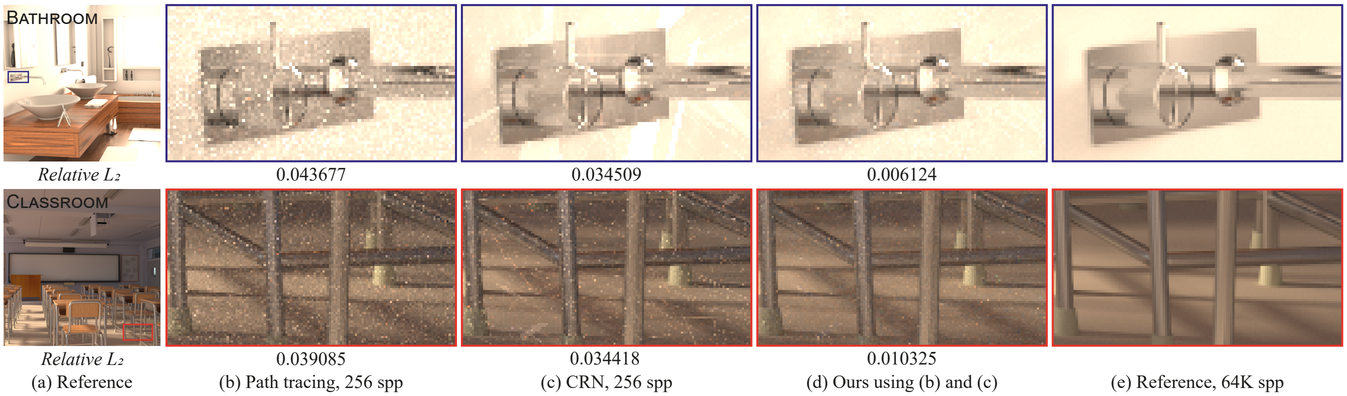
Input-Dependent Uncorrelated Weighting for Monte Carlo Denoising

Jonghee Back
Gwangju Institute of Science and Technology
South Korea
jongheeback@gm.gist.ac.kr

Binh-Son Hua
Trinity College Dublin
Ireland
binhson.hua@gmail.com

Toshiya Hachisuka
University of Waterloo
Canada
thachisu@uwaterloo.ca

Bochang Moon
Gwangju Institute of Science and Technology
South Korea
moonbochang@gmail.com



(a) Reference (b) Path tracing, 256 spp (c) CRN, 256 spp (d) Ours using (b) and (c) (e) Reference, 64K spp

Figure 1: Our denoising with a new input-dependent kernel locally weights the two input pixel estimates, path tracing with independent sampling (b) and common random numbers (CRN) (c) to produce a denoised output while preventing image edges from blurring (i.e., a limited denoising bias). The numerical accuracy is measured by a relative L_2 error using the reference images (a) and (e).

ABSTRACT

Image-space denoising techniques have been widely employed in Monte Carlo rendering, typically blending neighboring pixel estimates using a denoising kernel. It is widely recognized that a kernel should be adapted to characteristics of the input pixel estimates in order to ensure robustness to diverse image features and amount of noise. Denoising with such an input-dependent kernel, however, can introduce a bias that makes the denoised estimate even less accurate than the noisy input estimate. Consequently, it has been considered essential to balance the bias introduced by denoising and the reduction of noise. We propose a new framework to define an input-dependent kernel that departs from the existing approaches based on error estimation or supervised learning. Rather than seeking an optimal bias-noise balance as in those existing approaches, we propose to constrain the amount of bias introduced by denoising. Such a constraint is made possible by the concept of uncorrelated statistics, which has never been applied for denoising. By designing an input-dependent kernel with uncorrelated weights against the input pixel estimates, our denoising kernel can reduce data-dependent noise with a negligible amount of bias in most cases. We demonstrate the effectiveness of our method for various scenes.

CCS CONCEPTS

• **Computing methodologies → Ray tracing**.

KEYWORDS

Monte Carlo denoising, input-dependent weighting, uncorrelated weighting, unbiased denoising

ACM Reference Format:
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 Conference Papers '23), December 12–15, 2023, Sydney, NSW, Australia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3610548.3618177

1 INTRODUCTION

Monte Carlo (MC) integration is a widely adopted numerical method to estimate the light transport integral [Kajiya 1986] to simulate photorealistic rendering effects [Burley et al. 2018; Christensen et al. 2018; Kulla et al. 2018]. MC integration provides us with unbiased pixel estimates, meaning that the errors can be effectively characterized by examining the variances of these estimates. These errors

SA Conference Papers '23, December 12–15, 2023, Sydney, NSW, Australia

manifest in a rendered image as random variations among its pixels, commonly referred to as noise. Therefore, there has been a significant amount of study on reducing noise to improve the accuracy of MC rendering. A naïve approach to reducing noise is simply increasing the number of random samples for MC estimation, but it comes with the computational cost for additional samples.

One very promising alternative is image-space denoising, which takes an MC-rendered image as input and produces a denoised image as output without taking additional samples. A widely adopted denoising strategy is to exploit a denoising kernel to reduce the noise of a pixel by taking a weighted sum over its neighboring pixels. A careful definition of a denoising kernel is essential to achieve effective denoising, and virtually all the denoising kernels are designed to be adaptive to the noisy input, which makes the kernel *input-dependent.* Input-dependent kernels allow image denoising to handle pixel estimates robustly with non-uniform image features (i.e., edges) and noise, which is typical for MC-rendered images.

A major challenge is that an input-dependent kernel can introduce an arbitrarily large denoising bias to denoising results as a cost of its variance reduction. Hence, balancing the (squared) bias and variance has been considered necessary for image denoising. The latest approaches optimize the kernel using a mean squared error (MSE) estimation [Zwicker et al. 2015] or using a neural network trained over a set of training data [Huo and Yoon 2021].

There is also another class of image-space denoising approaches that reduce only the noise while avoiding introducing bias, i.e., unbiased denoising. A well-known example is the L2 reconstruction kernel in gradient-domain rendering [Kettunen et al. 2015] that takes two unbiased inputs (e.g., primal colors and image gradients) and reconstructs a combined image in an unbiased manner via solving an ordinary least squares, i.e., the screen-space Poisson reconstruction. This reconstruction kernel, however, assumes homogeneous errors in the input estimates and treats the color and gradient errors among pixels equally, i.e., uniform weighting, and thus input-independent. While this approach can produce unbiased results, it is not as efficient as input-dependent kernels and often results in visual artifacts when its assumption breaks. For example, L2 reconstruction in the presence of gradient outliers results in dipole-like artifacts.

We propose generalizing this unbiased denoising approach by relaxing the strict unbiasedness constraint. Unlike unbiased denoising, our kernel is *input-dependent* analogously to the other biased image denoisers with input-dependent kernels. Our formulation, built upon *uncorrelated statistics*, fundamentally differs from other biased denoising as we pursue a different objective where we aim at reducing noise while forcing denoising bias from an input-dependent kernel to be as small as possible instead of balancing the bias-variance. Our input-dependent kernel forces image denoising to have a bias only introduced by skewness in the distribution of input pixel estimates. Our main contributions are as follows.

- We revisit the theory of uncorrelated statistics, which states that two random variables with specific technical conditions can be dependent but uncorrelated, and we use this theory to devise an input-dependent denoising kernel that is uncorrelated with input pixel estimates.
- We show that image denoising using an input-dependent kernel, which has the conditions to be uncorrelated with input estimates, becomes unbiased when it takes input estimates symmetrically distributed.
- As a proof of concept, we present a practical implementation of such an input-dependent kernel.

The distributions of input estimates are often asymmetric in rendering, which violates the symmetric assumption required for the unbiasedness of our denoising. Nonetheless, we empirically show that this violation can be mild in practice, which leads to effective noise reduction in input pixel estimates compared to input-independent kernels while significantly restricting denoising bias.

2 RELATED WORK

Image-space denoising of MC ray tracing has become increasingly popular because of its effectiveness in variance reduction. As this approach operates in the image space, it does not require a significant modification to the underlying path-space sampling techniques required for MC ray tracing. The key idea of image-space denoising is to devise a denoising kernel that computes a weighted sum of pixel estimates so that the kernel outputs a better pixel estimate. Several weighting mechanisms exist; this section focuses on discussing existing weighting mechanisms designed for MC ray tracing and gradient-domain rendering.

