Image-space Control Variates for Rendering

Fabrice Rousselle[1*] Wojciech Jarosz[2] Jan Novák[1]

[1]Disney Research [2]Dartmouth College



Figure 1: *Image-space control variates allow leveraging coherence in renderings. We show here an example of our re-rendering application, leveraging temporal coherence. We used 1024/64 samples per pixel for rendering the control/difference images, and our final reconstruction (*Ours*, far right) offers a significant improvement over standard* Path tracing*, despite the magnitude of the changes.*

Abstract

We explore the theory of integration with control variates in the context of rendering. Our goal is to optimally combine multiple estimators using their covariances. We focus on two applications, re-rendering and gradient-domain rendering, where we exploit coherence between temporally and spatially adjacent pixels. We propose an image-space (iterative) reconstruction scheme that employs control variates to reduce variance. We show that recent works on scene editing and gradient-domain rendering can be directly formulated as control-variate estimators, despite using seemingly different approaches. In particular, we demonstrate the conceptual equivalence of screened Poisson image reconstruction and our iterative reconstruction scheme. Our composite estimators offer practical and simple solutions that improve upon the current state of the art for the two investigated applications.

Keywords: Monte Carlo integration, control variates, re-rendering, gradient-domain rendering

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

1 Introduction

Physically-based image synthesis often employs Monte Carlo (MC) integration, in particular path tracing, which has a number of attractive properties: the algorithm is conceptually simple, can be used to produce photo-realistic images, offers a predictable convergence rate, allows for rapid iterations, and can directly scale to final renders given more computation time. The main downside of MC rendering is that the computational cost of producing noise-free renders is often prohibitively expensive. Consequently, a number of techniques have been proposed to exploit the spatial and temporal coherence in rendered sequences. For instance, image-space denoising algorithms have proved to be very effective at reducing noise and are now commonly used in production environments. Most of these techniques are, however, intrinsically biased. In this paper, we explore the concept of control-variate integration, a technique that was specifically designed to exploit coherence, and which can do so while preserving the unbiased nature of MC integration. Control-variate integration has potential in the context of rendering animations, gradient-domain rendering, upsampling, and stereo and light field rendering. In this work, we consider applications to re-rendering and gradient-domain rendering.

The concept of integrating with control variates is simple. Suppose we want to estimate the expected value F of estimator $\langle F \rangle$, and there is another estimator $\langle H \rangle$ with a *known* expectation H. Then, we can use $\langle H \rangle$ as the *control variate* (CV) by formulating a new estimator $\langle F \rangle_\star = \langle F - \alpha H \rangle + \alpha H$, which, if $\langle F \rangle$ and $\langle H \rangle$ are correlated, will estimate F with lower variance. The parameter α should reflect the amount of correlation between $\langle F \rangle$ and $\langle H \rangle$, and, if chosen optimally, guarantees that variance will not increase.

Finding practical control variates in the context of image synthesis is challenging. This is because integrals governing light transport cannot typically be expressed in closed form, unless we simplify them heavily by neglecting certain components of the transport. Such simplifications, however, prevent the variate from correlating well with $\langle F \rangle$ and the gains of utilizing it diminish quickly. Instead, we turn to two-level MC integration, a generalization of control-variate integration that uses a MC estimate of the control variate instead of a closed-form expression. Previous works on multi-level MC (see e.g. Giles [2013] for an overview) assume perfect correlation between the signal and the control variate and set $\alpha = 1$, which can lead to increased variance in the estimator. We address this issue using the theory of optimally combining unbiased estimators. We show how this effectively offers the same control as the α in the control-variate framework, with the advantage of being trivially applicable to multiple estimators simultaneously.

In the re-rendering application, we use an existing rendering as the control variate, and re-render the scene after editing various material properties. The resulting scheme is similar to the one used by Günther and Grosch [2015], with the key difference that we rely on a principled combination of estimators instead of a heuristic one to reconstruct the final image, leading to significant improvements.

*e-mail:fabrice.rousselle@disneyresearch.com

This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in ACM Transactions on Graphics. © 2016 Copyright held by the owner/author(s). Publication rights licensed to ACM.

SA '16 Technical Papers,, December 05 - 08, 2016, , Macao
ISBN: 978-1-4503-4514-9/16/12
DOI: http://dx.doi.org/10.1145/2980179.2982443

In the gradient-domain rendering application, we use adjacent pixels as the control variates. These neighbors are themselves noisy and therefore poor control variates on their own. By using an iterated scheme, we gather data from a larger region in order to obtain a robust effective control variate, resulting in a very similar reconstruction to the one obtained with the L^2 Poisson solver used in recent gradient-domain rendering techniques pioneered by Lehtinen et al. [2013]. We also propose a weighted iterated reconstruction, based again on our principled combination of estimators, which leads to better results than the robust L^1 Poisson solver proposed in previous works. Lastly, we demonstrate how our iterated control-variate scheme and the Poisson reconstruction are two instances of the same general filtering scheme. This observation allows us to import our findings to reweight an L^2 Poisson solver, and achieve similar or better quality than the L^1 reconstruction proposed previously.

In summary, we formulate recent works on re-rendering (Section 4) and gradient-domain rendering (Section 5) as control-variate integrators leveraging the theory of optimally combining estimators. Our estimators achieve state-of-the-art reconstruction in both applications. We also demonstrate the theoretical connection between our iterated control-variate scheme and the L^2 Poisson solver commonly used in gradient-domain rendering techniques.

2 Previous Work

Control Variates. Control variates are frequently used as a variance reduction technique in many fields, e.g. finance [Kemna and Vorst 1990; Broadie and Glasserman 1998] or operations research [Hesterberg and Nelson 1998]. Lavenberg et al. [1982] and Nelson [1990] were the first to study the bias and loss of efficiency when α is obtained from a small set of correlated samples. Glynn and Szechtman [2002] present connections of CV to conditional Monte Carlo, antithetics, rotation sampling, stratification, and nonparametric maximum likelihood. An excellent summary of these findings is presented in the book by Glasserman [2004] and Loh [1995] provides a thorough review of the CV concept.

In graphics, CVs have remained relatively unexplored compared to variance reduction techniques like importance sampling. Some notable exceptions include Lafortune and Willems' work on using an ambient term [1994] or a 5D tree of radiance values [1995] as CVs during path tracing. Others have applied CVs to direct illumination or glossy reflections [Szécsi et al. 2004; Fan et al. 2006], average hemispherical visibility [Clarberg and Akenine-Möller 2008], or to computing transmittance and free-flight distances in heterogeneous participating media [Szirmay-Kalos et al. 2011; Novák et al. 2014]. These prior approaches apply CVs to specific rendering sub-problems, operating on carefully chosen path sub-spaces or hemispherical rendering integrals. In contrast, we leverage the CV concept in a very generic way in image space, showing practical applications in a variety of rendering problems, and establishing theoretical connections between CVs and seemingly unrelated prior rendering approaches.

If H is not known, but can be estimated more efficiently than F, we can still use $\langle H \rangle$ as a control variate: $\langle F \rangle_\star = \langle F - \alpha H \rangle + \alpha \langle H \rangle$. The technique is often referred to as two-level Monte Carlo, or separation of the main part. The idea of using numerically estimated control variates was originally formulated for hierarchical parametric integration, known as multi-level Monte Carlo (MLMC), by Heinrich [1998; 2000], and then extended to many applications in financial mathematics, collision physics, and stochastic modeling, e.g. for Brownian path simulation [Giles 2008] or solving elliptic and parabolic SPDEs [Barth et al. 2011]; see the survey by Giles [2013] for more examples. Very related to our work is the hierarchical image synthesis by Keller [2001], which builds upon MLMC integration; our gradient-domain rendering application uses an iterated scheme to similar effect but we only sample differences to very close locations, which can be done with less variance.

Variance-optimal Composite Estimators. Many applications estimate the mean of a random variable as a linear combination of multiple estimators. Deriving a set of linear weights that minimize the variance of the composite estimator is non-trivial when the constituent estimators have unequal variances and/or are correlated. Cochran's [1937] seminal paper on interpreting multiple independent observations seeded a growing interest in this problem. Graybill and Deal [1959] demonstrate that two independent estimators can be optimally weighted using estimates of their variances, if each variance is estimated with at least 9 samples. Zacks [1966] reduces the requirement to estimating the ratio of the variances, and Cohen and Sackrowitz [1974] base the weights only on sample variances and a normalized squared error loss function. Generalizations to multiple, normally distributed estimators were discussed by Norwood and Hinkelmann [1977], and extensions to multivariate normal distributions presented by Loh [1991]. Halperin [1961] and later Keller and Olkin [2004] proposed weighting schemes for unbiased estimators that are correlated deriving the set of optimal weights from the (estimated) covariance matrix. As we combine multiple estimators using the aforementioned approaches, we review the relevant theoretical background in Section 3.2.

Gradient-domain Rendering and Re-rendering. Lehtinen et al. [2013] recently introduced the idea of gradient-domain rendering (GDR), which was later improved and extended by Manzi et al. [2014; 2015] and Kettunen et al. [2015]. In our work, we show how GDR can be formulated as a control-variate integration, and propose an improved reconstruction using the theory of optimally combining estimators. Similarly, we formulate the recent work on re-rendering of Günther and Grosch [2015] as control-variate integration and show how its reconstruction can be improved.

3 Theoretical Background

In this section, we outline variance reduction techniques for Monte Carlo estimation that are related to or used directly by the techniques we will introduce in Sections 4 and 5.

