# Regression-based Monte Carlo Integration

CORENTIN SALAÜN, Max-Planck-Institut für Informatik, Germany
ADRIEN GRUSON, McGill University & École de Technologie Supérieure, Canada
BINH-SON HUA, VinAI Research, Vietnam
TOSHIYA HACHISUKA, University of Waterloo, Canada
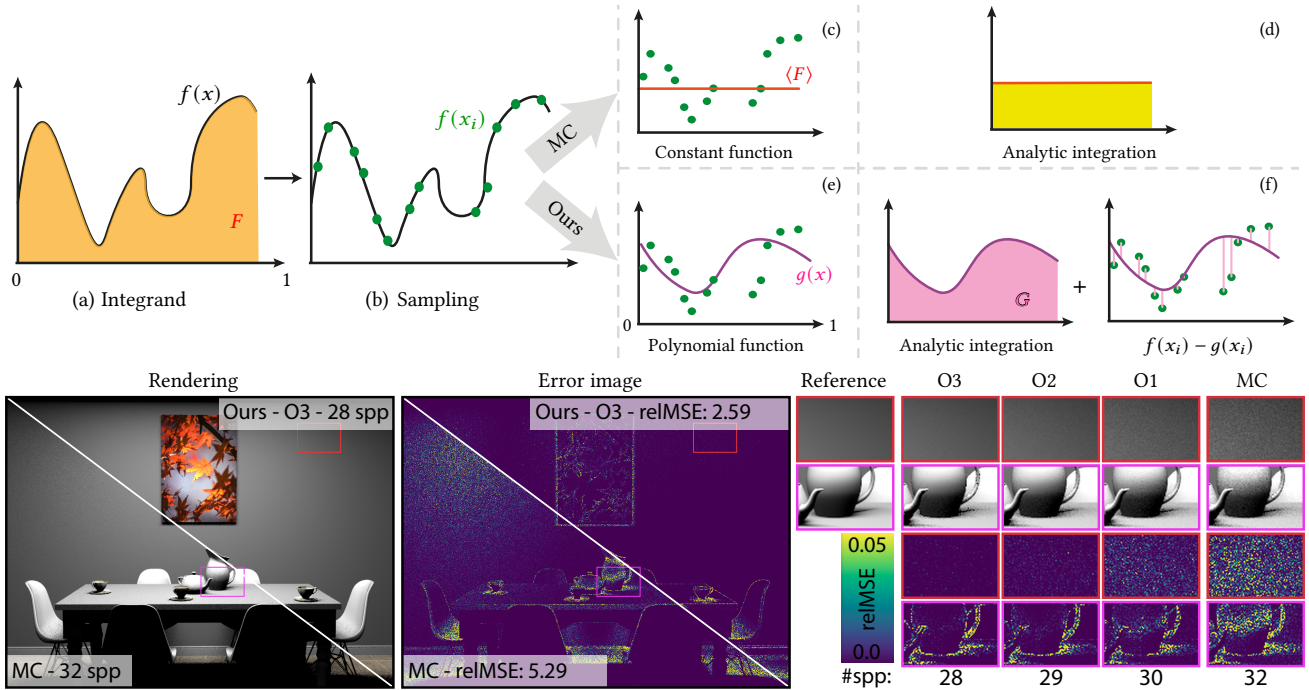GURPRIT SINGH, Max-Planck-Institut für Informatik, Germany

Fig. 1. Given an integrand (a), we first sample (b) $f(x)$ as in Monte Carlo (MC) integration. (c-d) Traditional MC estimator can be interpreted as fitting a *constant* model function to the sample values, with the integral of this constant function equals to $F$. (e) We, instead, propose to use a *non-constant* model function such as a polynomial, which is then fitted to the sampled values. (f) The resulting estimator is based on control variates; we add the analytical integral of the model function to MC integration of the difference between the original integrand and the model function. The bottom row shows renderings and the corresponding error images to demonstrate the impact of our regression approach against the traditional MC integration. The insets on the right compare our method with different orders ($O$x) of polynomials. Our method has significant error reduction at equal time.