Denoising for MC ray tracing. One weighting mechanism is to set a denoising kernel to be independent of its inputs, e.g., a Gaussian filter, which performs a weighted average over a local neighborhood of pixel estimates without considering the actual values of these pixels. This mechanism can reduce noise in smooth areas but works poorly in other regions, as it ignores image-space features (e.g., edges) in the inputs. It is well known that forming an input-dependent kernel, e.g., a bilateral filter, is necessary so that image denoising can handle MC-rendered images robustly with heterogeneous features and errors (e.g., noise).

A classical but well-known approach for conducting such input-dependent weighting is to exploit existing image filters, such as cross-bilateral filters [Li et al. 2012; Sen and Darabi 2012] and non-local means filtering [Rousselle et al. 2012, 2013]), which take the pixel estimates to be weighted as input to the filters, e.g., the squared difference between the colors of a center pixel and its neighborhood. One can also generate an input-dependent kernel using a weighted local regression [Bitterli et al. 2016; Moon et al. 2014, 2016]. We refer to a comprehensive survey [Zwicker et al. 2015] that reviews classical image denoisers with different input-dependent kernels.

Another common direction for input-dependent weighting relies on a deep neural network to optimize the kernel without the traditional mean squared error (MSE) optimization, e.g., a multilayer perceptron for inferring optimal bandwidths of classical filters [Kalantari et al. 2015] and normalized non-negative per-pixel kernels [Bako et al. 2017; Salehi et al. 2022; Vogels et al. 2018]. Recently, more advanced neural denoising frameworks were presented for weighting individual samples [Gharbi et al. 2019], a self-attention-based weighting using rendering-specific information [Yu et al. 2021], computationally efficient denoising using bilateral grids [Meng et al. 2020] and an affinity-based kernel [Işık et al. 2021]. Additionally, one can apply a post-denoising step that corrects a denoised image with an imbalanced bias-variance for an improved denoising result [Back et al. 2020, 2022; Firmino et al. 2022; Gu et al. 2022; Zheng et al. 2021]. We refer to a recent survey [Huo and Yoon 2021] for a comprehensive overview of learning-based denoisers.

Denoising for gradient-domain rendering. Gradient-domain rendering estimates image gradients in addition to pixel colors, and the estimated gradients can become less noisy than the primal colors through correlated sampling such as shift mapping [Kettunen et al. 2015; Lehtinen et al. 2013]. Then, the two estimated inputs can be combined via unweighted least squares, e.g., screened Poisson equation with the L2 norm [Pérez et al. 2003], a method commonly referred to as L2 reconstruction. A survey [Hua et al. 2019] thoroughly discusses correlated sampling and reconstruction.

The output image using the L2 reconstruction can have a much-reduced variance compared to the variance of the primal colors without introducing a reconstruction bias. However, the quality of the reconstruction relies entirely on the accuracy of the estimated gradients, as the reconstruction cannot robustly handle heterogeneous errors in the estimated gradients. As an alternative to the uniform weighting (i.e., L2 reconstruction), Rousselle et al. [2016] showed that an *input-independent* weighting using a half-buffer scheme [Rousselle et al. 2012] allowed an unbiased reconstruction. However, such input-independent weighting is not effective in addressing outliers caused by erroneous gradient estimates. To handle image gradients with heterogeneous errors, one can exploit input-dependent weighting using classical methods (e.g., a regularized L1 reconstruction with auxiliary features [Manzi et al. 2016]) and learning-based methods [Guo et al. 2019; Kettunen et al. 2019].

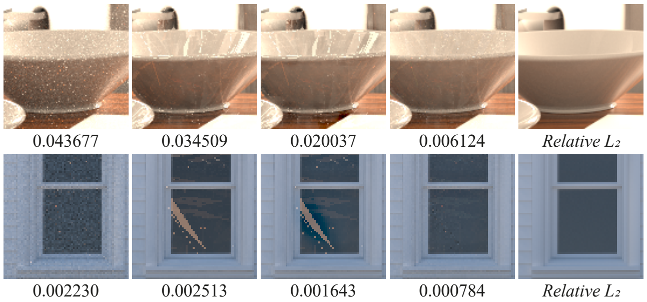
While both these classical and learning-based denoisers for primal- and gradient-domain renderings effectively reduced noise in MC-rendered images through sophisticated input-dependent weighting, they come at the cost of some additional bias as the overall denoising errors are minimized by balancing between bias and variance. In this paper, we devise the theoretical conditions of an input-dependent kernel, which makes image denoising unbiased for input estimates with symmetric distributions. It allows us to design a kernel whose denoising bias is only introduced by violating the conditions, unlike the existing biased denoisers that need a balance between bias and variance.

3 PROBLEM STATEMENT AND MOTIVATION

We aim to design an input-dependent denoising kernel that enables image denoising to restrict its bias without the help of additional optimization, such as optimal balancing of bias-variance trade-off.

We build upon a general combination function [Back et al. 2020] that takes an image y with independent pixel estimates and an image z with correlated pixel estimates to produce a combined image $\hat{\mu}$, i.e., estimates of the ground truth μ. The two inputs are assumed to be independent of each other. Specifically, a denoising estimate $\hat{\mu}_c$ at pixel c is computed as

$$\hat{\mu}_c = k_c y_c + \sum_{i\in\Omega_c} k_i y_i + \sum_{i\in\Omega_c} k_i \Delta z_{ci}, \tag{1}$$



(a) Input image (PT, 256 spp) (b) Input image (CRN, 256 spp) (c) Combination (w/ uni-kernel) (d) Combination (w/ our kernel) (e) Reference (64K spp)

Figure 2: Denoising results with a uniform kernel and our input-dependent kernel for the two inputs (a) and (b). While the uniform kernel generates an unbiased image (c), it is ineffective in reducing the heterogeneous variances inherent in the correlated input (b). Our denoising with an input-dependent kernel reduces the structural errors while avoiding excessive blurring on edges, as shown in (d).

where $\Delta z_{ci} \equiv z_c - z_i$. The combination above averages independent colors y_i and the differences Δz_{ci} of dependent colors between pixel c and pixel i. Ω_c is the set of neighboring pixels nearby pixel c, excluding the pixel c, and k is a normalized kernel that determines positive weights k_c and k_i for pixels c and i (i.e., $k_c + \sum_{i\in\Omega_c} k_i = 1$). The deep combiner by Back et al. [2020] determines the kernel k in an input-dependent manner via a deep neural network that takes the inputs y and z and is generally biased.

Multiple techniques exist to obtain a correlated image, e.g., using identical random number sequences across all pixels (path tracing with CRN), sharing traced photons via density estimation [Hachisuka et al. 2008; Jensen 1996], and blending neighboring pixel estimates using image gradients (L2 reconstruction). Our work focuses on input with unbiased estimates (e.g., path tracing with CRN) since we design a kernel for image denoising. Note that the combination (Eq. 1) serves as image denoising when it takes unbiased estimates. We will refer to the combined output $\hat{\mu}_c$ (in Eq. 1) as a denoised estimate.

Motivation. Eq. 1 provides one trivial recipe for unbiased denoising. If a denoising kernel is uncorrelated with its inputs y and z, e.g., $E[k_i\Delta z_{ci}] = E[k_i]E[\Delta z_{ci}]$ and $E[k_i y_i] = E[k_i]E[y_i]$, then the denoised estimate $\hat{\mu}_c$ becomes unbiased:

$$\begin{aligned} E[\hat{\mu}_c] &= E[k_c]E[y_c] + \sum_{i\in\Omega_c} E[k_i]E[y_i] + \sum_{i\in\Omega_c} E[k_i]E[\Delta z_{ci}] \\ &= E[k_c]\mu_c + \sum_{i\in\Omega_c} E[k_i]\mu_i + \sum_{i\in\Omega_c} E[k_i](\mu_c - \mu_i) \\ &= E\left[k_c + \sum_{i\in\Omega_c} k_i\right]\mu_c = \mu_c. \end{aligned} \tag{2}$$

A straightforward way to achieve this uncorrelatedness condition is via statistical independence, i.e., to construct an *input-independent* kernel such that the kernel is independent of inputs y and z. An example of such an input-independent kernel is a uniform kernel that treats each pixel estimate equally by $k_c = k_i = 1/(|\Omega_c| + 1)$.

While being unbiased, this type of independent weighting of pixel estimates is not effective as it does not consider the heterogeneous nature of noise (e.g., structural errors) in the correlated input z, as shown in Fig. 2. To handle the heterogeneous variances, it is desirable to have a kernel that considers input pixel estimates y and z while keeping it ideally unbiased. This observation on uncorrelated weights gives us a hint for such denoising kernels; we want them to be *input-dependent and uncorrelated at the same time*, which leads to our main idea of utilizing the concept of uncorrelated statistics.

