:9. doi:10.1145/3355089.3356515
- Toshiya Hachisuka, Wojciech Jarosz, Richard Peter Westroffer, Kevin Dale, Greg Humphreys, Matthias Zwicker, and Henrik Wann Jensen. 2008. Multidimensional adaptive sampling and reconstruction for ray tracing. *ACM Trans. Graph.* 27, 3 (2008), 33. doi:10.1145/1360612.1360632
- H. O. Hartley. 1950. The Maximum F-Ratio as a Short-Cut Test for Heterogeneity of Variance. *Biometrika* 37, 3-4 (Dec. 1950), 308–312. doi:10.1093/biomet/37.3-4.308 [arXiv:https://academic.oup.com/biomet/article-pdf/37/3-4/308/421685/37-3-4-308.pdf](https://academic.oup.com/biomet/article-pdf/37/3-4/308/421685/37-3-4-308.pdf)
- Jon Hasselgren, Jacob Munkberg, Marco Salvi, Anjul Patney, and Aaron E. Lefohn. 2020. Neural Temporal Adaptive Sampling and Denoising. *Comput. Graph. Forum* 39, 2 (2020), 147–155. doi:10.1111/CGF.13919
- Yuchi Huo, Rui Wang, Ruzahng Zheng, Hualin Xu, Hujun Bao, and Sung-Eui Yoon. 2020. Adaptive Incident Radiance Field Sampling and Reconstruction Using Deep Reinforcement Learning. *ACM Trans. Graph.* 39, 1 (2020), 6:1–6:17. doi:10.1145/3368313
- Yuchi Huo and Sung-Eui Yoon. 2021. A survey on deep learning-based Monte Carlo denoising. *Comput. Vis. Media* 7, 2 (2021), 169–185. doi:10.1007/S41095-021-0209-9
- 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. doi: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. doi:10.1111/CGF.12029
- Alexandr Kuznetsov, Nima Khademi Kalantari, and Ravi Ramamoorthi. 2018. Deep Adaptive Sampling for Low Sample Count Rendering. *Comput. Graph. Forum* 37, 4 (2018), 35–44. doi:10.1111/CGF.13473
- 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. doi:10.1145/2366145.2366213
- Yuzhi Liang, Tao Liu, Yuchi Huo, Rui Wang, and Hujun Bao. 2024. Adaptive sampling and reconstruction for gradient-domain rendering. *Comput. Vis. Media* 10, 5 (2024), 885–902. doi:10.1007/S41095-023-0361-5
- Yu Liu, Changwen Zheng, Quan Zheng, and Hongliang Yuan. 2018. Removing Monte Carlo noise using a Sobel operator and a guided image filter. *Vis. Comput.* 34, 4 (2018), 589–601. doi: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. doi:10.1145/3105762.3105774
- Jr. Massey, Frank J. 1951. The Kolmogorov-Smirnov Test for Goodness of Fit. *J. Amer. Statist. Assoc.* 46, 253 (1951), 68–78. doi:10.1080/01621459.1951.10500769 [arXiv:https://www.tandfonline.com/doi/pdf/10.1080/01621459.1951.10500769](https://www.tandfonline.com/doi/pdf/10.1080/01621459.1951.10500769)
- Michael D. McCool. 1999. Anisotropic diffusion for Monte Carlo noise reduction. *ACM Trans. Graph.* 18, 2 (1999), 171–194. doi: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) <https://arxiv.org/abs/1510.04923>
- Don P. Mitchell. 1987. Generating antialiased images at low sampling densities. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1987, Anaheim, California, USA, July 27-31, 1987*, Maureen C. Stone (Ed.). ACM, 65–72. doi:10.1145/37401.37410
- 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. doi: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. doi: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. doi: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. doi:10.1145/2897824.2925936
- Jerzy Neyman. 1934. On the Two Different Aspects of the Representative Method: The Method of Stratified Sampling and the Method of Purposive Selection. *Journal of the Royal Statistical Society* 97, 4 (1934), 558–625. doi:10.2307/2342192
- Ryan S. Overbeck, Craig Donner, and Ravi Ramamoorthi. 2009. Adaptive wavelet rendering. *ACM Trans. Graph.* 28, 5 (2009), 140. doi: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. doi:10.1109/CADGRAPHICS.2013.24