Monte Carlo integration is typically interpreted as an estimator of the expected value using stochastic samples. There exists an alternative interpretation in calculus where Monte Carlo integration can be seen as estimating a *constant* function—from the stochastic evaluations of the integrand—that integrates to the original integral. The integral mean value theorem states that this *constant* function should be the mean (or expectation) of the integrand. Since both interpretations result in

the same estimator, little attention has been devoted to the calculus-oriented interpretation. We show that the calculus-oriented interpretation actually implies the possibility of using a more *complex* function than a *constant* one to construct a more efficient estimator for Monte Carlo integration. We build a new estimator based on this interpretation and relate our estimator to control variates with least-squares regression on the stochastic samples of the integrand. Unlike prior work, our resulting estimator is *provably* better than or equal to the conventional Monte Carlo estimator. To demonstrate the strength of our approach, we introduce a practical estimator that can act as a simple drop-in replacement for conventional Monte Carlo integration. We experimentally validate our framework on various light transport integrals. The code is available at https://github.com/iribis/regressionmc.

CCS Concepts: • **Computing methodologies → Rendering**.

Additional Key Words and Phrases: Monte Carlo integration, Regression, Control Variates, Light transport simulation

Authors' addresses: Corentin Salaün, Max-Planck-Institut für Informatik, Saarbrücken, Germany, csalaun@mpi-inf.mpg.de; Adrien Gruson, McGill University & École de Technologie Supérieure, Montreal, Canada, adrien.gruson@gmail.com; Binh-Son Hua, VinAI Research, Vietnam, binhson.hua@gmail.com; Toshiya Hachisuka, University of Waterloo, Waterloo, Canada, toshiya.hachisuka@uwaterloo.ca; Gurprit Singh, Max-Planck-Institut für Informatik, Saarbrücken, Germany, gsingh@mpi-inf.mpg.de.

# 1 INTRODUCTION

Monte Carlo (MC) integration is a basic tool for many numerical integration problems. In textbooks, MC integration is typically interpreted as a statistical process where it samples the integrand and estimates the expected value by taking the average of these samples. While less common, there is another interpretation of MC integration based on the mean value theorem in calculus. The mean value theorem for definite integrals tells us that there exists a *constant* that integrates exactly to the same solution as the given definite integral (see Fig. 1). This constant, by definition, is equal to the mean of the integrand, which can be again estimated by taking the average of random samples, and thus both interpretations result in the same estimator. This alternative interpretation, however, raises interesting questions: *Can we use a non-constant function to construct a MC estimator? If so, what happens?* These questions do not appear in the statistics-oriented interpretation as the expected value is fundamentally tied to the idea of samples' average.

We propose a new MC estimator that takes the same input as MC integration (see Fig. 1b) with no extra assumptions. Instead of merely taking the average to find a constant function (Fig. 1c), we propose to fit a *non-constant* model function (Fig. 1e) to the samples via least-squares regression and utilize its analytical integral to define an estimator (Fig. 1f). We show that our approach can be formulated as the method of control variates, where control variates are obtained via least-squares regression.

While the use of regression in control variates is not new, we show, for the first time how regression, control variates, and MC integration are closely coupled. MC integration is a special case under our formulation where a constant function is the solution of regression. Our approach is provably shown to be better than or equivalent to MC integration; in the worst case, it will revert back to conventional MC integration as long as a constant function is included in regression. This theoretical guarantee has not been explored in prior work on regression-based control variates. To summarize our contributions:

- We introduce a novel formulation of MC integration which connects least-squares regression, control variates, and conventional MC integration.
- We prove that our regression-based formulation leads to a class of estimators that are better than or equal to conventional MC integration in terms of variance.
- We develop a practical algorithm based on our formulation and polynomials, which can work as drop-in improvement over conventional MC integration.
- We demonstrate the effectiveness of our algorithms over light transport integral estimations by performing experiments over multiple dimensions (2D to 15D).

# 2 RELATED WORK

## 2.1 Numerical integration

Integration is a classical subject in numerical analysis [Davis 2013; Trefethen 2012]. The calculus-oriented approach has lead to deterministic quadrature techniques such as a Riemann sum. The quadrature techniques sample the integrand at predetermined points and approximate the given integrand using an analytically integrable function (e.g., quadrangles, trapezoids, and certain basis functions). The calculus-oriented approach, in general, suffers from the exponential growth of the number of samples for higher dimensional functions. A statistics-oriented approach, *MC integration*, performs a similar summation but instead evaluates the integrand at stochastically sampled locations. This stochastic nature of sample generation extends well to higher dimensions [Niederreiter 1992; Owen 2013] without the exponential growth of the number of samples, which has led to its wide adoption in the rendering community. We take a step back from this trend and show how a calculus-oriented reformulation of MC integration can lead to more efficient numerical integration methods.

