Imperfect Image-Space Control Variates for Monte Carlo Rendering

CHANU YANG, Gwangju Institute of Science and Technology, South Korea
BOCHANG MOON, Gwangju Institute of Science and Technology, South Korea

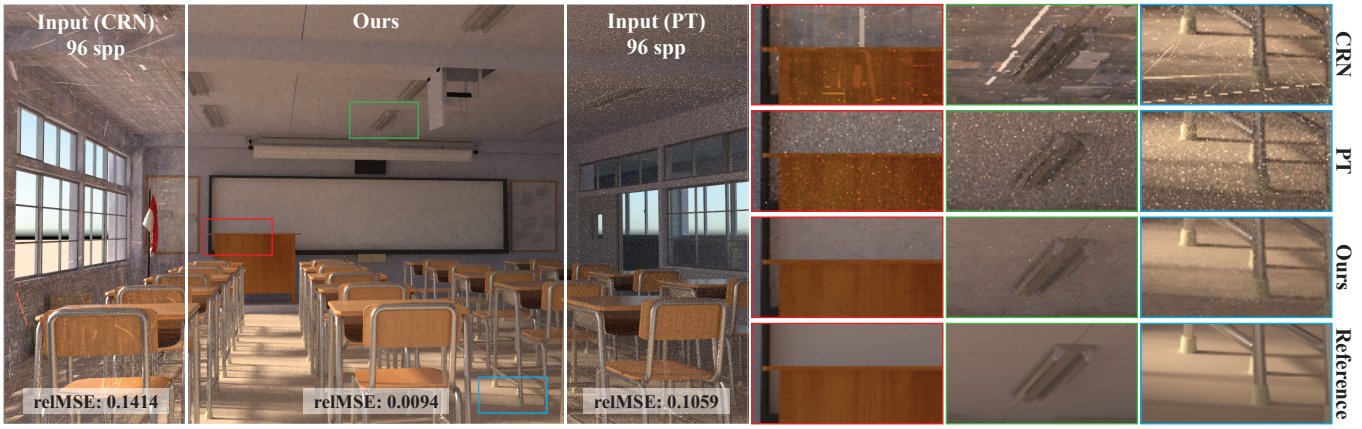


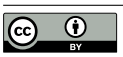
Fig. 1. Results of our image-space control variate method, which utilizes two image inputs: one generated via path tracing with independent sampling (PT) and the other via path tracing with common random numbers (CRN), where identical random seeds are assigned to all pixels, unlike in PT. Both input images are rendered with 96 samples per pixel (spp), and the relative mean squared error (relMSE) is reported to quantify numerical accuracy. Our method reduces the variance of each pixel estimate by exploiting correlations among spatially nearby pixel estimates in the CRN image, treating them as control variates. Since the expectations of these variates are unknown, they are approximated using unbiased pixel estimates from the PT image. The control variate coefficients, which determine their relative contributions, are then optimally adjusted by accounting for heterogeneous errors in the estimated expectations.

We present an image-space control variate technique to improve Monte Carlo (MC) integration-based rendering. Our method selects spatially nearby pixel estimates as control variates to exploit spatial coherence among pixel estimates in a rendered image without requiring analytic modeling of the control variate functions. Employing control variates is a classical and well-established technique for variance reduction in MC integration, typically relying on the assumption that the expectations of control variates are readily obtainable. When this condition is met, control variate theory offers a principled framework for optimizing their use by adjusting coefficients that determine the relative contribution of each control variate. However, our image-space approach introduces a technical challenge, as the expectations of the pixel-based control variates are unknown and must be estimated from additional MC samples, which are unbiased but inherently noisy. In this paper, we propose a control variate estimator designed to optimally leverage such imperfect control variates by relaxing the traditional requirement that their expectations are known. We demonstrate that our approach, which estimates the optimal coefficients while explicitly accounting for uncertainty in the expectation estimates, effectively reduces the variance of MC rendering across various test scenes.

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

Additional Key Words and Phrases: image-space control variates, multiple control variates, Monte Carlo rendering, penalized least squares

Authors' Contact Information: Chanu Yang, yangchanu@gm.gist.ac.kr, Gwangju Institute of Science and Technology, South Korea; Bochang Moon, bmoon@gist.ac.kr, Gwangju Institute of Science and Technology, South Korea.



ACM 1557-7368/2025/12-ART205
https://doi.org/10.1145/3763335

ACM Reference Format:
Chanu Yang and Bochang Moon. 2025. Imperfect Image-Space Control Variates for Monte Carlo Rendering. *ACM Trans. Graph.* 44, 6, Article 205 (December 2025), 11 pages. https://doi.org/10.1145/3763335

1 Introduction

Monte Carlo (MC) integration is widely used to numerically approximate the light transport integral [Kajiya 1986], which cannot be solved analytically. A key advantage of MC integration is that its convergence rate is independent of the dimensionality of the integral, thereby circumventing the curse of dimensionality. Since simulating global illumination involves high-dimensional integrals due to multiple light interreflections, MC integration-based rendering methods, such as path tracing [Kajiya 1986], have become the standard approach for generating photorealistic images.

In principle, reducing the approximation error in MC rendering is straightforward because the variance of the resulting estimates is reduced by half when the number of samples is doubled. However, in practice, the number of samples required to produce noise-free images is often prohibitively large, e.g., more than thousands of samples per pixel. This limitation motivates the development of optimization techniques that can effectively reduce MC variance without relying on exhaustive sampling.

One well-known variance reduction technique is the use of control variates. The core idea is to introduce an auxiliary function that closely approximates the light transport integrand in order to exploit the correlation between them. Although the control variate function can be freely designed, a common strategy is to choose one whose integral can be computed efficiently, such as a function with

an analytically solvable form (e.g., polynomials [Crespo et al. 2021; Salaün et al. 2022]). By evaluating both the original integrand and the control variate function using the same set of random samples, the control variate estimator can reduce MC variance due to their correlation.

An alternative strategy is to relax the requirement that control variates must have tractable integrals, i.e., known expectations. Instead, the unknown expectations are replaced with unbiased estimates. This approach offers greater flexibility in selecting control variate functions and removes the need to model them using simple analytical forms. A notable example is the image-space scheme proposed by Rousselle et al. [2016], which demonstrates effective variance reduction in re-rendering scenarios, such as editing specific parts of a scene. In this method, pixel estimates from a previously rendered image serve as control variates for corresponding pixels in a newly rendered image. This technique can effectively reduce variance, especially when the pre-rendered image was computed with more samples than the new one, as estimated expectations can be less noisy than the current pixel estimates.

In this paper, we adopt this flexible approach that does not rely on control variates with known expectations, and we present an image-space control variate framework in which pixel estimates themselves act as control variates, following the idea introduced by Rousselle et al. [2016]. However, unlike their method, our approach is designed for general rendering scenarios where free information, such as a pre-rendered image, is not available. The main technical contributions of this work are as follows:

- Regression-based control variate estimator: We introduce a per-pixel estimator that leverages neighboring pixel estimates as control variates using a least-squares regression. This regression locally adjusts the relative importance of each control variate based on its correlation with the original MC estimate.
- Penalized regression for heteroscedasticity: We formulate a penalized least-squares regression that estimates optimal coefficients for the control variates while accounting for heterogeneous uncertainty in the estimation of their expectations.

We demonstrate that our control variate framework, which incorporates both the varying correlation of control variates with the original MC estimates and the heterogeneity of estimation errors, achieves effective variance reduction, as illustrated in Fig. 1.

2 Related Work

Exploiting control variates has been widely adopted to reduce MC variance, which typically involves devising auxiliary functions, i.e., control variate functions. While the functions can be designed arbitrarily, a common practice is to select simple forms close to the target function so that their expectations can be computed analytically or at low cost, unlike the original function.

Such traditional design choices for control variates have been adopted for specific MC integrals in rendering, such as ambient lighting term [Lafortune and Willems 1995b], diffuse-only illumination (e.g., radiosity) [Szirmay-Kalos et al. 2001], and direct illumination, either under a constant visibility assumption [Szécsi et al. 2004] or using an approximate visibility field [Clarberg and Akenine-Möller 2008]. Lafortune and Willems [1995a] also explored a radiance approximation based on a five-dimensional tree structure, which was employed both as a control variate and as an importance sampling distribution. Spherical harmonics (SH) representations of environment lighting or BRDFs have also been adopted as control variates [Belcour et al. 2018; Mehta et al. 2012].

A more sophisticated approach to designing control variate functions, while still satisfying the requirement that their expectations can be computed easily, is to leverage neural networks [Müller et al. 2020; Subr 2021]. For example, Müller et al. [2020] integrated normalizing flows into a neural network that learns control variates and applied this neural control variate to estimate reflected radiance at a path vertex, i.e., a surface point intersected by a ray, along with a neural-network-based path-guiding [Müller et al. 2019].