4 INPUT-DEPENDENT AND UNCORRELATED KERNELS

In general, two random variables X and Y are uncorrelated when their covariance is zero, i.e., $cov(X,Y) = E[XY] - E[X]E[Y] = 0$, thus there is no explicit condition of dependency between X and Y. One subtlety is that X and Y being independent means they are uncorrelated, but X and Y being dependent does not necessarily mean they are correlated. Prior works showed that dependent random variables could be uncorrelated under certain conditions (e.g., [Lancaster 1959; Ostle and Steck 1959; Wolfe 1973]). We leverage this flexibility to define the weights k_i that are dependent on, but uncorrelated with, the input Δz_{ci} so that $E[k_i \Delta z_{ci}] = E[k_i]E[\Delta z_{ci}]$.

Hogg [1960] showed that the values of two functions f and g evaluated at random samples $x_1, \cdots, x_n$ satisfying the following conditions are uncorrelated:

Theorem 1 (Hogg's Theorem). *Let $x_i, \forall i \in [1,n]$ be a random sample taken from a symmetric distribution where n is the number of the random samples. Let f and g be functions of the random samples satisfying the following conditions:*

$$\begin{aligned} f(x_1+\alpha, x_2+\alpha, \cdots, x_n+\alpha) &= f(x_1, x_2, \cdots, x_n) + \alpha, \\ f(-x_1, -x_2, \cdots, -x_n) &= -f(x_1, x_2, \cdots, x_n), \end{aligned} \tag{3}$$

$$\begin{aligned} g(x_1+\alpha, x_2+\alpha, \cdots, x_n+\alpha) &= g(x_1, x_2, \cdots, x_n), \\ g(-x_1, -x_2, \cdots, -x_n) &= g(x_1, x_2, \cdots, x_n), \end{aligned} \tag{4}$$

where α is an arbitrary value. Then, two statistics from the functions, $f(x_1, x_2, \cdots, x_n)$ and $g(x_1, x_2, \cdots, x_n)$, are uncorrelated.

The theorem implies that two dependent statistics (on x_i) generated by the two different functions f and g are uncorrelated as long as they satisfy the conditions (Eqs. 3 and 4).

4.1 Input-Dependent Kernel using Uncorrelated Statistics

Let us show that image denoising via the combination (Eq. 1) is unbiased when its kernel weights the input estimates with a symmetric distribution, guided by the theory of uncorrelated statistics (Theorem 1). We first transform the combination (Eq. 1) into a simplified variant of the original one by setting $k_c = 1 - \sum_{i \in \Omega_c} k_i$:

$$\hat{\mu}_c = y_c + \sum_{i \in \Omega_c} k_i \left(\Delta z_{ci} - \Delta y_{ci} \right), \tag{5}$$

where $\Delta y_{ci} \equiv y_c - y_i$.

We then employ Theorem 1 to design a kernel function k that can take the input estimates (e.g., Δz_{ci}) as its input to produce data-dependent weights k_i. Note that one can make the kernel take sub-averages (not just sample averages such as Δz_{ci}) since we can access individual samples in rendering. To make our kernel compatible with the rendering-specific data (i.e., sub-averages), we introduce B ($B \geq 1$) sub-averages of correlated samples by $\Delta z_{ci}^1, \cdots, \Delta z_{ci}^B$, which follow an identical distribution $D(\mu_c - \mu_i, \sigma^2/n)$, where σ^2 is the variance of the correlated samples and n is the sample size for each sub-average. Note that $\Delta z_{ci}^1 = \Delta z_{ci}$ when $B = 1$.

The sub-averages can be trivially computed in rendering since individual samples per pixel can be split into B disjoint sets, i.e., $\Delta z_{ci}^j = \frac{1}{n} \sum_{s=1}^{n} \Delta z_{ci, j \times n + s}$ where $\Delta z_{ci, j \times n+s}$ is $(j \times n + s)$-th correlated sample for the estimate Δz_{ci}. We can consider that Δz_{ci} is an estimate of a function f that takes the average of B sub-averages:

$$\Delta z_{ci} = f(\Delta z_{ci}^1, \cdots, \Delta z_{ci}^B) = \frac{1}{B} \sum_{j=1}^{B} \Delta z_{ci}^j. \tag{6}$$

It is worth noting that the defined function f has the properties aforementioned in Eq. 3.

Next, we need to model the kernel k as the function g in Eq. 4, so that two statistics from the functions f and k are uncorrelated according to Theorem 1. The conditions for this kernel k (or g) can be stated in the following new theorem:

Theorem 2. *Let the kernel k be a bounded function of estimates $\Delta z_{ci}^j, \forall j \in [1,B]$, i.e., $k(\Delta z_{ci}^1, \cdots, \Delta z_{ci}^B)$ satisfying the following conditions:*

$$\begin{aligned} k(\Delta z_{ci}^1 + \alpha, \cdots, \Delta z_{ci}^B + \alpha) &= k(\Delta z_{ci}^1, \cdots, \Delta z_{ci}^B), \\ k(-\Delta z_{ci}^1, \cdots, -\Delta z_{ci}^B) &= k(\Delta z_{ci}^1, \cdots, \Delta z_{ci}^B), \end{aligned} \tag{7}$$

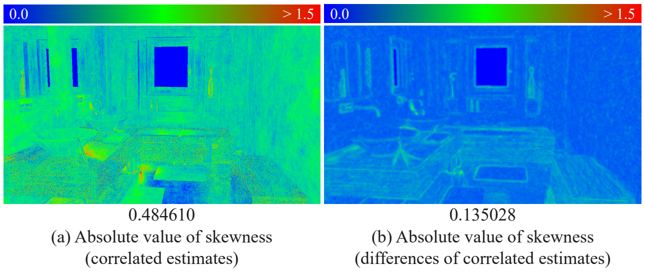
where α is an arbitrary value. By assuming that Δz_{ci}^j has a symmetric distribution, the denoised output $\hat{\mu}_c$ (Eq. 5) is an unbiased estimate of the ground truth μ_c, i.e., $E[\hat{\mu}_c] = \mu_c$.

This theorem shows how one can achieve unbiased denoising when the condition of the symmetric distribution is met. One can also set the kernel k to a function of another input Δy_{ci} or of both inputs Δy_{ci} and Δz_{ci}, analogously in the theorem. We include the proofs of Theorem 2 and these variations in the supplemental report.

The main assumption in Theorem 2 is a symmetric distribution of the difference (e.g., Δz_{ci}^j) between correlated pixel estimates, not the pixel estimates (e.g., z_i^j). It is well-known that the distributions of rendered pixel estimates with finite samples can have long right tails [Elek et al. 2019; Salehi et al. 2022], but the distribution of the differences can be less skewed than one of the pixel estimates since the two adjacent pixels tend to have a similar level of skewness (see Fig. 3-(a)). Additionally, the distribution of the Δz_{ci}^j converges to a symmetric distribution (i.e., normal distribution) as the sample size goes to infinity, according to the central limit theorem. Thus, the violation of the assumption becomes smaller with larger sample counts. This property, in turn, automatically makes the denoising with the kernel k *consistent*. We also numerically show that the bias coming from this assumption is typically much smaller than the bias in the existing methods. As such, we can use Theorem 2 as a guideline to design a new denoising kernel that has a negligible bias in most cases and is always consistent.

4.2 Example Denoising Kernel

Theorem 2 opens up a large design space of nearly unbiased denoising kernels, but as a proof of concept, we pick up a simple



0.484610
(a) Absolute value of skewness (correlated estimates)

0.135028
(b) Absolute value of skewness (differences of correlated estimates)

Figure 3: Visualization of the Pearson median skewness for the correlated estimates z_i and their differences Δz_{ci}. We compute the absolute values of skewness using 600 path-tracing images with CRN, each using 128 spp. As the skewness of differences Δz_{ci} per pixel c varies according to a neighbor i in a local window Ω_c, we visualize the average of the absolute values of the skewness. The reported numbers below the sub-figures are the averages for the visualized per-pixel skewness. The distribution of per-pixel correlated estimates is highly skewed (a), but the skewness tends to be spatially coherent, resulting in 3.6× lower skewness for the difference (b).