## 2.2 Control variates

Control variates (CVs) [Glynn and Szechtman 2002; Loh 1995] are variance reduction techniques which utilize MC integration of the difference between a given integrand $f$ and another function $g$ to perform numerical integration. This difference estimate is added to the analytical integral of the function $g$ to estimate the integral of $f$. The improvement over conventional MC integration is dependent on how well $g$ approximates $f$, and the improvement is *not guaranteed* in general.

Owen and Zhou [2000] discussed a practical approach to perform importance sampling with control variates for variance reduction. They showed that adaptive control variates using a mixture density cannot be *much worse* (i. e., can only be slightly worse) than conventional MC using the best mixture component. Our approach, on the other hand, is *provably never worse* than conventional MC using the same PDF (shown in Section 4.3 and Equation (25)). Fan et al. [2006] used the method of Owen and Zhou in rendering with an additional approximation.

Multi-level Monte Carlo [Heinrich 2001; Keller 2001] shows how low resolution renderings can be used as CVs for high-resolution synthesis. Rousselle et al. [2016] shows a connection between CVs and the reconstruction step in gradient-domain rendering [Hua et al. 2019]. Recently, Kondapaneni et al. [2019] showed a connection between weights in multiple importance sampling [Veach and Guibas 1995] and CVs. We show connections among control variates, least-squares regression, and MC integration to have a provable variance reduction. We will discuss more related work [Crespo et al. 2021; Hickernell et al. 2005; Nakatsukasa 2018; Owen and Zhou 2000; Pajot et al. 2014; Rubinstein and Marcus 1985] later once we introduced our approach.

*Construction of control variates in rendering.* Constructing control variates for complex integrands in light transport simulation is a challenging task. In rendering, the simplest form of control variate uses the constant ambient term to approximate incoming illumination [Lafortune and Willems 1994]. This, however, does not work well in complex lighting conditions. Clarberg and Akenine-Möller [2008] used visibility correlations to design control variates for direct illumination. Vévoda et al. [2018] used bayesian regression to construct a light clustering. For further improvement, a control variate function is derived from the respective estimated contribution of each cluster at any shading point. Kutz et al. [2017] proposed a method of control variate extension for heterogenous participating media rendering. It is based on the decomposition of the medium into a control and a residual components. These two parts can be sampled separately. The control variate has also been used by Belcour et al. [2018] to extend their method of integration of spherical harmonic expansions over polygonal domains to non-analytic integrands. Neural control variates [Müller et al. 2020] fit a neural network to define control variates. They also proposed to simultaneously train another network to importance sample the difference function. The resulting estimator shows remarkable improvements,

but with a significant computational overhead and changes to the sampling process that have to be accelerated by utilizing a high-performance multi-GPU workstation. Recently, Subr [2021] propose a neural network Q-NET that uses a neural proxy as a control variate. Our control variates are constructed from the samples within each pixel using least squares regression.

The expected improvement from these control variates is limited by construction due to the lack of complexity captured by these control variates. It is thus reasonable to assume that control variates usually do not approximate the integrand well in light transport simulation. Recognizing this limitation, our approach is designed to be provably better than MC integration as long as least-squares regression can find *any* non-constant solution. A major deviation from prior work is that we show how regression is closely connected to control variates and MC integration. Many prior work on control variates instead focus on introducing a sophisticated model function for control variates. This conceptual difference allows us to realize that we can use a *significantly simpler* model function, such as polynomials, and still provably outperforms MC integration. We demonstrate that our estimators are already practical without any special optimization for regression (e.g., use of high performance hardware like GPUs) and without any change to the sampling process.

## 2.3 Regression in rendering

Regression has been applied for adaptive sampling and reconstruction in rendering [Zwicker et al. 2015]. Moon et al. [2014] proposed to locally approximate the unknown image function by locally regressing a low-order polynomial on auxiliary buffers (normals, depth, textures). To alleviate the computational overhead of the method, Moon et al. [2015] used a iterative linear model to simultaneously reconstruct multiple pixels. These methods control the filtering bandwidths locally to increase numerical accuracy. Bitterli et al. [2016] analyze various zero-order and first-order regression models and use auxiliary data just to fit the data and considers only the pixel color to compute the regression weights. Our framework is largely agnostic to how regression is implemented, and we demonstrate multiple implementations as examples. The regression part of our framework can thus utilize findings in those previous.