The combination of control variates and multiple importance sampling (MIS) [Veach and Guibas 1995] has also been explored [Fan et al. 2006; Hua et al. 2023; Kondapaneni et al. 2019; Owen and Zhou 2000]. For example, Owen and Zhou [2000] demonstrated that MIS can be further improved using control variates defined over sampling distribution functions. Fan et al. [2006] applied this enhanced MIS framework to direct illumination. Additionally, Kondapaneni et al. [2019] derived optimal MIS weights without non-negativity constraints and established a theoretical connection between these weights and control variates. This framework was later extended into a practical solution for global illumination by Hua et al. [2023].

An alternative direction is to construct control variates using auxiliary functions of random samples in the primary sample space, rather than defining them in a local path space (e.g., at a path vertex). Crespo et al. [2021] and Salaün et al. [2022] demonstrated that designing such functions using polynomials can reduce MC noise while taking advantage of the tractable integrals of the polynomials. Crespo et al. [2021] further improved their control variate estimator by adaptively constructing the polynomials using a nested quadrature rule, and Salaün et al. [2022] showed that the control variate estimator can asymptotically outperform the straightforward MC estimator, i.e., the sample mean at a pixel, which corresponds to a constant approximation.

Image-space control variates. A simpler alternative to the aforementioned approaches is to use pixel estimates as control variates, replacing their expectations with unbiased estimates rather than explicitly constructing a control variate function with a tractable integral. Rousselle et al. [2016] adopted this approach and applied it to two specific rendering scenarios: (1) scene editing, where a pre-rendered image can be used as input, and (2) gradient-domain reconstruction, where image gradients computed by a path-shifting algorithm [Kettunen et al. 2015] are available. This control variate framework has also been extended into a recursive form for inverse rendering [Nicolet et al. 2023] and into a residual path integral for complex re-rendering scenarios, such as moving objects [Xu et al. 2024].

Inspired by prior image-space work [Rousselle et al. 2016], we introduce a control variate technique that leverages spatially nearby pixel estimates as control variates. However, our approach is designed for a more general rendering scenario, where we cannot exploit extra information (e.g., a pre-rendered image) or rely on a

strong correlation between the original pixel estimates and the control variates, as achieved via sophisticated path-shifting in gradient-domain rendering.

Instead, our approach constructs a per-pixel control variate estimator using least-squares regression. This formulation allows for the efficient adjustment of coefficients across multiple control variates, based on their varying degrees of correlation with the original MC estimates. We further extend this framework to a penalized least-squares regression that accounts for heterogeneous variances in the estimated expectations of the control variates.

Multilevel Monte Carlo. Multilevel Monte Carlo (MLMC) [Giles 2008, 2015], which performs an MC simulation with multiple levels of approximation, is related to control variate techniques, since a relatively cheap but coarse approximation serves as a control variate for a more expensive but accurate approximation [Li and Ware 2024]. In rendering, Keller [2001] introduced an interpolation scheme on an image hierarchy with different discretization levels. Recently, Dereviannykh et al. [2025] proposed a two-level MLMC framework in which radiance caching with a neural network serves as a low-level (and fast) approximation of path tracing. Unlike these MLMC techniques, our method does not require computing multiple levels of approximation with varying costs. Instead, we formulate neighboring pixel estimates as multiple control variates and investigate their optimal weighting.

3 Theoretical Background and Problem Specification

This section provides a brief theoretical overview of control variates and describes the target problem to which we apply image-space control variates. Consider a Monte Carlo (MC) integration that numerically approximates the integral of a function f over a unit domain $\mathcal{X}$:

$$F = \int_{\mathcal{X}} f(x)dx \approx \bar{f} = \frac{1}{n}\sum_{s=1}^{n} f(x_s), \tag{1}$$

where n is the number of samples, and the integrand $f(x)$ is evaluated at independent random samples x_s drawn uniformly from $\mathcal{X}$. This basic estimator, the sample mean $\bar{f}$, provides an unbiased estimate of the true value F, but it often requires a large number of samples to reduce its approximation error, i.e., the variance $\sigma^2(\bar{f}) = \sigma^2(f(x))/n$, to an acceptable level.

To reduce the variance of the MC estimator, let us consider the case where we have k control variate functions g_i, for $1 \le i \le k$, which may be correlated with the original function f. The MC integration can then be reformulated using these auxiliary functions as:

$$\begin{aligned}\hat{F} &= \frac{1}{n}\sum_{s=1}^{n}\left(f(x_s) - \sum_{i=1}^{k}\beta_i g_i(x_s)\right) + \sum_{i=1}^{k}\beta_i G_i \\ &= \bar{f} - \sum_{i=1}^{k}\beta_i(\bar{g}_i - G_i),\end{aligned} \tag{2}$$

where β_i is the coefficient assigned to the i-th control variate $\bar{g}_i$, and $G_i = E(\bar{g}_i)$ is its expectation.

Note that the same random samples x_s are used for both the target function f and control variate functions g_i to introduce correlation between $\bar{f}$ and $\bar{g}_i$. Since the strength of this correlation may vary across different variates, the coefficients β_i should be chosen accordingly. For instance, higher values should be assigned to variates that are more strongly correlated with the sample mean $\bar{f}$.

To this end, we derive the variance $\sigma^2(\hat{F})$ as a function of the coefficients β_i, assuming that the expectations G_i of the control variates $\bar{g}_i$ are known. Under this assumption, the variance can be expressed as:

$$\sigma^2(\hat{F}) = \frac{1}{n}\left(\sigma^2(f(x)) + \boldsymbol{\beta}^\top V_g \boldsymbol{\beta} - 2\boldsymbol{\beta}^\top V_{fg}\right), \tag{3}$$

where V_g is the $k\times k$ covariance matrix of the random variables $g_i(x)$, V_{fg} is the $k \times 1$ cross-covariance vector between $f(x)$ and $g_i(x)$, and $\boldsymbol{\beta} = [\beta_1, ..., \beta_k]^\top$. The derivation of this variance expression is provided in the supplementary material. Differentiating $\sigma^2(\hat{F})$ with respect to $\boldsymbol{\beta}$ yields:

$$\frac{\partial\sigma^2(\hat{F})}{\partial\boldsymbol{\beta}} = \frac{1}{n}\left(2V_g\boldsymbol{\beta} - 2V_{fg}\right). \tag{4}$$

We can then determine the optimal coefficients $\boldsymbol{\beta}^*$ by setting the derivative equal to zero, which yields:

$$\boldsymbol{\beta}^* = {V_g}^{-1}V_{fg}. \tag{5}$$

This formulation of the optimal coefficients also appears in prior work [Kondapaneni et al. 2019; Owen 2013]. Although the optimal coefficients have a closed-form solution (Eq. 5), it depends on unknown quantities (V_g and V_{fg}) that must be estimated. If this estimation is performed independently of the processed data, i.e., $\bar{f}$ and $\bar{g}_i$, then the control variate estimator (Eq. 2) becomes unbiased, since $E(\bar{g}_i) = G_i$.

In practice, however, the coefficients are typically estimated from the data itself, since the bias introduced by the statistical dependency between the coefficients β_i and the data is often negligible compared to the variance of the MC estimator [Owen 2013].

Once this bias is allowed, it becomes natural to estimate the unknown covariance terms using their sample covariances, leading to the following solution:

$$\begin{aligned}\hat{\boldsymbol{\beta}}^* &= \left(\frac{1}{n-1}X^\top X\right)^{-1}\frac{1}{n-1}X^\top\boldsymbol{y} \\ &= \left(X^\top X\right)^{-1}X^\top\boldsymbol{y},\end{aligned} \tag{6}$$

where $X^\top X/(n-1)$ and $X^\top\boldsymbol{y}/(n-1)$ serve as estimates for the unknown quantities V_g and V_{fg}, respectively. In the equation above, X is an $n\times k$ matrix whose s-th row is given by $[g_1(x_s)-\bar{g}_1, ..., g_k(x_s)-\bar{g}_k]$. The vector $\boldsymbol{y}$ is defined as $\boldsymbol{y} = [f(x_1) - \bar{f}, ..., f(x_n) - \bar{f}]^\top$.

After determining the estimated optimal coefficients $\hat{\boldsymbol{\beta}}^*$ using Eq. 6, we can substitute them into the control variate estimator (Eq. 2) to obtain improved estimates, rather than relying on the simple MC estimator (Eq. 1).