3.1 Control Variates

Suppose we want to numerically evaluate the following integral:

$$F = \int_\Omega f(x)\,\mathrm{d}x, \tag{1}$$

using a Monte Carlo estimator:

$$\langle F \rangle^n = \frac{1}{n}\sum_{i=1}^{n} \frac{f(X_i)}{p(X_i)}, \tag{2}$$

with the expected value $\mathrm{E}[\langle F \rangle] = F$[†] and probability density function (PDF) $p(x)$ for drawing the samples. Let us further assume the existence of a function $h(x)$—the *control variate*—for which the integral over Ω is known to be H. We can then rewrite F and its CV estimator $\langle F \rangle_\star$ as:

$$F = \int_\Omega f(x) - \alpha h(x)\,\mathrm{d}x + \alpha H, \tag{3}$$

$$\langle F \rangle_\star = \langle F - \alpha H \rangle + \alpha H, \tag{4}$$

[†] For brevity we drop the superscript of the estimator whenever possible.

where the absolute value of α can be interpreted as the strength of leveraging the control variate. The key feature of the estimator above is that functions $f(x)$ and $h(x)$ are estimated using the same sample set. As long as the functions are similar, their difference for any x will be relatively small and largely independent of the functions' actual shape; estimator (4) should thus exhibit low variance.

The optimal value of α depends on the correlation between $f(x)$ and $h(x)$ and can be found by minimizing the variance of Equation (4) w.r.t. α:

$$\begin{aligned}\operatorname{Var}[\langle F\rangle_\star] &= \operatorname{Var}[\langle F-\alpha H\rangle + \alpha H] \\ &= \operatorname{Var}[\langle F\rangle] + \alpha^2 \operatorname{Var}[\langle H\rangle] - 2\alpha \operatorname{Cov}[\langle F\rangle, \langle H\rangle], \quad (5)\end{aligned}$$

yielding $\alpha = \operatorname{Cov}[\langle F\rangle, \langle H\rangle] / \operatorname{Var}[\langle H\rangle]$. With this choice the variance of the estimator reads:

$$\operatorname{Var}[\langle F\rangle_\star] = \operatorname{Var}[\langle F\rangle](1 - \operatorname{Corr}[\langle F\rangle, \langle H\rangle]^2). \quad (6)$$

Discussion. Equation (6) clearly shows the necessary precondition for $h(x)$ being a useful control variate, i.e. it needs to be strongly (anti)correlated with $f(x)$. If the correlation is zero, then the variance of $\langle F\rangle_\star$ falls back to variance of $\langle F\rangle$ with no gain compared to the original estimator. If the correlation is perfect (i.e. 1 or -1) then the variance drops to zero. Choosing appropriate α, ideally proportional to the covariance, prevents the CV estimator from increasing the variance in cases when $h(x)$ serves as a poor control variate. In addition to correlation, the performance of the CV estimator hinges on the efficiency of $\langle F - \alpha H\rangle$; indeed, the variance of the CV estimator can be significantly reduced by drawing samples from a PDF $\propto f(x) - \alpha h(x)$. It is also worth noting that true population parameters of $\langle F\rangle$ and $\langle H\rangle$ are rarely known and are frequently replaced by their (dependent) estimates, consequently turning α into a (correlated) random variable.

So far, we only considered control variates with known antiderivatives. However, even if the integral $H = \int_\Omega h(x)\,\mathrm{d}x$ is not known, we can still use $h(x)$ as a control variate, provided that the estimation of H is relatively inexpensive. We simply replace αH in estimator (4) by its estimate $\langle \alpha H\rangle^m$:

$$\langle F\rangle_\star = \langle F - \alpha H\rangle^n + \langle \alpha H\rangle^m. \quad (7)$$

The variance of such an estimator reads:

$$\begin{aligned}\operatorname{Var}[\langle F\rangle_\star] = \operatorname{Var}[\langle F\rangle^n] + \alpha^2 \operatorname{Var}[\langle H\rangle^n] - 2\alpha \operatorname{Cov}[\langle F\rangle^n, \langle H\rangle^n] \\ + \alpha^2 \operatorname{Var}[\langle H\rangle^m] + 2\alpha \operatorname{Cov}[\langle F - \alpha H\rangle^n, \langle H\rangle^m]. \quad (8)\end{aligned}$$

We skip the derivation of the optimal α for brevity; it can be computed analogously as before by setting the first derivative of Equation (8) to zero and solving for α. Estimator (7), typically with α assumed to be 1, is referred to as the two-level MC estimator.

3.2 Variance-minimizing Combination of Estimators

There are cases when a certain quantity Q can be estimated using several estimators. The question that arises immediately is: "What combination of these will minimize the variance of the Q estimate?" Given k unbiased *independent* estimators $\langle Q\rangle_1, ..., \langle Q\rangle_k$, where the i-th estimator has a normal distribution $\mathcal{N}(Q, \sigma_i^2)$, we can define the *variance-optimal* composite estimator as:

$$\langle Q\rangle = \sum_{i=1}^{k} w_i \langle Q\rangle_i, \quad (9)$$

where the weight w_i is defined as the relative reciprocal of the respective variance:

$$w_i = \frac{\sigma_i^{-2}}{\sum_{j=1}^{k} \sigma_j^{-2}}. \quad (10)$$

The variances σ_i^2 are often unknown in practice. A popular alternative is to substitute an independently estimated sample variance $\langle \sigma_i^2\rangle$ for each σ_i^2. However, since the sample variance is a random variable itself, the above weight may no longer be optimal and the variance of the composite can theoretically exceed the variance of individual $\langle Q\rangle_i$. Interestingly, Norwood and Hinkelmann [1977] proved that estimating σ_i^2 with at least 9 samples ensures that $\operatorname{Var}\langle Q\rangle \leq \min(\sigma_1^2, ..., \sigma_k^2)$. For this to hold, $\langle Q\rangle_i$ and $\langle \sigma_i^2\rangle$ must use independent samples, otherwise $\langle Q\rangle$ will be biased.

In cases when $\langle Q\rangle_1, ..., \langle Q\rangle_k$ are *dependent*, an optimal weighting scheme needs to acknowledge their correlation. Keller and Olkin [2004] define the weights $w = (w_1, ..., w_k)$ for the optimal composed estimator as:

$$w = \frac{e\Sigma^{-1}}{e\Sigma^{-1}e^\mathsf{T}}, \quad (11)$$

where $e = (1, ..., 1)$ and Σ is the covariance matrix. Using such weights, the variance of estimator (9) equals $(e\Sigma^{-1}e^\mathsf{T})^{-1}$, and it grows by $(n-1)/(n-k)$ if the entries of the covariance matrix are estimated numerically using n samples [Keller and Olkin 2004].

In the next two sections, we present image-space control variates, a technique for leveraging information from "nearby" pixels to reduce the variance of MC renderings. We strive to introduce the concept in concrete terms demonstrating its tangible benefits on applications to re-rendering and gradient-domain rendering.

4 Re-rendering

The workflow of a re-rendering application can be directly mapped to the concept of two-level MC integration. Given a rendered image of a scene, we want to produce a new image of the scene with its material properties changed, while reusing the computation done for the original image. The original image can be used as a control variate and, during the re-rendering of the scene, we would only sample the difference in light transport between the original and the edited scene. This constitutes our CV estimator, which we combine with a standard (non-CV) estimator for increased robustness.

More formally, denoting h the path contribution function with the old material, and f the path contribution function with the new material, we readily have the control image, $\langle H\rangle_0$, obtained using m samples per pixel,

$$\langle H\rangle_0 = \frac{1}{m}\sum_{i=1}^{m} \frac{h(\bar{x}_i)}{p_0(\bar{x}_i)} \approx \int_\Omega h(\bar{x})d\bar{x}, \quad (12)$$

where $\bar{x} = \mathbf{x}_0 \cdots \mathbf{x}_k$, with $1 \leq k \leq \infty$, defines a path of length k, Ω_k is the set of all paths of length k, $\Omega = \cup_{k=1}^{\infty}\Omega_k$ is the set of all paths of all lengths, and p_0 is the sampling PDF defined for the old material properties. We are now interested in computing the new image using a CV estimator:

$$\langle F\rangle_\star = \langle F - H\rangle_1 + \langle H\rangle_0. \quad (13)$$