- Philippe Pierre Pébay. 2008. *Formulas for robust, one-pass parallel computation of covariances and arbitrary-order statistical moments*. Technical Report. Sandia National Laboratories. doi:10.2172/1028931
- Matt Pharr, Wenzel Jakob, and Greg Humphreys. 2016. *Physically Based Rendering: From Theory to Implementation* (3rd ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. 1266 pages. doi:10.1016/C2013-0-15557-2
- Mahesh Ramasubramanian, Sumanta N. Pattanaik, and Donald P. Greenberg. 1999. A Perceptually Based Physical Error Metric for Realistic Image Synthesis. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1999, Los Angeles, CA, USA, August 8-13, 1999*, Warren N. Waggenspack

- (Ed.). ACM, 73–82. doi:10.1145/311535.311543
- Fabrice Rousselle, Claude Knaus, and Matthias Zwicker. 2011. Adaptive sampling and reconstruction using greedy error minimization. *ACM Trans. Graph.* 30, 6 (2011), 159. doi: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. doi: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. doi:10.1111/CGF.12219
- 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 2024, Tokyo, Japan, December 3-6, 2024*, Takeo Igarashi, Ariel Shamir, and Hao (Richard) Zhang (Eds.). ACM, 68:1–68:11. doi:10.1145/3680528.3687591
- Farnood Salehi, Marco Manzi, Gerhard Rothlin, Romann M. Weber, Christopher Schroers, and Marios Papas. 2022. Deep Adaptive Sampling and Reconstruction Using Analytic Distributions. *ACM Trans. Graph.* 41, 6 (2022), 259:1–259:16. doi:10.1145/3550454.3555515
- 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. doi: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. doi: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. doi:10.1145/2167076.2167083
- George W. Snedecor and William G. Cochran. 1989. *Statistical Methods* (8th ed.). Iowa State University Press, Ames, Iowa.
- Carlos M. Urzúa. 1996. On the correct use of omnibus tests for normality. *Economics Letters* 53, 3 (1996), 247–251. doi:10.1016/S0165-1765(96)00923-8
- Eric Veach. 1997. *Robust Monte Carlo methods for light transport simulation*. Ph.D. Dissertation. Stanford University, USA. <https://searchworks.stanford.edu/view/3911108>
- 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. doi:10.1111/CGF.13533
- Thijs Vogels, Fabrice Rousselle, Brian McWilliams, Gerhard Röhlin, Alex Harvill, David Adler, Mark Meyer, and Jan Novák. 2018. Denoising with kernel prediction and asymmetric loss functions. *ACM Trans. Graph.* 37, 4 (2018), 124. doi:10.1145/3197517.3201388
- B. L. Welch. 1947. The Generalization of ‘Student’s’ Problem when Several Different Population Variances are Involved. *Biometrika* 34, 1/2 (1947), 28–35. doi:10.2307/2332510
- B. P. Welford. 1962. Note on a Method for Calculating Corrected Sums of Squares and Products. *Technometrics* 4, 3 (1962), 419–420. doi:10.1080/00401706.1962.10490022
- Qiwei Xing and Chunyi Chen. 2020. Path Tracing Denoising Based on SURE Adaptive Sampling and Neural Network. *IEEE Access* 8 (2020), 116336–116349. doi:10.1109/ACCESS.2020.2999891
- Ruifeng Xu and Sumanta N. Pattanaik. 2005. A Novel Monte Carlo Noise Reduction Operator. *IEEE Computer Graphics and Applications* 25, 2 (2005), 31–35. doi: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. doi: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. doi:10.1007/S00371-017-1360-2
- 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. doi: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. doi:10.1111/CGF.12592