Notably, the closed-form solution (Eq. 6) corresponds to the normal equation of ordinary least squares applied to mean-centered data, i.e., $g_i(x_s) - \bar{g}_i$ and $f(x_s) - \bar{f}$. Since the optimal coefficients can be estimated directly from the data via this closed-form expression, this regression approach has been widely adopted to estimate the optimal coefficients for control variates (e.g., [Fan et al. 2006; Owen and Zhou 2000; Salaün et al. 2022]).

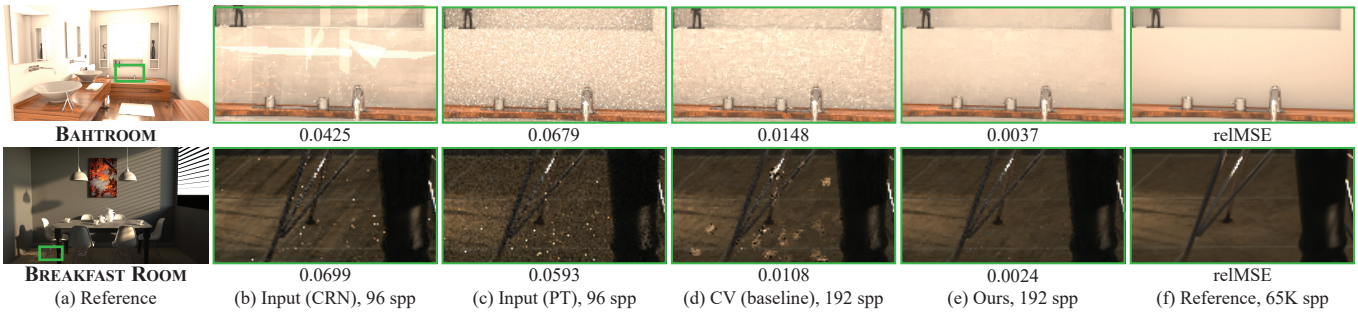


Fig. 2. Results of the image-space control variate techniques, (d) and (e). These techniques select spatially close pixel estimates from the CRN input (b) as control variates and use the corresponding pixel estimates from the PT input (c) as unbiased estimates of their expectations. The CV (baseline) method (d) estimates the optimal control variate coefficients while ignoring errors in these expectations, i.e., the variances in the PT image. In contrast, our method (e) explicitly accounts for such errors when determining the coefficients. This technical distinction allows our approach to achieve lower errors than the baseline by robustly handling heterogeneous noise in the PT image, such as fireflies. The relMSE values are computed over the full images rather than cropped regions.

Problem specification. We aim to design an image-space control variate framework grounded in the aforementioned theories related to control variates. Specifically, we use two input images, CRN and PT (as shown in Fig. 1), both generated using identical Monte Carlo rendering settings but with different random number sequences. In the CRN image, common random numbers are used across all pixels, while in the PT image, independent random sequences are applied.

Given these inputs, we apply the control variate estimator $\hat{F}$ at each pixel as a pixel-wise estimator, where $\bar{f}$ and $\bar{g}_i$ in Eq. 2 represent the sample mean estimate at the target pixel and the estimate at its i-th neighboring pixel in the CRN image, respectively. Because the estimator requires the unknown expectation G_i of the variate $\bar{g}_i$, we approximate this expectation using the corresponding pixel value from the PT image. The control variate coefficients are then computed using the least-squares regression (Eq. 6), and the estimator $\hat{F}$ is evaluated using the coefficients to generate the output. This method can be viewed as a direct image-space adaptation of classical control variate theory. Throughout this paper, we refer to this straightforward adaptation as CV (baseline).

Fig. 2(d) shows the results of CV (baseline), which produces reduced noise compared to the two input images, CRN and PT. However, the output still exhibits noticeable residual noise, primarily due to errors in the expectation estimates, especially from noisy pixels in the PT image. It is important to note that this baseline relies on optimal coefficients (Eq. 5) derived under the assumption that the expectations of the control variates are known. This assumption is violated in our image-space case since the expectations are replaced with their unbiased estimates. In the following section, we present a generalization of this baseline framework.

4 Imperfect Image-Space Control Variates

This section presents our control variate framework (Fig. 3), which explicitly accounts for uncertainty in the expectations of image-space control variates.

Given a user-specified number of samples per pixel (spp), denoted as $2n$, we perform MC rendering twice, each with n samples, to generate two input images: CRN and PT. Since we work with imperfect control variates whose expectations are unknown and

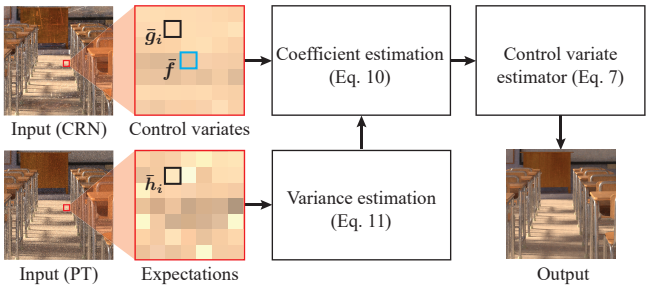


Fig. 3. Overview of our image-space control variates framework. We employ path tracing to generate two types of inputs: PT, which uses independent random samples across pixels, and CRN, which uses the same random sequences across pixels. A control variate estimator $\hat{F}$ is then applied to each pixel (indicated by a small cyan square), leveraging neighboring pixel estimates $\bar{g}_i$ from the CRN image as control variates. Corresponding pixel estimates $\bar{h}_i$ from the PT image are used in place of the true expectations of the control variates. We estimate the variances of the estimates $\bar{h}_i$ using Eq. 11, and compute the estimated optimal coefficients $\hat{\boldsymbol{\beta}}^*$ through Eq. 10, taking into account both the correlation between $\bar{f}$ and $\bar{g}_i$ and the variances of the $\bar{h}_i$. Finally, the control variate estimator (Eq. 7) is evaluated using the coefficients to produce the final output for each pixel.

must be estimated, we redefine the control variate estimator (Eq. 2) accordingly:

$$\hat{F} = \bar{f} - \sum_{i=1}^{k} \beta_i \left(\bar{g}_i - \bar{h}_i\right), \tag{7}$$

where $\bar{h}_i$ represents the estimated pixel value in the PT image, i.e., the sample mean $\bar{h}_i = 1/n \sum_{s=1}^{n} h_i(x_{i,s})$ that serves as an estimate of the unknown expectation G_i.

To apply this estimator $\hat{F}$ to each pixel, we select k neighboring pixels from the CRN image to define k control variates $\bar{g}_i$ for $1 \le i \le k$. These neighboring estimates serve as control variates to reduce the noise in the estimate $\bar{f}$ at the target pixel in the CRN. Specifically, we use an 11×11 search window centered on the target pixel, and compute the squared differences $\|\bar{f} - \bar{g}_i\|^2$ between the estimates

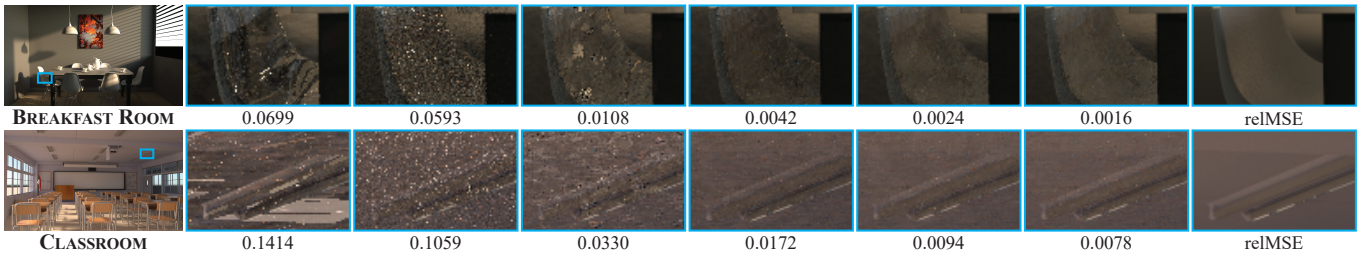


Fig. 4. Ablation study of our method. We evaluate two strategies for estimating the variances of the estimates $\bar{h}_i$: (e) using sample variances, and (f) using a pilot estimator (i.e., a cross-bilateral filter), as defined in Eq. 11. To more clearly assess the accuracy of these error estimation strategies, we also include the results (g), where we use the ground truth errors of $\bar{h}_i$ in place of the estimated variances $\hat{\sigma}^2(\bar{h}_i)$ in Eq. 11. These oracle errors are computed as the squared differences between $\bar{h}_i$ and their corresponding ground truth values in (h). In both cases, (e) and (f), our method outperforms the baseline approach (d), which does not account for the variance in $\bar{h}_i$, highlighting the practical importance of incorporating such errors when estimating the optimal control variate coefficients. Our chosen strategy (f), which estimates these errors via Eq. 11, yields better results than using per-pixel sample variances (e).

within this window and the target value $\bar{f}$. We then select the k-nearest neighbors based on the differences. In our implementation, we set $k = 25$ (an analysis of k is provided in Sec. 5).