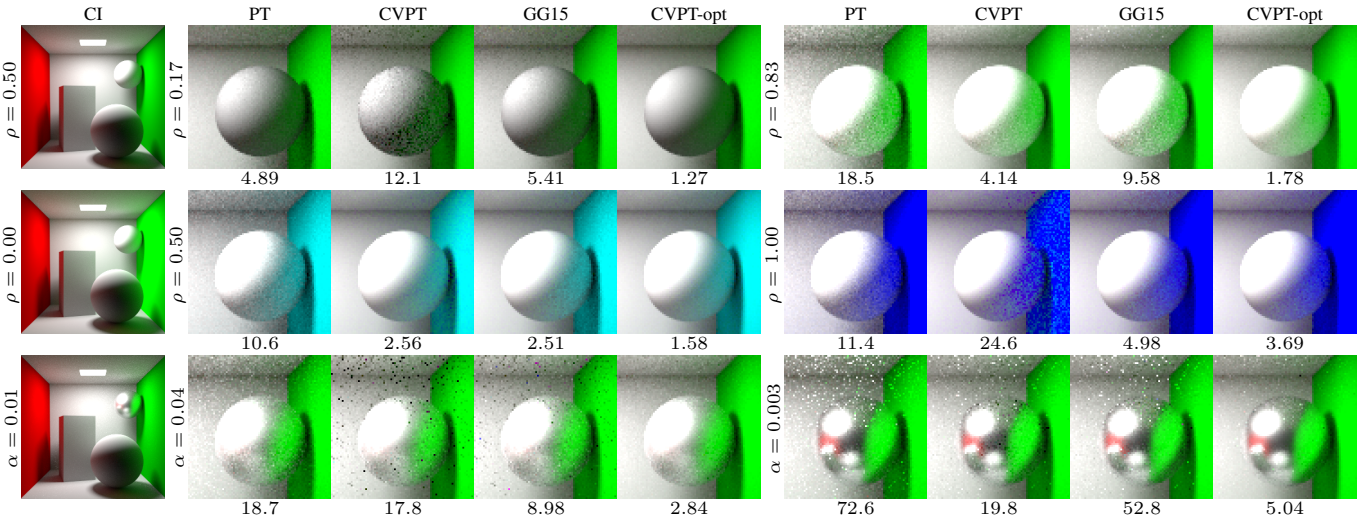


Figure 2: *Re-rendering of the* CORNELL BOX *scene with three types of changes: 1) we modify the brightness of the two spheres and the block (top row); 2) we modify the chromaticity of the right wall using the parameter ρ to interpolate between green and blue (middle row); 3) we use a metal material for the small sphere and change its roughness (bottom row). We used 1024 samples per pixel for the control image (CI), and 64 for the difference image. For every change, we provide the two input estimators: $\langle F\rangle_1$, which corresponds to a path tracer output (PT), and $\langle F\rangle_\star$, which corresponds to our control-variate integrator (CVPT) with $\alpha = 1$. Our optimally-weighted, composite estimator, $\langle F\rangle$ (CVPT-opt), leverages the (anti)correlation of the PT and CVPT estimates to produce a result with lower variance than both inputs, and improves significantly upon the reconstruction obtained using the heuristic (GG15, using a hand-tuned threshold of $\tau = 0.4$) proposed by Günther and Grosch [2015]. The numbers at the bottom of each image report the relative MSE $\times 10^{-3}$ of the respective estimator.*

For convenience, we define the following estimators,

$$\langle F\rangle_1 = \frac{1}{n}\sum_{i=1}^{n}\frac{f(\bar{x}_i)}{p_1(\bar{x}_i)},$$

$$\langle H\rangle_1 = \frac{1}{n}\sum_{i=1}^{n}\frac{h(\bar{x}_i)}{p_1(\bar{x}_i)},$$

$$\langle D\rangle_1 = \frac{1}{n}\sum_{i=1}^{n}\frac{f(\bar{x}_i) - h(\bar{x}_i)}{p_1(\bar{x}_i)},$$

and evaluate them using the *same* set of paths samples $\bar{x}_i$ generated from PDF p_1, defined for the new material properties. As such, $\langle F - H\rangle_1 = \langle F\rangle_1 - \langle H\rangle_1 = \langle D\rangle_1$.

In order to optimally combine the CV estimator $\langle F\rangle_\star$ and the standard estimator $\langle F\rangle_1$, we first need to evaluate their respective variances. For the CV estimator $\langle F\rangle_\star$ we have

$$\mathrm{Var}[\langle F\rangle_\star] = \mathrm{Var}[\langle D\rangle_1] + \mathrm{Var}[\langle H\rangle_0]\,. \tag{14}$$

The covariance between the two estimators is

$$\begin{aligned}\mathrm{Cov}[\langle F\rangle_\star, \langle F\rangle_1] &= \mathrm{Cov}[\langle F\rangle_1 - \langle H\rangle_1 + \langle H\rangle_0, \langle F\rangle_1] \\ &= \mathrm{Var}[\langle F\rangle_1] - \mathrm{Cov}[\langle H\rangle_1, \langle F\rangle_1]\,,\end{aligned} \tag{15}$$

with

$$\mathrm{Cov}[\langle H\rangle_1, \langle F\rangle_1] = \frac{\mathrm{Var}[\langle F\rangle_1] + \mathrm{Var}[\langle H\rangle_1] - \mathrm{Var}[\langle D\rangle_1]}{2}. \tag{16}$$

Given the covariance matrix of the two estimators, $\langle F\rangle_\star$ and $\langle F\rangle_1$, we can now use the weights computed using Equation (11) to obtain the final composite estimator $\langle F\rangle = w_\star\langle F\rangle_\star + w_1\langle F\rangle_1$.

4.1 Implementation

In order to evaluate the performance of the composite estimator, we extended the PBRT renderer [Pharr and Humphreys 2010] by adding a new material type that can hold two material settings. We also added a modified path tracing integrator, which outputs results with both materials, $\langle H\rangle_1$ and $\langle F\rangle_1$, as well as the sampled difference, $\langle D\rangle_1$. Additionally, we collect the sample mean variances of $\langle H\rangle_0$, $\langle H\rangle_1$, $\langle F\rangle_1$, and $\langle D\rangle_1$. Since we use the same PDF when sampling $\langle H\rangle_1$ and $\langle F\rangle_1$, all estimators used during re-rendering can be evaluated with the same set of paths; we simply need to keep track of the two path contributions corresponding to the material settings before and after editing. The only extra processing cost, compared to a standard path tracer, occurs at path vertices where the material has changed and the BSDF needs to be evaluated twice.

Robustness at Low Sampling Rates. The covariance matrix estimate can be too noisy at low sampling rates. We therefore prefilter all sample variances using an NL-Means filter [Buades et al. 2005; Rousselle et al. 2012] over a 3×3 neighborhood, guided by the corresponding color buffer. Furthermore, since covariance matrices are positive, semi-definite, we detect cases when the eigenvalues in the matrix are negative and switch to a simpler weighting scheme using Equation 10, that is, assuming independent estimators. Finally, we specifically handle pixels where the (noisy) variance estimates of $\langle F\rangle_1$ and $\langle D\rangle_1$ are both zero. In such cases, the covariance-based weighting fully discards the CV estimator, which is undesired. Instead, we employ sample rate-based weights, where the weight of an estimator is directly proportional to its sampling rate.

Unbiased Variant. Since we use *sample* covariances computed while rendering, the resulting weighted reconstruction will be biased. To remove the bias, we need to decouple the covariance estimation from the estimation of pixel colors. We thus employ a cross-weighting scheme splitting the samples evenly between two half-

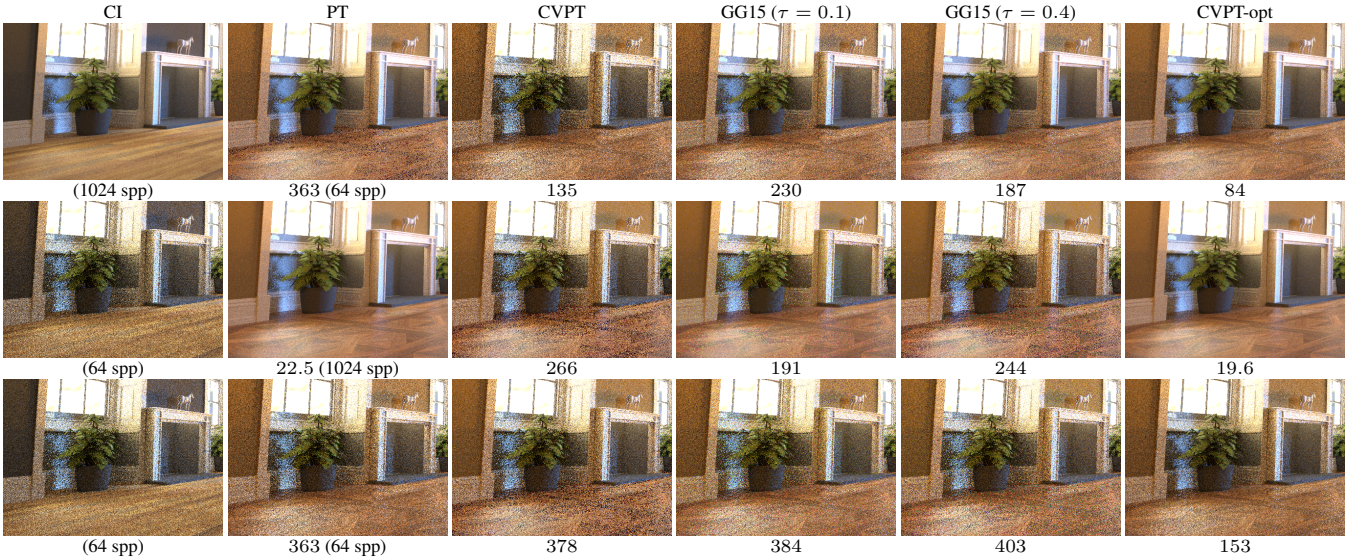


Figure 3: *Re-rendering of the* HORSE ROOM *scene of Figure 1 at varying sampling rates: 1024/64, 64/1024, and 64/64 samples per pixel for rendering the control/difference images. Our composite estimator (CVPT-opt) offers consistently improved results, whereas the heuristic of Günther and Grosch is sensitive to its threshold setting—we tested $\tau = 0.1$ and $\tau = 0.4$—and fails when the control image is noisy. The numbers at the bottom of each image report the relative MSE $\times 10^{-3}$ of the respective estimator.*

buffers, as proposed by Rousselle et al. [2012]. Using covariance matrices of one buffer to weight the reconstruction of the second buffer, and vice-versa, yields unbiased reconstructions. It is worth noting that the unbiased reconstruction is not robust in the presence of fireflies, as can be seen in our supplemental material, since the weighting assumes normally distributed random variables.