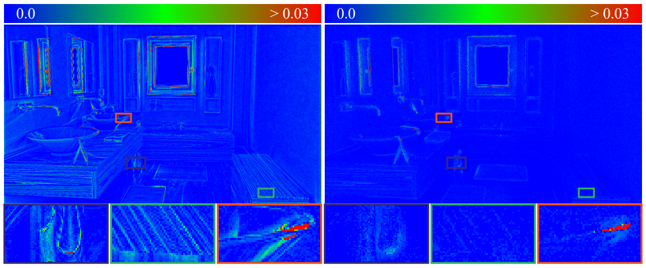
variance-based weighting with $B = 2$:

$$k(\Delta z^1_{ci}, \Delta z^2_{ci}) = \frac{1}{|\Omega_c|} \exp\left(-\gamma n \left(\Delta z^1_{ci} - \Delta z^2_{ci}\right)^2\right), \tag{8}$$

which satisfies the conditions in Theorem 2. γ is a scale parameter shared for all center pixels c. We set the denoising window Ω_c to 15×15. The squared term $(\Delta z^1_{ci} - \Delta z^2_{ci})^2$ serves as an estimated variance for Δz_{ci}. We apply the combination (Eq. 5) using this input-dependent kernel (Eq. 8) to produce a denoised output $\hat{\mu}$ from two unbiased inputs y and z.

Fig. 4 compares the denoising bias of the combination results from our input-dependent kernel and the existing learning-based scheme, deep combiner (DC) [Back et al. 2020]. As shown in the figure, our method shows a limited bias only for small regions where our assumption is violated (e.g., in the orange box of Fig. 4), unlike the existing kernel with a noticeable bias around image edges. As a result, our bias is 21× lower than DC. Note that such bias reduction is achieved via a newly designed input-dependent kernel without relying on rendering-specific information (e.g., G-buffers) and neural network based supervised learning.

Details for the parameter γ. To select the scale parameter γ (in Eq. 8), we estimated our denoising variance using the dual-buffer scheme [Rousselle et al. 2012]. Specifically, we generated four pairs of input images, i.e., four independent and correlated image pairs, and conducted our denoising with the example kernel (Eq. 8) using the first two and the other two pairs while varying the parameter $\gamma = \{0.01, 0.025, 0.05, 0.1, 0.2, 0.5, 1, 1.5, 2, 2.5\}$. Then, we selected the best γ that has the smallest relative variance, $\frac{1}{P}\sum_{c=1}^{P}(\hat{\mu}^a_c - \hat{\mu}^b_c)^2/(\bar{y}^2_c + 0.01)$. $\hat{\mu}^a$ and $\hat{\mu}^b$ are the two denoising outputs using a candidate γ. P is the total pixel count, and $\bar{y}_c = (y_c + \sum_{i\in\Omega_c} y_i)/(1 + |\Omega_c|)$. We computed the relative variance for each color channel and then averaged them.



0.011265
(a) Squared bias (DC, 64 spp)

0.000524
(b) Squared bias (ours, 64 spp)

Figure 4: Bias comparison between two input-dependent kernels: an existing learning-based kernel (DC [Back et al. 2020] (a)) and ours (b). The biases are computed using 300 input image pairs, i.e., path-tracing results with independent sampling and CRN, and the numbers below the sub-figures are the averages of per-pixel squared biases. Our method has a drastically lower bias than DC since we design an input-dependent kernel to have a restricted bias only associated with the skewness in the distribution of input estimates.

5 RESULTS AND DISCUSSION

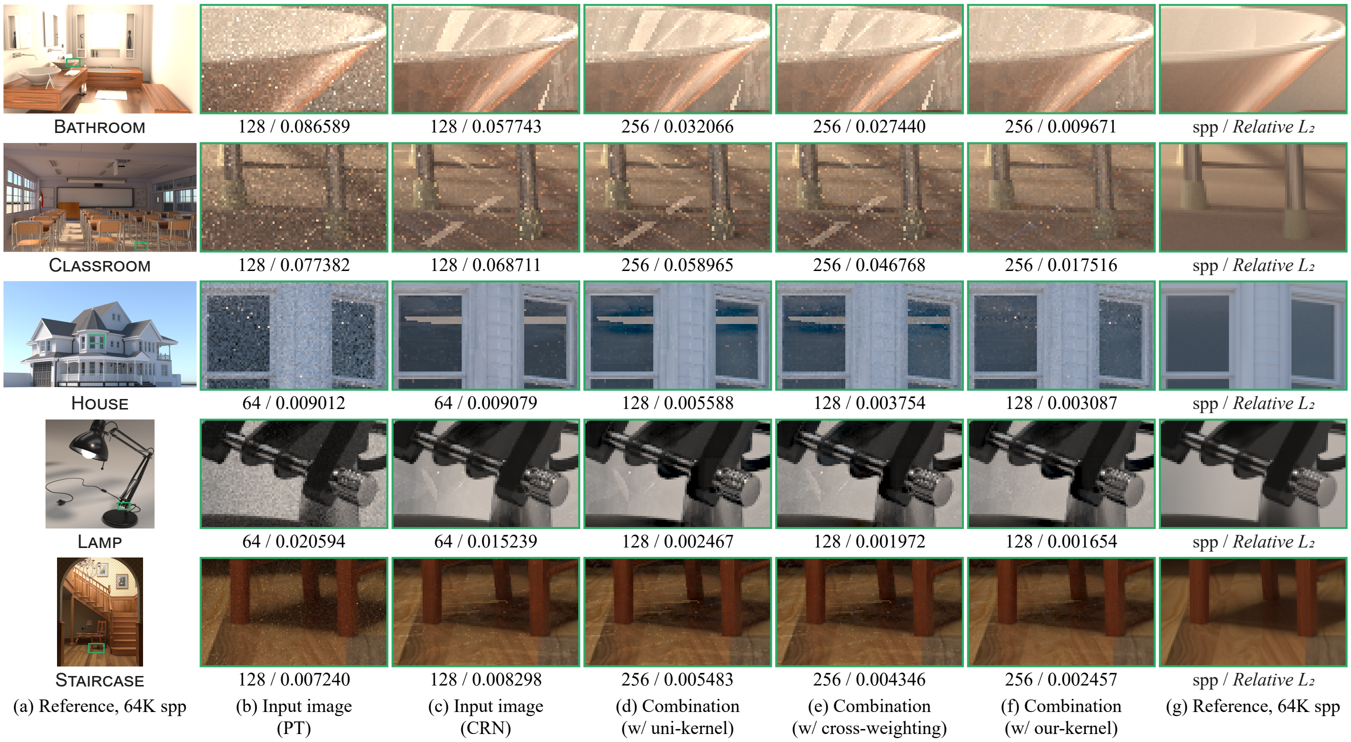
We have compared our simple denoising kernel with other baselines for Eq. 1: the two input-independent kernels (a uniform kernel and a cross-weighting kernel) that always produce unbiased results, and the input-dependent kernel predicted by a neural network [Back et al. 2020] using the pre-trained model publicly released by the authors. These baselines are selected as two extremes of input-independent and uncorrelated weights (always unbiased), and input-dependent and correlated weights (biased). Our kernel is input-dependent, and it would be uncorrelated (and thus unbiased) when the assumption in Theorem 2 holds. The inputs to all these methods are obtained by path tracing: input y with independent sampling and input z with correlated sampling using CRN.

We have compared our denoising with a recent biased denoiser, auxiliary feature guided self-attention (AFGSA) [Yu et al. 2021]. This biased denoiser takes only independent pixel estimates as input, unlike our input (i.e., independent and correlated inputs). Thus, we set the *total* sample budgets for both methods to be the same. This comparison can also be seen as a same-time comparison since our computational overhead (135 ms) is much smaller than their overhead (2 secs), given the tested image resolution (1280 × 720).

We also demonstrate that our method can be used to improve the unbiased result of gradient-domain rendering [Kettunen et al. 2015]. Specifically, we use the unbiased L2 reconstruction result from gradient-domain rendering [Kettunen et al. 2015] as our correlated estimates.

We tested five scenes (Bathroom, Classroom, House, Lamp, and Staircase) using Mitsuba renderer [Jakob 2010]. We used the Intel Xeon CPU E5-2687W and the NVIDIA GeForce RTX 3090 graphics card for all comparisons. We used the relative L_2 error [Rousselle et al. 2011] to compare the results quantitatively.