For brevity, we omit the subscript that denotes the target pixel index (e.g., in $\hat{F}$ and $\bar{f}$), as the control variate estimator is applied independently to each pixel.

Given the new estimator (Eq. 7), which employs estimated expectations $\bar{h}_i$, we derive its variance $\sigma^2(\hat{F})$ in a manner analogous to Eq. 3, while additionally accounting for the variance introduced by $\bar{h}_i$. The resulting expression is:

$$\sigma^2(\hat{F}) = \frac{1}{n}\left(\sigma^2(f(x)) + \boldsymbol{\beta}^\top V_g \boldsymbol{\beta} - 2\boldsymbol{\beta}^\top V_{fg} + \boldsymbol{\beta}^\top V_h \boldsymbol{\beta}\right), \quad (8)$$

where V_h is the $k \times k$ covariance matrix of the random variables $h_i(x)$ in the PT image. This formulation introduces an additional error term, $\boldsymbol{\beta}^\top V_h \boldsymbol{\beta}$, compared to the original variance formulation (Eq. 3), which assumes that the expectations G_i are exact.

Since the $\bar{h}_i$ values in the PT image are independently estimated across pixels, the corresponding covariance matrix V_h becomes a diagonal matrix, where the i-th diagonal element is given by $\sigma^2(h_i(x))$. The derivation of this variance expression is provided in the supplementary material.

We then compute the derivative of the variance with respect to $\boldsymbol{\beta}$, which is given by $(2V_g\boldsymbol{\beta} - 2V_{fg} + 2V_h\boldsymbol{\beta})/n$. By setting the derivative to zero, we obtain a closed-form solution for the optimal coefficients:

$$\boldsymbol{\beta}^* = \left(V_g + V_h\right)^{-1} V_{fg}. \quad (9)$$

The formulation above clearly indicates that the optimal coefficients should be adjusted according to the error structure V_h in the unbiased but noisy estimates $\bar{h}_i$. Since this formulation involves unknown terms, we estimate the optimal coefficients using the following expression:

$$\hat{\boldsymbol{\beta}}^* = \left(\frac{1}{n-1}X^\top X + \hat{V}_h\right)^{-1} \frac{1}{n-1}X^\top \boldsymbol{y}. \quad (10)$$

Compared to the original formulation in Eq. 6, this revised expression incorporates an additional diagonal matrix, $\hat{V}_h$, whose i-th diagonal element represents the estimated variance $\hat{\sigma}^2(h_i(x))$.

The resulting closed-form expression is closely related to Tikhonov regularization, a widely used technique for stabilizing least-squares problems that are ill-posed or sensitive to noise by introducing a penalty matrix [Hansen 2010; Tikhonov and Arsenin 1977].

From this perspective, the estimated covariance $\hat{V}_h$ serves as a penalty term that suppresses the magnitude of the i-th coefficient β_i when the corresponding error $\hat{\sigma}^2(h_i(x))$ is large. This regularization property offers a principled approach for robustly handling pixel estimates $\bar{h}_i$, whose errors may vary significantly across pixels, as in the case of fireflies (see Fig. 2).

Variance estimation. To compute the estimated optimal coefficients using Eq. 10, it is necessary to compute the estimated variances $\hat{\sigma}^2(h_i(x))$. A straightforward approach is to use the sample variance, i.e., $\hat{\sigma}^2(h_i(x)) = \sum_{s=1}^{n} (h_i(x_{i,s}) - \bar{h}_i)^2/(n-1)$.

Alternatively, a simple denoising method can be used to generate pilot estimates, providing a more robust means of estimating the error in $\bar{h}_i$ than direct sample variances. For this purpose, we apply a cross-bilateral filter [Eisemann and Durand 2004] and estimate $\sigma^2(h_i(x))$ using the squared difference between $\bar{h}_i$ and the corresponding denoised value:

$$\hat{\sigma}^2(h_i(x)) = n\hat{\sigma}^2(\bar{h}_i) = n\left\{\left(\frac{1}{\sum_{j\in\Omega_i} w_j}\sum_{j\in\Omega_i} w_j \bar{h}_j\right) - \bar{h}_i\right\}^2. \quad (11)$$

The weight w_j allocated to pixel j within an 11×11 denoising window is set by $w_j = \exp(-\|\bar{g}_j - \bar{g}_i\|^2/\hat{\sigma}^2(\bar{g}_i))$, where the bandwidth $\hat{\sigma}^2(\bar{g}_i)$ is the estimated variance of the sample mean $\bar{g}_i$.

We also found that enforcing a symmetric penalty by setting the i-th diagonal element of $\hat{V}_h$ to $0.5\left(\hat{\sigma}^2(h(x)) + \hat{\sigma}^2(h_i(x))\right)$, where $\hat{\sigma}^2(h(x))$ denotes the estimated variance at the target pixel computed using Eq. 11, can further improve the results compared to using $\hat{\sigma}^2(h_i(x))$ alone (see the supplementary material for analysis).

After defining the matrix $\hat{V}_h$ either from sample variances or via cross-bilateral filtering, we compute the estimated optimal coefficients $\hat{\boldsymbol{\beta}}^*$ using Eq. 10, and evaluate the final per-pixel output using the control variate estimator $\hat{F}$ in Eq. 7.

As shown in Fig. 4, both approaches that incorporate uncertainty in the control variates result in a noticeable improvement over CV (baseline), which does not consider such imperfections. Since the alternative method using the cross-bilateral filter produces better results than the sample variance approach, we adopt it for all other experiments presented in this paper.

Implementation details. Implementing our control variate is straightforward due to its image-space formulation, which does not require modifications to existing sampling strategies such as importance sampling. However, it does require access to individual samples in the CRN image (e.g., $g_i(x_s)$) to compute the terms $X^\top X$ and $X^\top \boldsymbol{y}$ in Eq. 10. As a result, the computational overhead of constructing these terms increases with the number of samples n, which is undesirable in offline rendering scenarios where n is typically large.

To mitigate this issue, we adopt a practical surrogate by partitioning individual samples into a fixed number of groups and using the sample mean of each group as a proxy sample for computing $X^\top X$ and $X^\top \boldsymbol{y}$. These per-pixel group means can be computed without storing the full set of individual samples. Moreover, the computational overhead associated with evaluating the control variate estimator in Eq. 7, using the coefficients from Eq. 10, depends only on the number of groups rather than on the total number of samples.

We empirically found that using twelve groups offers a good balance between accuracy and efficiency. For example, the overheads with 4, 12, and 24 groups were 31.9 ms, 47.7 ms, and 72.2 ms, respectively, when testing the ten scenes in Table 1. Increasing the group count from 4 to 12 yielded a 16.8% improvement in relMSE, whereas a further increase from 12 to 24 groups provided only an additional 2.6% improvement on average. Based on this trade-off, we adopt twelve groups for all other experiments in the paper.

5 Results and Discussion

This section evaluates the effectiveness of our method in reducing MC noise compared to other control variate techniques. Our straightforward image-space adaptation, built upon traditional control variate theory (described in Sec. 3), is considered a baseline, referred to as CV (baseline). We performed experiments on ten scenes from the public repository [Bitterli 2016], as summarized in Table 1. All tests were conducted on a PC with an AMD Ryzen Threadripper PRO 5995WX CPU and a single Nvidia GeForce RTX 4090 GPU. The experiments employed the path tracing implementation in Mitsuba [Jakob 2010], unless stated otherwise. All images were rendered at a resolution of 1280×720. Numerical accuracy was measured using the relative mean squared error (relMSE), computed using reference images rendered with 65K samples per pixel (spp). Specifically, the relMSE of an estimated image $\hat{F}$ with respect to its reference F is computed as $\frac{1}{3N}\sum_{i=1}^{N}\|\hat{F}_i - F_i\|^2/(\bar{F}_i^2 + \epsilon)$, where N is the number of pixels, $\bar{F}_i$ is the grayscale value of the pixel color F_i, and $\epsilon = 0.01$.

Comparisons with image-space control variate techniques. We compare three image-space control variate techniques: CV (baseline), IDUW [Back et al. 2023], and our method, in terms of numerical accuracy (Table 1), visual quality (Fig. 8). Both CV (baseline) and IDUW use the same input (CRN and PT) as our technique. For IDUW, we used the public implementation provided by the authors.

The computational overheads of CV (baseline), IDUW, and our method are 47.2 ms, 67.4 ms, and 47.7 ms, respectively. These differences are negligible when compared to the sampling time, which ranges from 8.9 s (Teapot) to 37.7 s (Bathroom) across the five scenes shown in Fig. 8. Therefore, the comparisons correspond to equal-time evaluations.