4.2 Results and Discussion

We performed multiple experiments to evaluate the robustness of our re-rendering technique and analyzed a number of estimators: the standard path tracer $\langle F \rangle_1$, the control-variate estimator $\langle F \rangle_\star$, the composite estimator $\langle F \rangle$, and its unbiased variant using cross-weighting $\langle F \rangle_\times$. All results were produced at three different sampling rates: 1024/64, 64/1024, and 64/64 samples per pixel, for rendering the control/difference images with the gathered statistics prefiltered using an NL-Means filter. Due to space constraints, we only present a subset of the results here; see the supplemental material for the full set and results without prefiltering.

We also compare our estimators to the work of Günther and Grosch [2015], who address scene re-rendering also by estimating differences to previous renders. The key difference in our work is that we obtain the composite estimator by applying the theory of an optimal combination and resort to heuristics only when the statistics are not reliable. The solution proposed by Günther and Grosch uses only the standard or only the difference-based estimator, with a heuristic selection criterion based solely on the magnitude of the sampled difference with respect to a prescribed threshold. For their method, we used two thresholds: $\tau = 0.1$, as suggested by the authors, and $\tau = 0.4$ that performed better in some of our tests.

CORNELL BOX. Figure 2 compares the aforementioned estimators in a simple scene modified by independently changing the brightness, chromaticity, and roughness of certain materials. For each modification, we performed small and large changes, but only present results with the largest modification and 1024/64 samples per pixel for the control/difference images; see the supplementary material for a complete evaluation. Changing the brightness illustrates the underlying tradeoff of using control variates: if the difference to be estimated has a larger magnitude than the signal itself, e.g. when changing the brightness to 0.17 (top row), then the CV estimator $\langle F \rangle_\star$ exhibits larger variance. Note, however, that the standard estimator $\langle F \rangle_1$ and $\langle F \rangle_\star$ are still correlated (or rather, anticorrelated here). This allows the weighted reconstruction $\langle F \rangle$ to further reduce the error and obtain results that are better than with each estimator in isolation. In contrast, the heuristic of Günther and Grosch cannot leverage this (anti)correlation and produces worse results. The chromaticity experiment (middle row) demonstrates general robustness of two-level MC integration under fairly large hue changes, while the roughness change (bottom row) illustrates the behavior when handling glossy materials. In particular, our weighted reconstruction is more robust to fireflies than the heuristic proposed by Günther and Grosch.

HORSE ROOM. In Figures 1 and 3, we apply our re-rendering application to an interior scene lit by an environment map, where we increased the albedo of walls and changed the albedo texture and the coating roughness of the floor, all at the same time. We show results at different sampling rates to highlight the robustness of our composite estimator to noisy control/difference images. The heuristic proposed by Günther and Grosch succeeds when the control image is of high quality (top row), but preserves some of its noise, which results in suboptimal results in the bottom row. Our composite estimator $\langle F \rangle$ yields improvements at all sampling rates. The cost of the reconstruction scales with the image size; computing the weights for 1280×720 image took 4 core seconds on a 3.2GHz Intel Core i7 CPU, using a prototype Python implementation, while rendering the scene at 64 samples per pixel takes 38 core minutes.

Discussion. While our composite estimator performed best in all experiments, it is worth pointing out that extensive editing increases the difference to the control image and the benefits of using CV estimators diminish. Consequently, the proposed re-rendering scheme is better suited for fine-tuning shader parameters than for large-scale modifications.

5 Gradient-Domain Rendering

Monte Carlo rendering typically suffers from high variance. One approach to reduce the noise is to estimate the color F_p of a pixel p as a weighted sum of its neighborhood $\mathbb{N}_p$:

$$\langle F_p \rangle = \sum_{q \in \mathbb{N}_p} w_{p,q} \langle F_q \rangle, \tag{17}$$

where $w_{p,q}$ is the weight of neighbor q w.r.t. p.

Image-space denoising algorithms [Zwicker et al. 2015] commonly use the neighbor colors directly, neglecting the fact that the expectation of the difference between the pixel and its neighbor may not be zero. A notable exception is the work on gradient-domain rendering (GDR), pioneered by Lehtinen et al. [2013], where the contributions of neighbors are "corrected" by taking into account horizontal and vertical image gradients. Our approach is very similar: we use the neighbors as control variates, i.e. we add their weighted contribution adjusted by an estimate of the color difference between p and q:

$$\langle F_p \rangle = \frac{1}{|\mathbb{N}_p|} \sum_{q \in \mathbb{N}_p} \langle F_p - \alpha_{p,q} F_q \rangle^n + \alpha_{p,q} \langle F_q \rangle^m. \tag{18}$$

Estimator (18) is a straightforward extension of the two-level MC estimator (7) to multiple control variates. We could in theory compute the individual variance-optimal α coefficients according to the recipe given in Section 3.1. This would not, however, take into account the correlation between individual control variates and might result in a poor composite estimator.

Instead, we rewrite Equation (18) to factor out the weights from the estimators:

$$\begin{aligned}\langle F_p \rangle &= \frac{1}{|\mathbb{N}_p|} \sum_{q \in \mathbb{N}_p} \langle F_p - \alpha_{p,q} F_p + \alpha_{p,q} F_p - \alpha_{p,q} F_q \rangle^n + \alpha_{p,q} \langle F_q \rangle^m \\ &= \frac{1}{|\mathbb{N}_p|} \sum_{q \in \mathbb{N}_p} (1 - \alpha_{p,q}) \langle F_p \rangle^n + \alpha_{p,q} \left(\langle F_p - F_q \rangle^n + \langle F_q \rangle^m \right).\end{aligned} \tag{19}$$

Interestingly, this formulation closely resembles Equation (9): a weighted sum of estimators of the same quantity, F_p. As such, we can rewrite the composite estimator in a more general form:

$$\langle F_p \rangle = \sum_{q \in \mathbb{N}_p} w_{p,q} \left(\langle F_p - F_q \rangle^n + \langle F_q \rangle^m \right), \tag{20}$$

where $\sum w_{p,q} = 1$, and the individual generic weights w can be set using formulas in Section 3.2. Please note that we did not lose the baseline estimator $\langle F_p \rangle$, as it is implicitly included when $q = p$.

In most denoising applications, $\mathbb{N}_p$ is fairly large, e.g. spanning over a window of 21×21 pixels [Rousselle et al. 2012]. Considering such a large neighborhood would require, in addition to evaluating $\langle F_q \rangle$, estimating the difference $\langle F_p - F_q \rangle$ for 441 neighbor pixels; this is rather impractical. We instead restrict the neighborhood to the four nearest neighbors and then perform the estimation of final pixel colors by iteratively propagating contributions from distant neighbors.

Neighborhood. First, we constrain the neighborhood. In addition to p, we consider its immediate left, right, top, and bottom neighbors, $\mathbb{N}_p = \{p, l, r, t, b\}$, yielding the following five estimators:

$$\begin{aligned}&\langle F_p \rangle_p = \langle F_p \rangle, \\ &\langle F_p \rangle_l = \langle F_p - F_l \rangle + \langle F_l \rangle, \quad \langle F_p \rangle_r = \langle F_p - F_r \rangle + \langle F_r \rangle, \\ &\langle F_p \rangle_t = \langle F_p - F_t \rangle + \langle F_t \rangle, \quad \langle F_p \rangle_b = \langle F_p - F_b \rangle + \langle F_b \rangle.\end{aligned}$$

Provided that we appropriately weight these estimators (we discuss two possible schemes in Sections 5.2 and 5.3), we obtain a CV estimator for every p in the image. Evaluating these estimators constitutes one step of the iterative estimation.

Iterative Estimation. We apply the CV estimators iteratively formulating the estimate of F_p^{i+1} in the $(i+1)$-th iteration as a function of the estimates in iteration i:

$$\langle F_p^{i+1} \rangle = \sum_{q \in \mathbb{N}_p} w_{p,q}^i \left(\langle F_p^i - F_q^i \rangle + \langle F_q^i \rangle \right). \tag{21}$$

Provided that the estimators on the right-hand side are unbiased, and the weights $w_{p,q}$ computed independently, estimator (21) is unbiased, and so is by transitivity the entire iterative reconstruction.

Discussion. The iterative formulation is effectively very similar to screened Poisson solvers employed in gradient-domain rendering, but formulated as an integration with control variates, which leads to a slightly different reconstruction strategy; we relate the two more precisely in Section 5.4. Our reconstruction is also related to multi-level MC methods, the main difference being that MLMC use a different parameterization for each level (iteration). For instance, the hierarchical image synthesis proposed by Keller [2001] uses a mipmap where each level has a different set of pixels; hence the difference $F_p - F_q$ needs to be estimated at every level. In contrast, we use the same set of pixels in every iteration. Consequently, the difference $F_p - F_q$ can be estimated only once, a-priori, and reused across all iterations.

5.1 Estimation of Pixel Differences

Since our goal is to produce physically-correct images, we rely on path tracing algorithms that synthesize the color of a pixel by averaging many path samples. In addition to estimating the initial color of each pixel p, denoted $\langle F_p^0 \rangle$, we also need to estimate the difference $F_p - F_q$ between any pair of adjacent pixels using correlated (path) samples. This can be trivially achieved by formulating the integration in the primary sample space (PSS) [Kelemen et al. 2002], which in practice amounts to constructing paths through two adjacent pixels using the same sequence of random numbers. Unfortunately, the correlation of such paths can still be fairly weak because the paths can diverge significantly, especially at higher bounces. Furthermore, estimating the difference involves computing two full paths; this is relatively expensive.