Implementation details for the comparisons. For the cross-weighting strategy, we have applied the same form of our kernel (i.e., Eq. 8) and also the same scheme for selecting the parameter γ (in Sec. 4), but



(a) Reference, 64K spp (b) Input image (PT) (c) Input image (CRN) (d) Combination (w/ uni-kernel) (e) Combination (w/ cross-weighting) (f) Combination (w/ our-kernel) (g) Reference, 64K spp

Figure 5: Comparisons with input-independent kernels, a uniform kernel (d) and a cross-weighting (e). Since these alternatives are not input-dependent kernels that take into account the heterogeneous errors of their inputs, both approaches (i.e., (d) and (e)) tend to inherit residual noise in the input estimates (b) and (c). On the other hand, our input-dependent kernel (f) allows image denoising to handle the heterogeneous errors in the input estimates robustly.

we configured the kernel and the input estimates to be independent of each other. For example, we prepared four pairs of input images, e.g., $(y^1, z^1), \cdots, (y^4, z^4)$, and then computed the kernel using the first two pairs (y^1, z^1) and (y^2, z^2) and denoised the average of the next two, i.e., (y^3, z^3) and (y^4, z^4), using the kernel, and vice-versa. Then, the two denoised outputs are averaged to produce a final denoised image.

For our denoising with the unbiased L2 reconstruction, it is required to remove the dependencies between our two inputs, independent and correlated images (i.e., the L2 reconstruction result), since the L2 reconstruction is solved from the independent estimates and their gradients. We achieve this by producing four pairs of sub-averaged independent and correlated images (y^j, z^j) with $j \in \{1..4\}$ and applying our approach twice on each decorrelated input set, i.e., one denoising using the sub-averaged independent estimates (y^1, y^2) and the sub-averaged correlated estimates (z^3, z^4), and similarly another denoising using the sub-averaged independent estimates (y^3, y^4) and the sub-averaged correlated estimates (z^1, z^2). Then, we average two denoised results to get a final output.

Comparisons with input-independent kernels. Fig. 5 compares our kernel to two input-independent kernels that result in an unbiased combination. Since the correlated estimates with high and low inter-pixel correlation are equally utilized for its denoising, uniform weighting fails to reduce the structured noise in the correlated estimates (see Fig. 5-(c) and (d)). The cross-weighting strategy slightly alleviates this issue compared to the uniform kernel. However, some residual errors from the input estimates remain since the weights are not determined by considering the inputs being actually weighted. While being slightly biased, our method can reduce the random noise from the independent estimates and the structured noise in the correlated estimates simultaneously, compared to the other input-independent kernels, thanks to our input-dependent weighting.

Comparisons with an input-dependent kernel. Fig. 6 compares our input-dependent kernel and the learning-based kernel (DC) [Back et al. 2020] for the same inputs. Our technique and the learning-based method by Back et al. [2020] have different goals. Back et al. [2020] aim at reducing relative L_2 error overall based on supervised learning. On the other hand, we theoretically identify the conditions for unbiased denoising in general and pick up one kernel that satisfies the conditions (and expect that the bias will be small in practice for those cases where the conditions are not met). Thus, we compare the techniques with respect to both the relative L_2 errors and denoising biases. DC produces 1.9× to 3.6× smaller relative L_2 errors than ours since its supervised loss is designed to reduce the L_2 errors directly. On the other hand, our method has significantly lower biases than DC, e.g., 18.2× to 39.3× lower (squared) biases.

Note that the denoising kernel we tested is merely a proof of concept - we did not employ sophisticated optimization or supervised learning to design our kernel. Nonetheless, our reduction in the squared biases (18.2× to 39.3× lower than DC) is more significant than the increase in the L_2 errors (1.9× to 3.6× higher than DC).

Comparisons with a recent biased denoiser. Fig. 7 compares our denoising with a state-of-the-art biased denoiser, AFGSA [Yu et al. 2021]. The biased denoiser, which relies on a denoising neural network trained for minimizing denoising errors without a bias constraint, produces 1.5× to 2.6× lower L_2 errors than our method. However, their improvements over our technique come with significantly higher biases than ours, i.e., 54.3× to 292.0×, which are visible as blurry features. It indicates that our method can be preferable for scenarios where high-quality rendering, which does not allow a noticeable bias (e.g., blurred edges), is necessary. Denoising with a low bias can also make visual inspection and error estimation of denoised images straightforward for practical scenarios where ground truth is unavailable. For example, visually inspecting noise (variance) is much easier than systematic errors (bias), and denoising variance can be easily estimated using the sample variance or dual-buffer estimation (Sec. 4.2).

Results for gradient-domain rendering. Fig. 8 shows denoising results for L2 reconstruction in gradient-domain rendering [Kettunen et al. 2015]. Note that the combination (Eq. 1) allows us to take any unbiased correlated estimates, such as L2 reconstructed images, as its input [Back et al. 2020]. As can be seen, the combination using our kernel mitigates heterogeneous variances in the reconstructed images, such as visual artifacts in the Bathroom and Classroom scenes and spike noise in the House scene. Our method can be regarded as a generalization of Poisson reconstruction with a more relaxed constraint on unbiasedness than the L2 reconstruction.

Convergence. Fig. 9 shows the convergences of our denoising for primal and gradient-domain renderings across different sample counts. Our approach reduces the variances of input estimates while restricting our denoising bias, which gradually diminishes with an increase in the sample size since our assumption on the input distribution, i.e., symmetrically distributed estimates, becomes exact with an infinite sample count. It makes our method consistently reduce the errors of our input estimates across the sample sizes. In other words, our denoising kernel will be consistent by design.

Computational overhead. Given a default configuration of our approach that takes path-traced images with independent and correlated sampling using CRN, our denoising overhead, including the scale parameter selection and final denoising, is 135 milliseconds for an image with 1280 × 720 resolution. Considering that the sampling time is dominant in offline rendering (e.g., 872, 844, 355, 494, and 988 milliseconds for 1 spp on Bathroom, Classroom, House, Lamp, and Staircase, respectively), the computational overhead imposed by our method is negligible.

Limitations and future work. Our denoising framework is built on top of the combination (Eq. 1) that explicitly utilizes a positive correlation between its correlated pixel estimates. Hence, our noise reduction can be bounded by the amount of inter-pixel correlation determined by a chosen correlated sampling, e.g., path tracing with CRN. As can be observed in Figs. 5, 6, and 7, our method reduces the actual errors in the input estimates but leaves some residual noise in our output estimates when the inter-pixel correlation is low. It would be interesting to adopt more advanced correlated sampling that introduces a higher correlation among pixels, e.g., path-reusing techniques via a generalized shift mapping [Bauszat et al. 2017] or resampling-based techniques [Bitterli et al. 2020; Lin et al. 2022; Ouyang et al. 2021], for more effective noise reduction. We leave this investigation for future work.

We implemented a simple variance-based weighting (Eq. 8) as an example of our input-dependent kernel (Theorem 2) and showed variance reduction with limited bias. Nevertheless, our example kernel is likely far from the best possible kernel within our framework, and the numerical results shown in this paper might not represent such best possible outcomes, e.g., higher relative L_2 errors than recent biased methods. Future research would be to derive a more sophisticated kernel with auxiliary information (e.g., albedos) or to implicitly predict a kernel that satisfies the conditions (Eq. 7) via a kernel-predicting neural network [Bako et al. 2017].

6 CONCLUSION

We have proposed a new denoising strategy for an input-dependent kernel, which allows image denoising to reduce noise in input pixel estimates while limiting denoising bias. It has been commonly believed that relying on an input-dependent kernel requires controlling a bias-variance trade-off as a necessary step since image denoising can have an arbitrarily large denoising bias. Our paper, however, shows that such an input-dependent kernel can be uncorrelated with input pixel estimates thanks to the theory of uncorrelated statistics, and one can reduce noise in input pixel estimates without the tedious task of finding an optimal bias and variance trade-off.

Empirically, our method is not a perfect unbiased denoising method. However, the source of our denoising bias becomes technically limited and explainable since it is only related to the violation of the symmetric distribution assumption of input pixel estimates, unlike existing biased image denoisers. We believe that our finding for an input-dependent kernel, which can be uncorrelated with the input estimates, can become an essential step toward unbiased denoising methods with an optimal input-dependent kernel.

ACKNOWLEDGMENTS

We appreciate the anonymous reviewers for their valuable comments and the following artists for each scene: nacimus (Bathroom), NovaZeeke (Classroom), MrChimp2313 (House), UP3D (Lamp) and Wig42 (Staircase). Bochang Moon is the corresponding author of the paper. This work was supported in part by the National Research Foundation of Korea (NRF) funded by the Korea government (MSIT) (No. RS-2023-00207939) and Ministry of Culture, Sports and Tourism and Korea Creative Content Agency (No. R2021080001).