Although IDUW was originally introduced as a low-bias denoising technique, we include it in our evaluation because its image-space formulation can be reinterpreted as a control variate estimator (see the supplementary material). The key distinction lies in the computation of the control variate coefficients: IDUW heuristically determines them using the variances of pixel estimates in the CRN image, while our method derives and estimates optimal coefficients.

This technical distinction enables our method to produce both quantitatively and qualitatively superior results, as shown in Table 1 and Fig. 8. Moreover, our improvement over CV (baseline) highlights the importance of accounting for uncertainty in the estimated expectations of control variates.

We further evaluate the numerical convergence of all tested image-space techniques in Fig. 9. Each method estimates its coefficients from the processed pixel estimates, which introduces bias. However, as demonstrated in the convergence plots, this bias does not diminish the practical benefits of these techniques, as they consistently outperform the standard MC estimator (PT without control variates) up to high sample counts (for example, 1536 spp).

The convergence results further demonstrate the practical significance of accounting for uncertainty in the estimated expectations of control variates. While the performance gap between CV (baseline) and our method narrows with increasing sample counts in the House, Staircase, and Teapot scenes, our method consistently maintains a significant advantage in all other scenes where MC noise, as reflected in the errors of the PT estimates, remains relatively high, even at 1536 spp.

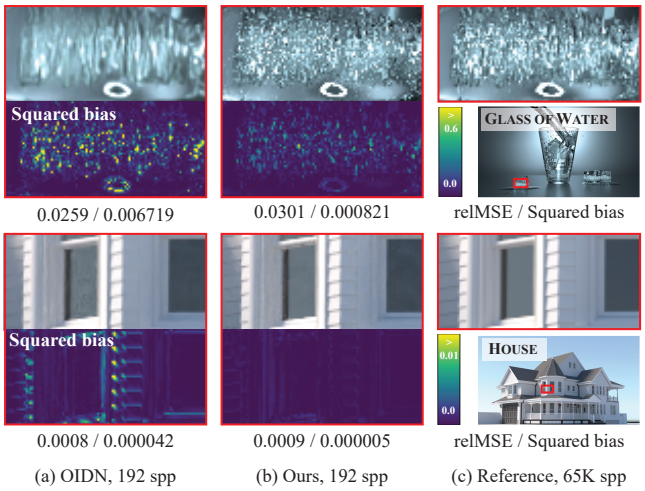
Comparisons with control variates using tractable expectations. Fig. 10 compares with the control variate method proposed in [Salaün et al. 2022], which uses polynomial functions in the primary sample space. We refer to this method as CV (polynomials). For testing CV (polynomials), we used the publicly available code provided by the authors, which was implemented on top of the PBRT [Pharr et al. 2016] framework. To ensure a fair comparison, we also used the same framework (PBRT) when evaluating our method.

CV (polynomials) enables analytical computation of the expectations of control variates, effectively eliminating expectation errors. However, it requires the dimensionality of the analytic function to be fixed and relatively low. This contrasts with image-space methods, including ours, which can accommodate arbitrarily high-dimensional scenarios resulting from multiple secondary ray bounces in path tracing. As a result, a direct comparison with CV (polynomials) is not meaningful for the general experiments shown in Table 1 and Figs. 8 and 9.

To allow for a fair comparison, we selected two scenes from Table 1, House and Teapot, which are primarily lit by direct illumination from environmental lights. We further restricted the

Table 1. relMSE comparisons with IDUW [Back et al. 2023] and CV (baseline) across different sample counts for ten tested scenes.

Bathroom				Breakfast Room			Classroom			Door			Glass of Water		
spp	IDUW	Baseline	Ours	IDUW	Baseline	Ours	IDUW	Baseline	Ours	IDUW	Baseline	Ours	IDUW	Baseline	Ours
48	0.0257	0.0535	**0.0087**	0.0208	0.0306	**0.0061**	0.0650	0.1226	**0.0200**	0.1518	0.2296	**0.0705**	0.0951	0.1583	**0.0478**
96	0.0150	0.0290	**0.0056**	0.0158	0.0206	**0.0036**	0.0421	0.0688	**0.0143**	0.1039	0.1673	**0.0363**	0.0692	0.0955	**0.0394**
192	0.0088	0.0148	**0.0037**	0.0086	0.0108	**0.0024**	0.0230	0.0330	**0.0094**	0.0589	0.1047	**0.0202**	0.0529	0.0611	**0.0301**
384	0.0056	0.0081	**0.0025**	0.0046	0.0046	**0.0014**	0.0136	0.0156	**0.0063**	0.0344	0.0600	**0.0117**	0.0362	0.0373	**0.0201**
Grey & White Room				**House**			**Kitchen**			**Staircase**			**Teapot**		
spp	IDUW	Baseline	Ours	IDUW	Baseline	Ours	IDUW	Baseline	Ours	IDUW	Baseline	Ours	IDUW	Baseline	Ours
48	0.1270	0.0930	**0.0358**	0.0085	0.0059	**0.0032**	0.0456	0.0396	**0.0152**	0.0084	0.0073	**0.0041**	0.0065	0.0054	**0.0020**
96	0.0688	0.0564	**0.0211**	0.0036	0.0030	**0.0015**	0.0299	0.0244	**0.0083**	0.0050	0.0037	**0.0024**	0.0034	0.0027	**0.0012**
192	0.0364	0.0313	**0.0127**	0.0023	0.0015	**0.0009**	0.0156	0.0132	**0.0047**	0.0028	0.0019	**0.0015**	0.0015	0.0013	**0.0006**
384	0.0186	0.0165	**0.0073**	0.0012	0.0007	**0.0005**	0.0085	0.0071	**0.0029**	0.0015	0.0010	**0.0008**	0.0009	0.0006	**0.0003**



(a) OIDN, 192 spp (b) Ours, 192 spp (c) Reference, 65K spp

Fig. 5. Comparison with a neural denoiser, OIDN [Áfra 2025]. While our technique produces noisier results and higher relMSEs than OIDN, it achieves a substantially lower squared bias compared to the denoiser. To compute the bias, we used 300 input images generated with different random seeds per scene for both our method and OIDN.

maximum ray depth to exclude indirect illumination, ensuring that MC noise originates mainly from direct lighting. The corresponding reference images were also re-rendered under this restricted setting for consistency. The computational overhead of this method was approximately 1100 ms, which is higher than the 47.7 ms required by our method under the same conditions shown in Fig. 10. Since the provided code is CPU-based, unlike our GPU-based implementation, we ensured fairness by conducting same-sample comparisons.

As shown in Fig. 10, polynomial approximations of the light transport integral, such as the tested direct lighting integral, outperform the standard MC estimator (PT). Nevertheless, our method achieves more effective noise reduction by leveraging spatial correlations across pixel estimates, in contrast to CV (polynomials), which processes each pixel independently without incorporating information from neighboring pixels. More importantly, our method offers greater generality, as it does not impose any constraints on the dimensionality of the integrand.

Comparisons with image denoisers. Our method shares high-level similarities with denoising methods, as both leverage spatial coherence in image space, i.e., the similarity of ground-truth colors among spatially close pixels. However, unlike denoisers, whose bias increases when blending pixels with different ground-truth colors, our bias arises solely from the statistical dependency between control variate coefficients and the input data. This technical distinction can lead our technique to yield lower bias than image denoisers.

To analyze this difference, we compare our method with a recent neural denoiser, Intel's Open Image Denoise (OIDN) [Áfra 2025], using its publicly released pre-trained neural network. For OIDN, we generated and provided PT images (excluding CRN images), along with G-buffers, as input to the OIDN network.

When testing the ten scenes using 192 spp in Table 1, we observed that OIDN achieved 2.27× lower relMSE than our method on average, whereas our squared bias was 15.25× lower. We also include qualitative comparisons for the Glass of Water and House scenes in Fig. 5, where the strengths and weaknesses of each method are evident: OIDN produces smoother but more biased results, while our technique yields noisier but less biased results.

Analysis of the number of control variates. We utilize k-nearest pixel estimates selected from a search window centered at each pixel. In our tests, we set $k = 25$ using an 11×11 search window. Fig. 11 presents our results where we vary k values, ranging from 6 to 100. For higher values of k (specifically, 50 and 100), we increased the search window size to 21×21.

The changes in the accuracy of our control variate technique remain moderate compared to the overall noise reduction achieved over the input (e.g., CRN). This is because the relative influence of each control variate is adaptively balanced through the estimated optimal coefficients $\hat{\boldsymbol{\beta}}^*$ (Eq. 10). Nonetheless, accuracy generally improves with increasing k up to an intermediate value ($k = 25$) without significantly affecting computational overhead. Therefore, we chose $k = 25$ and used this setting for all other experiments. We provide additional analyses (e.g., using alternative strategies for selecting the k-nearest neighbors) in the supplementary material.