A better alternative is to use the *shift* operator proposed in the recent gradient-domain path tracing algorithm by Kettunen et al. [2015]. The shift operator addresses both of the aforementioned issues, and allows for estimating the finite differences with lower variance; see Figure 4. The estimated horizontal and vertical differences are stored in two buffers, $\langle X \rangle$ and $\langle Y \rangle$, where $\langle X_p \rangle = \langle F_r - F_p \rangle$ and $\langle Y_p \rangle = \langle F_b - F_p \rangle$.

5.2 Uniform Reconstruction

A trivial way of linearly combining k estimators is to average them, i.e. to weight each by $1/k$. We will refer to this as the *uniform* reconstruction, and while not being very practical, the uniform reconstruction will help us relate our work to a screened Poisson solver in

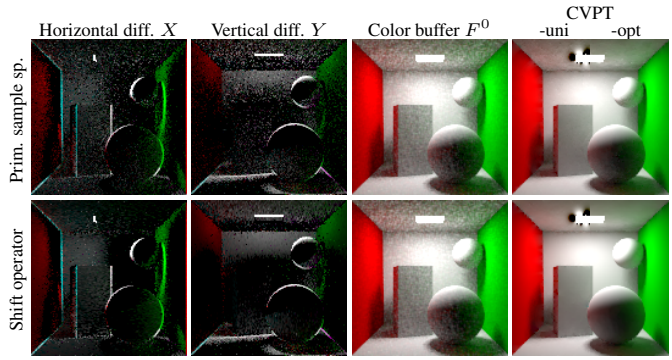


Figure 4: *Control-variate Monte Carlo rendering of the* CORNELL BOX *scene. We sample the differences X, Y using paths with the same PSS coordinates (top row) and the shift operator of Kettunen et al. [2015] (bottom row), which better correlates the samples. The shift operator estimates X, Y with lower variance and results in a better reconstruction (CVPT-uni), which can be further improved using our weighted reconstruction (CVPT-opt).*

Section 5.4. With uniform weights and using F, X, and Y buffers, estimator (20) can be written as:

$$\begin{aligned}\langle F_p^{i+1}\rangle = &\frac{1}{5}\langle F_p^i\rangle \\ &+ \frac{1}{5}\left(\langle F_l^i\rangle + \langle X_l\rangle\right) + \frac{1}{5}\left(\langle F_r^i\rangle - \langle X_p\rangle\right) \\ &+ \frac{1}{5}\left(\langle F_t^i\rangle + \langle Y_t\rangle\right) + \frac{1}{5}\left(\langle F_b^i\rangle - \langle Y_p\rangle\right). \end{aligned} \tag{22}$$

It is important to note that, while we update the color values F, the finite differences stay fixed. If we perform an infinite number of iterations, the uniform reconstruction will converge to the integrated-gradient image. Adjusting the number of iterations therefore allows "interpolating" between the noisy input image $\langle F^0\rangle$ and the image of integrated gradients, see Figure 5 for a 1D illustration.

5.3 Weighted Reconstruction

In this section, we leverage the theory from Section 3.2 and discuss the possibilities of deriving more optimal weights. It is worth noting that even if we estimate F^0, X, and Y independently, the estimators in subsequent iterations become increasingly correlated as they combine (be it using different weights) values from similar sets of pixels.

Ideally, we would compute the weights as proposed in Equation (11) from the per-pixel covariance matrix. While building the full matrix for our five estimators in $\mathbb{N}_p$ is feasible, updating it through the iterative reconstruction would be non-trivial. This is because estimators at higher iterations combine values from large regions. Estimating their covariance is thus computationally demanding and would require significant book keeping. As such, we propose a simplified scheme, which can be efficiently implemented and still provides tangible benefits over the state of the art.

During rendering, we progressively compute the sample mean variances $\mathrm{Var}\left[\langle F_p^0\rangle\right]$ of pixel colors and sample mean variances $\mathrm{Var}[\langle X_p\rangle]$ and $\mathrm{Var}[\langle Y_p\rangle]$ of pixel differences for every pixel p. Using Equation (23), we use them to approximate the variances of estimators in the initial iteration; these are in turn used in Equation (10) to weight our estimators. The weights in higher iterations are obtained analogously, but using reduced estimator variances. We detail the reduction of estimator variance and other assumptions needed for the aforementioned approach in the following paragraphs.

Independent Estimators. Since we cannot track the estimator covariances through our iterated scheme, we instead simply assume they are independent. This assumption leads to sub-optimal weights, but still significantly improves upon the uniform weights. Note that our deterministic variance-reduction model defined below accounts to some extent for the increased correlation as we iterate our reconstruction, and mitigates the impact of assuming independent estimators.

Locally Uniform Variance. The variance of a neighbor estimator can be computed as the variance of that neighbor's color increased by the variance of the corresponding finite difference, i.e. $\mathrm{Var}\left[\langle F_q^i\rangle\right] + \mathrm{Var}[\langle F_q - F_p\rangle]$ for neighbor q. However, we found that using the neighbor's color variance $\mathrm{Var}\left[\langle F_q^i\rangle\right]$ is detrimental in practice because bright neighbors tend to have higher variance and therefore lower weights; this leads to significant energy loss as shown in Figure 6. To address this issue, we observe that our iterated reconstruction leads to a locally uniform variance in the limit. Consequently, we assume the neighbor pixel has the same variance as the center pixel, and compute the neighbor estimator variance as:

$$\mathrm{Var}\left[\langle F_p^i\rangle_q\right] = \mathrm{Var}\left[\langle F_p^i\rangle\right] + \mathrm{Var}[\langle F_q - F_p\rangle]\,. \tag{23}$$

Additionally, we update each pixel variance using the median variance of its neighborhood $\mathbb{N}_p$ at the start of the first iteration, since the input sample variance can be fairly noisy at low sampling rates. As can be seen in Figure 6, using our locally uniform variance assumption results in a much more robust energy preservation.

Deterministic Variance Model. Computing the variance of pixel colors after the first iteration is challenging, as the weights are random variables correlated with the data. We thus propose to update the variance of pixel colors using an idealized model:

$$\mathrm{Var}\left[\langle F_p^i\rangle\right] = \frac{\mathrm{Var}\left[\langle F_p^{i-1}\rangle\right]}{\epsilon + 1 + 4\times 0.5^{i-1}}, \tag{24}$$

which effectively assumes an ideal variance reduction at every iteration. At the first iteration $i = 1$, and the variance is reduced by a factor of $5\times$ (ignoring ϵ), which corresponds to the number of independent estimators that we are combining. The factor $4\times 0.5^{i-1}$ roughly models the amount of new (independent) information that the four neighbor estimators contribute in each iteration. To ensure that the variance converges to zero in the limit, we increase the denominator by a small epsilon; we use $\epsilon = 0.01$.

5.4 Relation to Screened Poisson Equation

In this section, we relate our reconstruction to the screened Poisson solver used in previous GDR algorithms [Lehtinen et al. 2013; Manzi et al. 2014; Kettunen et al. 2015; Manzi et al. 2015]. Without loss of generality, we will restrict our discussion to a 1D case of reconstructing a single scanline of an input image.

Previous GDR techniques reconstruct the final image using the Screened Poisson Equation to compute the estimate U that best matches the input data F and gradient X in the L^2 sense,

$$\int (\lambda(U-F))^2 + (\nabla U - X)^2 dx, \tag{25}$$

where λ is a parameter that controls the weight of the screening function. As λ approaches 0, the reconstruction converges to the

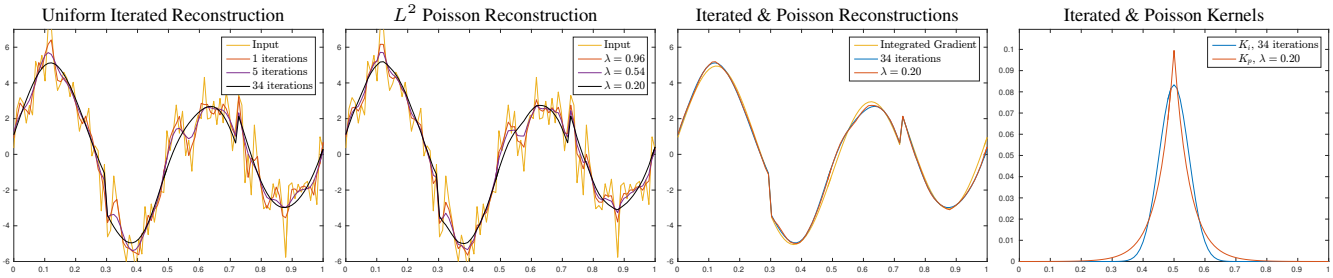


Figure 5: *Reconstruction of a 1D signal using our uniform iterated scheme and an L^2 Poisson solver. Both reconstructions use noisy values and noise-free differences. The λ values for the Poisson solver were chosen to give a similar results to our scheme. The corresponding filtering kernels are shown on the right: $K_p = \lambda^2 \exp(-\lambda|r|)$ for the Poisson solver, and K_i is the i-times self-convolved box filter $[1,1,1]/3$.*

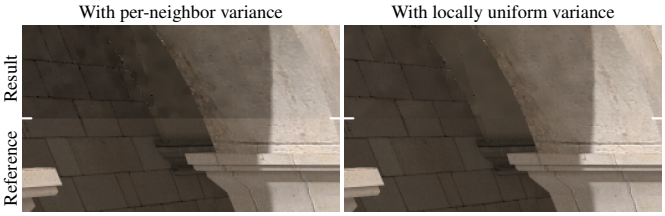


Figure 6: *Reconstruction of the* SPONZA *scene with per-neighbor variance to compute the reconstruction weights (left) and with locally uniform assumption (right). The upper part of each image is the reconstruction output, while the lower part is the reference rendering. The energy loss results in a darkened reconstruction with per-neighbor variance.*

integrated gradient I, such that $\nabla I = X$, and as λ tends to infinity, the reconstruction converges to the input F. The parameter λ has a very similar impact as the number of iterations in the iterated reconstruction described in Section 5.2.