## 3 TWO INTERPRETATIONS OF MC INTEGRATION

Let us motivate our main question by recapitulating the two different interpretations of MC integration. Suppose that we want to find a definite integral $F$ of a non-negative function $f(x)$ over the domain $\Omega$ where

$$F = \int_{\Omega} f(x)\mathrm{d}x. \qquad (1)$$

In many practical problems, it is difficult to find the analytical solution to the above. In light transport simulation, the function $f(x)$ is defined as the measurement contribution function for a light transport path $x$ [Veach 1997]. The definite integral in this case is the solution to light transport simulation, which is generally unavailable in an analytical form.

Rather than trying to find an analytical solution, MC integration uses a numerical estimator $\langle F \rangle$ where

$$F \approx \langle F \rangle = \frac{1}{N} \sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)}. \qquad (2)$$

The variables $x_i$ are $N$ independent random samples distributed according to the probability density function $p(x)$ which satisfies

$p(x) \neq 0$ whenever $f(x) \neq 0$ in the domain $\Omega$. This estimator $\langle F \rangle$ is said to be *unbiased* in the sense that its expected value is exactly equal to the integral $F$:

$$\mathrm{E}\left[\langle F \rangle\right] = \frac{1}{N} \sum_{i=1}^{N} \mathrm{E}\left[\frac{f(x_i)}{p(x_i)}\right] = \frac{1}{N} \sum_{i=1}^{N} \int_{\Omega} \frac{f(x)}{p(x)} p(x)\mathrm{d}x = F. \qquad (3)$$

*Change of variables.* The same estimator can also be redefined in a unit hypercube $U$ of uniformly distributed random numbers $u_i$ that are used for generating samples $x_i$. Suppose that we have a mapping $x = \Phi(u)$ where the Jacobian $|\mathrm{d}x/\mathrm{d}u| = 1/p(x)$. By change of variables, we can rewrite the integral into

$$F = \int_{\Omega} f(x)\mathrm{d}x = \int_{U} f(\Phi(u)) \left|\frac{\mathrm{d}x}{\mathrm{d}u}\right| \mathrm{d}u = \int_{U} \hat{f}(u)\mathrm{d}u, \qquad (4)$$

where $\hat{f}(u) = \frac{f(\Phi(u))}{p(\Phi(u))}$, and rewrite the estimator $\langle F \rangle$ into

$$F \approx \langle F \rangle = \frac{1}{N} \sum_{i=1}^{N} \frac{\hat{f}(u_i)}{p(u_i)} = \frac{1}{N} \sum_{i=1}^{N} \hat{f}(u_i). \qquad (5)$$

In other words, by properly redefining the integrand as above, one can always consider having uniformly distributed samples $u_i \sim p(u) = 1$ in this unit hypercube $U$, instead of samples $x_i \sim p(x)$ in the domain $\Omega$ which are generally not uniformly distributed. This definition is known as as primary sample space [Kelemen et al. 2002] in rendering. We use this formulation in the rest of the paper. One can generally convert Equation 1 to Equation 4 as long as change of variables from $x$ to $u$ is possible.

## 3.1 Statistic-oriented interpretation

The process of MC integration can be seen as an estimation of the expected value of $\hat{f}(u)$ since we have

$$F = \int_{U} \hat{f}(u)p(u)\mathrm{d}u = \mathrm{E}\left[\hat{f}(u)\right] \approx \frac{1}{N} \sum_{i=1}^{N} \hat{f}(u_i) = \langle F \rangle \qquad (6)$$

where $u$ is distributed according to $p(u) = 1$. Under this interpretation, one can also derive an expression of the expected squared errors of $\langle F \rangle$ as

$$\mathrm{E}\left[(\langle F \rangle - F)^2\right] = \mathrm{Var}\left[\langle F \rangle\right] = \frac{1}{N} \mathrm{Var}\left[\hat{f}(u)\right] \qquad (7)$$

where we used the unbiasedness of the estimator $\mathrm{E}\left[\langle F \rangle\right] = F$. It is a well known expression that shows that the expected squared error of MC integration is proportional to the variance of $\hat{f}(u)$ and inversely proportional to the number of samples $N$.