Analysis of the difference in sample rates between CRN and PT images. By default, we use the same sample count for both input images, CRN and PT. In Fig. 6, we vary the sample allocation between the two images and present the results from each configuration.

We observe a trade-off between correlated noise inherited from CRN and independent noise introduced by PT. For example, when allocating more samples to PT (Fig. 6(b)), the results exhibit less independent noise but more correlated noise, compared to the case of allocating more samples to CRN (Fig. 6(d)). While the optimal sample allocation between the two inputs depends on the scene, we find that our default choice of equal allocation (1:1) provides a reasonable balance in practice.

Limitations and future work. The effectiveness of control variate estimators depends on the degree of correlation between the original MC estimates and the selected control variates. Consequently, our method may fail to reduce noise effectively when neighboring pixels in the CRN image exhibit weak correlation, such as in regions with random noise patterns (e.g., in Fig. 12). In these scenarios, our error estimation (Eq. 11) can also become inaccurate, as it relies on a cross-bilateral filter whose weights w_j are determined by differences in CRN estimates (e.g., $\bar{g}_j - \bar{g}_i$). We found that this can result in energy loss in regions where the estimates are particularly noisy. Although this issue diminishes at higher sample counts because our control variate technique is a consistent estimator, improving error estimation accuracy through enhanced weighting could further increase robustness. We leave this exploration for future work.

An additional limitation stems from the image-space nature of our approach. By leveraging spatial correlation, our method inherently introduces statistical dependencies among output pixel estimates, resulting in correlated noise. This violates a common assumption in image denoising techniques [Zwicker et al. 2015], which typically treat pixel estimates as statistically independent. Therefore, applying a denoiser designed under this assumption to our output is not ideal (see Fig. 7). A promising direction for future work is to design a method-specific denoiser that explicitly accounts for the correlated errors produced by our method, similar to the recent denoiser proposed in [Chen et al. 2024] for handling correlated noise generated by Metropolis light transport [Veach and Guibas 1997]. Furthermore, since our method enables the estimation of per-pixel errors via Eq. 8, another potential extension is to incorporate an image-space adaptive sampling strategy [Zwicker et al. 2015] to effectively reduce the errors in $\bar{h}_i$. Finally, it would also be interesting to explore combining our technique with path-reusing frameworks that exploit image-space correlation through correlated sampling, such as gradient-domain rendering [Kettunen et al. 2015; Lehtinen et al. 2013] and image-space splatting [Tong and Hachisuka 2024].

6 Conclusion

This paper introduces a new per-pixel estimator that operates in image space and does not require modifications to existing sampling optimizations, such as importance sampling. Specifically, neighboring pixel estimates from a correlated input (CRN) are selected as control variates without explicitly modeling their analytic forms, and their expectations are estimated using corresponding pixel values from an independent input (PT). While this design keeps our method simple and broadly applicable to general rendering scenarios, it introduces a key challenge: the control variates can be unreliable since their estimated expectations are as noisy as the variates. To address this issue, we propose a strategy based on penalized least-squares regression that accounts for the heterogeneous variances in the estimated expectations. This approach enables robust exploitation of multiple imperfect control variates.

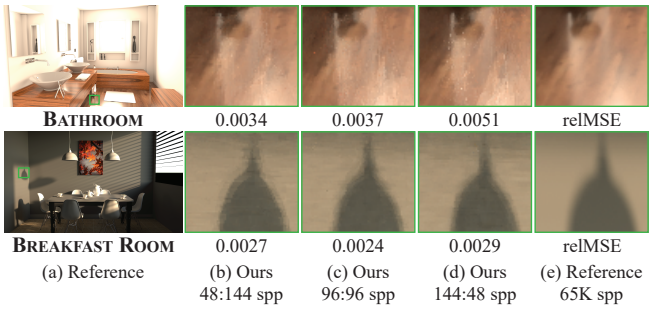


Fig. 6. Results with different sample allocations between the two input images (CRN and PT). Given a user-specified total sample count (196 spp), we divide it between the inputs as follows: (b) 48:144, (c) 96:96, and (d) 144:48 for CRN:PT, respectively.

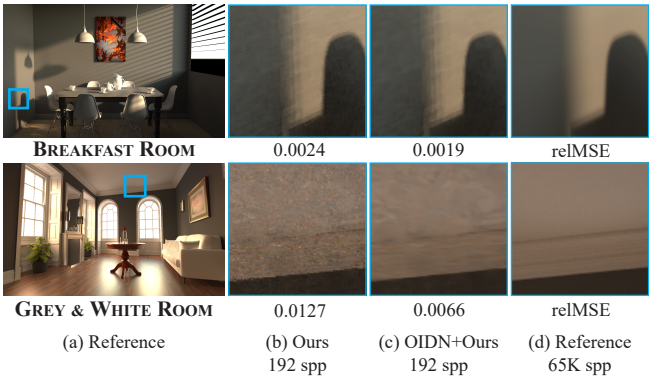


Fig. 7. Image denoising results on our output. We apply a recent denoiser, OIDN [Áfra 2025], to our output (b). While the denoised images (c) show reduced errors from the suppression of independent noise, another type of noise (i.e., correlated noise) remains.

Acknowledgments

We would like to thank the anonymous reviewers and Jonghee Back for their valuable comments, as well as the following artists for each scene: nacimus (Bathroom), Wig42 (Breakfast Room; Grey & White Room; Staircase), NovaAshbell (Classroom), BhaWin (Glass of Water), MrChimp2313 (House), and Jay-Artist (Kitchen). Bochang Moon is the corresponding author of this paper. This work was supported by the Institute of Information & Communication Technology Planning & Evaluation (IITP) and the National Research Foundation of Korea (NRF), funded by the Korean government (MSIT) under grants RS-2022-II220566 and RS-2023-00207939.

References

Attila T. Áfra. 2025. Intel® Open Image Denoise. https://www.openimagedenoise.org

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* (Sydney, NSW, Australia) *(SA '23)*. ACM, New York, NY, USA, Article 9, 10 pages.

Laurent Belcour, Guofu Xie, Christophe Hery, Mark Meyer, Wojciech Jarosz, and Derek Nowrouzezahrai. 2018. Integrating clipped spherical harmonics expansions. *ACM Trans. Graph.* 37, 2, Article 19 (March 2018), 12 pages.

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

Chuhao Chen, Yuze He, and Tzu-Mao Li. 2024. Temporally stable metropolis light transport denoising using recurrent transformer blocks. *ACM Trans. Graph.* 43, 4, Article 123 (July 2024), 14 pages.

Petrik Clarberg and Tomas Akenine-Möller. 2008. Exploiting visibility correlation in direct illumination. *Computer Graphics Forum* 27, 4 (2008), 1125–1136.

Miguel Crespo, Adrian Jarabo, and Adolfo Muñoz. 2021. Primary-space adaptive control variates using piecewise-polynomial approximations. *ACM Trans. Graph.* 40, 3, Article 25 (July 2021), 15 pages.

Mikhail Dereviannykh, Dmitrii Klepikov, Johannes Hanika, and Carsten Dachsbacher. 2025. Neural two-level Monte Carlo real-time rendering. *Computer Graphics Forum* 44, 2 (2025), e70050.

Elmar Eisemann and Frédo Durand. 2004. Flash photography enhancement via intrinsic relighting. *ACM Trans. Graph.* 23, 3 (Aug. 2004), 673–678.

Shaohua Fan, Stephen Chenney, Bo Hu, Kam-Wah Tsui, and Yu-chi Lai. 2006. Optimizing control variate estimators for rendering. *Computer Graphics Forum* 25, 3 (2006), 351–357.

Michael B. Giles. 2008. Multilevel Monte Carlo path simulation. *Operations Research* 56, 3 (2008), 607–617.

Michael B. Giles. 2015. Multilevel Monte Carlo methods. *Acta Numerica* 24 (2015), 259–328.

Per Christian Hansen. 2010. *Discrete inverse problems: insight and algorithms.* Society for Industrial and Applied Mathematics, USA.

Qingqin Hua, Pascal Grittmann, and Philipp Slusallek. 2023. Revisiting controlled mixture sampling for rendering applications. *ACM Trans. Graph.* 42, 4, Article 64 (July 2023), 13 pages.

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

James T. Kajiya. 1986. The rendering equation. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '86).* ACM, New York, NY, USA, 143–150.

Alexander Keller. 2001. Hierarchical Monte Carlo image synthesis. *Math. Comput. Simul.* 55, 1-3 (Feb. 2001), 79–92.