We will now show that our iterated CV reconstruction and the screened Poisson reconstruction are specific instances of a more general scheme. Minimizing the energy function (25) amounts to solving the screened Poisson equation (see Bhat et al. [2008] for a full derivation),

$$\left[\Delta - \lambda^2\right] U = \nabla X - \lambda^2 F = -S, \tag{26}$$

where $S = \lambda^2 F - \nabla X$ is called the *source* function. Using Green's theory, we can directly compute the solution U from the source S,

$$U = G * S = G * (\lambda^2 F - \nabla X) = G * (\lambda^2 F - \Delta I),$$

where the Green's function G is the response of the screened Poisson operator $\left[\Delta - \lambda^2\right]$ to an impulse, and I is the integrated gradient, such that $\Delta I = \nabla X$. Equivalently, we can write

$$U = \lambda^2 G * \left(F - \frac{\Delta I}{\lambda^2}\right). \tag{27}$$

In particular, we have

$$I = \lambda^2 G * \left(I - \frac{\Delta I}{\lambda^2}\right), \tag{28}$$

which follows directly from the application of the Green's function. This is because applying the screened Poisson equation to I gives:

$$[\Delta - \lambda^2] I = \Delta I - \lambda^2 I = -(\lambda^2 I - \Delta I), \tag{29}$$

and I is therefore the result of convolving the source $(\lambda^2 I - \Delta I)$ with the Green's function, $I = G * (\lambda^2 I - \Delta I) = \lambda^2 G * \left(I - \frac{\Delta I}{\lambda^2}\right)$, which is Equation (28).

We can now combine Equations (27) and (28),

$$\begin{aligned} U &= \lambda^2 G * \left(F - \frac{\Delta I}{\lambda^2}\right) - \lambda^2 G * \left(I - \frac{\Delta I}{\lambda^2}\right) + I \\ &= \lambda^2 G * (F - I) + I. \end{aligned} \tag{30}$$

In Equation (30), the output U can be interpreted as filtering the residual $(F - I)$, which represents the disagreement between the throughput and gradient data, and adding it back to the integrated gradient. Let us now define a general filtering kernel K, such that

$$U = K * (F - I) + I. \tag{31}$$

We can now set K to be any normalized kernel, in order to get an unbiased reconstruction. For instance, if we set K to be the identity filter, then $U = F$. If we set K to be a uniform filter with infinite support, then $U = I$, since we have $\int (F - I) = 0$ by construction. For the L^2 Poisson solver, we have $K_p = \lambda^2 G$, and for our iterated reconstruction, K_i is the result of i-times self-convolving the averaging kernel, $[1,1,1]/3$. We show in Figure 5 reconstructions obtained using an L^2 Poisson solver and our uniform iterated scheme, as well as the corresponding filtering kernels. It is important to note that only the kernel $K_p = \lambda^2 G$ corresponds to solving the screened Poisson equation. Consequently, our proposed iterated reconstruction is *not* a screened Poisson solver, but rather an alternate reconstruction with similar properties; this similarity stems from the similarity of the kernel shapes for some combinations of λ and iteration count.

We can also define a different filtering kernel for each pixel of the image, which is effectively what our weighted iterated scheme and the L^1 Poisson solver are doing. We will now show how we can use the weights of the first iteration of our iterated scheme to weight an L^2 Poisson solver, and directly get results of similar or better quality than the L^1 reconstruction.

Weighted L^2 reconstruction. Our iterated reconstruction using uniform weights differs from solving the screened Poisson equation only in the nature of the noise-filtering kernel used: a self-convolved box filter in the case of our iterated reconstruction, and the corresponding Green's function for the screened Poisson reconstruction. Conceptually, our weighted reconstruction is similar to the L^1 Poisson reconstruction used in previous GDR techniques, since the L^1 reconstruction is implemented as an iterated weighted least square solver, i.e. the L^1 solution corresponds to the solution of a specifically weighted L^2 solver. A similar result can be

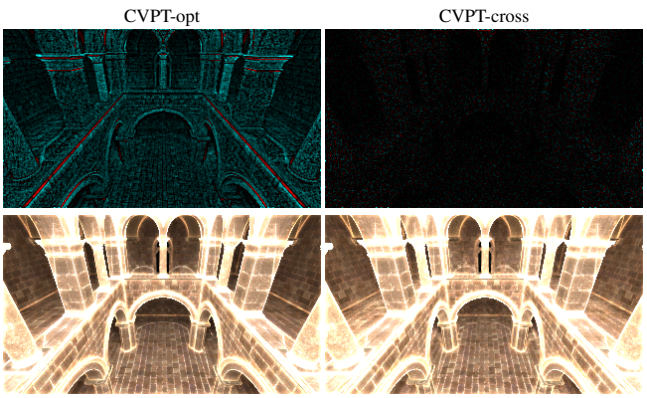


Figure 7: *Reconstruction bias and standard deviation in the* SPONZA *scene (rendered with 64 samples per pixel) after 20 iterations of our proposed reconstruction scheme. We visualize the bias (top row; red for positive, cyan for negative) and standard deviation (bottom row) of our weighted reconstruction (CVPT-opt), and an unbiased variant (CVPT-cross) using a cross-weighting scheme. Images are scaled by 100 for visualization. The cross-weighting scheme eliminates the bias, at the cost of an increase in the standard deviation of the reconstruction.*

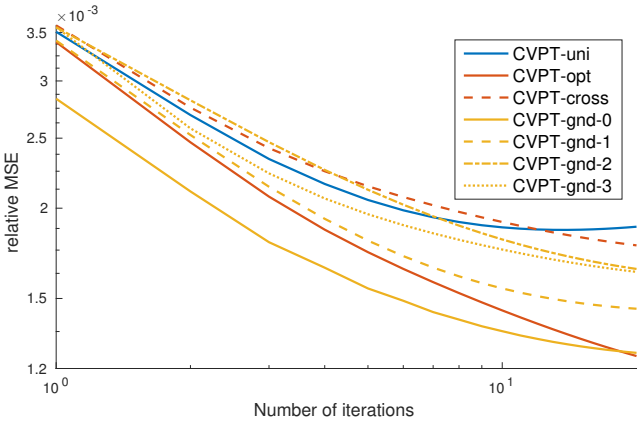


Figure 8: *We plot the relative MSE as a function of the number of iterations, when reconstructing the* SPONZA *scene (rendered with 64 samples per pixel) with our iterated schemes driven by noisy statistics, and weighted reconstructions driven by ground truth statistics, using zero, one, two, or all three of our assumptions (CVPT-gnd-0 to CVPT-gnd-3), illustrating the cumulative impact of our assumptions on the reconstruction quality.*

obtained by weighting the gradient constraints in the Poisson reconstruction, while always using a unit weight for the throughput constraints. For this, we directly leverage the weights computed for our first reconstruction iteration. Each gradient is used to compute two neighbor estimates (left-right pair, or top-bottom pair). In each case, we compute the relative weight of the neighbor estimate as the ratio between the corresponding neighbor estimator and the current value estimator, and the final gradient weight is taken as the minimum of the two ratios. For brevity, we consider only the horizontal case. The difference X_p between neighbors p and r is used to estimate the color of p using r, $\langle F_p^0 \rangle_r$, and vice versa, $\langle F_r^0 \rangle_p$. The weight that we assign to X_p in our weighted L^2 Poisson solver reads:

$$w\left(X_p\right) = \min\left(\frac{\operatorname{Var}\langle F_p^0\rangle_r}{\operatorname{Var}\langle F_p^0\rangle}, \frac{\operatorname{Var}\langle F_r^0\rangle_p}{\operatorname{Var}\langle F_r^0\rangle}\right). \tag{32}$$

We only reweight the gradient entries of the system, in order to prevent energy loss. Manzi et al. [2016] similarly observed that the L^1 solver suffers from energy loss and proposed a mixed L^1 – L^2 reconstruction, where the throughput entries are left untouched; see our supplemental material for a comparison to that solver.

5.5 Results

We performed a number of experiments and comparisons to evaluate our proposed weighted iterated reconstruction. We consider the source of bias in our reconstruction, the impact of our simplifying assumptions on the reconstruction, and also compared our results to reconstructions used in previous works.

Reconstruction Bias. As in the re-rendering application of Section 4, our weighted reconstruction is inherently biased, since the weights are computed according to the statistics of the rendering itself. Here again, we can obtain independent statistics by distributing our samples evenly over two half buffers, as proposed by Rousselle et al. [2012]. By using the first buffer statistics to weight the reconstruction of the second buffer, and vice versa, we can obtain an unbiased reconstruction. In Figure 7, we visualize the average difference between 1000 weighted reconstructions of the SPONZA scene (each rendered with a different random number generator seed) and a ground truth rendering. The bias of our weighted reconstruction (CVPT-opt) is clearly visible, but goes away when using independent statistics by cross weighting two half buffers (CVPT-cross). This unbiased variant however typically performs worse, in particular in the presence of outliers. This translates into an increased standard deviation in the unbiased reconstruction, as shown in Figure 7; see our supplemental materials for additional results.