REFERENCES

Jonghee Back, Binh-Son Hua, Toshiya Hachisuka, and Bochang Moon. 2020. Deep Combiner for Independent and Correlated Pixel Estimates. *ACM Trans. Graph.* 39, 6, Article 242 (Nov. 2020), 12 pages.

Jonghee Back, Binh-Son Hua, Toshiya Hachisuka, and Bochang Moon. 2022. Self-Supervised Post-Correction for Monte Carlo Denoising. In *ACM SIGGRAPH 2022*

Conference Proceedings (Vancouver, BC, Canada) *(SIGGRAPH '22)*. Article 18, 8 pages.

Steve Bako, Thijs Vogels, Brian Mcwilliams, Mark Meyer, Jan NováK, Alex Harvill, Pradeep Sen, Tony Derose, and Fabrice Rousselle. 2017. Kernel-Predicting Convolutional Networks for Denoising Monte Carlo Renderings. *ACM Trans. Graph.* 36, 4, Article 97 (July 2017), 14 pages.

Pablo Bauszat, Victor Petitjean, and Elmar Eisemann. 2017. Gradient-Domain Path Reusing. *ACM Trans. Graph.* 36, 6, Article 229 (Nov. 2017), 9 pages.

Benedikt Bitterli, Fabrice Rousselle, Bochang Moon, José A. Iglesias-Guitián, David Adler, Kenny Mitchell, Wojciech Jarosz, and Jan Novák. 2016. Nonlinearly Weighted First-order Regression for Denoising Monte Carlo Renderings. *Computer Graphics Forum* 35, 4 (2016), 107–117.

Benedikt Bitterli, Chris Wyman, Matt Pharr, Peter Shirley, Aaron Lefohn, and Wojciech Jarosz. 2020. Spatiotemporal Reservoir Resampling for Real-Time Ray Tracing with Dynamic Direct Lighting. *ACM Trans. Graph.* 39, 4, Article 148 (Aug. 2020), 17 pages.

Brent Burley, David Adler, Matt Jen-Yuan Chiang, Hank Driskill, Ralf Habel, Patrick Kelly, Peter Kutz, Yining Karl Li, and Daniel Teece. 2018. The Design and Evolution of Disney's Hyperion Renderer. *ACM Trans. Graph.* 37, 3, Article 33 (July 2018), 22 pages.

Per Christensen, Julian Fong, Jonathan Shade, Wayne Wooten, Brenden Schubert, Andrew Kensler, Stephen Friedman, Charlie Kilpatrick, Cliff Ramshaw, Marc Bannister, Brenton Rayner, Jonathan Brouillat, and Max Liani. 2018. RenderMan: An Advanced Path-Tracing Architecture for Movie Rendering. *ACM Trans. Graph.* 37, 3, Article 30 (Aug. 2018), 21 pages.

Oskar Elek, Manu M. Thomas, and Angus Forbes. 2019. Learning Patterns in Sample Distributions for Monte Carlo Variance Reduction. arXiv:1906.00124 [cs.GR]

Arthur Firmino, Jeppe Revall Frisvad, and Henrik Wann Jensen. 2022. Progressive Denoising of Monte Carlo Rendered Images. *Computer Graphics Forum* 41, 2 (2022), 1–11.

Michaël Gharbi, Tzu-Mao Li, Miika Aittala, Jaakko Lehtinen, and Frédo Durand. 2019. Sample-Based Monte Carlo Denoising Using a Kernel-Splatting Network. *ACM Trans. Graph.* 38, 4, Article 125 (July 2019), 12 pages.

Jeongmin Gu, Jose A. Iglesias-Guitian, and Bochang Moon. 2022. Neural James-Stein Combiner for Unbiased and Biased Renderings. *ACM Trans. Graph.* 41, 6, Article 262 (Nov. 2022), 14 pages.

Jie Guo, Mengtian Li, Quewei Li, Yuting Qiang, Bingyang Hu, Yanwen Guo, and Ling-Qi Yan. 2019. GradNet: Unsupervised Deep Screened Poisson Reconstruction for Gradient-Domain Rendering. *ACM Trans. Graph.* 38, 6, Article 223 (Nov. 2019), 13 pages.

Toshiya Hachisuka, Shinji Ogaki, and Henrik Wann Jensen. 2008. Progressive Photon Mapping. *ACM Trans. Graph.* 27, 5, Article 130 (Dec. 2008), 8 pages.

Robert V. Hogg. 1960. Certain Uncorrelated Statistics. *J. Amer. Statist. Assoc.* 55, 290 (1960), 265–267.

Binh-Son Hua, Adrien Gruson, Victor Petitjean, Matthias Zwicker, Derek Nowrouzezahrai, Elmar Eisemann, and Toshiya Hachisuka. 2019. A Survey on Gradient-Domain Rendering. *Computer Graphics Forum* 38, 2 (2019), 455–472.

Yuchi Huo and Sung-eui Yoon. 2021. A survey on deep learning-based Monte Carlo denoising. *Computational Visual Media* 7, 2 (2021), 169–185.

Mustafa Işık, Krishna Mullia, Matthew Fisher, Jonathan Eisenmann, and Michaël Gharbi. 2021. Interactive Monte Carlo Denoising Using Affinity of Neural Features. *ACM Trans. Graph.* 40, 4, Article 37 (July 2021), 13 pages.

Wenzel Jakob. 2010. Mitsuba renderer. http://www.mitsuba-renderer.org.

Henrik Wann Jensen. 1996. Global Illumination using Photon Maps. In *Rendering Techniques '96*. Springer Vienna, Vienna, 21–30.

James T. Kajiya. 1986. The rendering equation. In *ACM SIGGRAPH '86*. 143–150.

Nima Khademi Kalantari, Steve Bako, and Pradeep Sen. 2015. A Machine Learning Approach for Filtering Monte Carlo Noise. *ACM Trans. Graph.* 34, 4, Article 122 (July 2015), 12 pages.

Markus Kettunen, Erik Härkönen, and Jaakko Lehtinen. 2019. Deep Convolutional Reconstruction for Gradient-Domain Rendering. *ACM Trans. Graph.* 38, 4, Article 126 (July 2019), 12 pages.

Markus Kettunen, Marco Manzi, Miika Aittala, Jaakko Lehtinen, Frédo Durand, and Matthias Zwicker. 2015. Gradient-Domain Path Tracing. *ACM Trans. Graph.* 34, 4, Article 123 (July 2015), 13 pages.

Christopher Kulla, Alejandro Conty, Clifford Stein, and Larry Gritz. 2018. Sony Pictures Imageworks Arnold. *ACM Trans. Graph.* 37, 3, Article 29 (Aug. 2018), 18 pages.

H. O. Lancaster. 1959. Zero Correlation and Independence. *Australian Journal of Statistics* 1, 2 (1959), 53–56.

Jaakko Lehtinen, Tero Karras, Samuli Laine, Miika Aittala, Frédo Durand, and Timo Aila. 2013. Gradient-Domain Metropolis Light Transport. *ACM Trans. Graph.* 32, 4, Article 95 (July 2013), 12 pages.

Tzu-Mao Li, Yu-Ting Wu, and Yung-Yu Chuang. 2012. SURE-Based Optimization for Adaptive Sampling and Reconstruction. *ACM Trans. Graph.* 31, 6, Article 194 (Nov. 2012), 9 pages.

Daqi Lin, Markus Kettunen, Benedikt Bitterli, Jacopo Pantaleoni, Cem Yuksel, and Chris Wyman. 2022. Generalized Resampled Importance Sampling: Foundations of ReSTIR. *ACM Trans. Graph.* 41, 4, Article 75 (July 2022), 23 pages.

M. Manzi, D. Vicini, and M. Zwicker. 2016. Regularizing Image Reconstruction for Gradient-Domain Rendering with Feature Patches. *Computer Graphics Forum* 35, 2 (2016), 263–273.

Xiaoxu Meng, Quan Zheng, Amitabh Varshney, Gurprit Singh, and Matthias Zwicker. 2020. Real-time Monte Carlo Denoising with the Neural Bilateral Grid. In *Eurographics Symposium on Rendering - DL-only Track*, Carsten Dachsbacher and Matt Pharr (Eds.). The Eurographics Association.

Bochang Moon, Nathan Carr, and Sung-Eui Yoon. 2014. Adaptive Rendering Based on Weighted Local Regression. *ACM Trans. Graph.* 33, 5, Article 170 (Sept. 2014), 14 pages.