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.

Ivo Kondapaneni, Petr Vevoda, Pascal Grittmann, Tomáš Skřivan, Philipp Slusallek, and Jaroslav Křivánek. 2019. Optimal multiple importance sampling. *ACM Trans. Graph.* 38, 4, Article 37 (July 2019), 14 pages.

Eric P. Lafortune and Yves D. Willems. 1995a. A 5D tree to reduce the variance of Monte Carlo ray tracing. In *Rendering Techniques '95*, Patrick M. Hanrahan and Werner Purgathofer (Eds.). Springer Vienna, Vienna, 11–20.

Eric P. Lafortune and Yves D. Willems. 1995b. The ambient term as a variance reducing technique for Monte Carlo ray tracing. In *Photorealistic Rendering Techniques*, Georgios Sakas, Stefan Müller, and Peter Shirley (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 168–176.

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.

Yu Li and Antony Ware. 2024. A weighted multilevel Monte Carlo method. arXiv:2405.03453

Soham Uday Mehta, Ravi Ramamoorthi, Mark Meyer, and Christophe Hery. 2012. Analytic tangent irradiance environment maps for anisotropic surfaces. *Computer Graphics Forum* 31, 4 (2012), 1501–1508.

Thomas Müller, Brian Mcwilliams, Fabrice Rousselle, Markus Gross, and Jan Novák. 2019. Neural importance sampling. *ACM Trans. Graph.* 38, 5, Article 145 (Oct. 2019), 19 pages.

Thomas Müller, Fabrice Rousselle, Alexander Keller, and Jan Novák. 2020. Neural control variates. *ACM Trans. Graph.* 39, 6, Article 243 (Nov. 2020), 19 pages.

Baptiste Nicolet, Fabrice Rousselle, Jan Novak, Alexander Keller, Wenzel Jakob, and Thomas Müller. 2023. Recursive control variates for inverse rendering. *ACM Trans. Graph.* 42, 4, Article 62 (July 2023), 13 pages.

Art Owen and Yi Zhou. 2000. Safe and effective importance sampling. *J. Amer. Statist. Assoc.* 95, 449 (2000), 135–143.

Art B. Owen. 2013. *Monte Carlo theory, methods and examples.* https://artowen.su.domains/mc/.

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.

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.

Corentin Salaün, Adrien Gruson, Binh-Son Hua, Toshiya Hachisuka, and Gurprit Singh. 2022. Regression-based Monte Carlo integration. *ACM Trans. Graph.* 41, 4, Article 79 (July 2022), 14 pages.

Kartic Subr. 2021. Q-NET: A network for low-dimensional integrals of neural proxies. *Computer Graphics Forum* 40, 4 (2021), 61–71.

László Szirmay-Kalos, Ferenc Csonka, and György Antal. 2001. Global illumination as a combination of continuous random walk and finite-element based iteration. *Computer Graphics Forum* 20, 3 (2001), 288–298.

László Szécsi, Mateu Sbert, and László Szirmay-Kalos. 2004. Combined correlated and importance sampling in direct light source computation and environment mapping. *Computer Graphics Forum* 23, 3 (2004), 585–593.

Andrey N. Tikhonov and Vasiliy Y. Arsenin. 1977. *Solutions of ill-posed problems.* V. H. Winston & Sons, Washington, D.C. and New York. xiii+258 pages.

Xiaochun Tong and Toshiya Hachisuka. 2024. Efficient image-space shape splatting for Monte Carlo rendering. *ACM Trans. Graph.* 43, 6, Article 233 (Nov. 2024), 11 pages.

Eric Veach and Leonidas J. Guibas. 1995. Optimally combining sampling techniques for Monte Carlo rendering. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '95).* ACM, New York, NY, USA, 419–428.

Eric Veach and Leonidas J Guibas. 1997. Metropolis light transport. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques.* 65–76.

Bing Xu, Tzu-Mao Li, Iliyan Georgiev, Trevor Hedstrom, and Ravi Ramamoorthi. 2024. Residual path integrals for re-rendering. *Computer Graphics Forum* 43, 4 (2024), e15152.

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 (May 2015), 667–681.

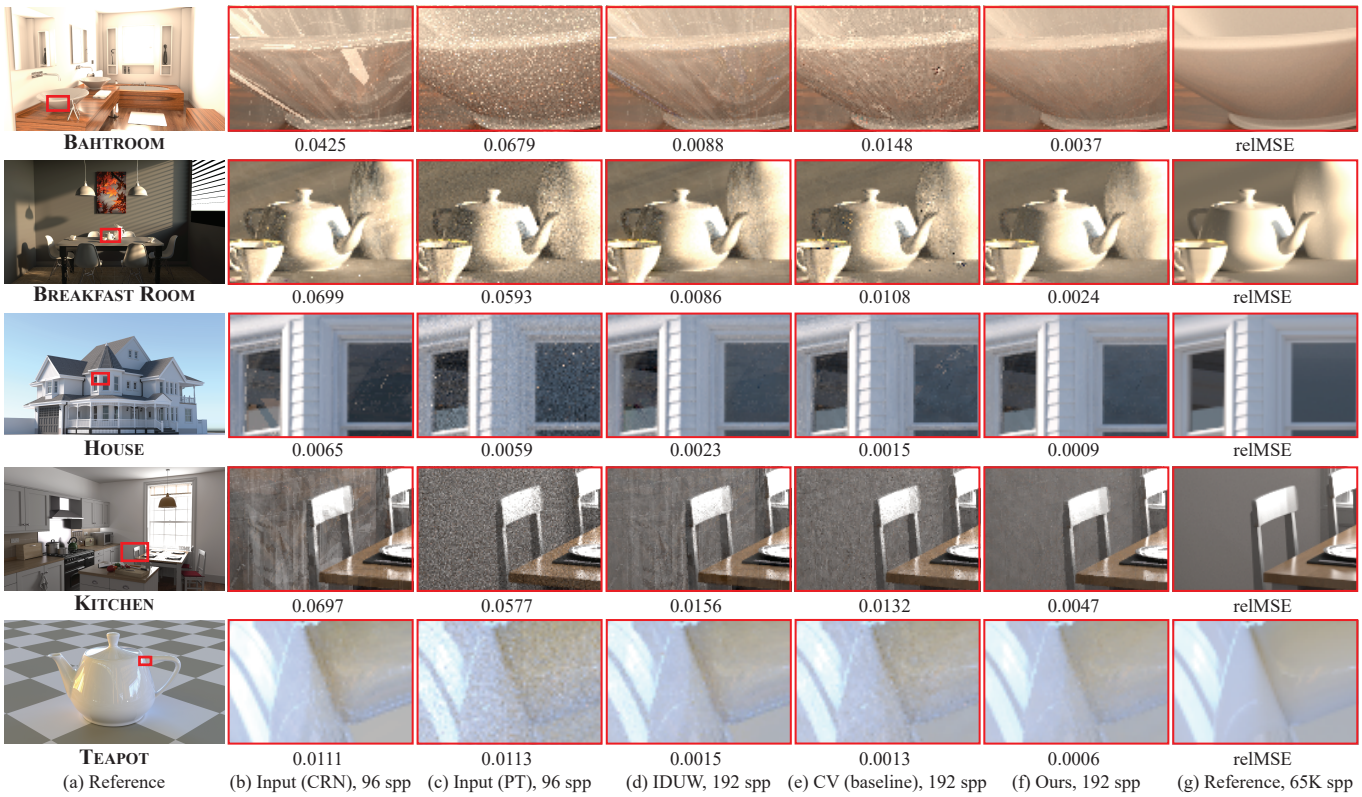


Fig. 8. We compare our method with two image-space control variate techniques: IDUW [Back et al. 2023] and CV (baseline). All methods use the same input images, as shown in (b) and (c), but differ in how they compute the control variate coefficients. IDUW (d) sets the coefficients based on the variances of pixel estimates in the CRN image, while CV (baseline) (e) employs least-squares regression but ignores the errors in the estimated expectations, i.e., the noise of pixel estimates in the PT input. In contrast, our method (f) derives and estimates optimal control variate coefficients by explicitly accounting for these errors. This distinction enables our approach to achieve more accurate and effective variance reduction.