Assumptions. In Figure 8, we evaluate the impact of the simplifying assumptions of our weighted iterated reconstruction (see Section 5.3) on the SPONZA scene. Again we use 1000 independent renderings, from which we can compute ground truth covariance matrices for every pixel at each step of our iterative reconstruction. We then gradually apply our approximations to these ground truth statistics: with no assumption (CVPT-gnd-0), only assuming independent estimators (CVPT-gnd-1), also assuming locally uniform variance (CVPT-gnd-2), and with all three assumptions (CVPT-gnd-3). We also provide errors for the uniform (CVPT-uni) and weighted (CVPT-opt and CVPT-cross) reconstructions using noisy statistics. The ground truth reconstructions are unbiased and they should therefore be compared to our unbiased weighted reconstruction variant (CVPT-cross). In practice, our assumptions do impact the reconstruction quality, but strike a reasonable balance between practicality and accuracy. Also, as illustrated in Figure 7, our assumptions only affect the quality of the reconstruction, and do not prevent us from achieving unbiased reconstructions, provided we use independent statistics.

Comparisons to Screened Poisson Solvers. In Figure 9, we compare our proposed uniform and weighted iterated reconstructions (CVPT-uni and CVPT-opt) to different variants of (weighted) screened Poisson solvers: the L^2 and L^1 reconstructions (GDPT-L2 and GDPT-L1) used in the GDPT algorithm of Kettunen et al. [2015], and our proposed weighted L^2 solver (GDPT-WL2). In practice, our weighted iterated reconstruction gives similar or slightly better results than the L^1 solver, and takes 16 core seconds on a 3.2GHz Intel Core i7 CPU to process a 1280×720 rendering, compared to 19 core seconds for the L^1 solver (using

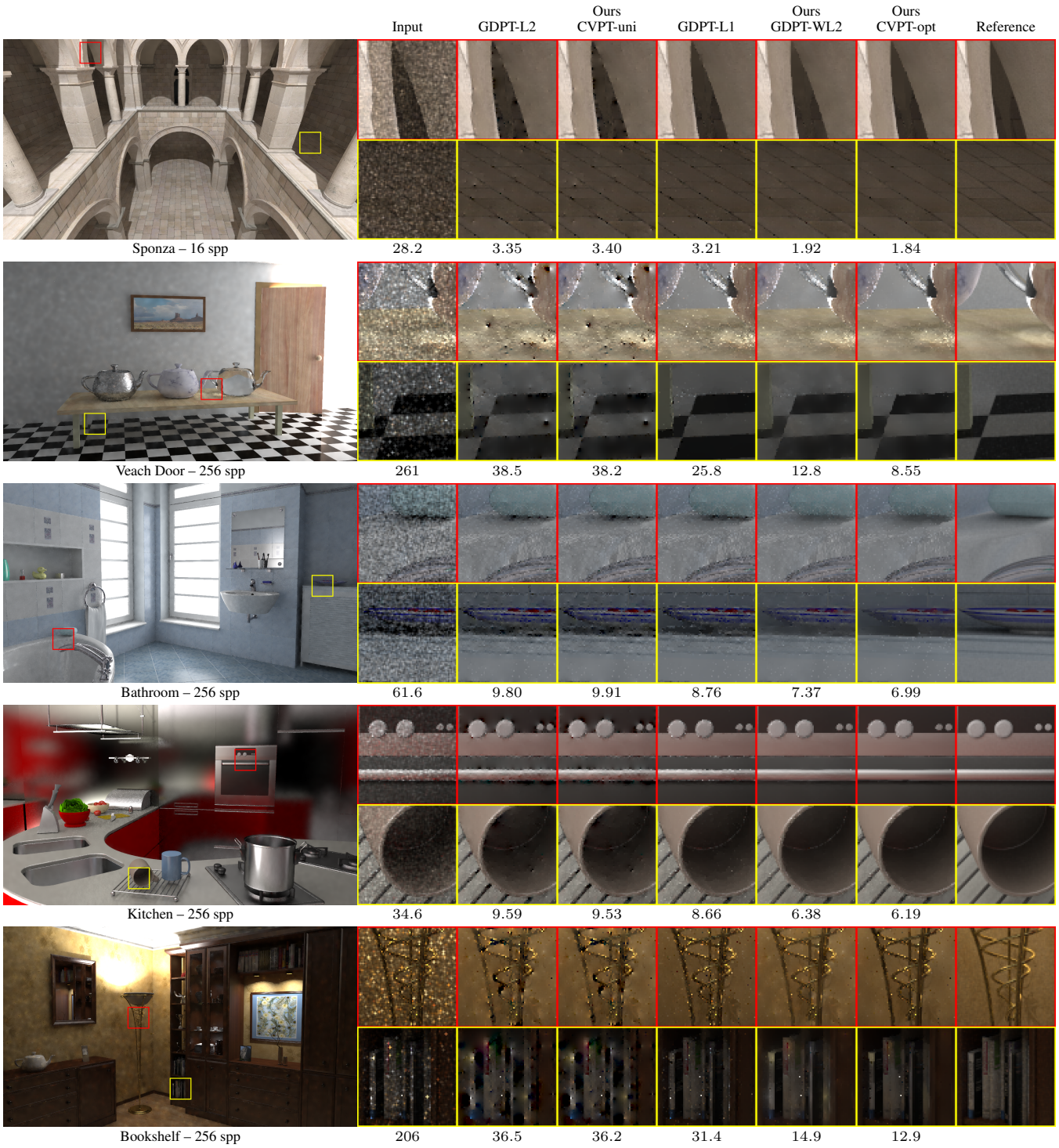


Figure 9: *Denoising results using our interated uniform (CVPT-uni, 50 iterations) and weighted (CVPT-opt, 100 iterations) reconstructions, compared to the L^2 and L^1 Poisson reconstructions of the GDPT algorithm (GDPT-L2 and GDPT-L1, $\lambda = 0.2$). Our uniform reconstruction is very close the the L^2 Poisson reconstruction, and our weighted reconstruction offers a significantly improved results, with better energy preservation and fewer residual noise than the L^1 Poisson reconstruction. Our weighted L^2 Poisson reconstruction (GDPT-WL2, $\lambda = 0.2$) offers similar or better results than the L^1 reconstruction, but does not require an iterated least square solver. The numbers at the bottom of each image report the relative MSE $\times 10^{-3}$ of the respective estimator.*

a C++ implementation for both, the GPU implementation of the GDPT-L1 reconstruction takes only 1 second). In general, the reconstruction time is only a small fraction of the rendering time. For instance, rendering the SPONZA and BATHROOM scenes at 16 samples per pixel respectively takes 12 and 13 core minutes. Our weighted L^2 solver, where gradients constraints are weighted using the weights designed for our iterated reconstruction gives similarly robust results as with the L^1 reconstruction, but with a better energy preservation and at a lower cost. Our supplemental material provides results at 16, 64, 256, and 1024 samples per pixel, as well as results using the unbiased variant of our weighted iterated reconstruction, and results with the mixed L^1 –L^2 screened Poisson solver proposed by Manzi et al. [2016], which was designed to better preserve energy than the standard L^1 solver.

Comparisons to Image-space Denoisers. While GDR can be seen as a form of denoising, gradient-domain reconstructions (including our own) are not competitive with state-of-the-art image-space denoisers. We refer the reader to the work of Manzi et al. [2016] for a discussion on the subject, and to our supplemental material for comparisons with two recent image-space denoisers [Rousselle et al. 2013; Bitterli et al. 2016].

6 Conclusion

We have presented an overview of the two-level Monte Carlo integration framework, a very simple but powerful tool, and applied it in the context of re-rendering and gradient-domain rendering. In our gradient-domain rendering application, we have a fairly noisy set of control variates (the four adjacent pixels), and use an iterated scheme to gather information from a larger neighborhood, which effectively offers useful control data. In our re-rendering application, we directly use a control image rendered with a larger number of samples, and propose a full reconstruction scheme that exploits the correlation between the standard path tracer estimator and the control-variate estimator, to offer high-quality re-rendering results at low sample rates, even for fairly large edits.

Our work also demonstrate how previous works, Consistent Scene Editing by Progressive Difference Image [Günther and Grosch 2015] for the re-rendering application, and Gradient-domain Path Tracing [Kettunen et al. 2015] for the gradient-domain rendering application, were both actually implementing control-variate integration schemes. Both previous works used some form of weighted reconstruction designed to adress the limitations of the control-variate estimator, and we showed how these weighting schemes can be interpreted as instances of a more general framework. We also proposed alternative weighted reconstructions, based on the theory of optimal estimator combinations, which offer consistent improvements on the state-of-the-art.

We intend to further explore potential applications of control-variate integration. In particular, our re-rendering application assumes a static scene, and we expect that leveraging the shift operator proposed by Kettunen et al. [2015] would increase the usefulness of re-rendering for dynamic scene edits. It would also be interesting to explore adaptive sampling strategies to improve our denoising application, as well as the ideal tradeoff between sampling the pixel colors and their gradients. Given the significant improvement offered by the shift operator, compared to correlated sampling in primary sample space, it is clear that further research in the design of improved shifts holds great potential. Lastly, we believe that control-variate integration could be leveraged to perform sparse spatio-temporal sampling of animated sequences, which could offer some interesting scaling to high framerate and high resolution renders.