Bochang Moon, Steven McDonagh, Kenny Mitchell, and Markus Gross. 2016. Adaptive Polynomial Rendering. *ACM Trans. Graph.* 35, 4, Article 40 (July 2016), 10 pages.

Bernard Ostle and George P. Steck. 1959. Correlation Between Sample Means and Sample Ranges. *J. Amer. Statist. Assoc.* 54, 286 (1959), 465–471.

Y. Ouyang, S. Liu, M. Kettunen, M. Pharr, and J. Pantaleoni. 2021. ReSTIR GI: Path Resampling for Real-Time Path Tracing. *Computer Graphics Forum* 40, 8 (2021), 17–29.

Patrick Pérez, Michel Gangnet, and Andrew Blake. 2003. Poisson Image Editing. *ACM Trans. Graph.* 22, 3 (July 2003), 313–318.

Fabrice Rousselle, Wojciech Jarosz, and Jan Novák. 2016. Image-Space Control Variates for Rendering. *ACM Trans. Graph.* 35, 6, Article 169 (Dec. 2016), 12 pages.

Fabrice Rousselle, Claude Knaus, and Matthias Zwicker. 2011. Adaptive Sampling and Reconstruction Using Greedy Error Minimization. *ACM Trans. Graph.* 30, 6 (Dec. 2011), 1–12.

Fabrice Rousselle, Claude Knaus, and Matthias Zwicker. 2012. Adaptive Rendering with Non-Local Means Filtering. *ACM Trans. Graph.* 31, 6, Article 195 (Nov. 2012), 11 pages.

Fabrice Rousselle, Marco Manzi, and Matthias Zwicker. 2013. Robust Denoising using Feature and Color Information. *Computer Graphics Forum* 32, 7 (2013), 121–130.

Farnood Salehi, Marco Manzi, Gerhard Roethlin, Romann Weber, Christopher Schroers, and Marios Papas. 2022. Deep Adaptive Sampling and Reconstruction Using Analytic Distributions. *ACM Trans. Graph.* 41, 6, Article 259 (Nov. 2022), 16 pages.

Pradeep Sen and Soheil Darabi. 2012. On Filtering the Noise from the Random Parameters in Monte Carlo Rendering. *ACM Trans. Graph.* 31, 3, Article 18 (May 2012), 15 pages.

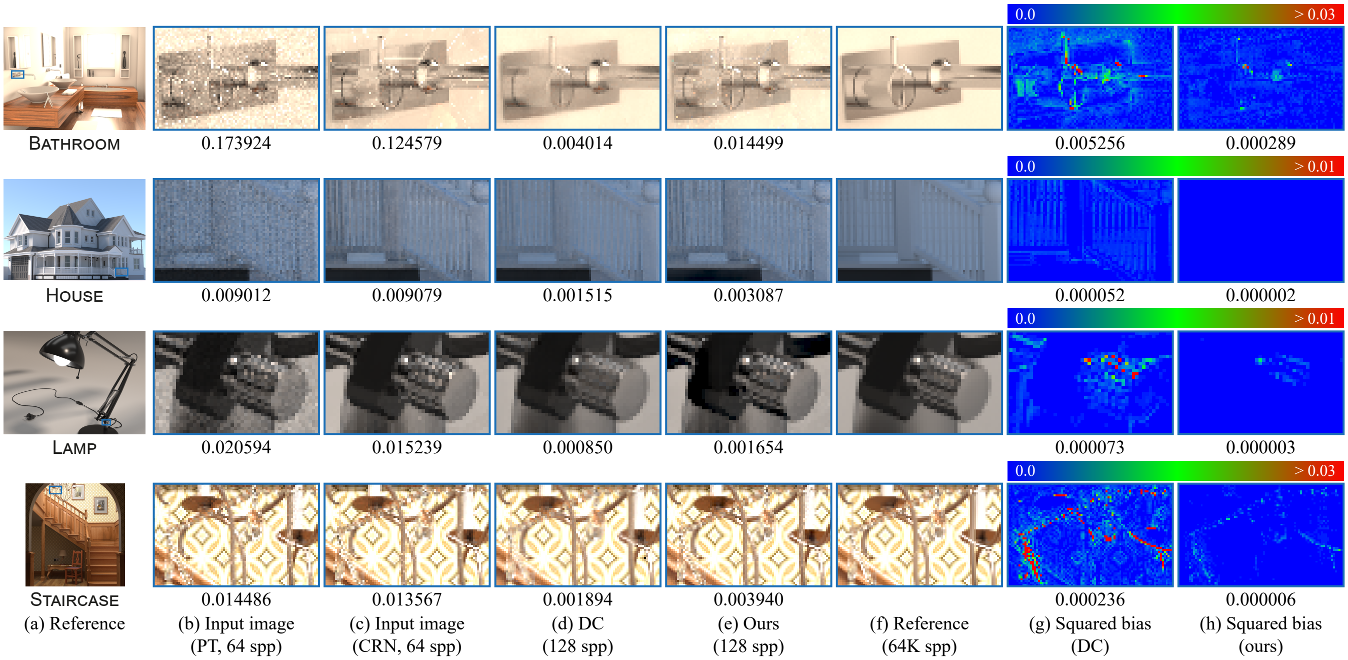
Thijs Vogels, Fabrice Rousselle, Brian Mcwilliams, Gerhard Röthlin, Alex Harvill, David Adler, Mark Meyer, and Jan Novák. 2018. Denoising with Kernel Prediction and Asymmetric Loss Functions. *ACM Trans. Graph.* 37, 4, Article 124 (July 2018), 15 pages.

Douglas A. Wolfe. 1973. Some General Results about Uncorrelated Statistics. *J. Amer. Statist. Assoc.* 68, 344 (1973), 1013–1018.

Jiaqi Yu, Yongwei Nie, Chengjiang Long, Wenju Xu, Qing Zhang, and Guiqing Li. 2021. Monte Carlo Denoising via Auxiliary Feature Guided Self-Attention. *ACM Trans. Graph.* 40, 6, Article 273 (Dec. 2021), 13 pages.

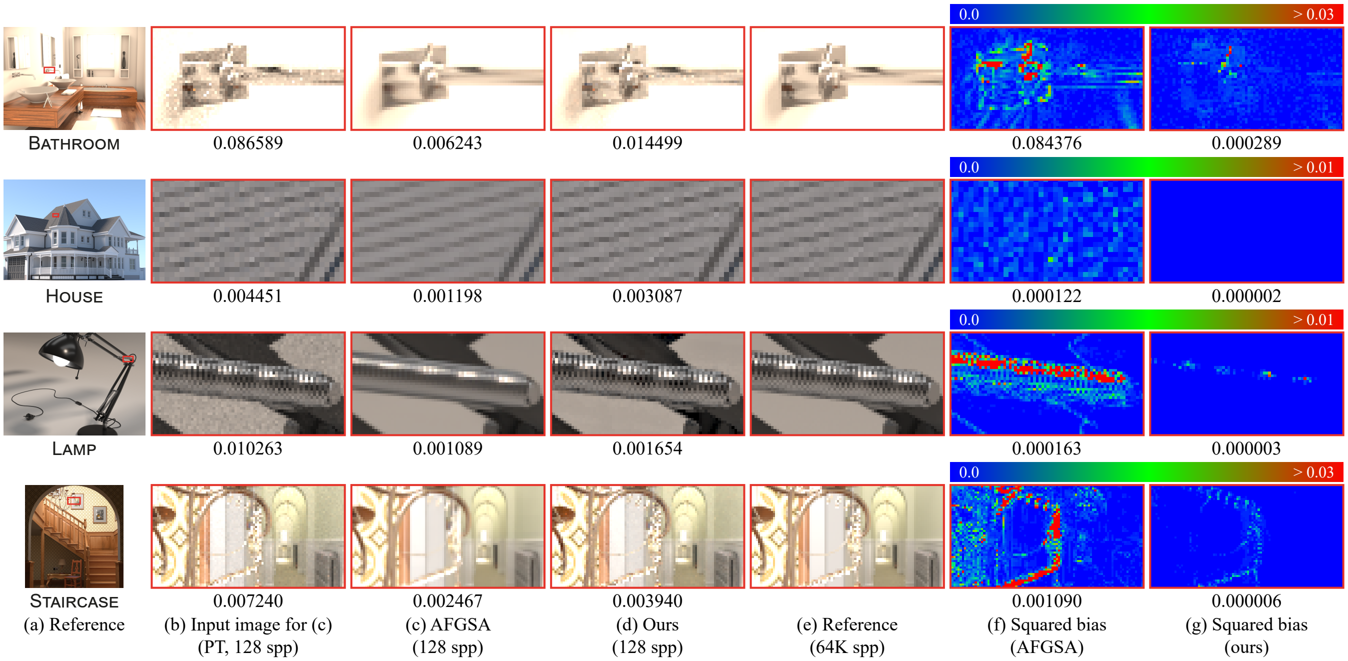
Shaokun Zheng, Fengshi Zheng, Kun Xu, and Ling-Qi Yan. 2021. Ensemble Denoising for Monte Carlo Renderings. *ACM Trans. Graph.* 40, 6, Article 274 (Dec. 2021), 17 pages.