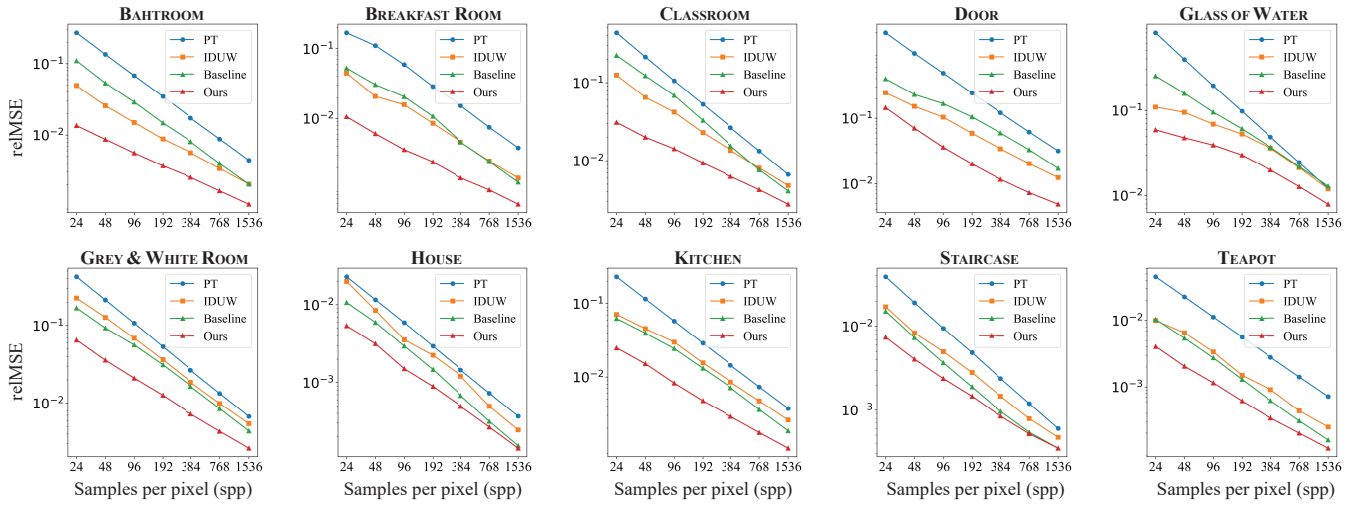


Fig. 9. Numerical convergence of IDUW, CV (baseline), and our method, plotted on a log-log scale. The convergence of the standard MC approach (PT) is also shown. All control variate techniques introduce bias because their coefficients are estimated from processed data (i.e., pixel estimates). Nevertheless, these bias errors can be acceptable in practice, as the methods consistently yield lower errors than PT. Moreover, the results demonstrate that accounting for imperfections in control variates remains important even at high sample counts (e.g., 1536), particularly in scenes with relatively high MC variance.

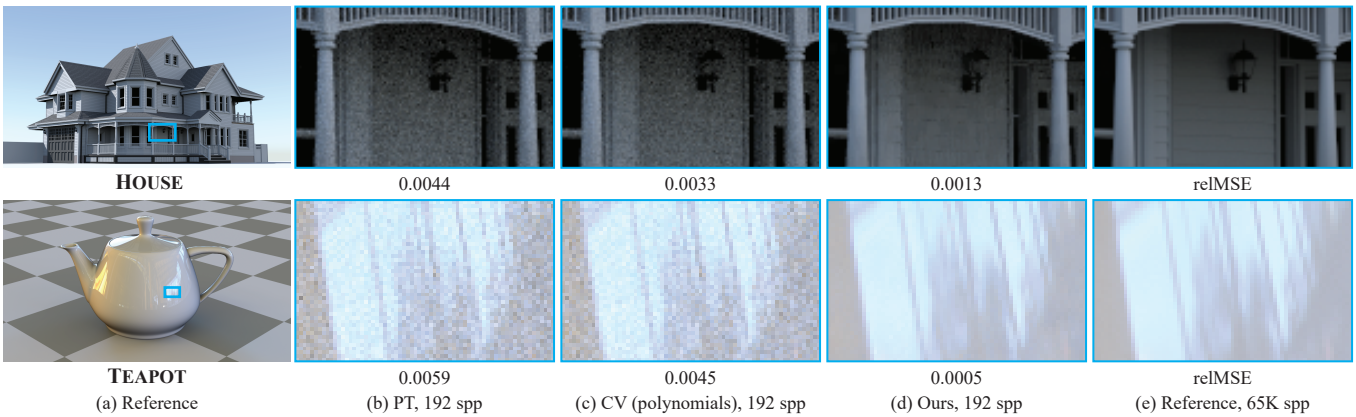


Fig. 10. Comparison with CV (polynomials) [Salaün et al. 2022], which constructs high-dimensional polynomials in the primary sample space and applies them as control variates independently at each pixel. This approach (c) is effective when the dimensionality of the light transport integrand is low, in contrast to our image-space method. To ensure a fair comparison, we reduce the dimensionality by disabling indirect illumination in the two tested scenes. Unlike CV (polynomials), our method (d) leverages spatial correlations across pixels rather than treating each pixel estimate in isolation. This distinction allows our technique to achieve more effective variance reduction without relying on control variate functions with tractable integrals.

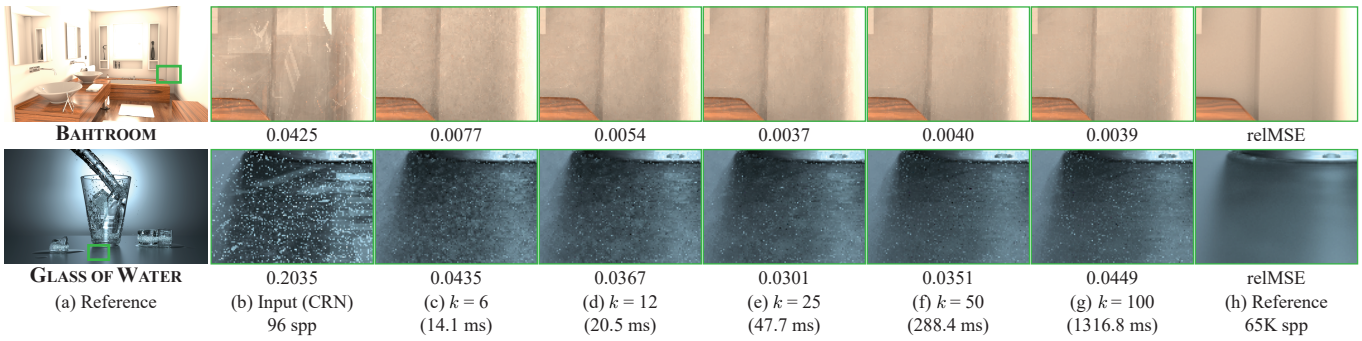


Fig. 11. Analysis of our method with varying numbers of control variates. We vary the number of control variates, k, from 6 (c) to 100 (g), and report both the corresponding errors and computational overhead. The results indicate that varying k does not significantly impact the final accuracy of our method relative to the improvements achieved over the input (b). It is also observed that increasing k tends to improve overall accuracy up to $k = 25$ without introducing a substantial increase in computational cost. Based on this analysis, we adopt $k = 25$ as the default setting for all other experiments.

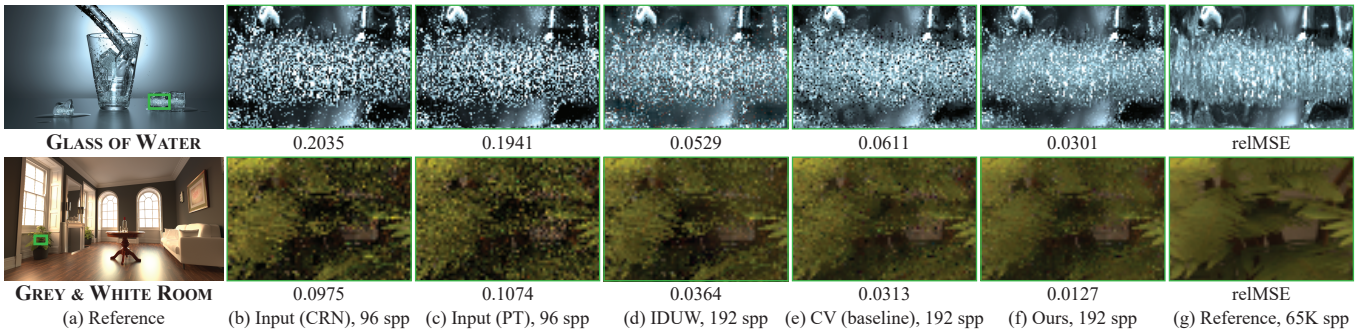


Fig. 12. Failure cases of image-space control variate techniques. All tested methods, including IDUW (d), CV (baseline) (e), and ours (f), rely on the correlation between spatially neighboring pixel estimates in the CRN input (b). However, in regions where such correlations cannot be established by sharing the same random sequences across pixels, the CRN input exhibits noise patterns similar to those in the PT input (c). Although our method adjusts the control variate coefficients by accounting for heterogeneous variance in the PT input and achieves lower errors than the other methods, it still produces noticeable residual noise in areas where spatial correlation is absent.