Acknowledgements

We thank Thijs Vogel and Delio Vicini for helpful discussions, the reviewers for their valuable comments, as well as the authors of the various scenes we used: SPONZA modeled by Marko Dabrovic; VEACH DOOR modeled after Eric Veach by Miika Aittala, Samuli Laine, and Jaakko Lehtinen; scenes from Evermotion Archinterior vol. 1, ported to the Mitsuba renderer by Anton Kaplanyan (KITCHEN) and Tiziano Portenier (BATHROOM and BOOKSHELF); HORSE ROOM modeled by Wig42 and ported to the PBRT renderer by Delio Vicini.

References

BARTH, A., SCHWAB, C., AND ZOLLINGER, N. 2011. Multi-level monte carlo finite element method for elliptic pdes with stochastic coefficients. *Numerische Mathematik 119*, 1, 123–161.

BHAT, P., CURLESS, B., COHEN, M., AND ZITNICK, C. L. 2008. *Fourier Analysis of the 2D Screened Poisson Equation for Gradient Domain Problems*. Springer Berlin Heidelberg, Berlin, Heidelberg, 114–128.

BITTERLI, B., ROUSSELLE, F., MOON, B., IGLESIAS-GUITIÁN, J. A., ADLER, D., MITCHELL, K., JAROSZ, W., AND NOVÁK, J. 2016. Nonlinearly weighted first-order regression for denoising monte carlo renderings. *Computer Graphics Forum 35*, 4, 107–117.

BROADIE, M., AND GLASSERMAN, P. 1998. *Risk Management and Analysis, Volume 1: Measuring and Modelling Financial Risk*. Wiley, New York, ch. Simulation for option pricing and risk management, 173–208.

BUADES, A., COLL, B., AND MOREL, J.-M. 2005. A review of image denoising algorithms, with a new one. *Multiscale Modeling & Simulation 4*, 2, 490–530.

CLARBERG, P., AND AKENINE-MÖLLER, T. 2008. Exploiting visibility correlation in direct illumination. *Computer Graphics Forum 27*, 4, 1125–1136.

COCHRAN, W. G. 1937. Problems arising in the analysis of a series of similar experiments. *Supplement to the Journal of the Royal Statistical Society 4*, 1, 102–118.

COHEN, A., AND SACKROWITZ, H. B. 1974. On estimating the common mean of two normal distributions. *The Annals of Statistics 2*, 6, 1274–1282.

FAN, S., CHENNEY, S., HU, B., TSUI, K.-W., AND LAI, Y.-C. 2006. Optimizing control variate estimators for rendering. *Computer Graphics Forum (Proceedings of Eurographics) 25*, 3, 351–358.

GILES, M. B. 2008. Monte carlo and quasi-monte carlo methods 2006. Springer, Berlin, Heidelberg, ch. Improved Multilevel Monte Carlo Convergence using the Milstein Scheme, 343–358.

GILES, M. B. 2013. Multilevel Monte Carlo methods. In *Monte Carlo and Quasi-Monte Carlo Methods 2012*, J. Dick, Y. F. Kuo, W. G. Peters, and H. I. Sloan, Eds. Springer, Berlin, Heidelberg, 83–103.

GLASSERMAN, P. 2004. *Monte Carlo Methods in Financial Engineering*. Applications of mathematics: stochastic modelling and applied probability. Springer.

GLYNN, P. W., AND SZECHTMAN, R. 2002. *Monte Carlo and Quasi-Monte Carlo Methods 2000*. Springer, Berlin, Heidelberg,

ch. Some New Perspectives on the Method of Control Variates, 27–49.

GRAYBILL, F. A., AND DEAL, R. 1959. Combining unbiased estimators. *Biometrics 15*, 4, 543–550.

GÜNTHER, T., AND GROSCH, T. 2015. Consistent scene editing by progressive difference images. *Computer Graphics Forum 34*, 4, 41–51.

HALPERIN, M. 1961. Almost linearly-optimum combination of unbiased estimates. *Journal of the American Statistical Association 56*, 293, 36–43.

HEINRICH, S. 1998. Monte carlo complexity of global solution of integral equations. *Journal of Complexity 14*, 2, 151 – 175.

HEINRICH, S. 2000. *Advances in Stochastic Simulation Methods*. Birkhäuser Boston, Boston, MA, ch. The Multilevel Method of Dependent Tests, 47–61.

HESTERBERG, T. C., AND NELSON, B. L. 1998. Control variates for probability and quantile estimation. *Management Science 44*, 9 (Sept.), 1295–1312.

KELEMEN, C., SZIRMAY-KALOS, L., ANTAL, G., AND CSONKA, F. 2002. A simple and robust mutation strategy for the Metropolis light transport algorithm. *Computer Graphics Forum 21*, 3, 531–540.

KELLER, T., AND OLKIN, I. 2004. Combining correlated unbiased estimators of the mean of a normal distribution. *Lecture Notes-Monograph Series 45*, 218–227.

KELLER, A. 2001. Hierarchical monte carlo image synthesis. *Mathematics and Computers in Simulation 55*, 1–3, 79 – 92. The Second {IMACS} Seminar on Monte Carlo Methods.

KEMNA, A., AND VORST, A. 1990. A pricing method for options based on average asset values. *Journal of Banking & Finance 14*, 1, 113–129.

KETTUNEN, M., MANZI, M., AITTALA, M., LEHTINEN, J., DURAND, F., AND ZWICKER, M. 2015. Gradient-domain path tracing. *ACM Trans. Graph. 34*, 4 (July), 123:1–123:13.

LAFORTUNE, E. P., AND WILLEMS, Y. D. 1994. The ambient term as a variance reducing technique for Monte Carlo ray tracing. In *Rendering Techniques (Proceedings of the Eurographics Workshop on Rendering)*, 163–171.

LAFORTUNE, E. P., AND WILLEMS, Y. D. 1995. A 5d tree to reduce the variance of Monte Carlo ray tracing. In *Rendering Techniques (Proceedings of the Eurographics Workshop on Rendering)*, 11–20.

LAVENBERG, S. S., MOELLER, T. L., AND WELCH, P. D. 1982. Statistical results on control variables with application to queueing network simulation. *Operations Research 30*, 1, 182–202.

LEHTINEN, J., KARRAS, T., LAINE, S., AITTALA, M., DURAND, F., AND AILA, T. 2013. Gradient-domain Metropolis light transport. *ACM Trans. Graph. 32*, 4 (July), 95:1–95:12.

LOH, W.-L. 1991. Estimating the common mean of two multivariate normal distributions. *The Annals of Statistics 19*, 1, 297–313.

LOH, W. W. 1995. *On the method of control variates*. PhD thesis, Department of Operations Research, Stanford University. PhD thesis.

MANZI, M., ROUSSELLE, F., KETTUNEN, M., LEHTINEN, J., AND ZWICKER, M. 2014. Improved sampling for gradient-domain Metropolis light transport. *ACM Trans. Graph. 33*, 6 (Nov.), 178:1–178:12.

MANZI, M., KETTUNEN, M., AITTALA, M., LEHTINEN, J., DURAND, F., AND ZWICKER, M. 2015. Gradient-domain bidirectional path tracing. In *Eurographics Symposium on Rendering - Experimental Ideas & Implementations*, The Eurographics Association.

MANZI, M., VICINI, D., AND ZWICKER, M. 2016. Regularizing image reconstruction for gradient-domain rendering with feature patches. *Computer Graphics Forum 35*, 2, 263–273.

NELSON, B. L. 1990. Control variate remedies. *Operations Research 38*, 6, 974–992.

NORWOOD, T. E., AND HINKELMANN, K. 1977. Estimating the common mean of several normal populations. *The Annals of Statistics 5*, 5 (09), 1047–1050.

NOVÁK, J., SELLE, A., AND JAROSZ, W. 2014. Residual ratio tracking for estimating attenuation in participating media. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia) 33*, 6 (Nov.).

PHARR, M., AND HUMPHREYS, G. 2010. *Physically Based Rendering: From Theory to Implementation*, 2nd ed. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

ROUSSELLE, F., KNAUS, C., AND ZWICKER, M. 2012. Adaptive rendering with non-local means filtering. *ACM Trans. Graph. 31*, 6 (Nov.), 195:1–195:11.

ROUSSELLE, F., MANZI, M., AND ZWICKER, M. 2013. Robust denoising using feature and color information. *Computer Graphics Forum 32*, 7, 121–130.

SZÉCSI, L., SBERT, M., AND SZIRMAY-KALOS, L. 2004. Combined correlated and importance sampling in direct light source computation and environment mapping. *Computer Graphics Forum (Proceedings of the Eurographics Symposium on Rendering) 23*, 585–594.

SZIRMAY-KALOS, L., TÓTH, B., AND MAGDICS, M. 2011. Free path sampling in high resolution inhomogeneous participating media. *Computer Graphics Forum 30*, 1, 85–97.

ZACKS, S. 1966. Unbiased estimation of the common mean of two normal distributions based on small samples of equal size. *Journal of the American Statistical Association 61*, 314, 467–476.

ZWICKER, M., JAROSZ, W., LEHTINEN, J., MOON, B., RAMAMOORTHI, R., ROUSSELLE, F., SEN, P., SOLER, C., AND YOON, S.-E. 2015. Recent advances in adaptive sampling and reconstruction for Monte Carlo rendering. 667–681.