M. Zwicker, W. Jarosz, J. Lehtinen, B. Moon, R. Ramamoorthi, F. Rousselle, P. Sen, C. Soler, and S.-E. Yoon. 2015. Recent Advances in Adaptive Sampling and Reconstruction for Monte Carlo Rendering. *Computer Graphics Forum* 34, 2 (2015), 667–681.



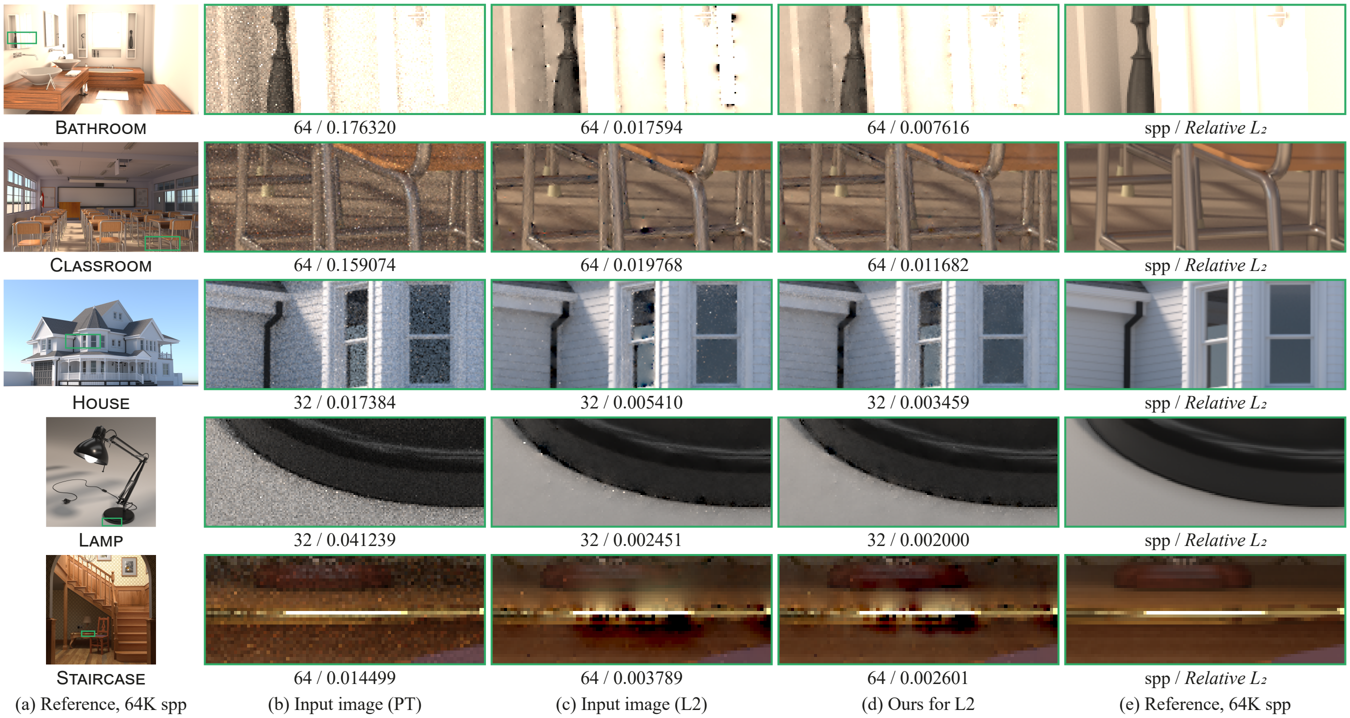
(a) Reference (b) Input image (PT, 64 spp) (c) Input image (CRN, 64 spp) (d) DC (128 spp) (e) Ours (128 spp) (f) Reference (64K spp) (g) Squared bias (DC) (h) Squared bias (ours)

Figure 6: Comparisons with a learning-based kernel (DC) [Back et al. 2020]. The numbers in (g) and (h) are the averages of the visualized per-pixel squared biases. We also report the relative L_2 errors under the input and denoised images, (b) to (e), to show the strength and weakness of the input-dependent kernels with different denoising objectives. DC shows 1.9× to 3.6× lower relative L_2 errors than ours since it optimizes its input-dependent kernel to minimize the errors via supervised learning. On the other hand, our technique, aiming at reducing denoising bias as small as possible, shows 18.2× to 39.3× lower squared biases.



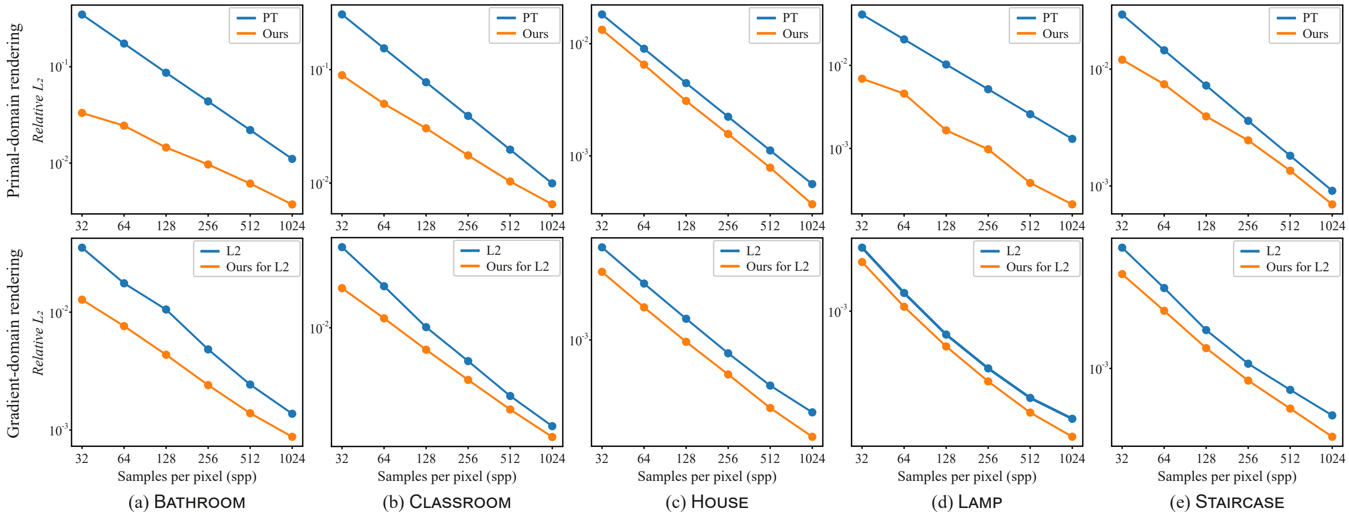
(a) Reference (b) Input image for (c) (PT, 128 spp) (c) AFGSA (128 spp) (d) Ours (128 spp) (e) Reference (64K spp) (f) Squared bias (AFGSA) (g) Squared bias (ours)

Figure 7: Comparisons with a recent biased denoiser (AFGSA) [Yu et al. 2021]. The numbers reported under the sub-figures (b) to (d) mean the relative L_2 errors, and the numbers in (f) and (g) represent the average of the per-pixel squared biases. While AFGSA shows 1.5× to 2.6× lower relative L_2 errors than our approach, our method exhibits 54.3× to 292.0× lower squared biases than AFGSA.



(a) Reference, 64K spp (b) Input image (PT) (c) Input image (L2) (d) Ours for L2 (e) Reference, 64K spp

Figure 8: Our denoising results for the L2 reconstruction [Kettunen et al. 2015]. Our method (d) mitigates the residual errors in our input image (c) without a noticeable blur on image edges.



(a) Bathroom (b) Classroom (c) House (d) Lamp (e) Staircase

Figure 9: Numerical convergence plots of our denoising for primal and gradient-domain renderings in a log-log scale.