## 3.2 Calculus-oriented interpretation

One can also interpret MC integration as a two-step process; *replacement* of the integrand $\hat{f}$ by a model function $\hat{g}$ and succeeding *analytical* integration of $\hat{g}$. Let us consider a constant function $\hat{g}(u) = c$ as the model function. To keep its integral $G$ equal to the original integral $F$, we would like to find a value for $c$ such that

$$F = \int_{U} \hat{f}(u)\mathrm{d}u = \int_{U} \hat{g}(u)\mathrm{d}u = G. \qquad (8)$$

The integral mean value theorem in calculus shows that there exist such a constant $c$ for a given definite integral. Since the function $\hat{g}(u)$ can be analytically integrated, we have

$$F = G = \int_{U} \hat{g}(u)\mathrm{d}u = c|U| \quad \therefore c = \frac{1}{|U|} \int_{U} \hat{g}(u)\mathrm{d}u \qquad (9)$$

where $|U| = 1$ is the volume of the unit hypercube $U$. Therefore, the constant $c$ can be derived as the *average* of $\hat{f}(u)$ since

$$c = \frac{1}{|U|} \int_U \hat{g}(u) du = \frac{1}{|U|} \int_U \hat{f}(u) du \qquad (10)$$

where the last expression is the definition of the average of $\hat{f}(u)$ over the unit hypercube $U$. Note that there is no approximation introduced until this point. Since $c$ is defined as the average of $\hat{f}(u)$, we can estimate $c$ by taking the average of samples;

$$c \approx \frac{1}{N} \sum_{i=1}^{N} \hat{f}(u_i) = \langle F \rangle. \qquad (11)$$

This interpretation thus leads to the same estimator as the statistic-oriented interpretation since

$$F = \int_U \hat{g}(u) du = \int_U c \, du \approx \int_U \left( \frac{\langle F \rangle}{|U|} \right) du = \langle F \rangle. \qquad (12)$$

Under this interpretation, MC integration can be seen as the process of first estimating a function $\hat{g}(u) = c$, and then analytically integrating (the approximation of) the function $\hat{g}(u)$.

### 3.3 Problem statement

One might see that the calculus-oriented interpretation is rather redundant at first, since the end result is the same. However, this interpretation actually points out the existence of $\hat{g}$ in the definition of the estimator, and also the fact that MC integration is derived by using a constant function $\hat{g} = c$. This existence of $\hat{g}$ is not apparent in the statistics-oriented interpretation.

Our main question is whether making $\hat{g}$ *non-constant* is possible and whether it is beneficial to consider such $\hat{g}$ to begin with. We show that such a $\hat{g}$ can be defined based on *control variates* and *least-squares regression* of $\hat{g}$ to $\hat{f}$. Our formulation shows that conventional MC integration is a special case where regression *happens to return* a constant function for $\hat{g}$. We also show that considering a more complex $\hat{g}$ is *provably better* than or equal to simply having a constant $\hat{g}$ in terms of its expected error.

## 4 REGRESSION-BASED MC INTEGRATION

We introduce a novel formulation of MC integration, *regression-based MC integration*, which allows us to consider any analytically integrable functions for $\hat{g}$, in place of the constant function. Fig. 1 illustrates our approach for a 1D function.

We start by considering an arbitrary model function $\hat{g}(u)$ which can be analytically integrated. Unlike the calculus-oriented interpretation, our function $\hat{g}(u)$ can be non-constant and the integral $G$ *does not need to be* equal to the original integral $F$. We can use such $\hat{g}(u)$ by adding the *difference* from the original integral as

$$F = G + (F - G) \qquad (13)$$

where $(F - G)$ is estimated by MC integration;

$$F - G \approx \frac{1}{N} \sum_{i=1}^{N} \left( \hat{f}(u_i) - \hat{g}(u_i) \right) = \langle D \rangle. \qquad (14)$$

The final estimator is defined as a sum of two terms where

$$F \approx \langle F \rangle^* = G + \langle D \rangle. \qquad (15)$$

The first term $G$ is analytically evaluated for this given $\hat{g}(u)$ in a deterministic manner. This estimator is not new and is known as the method of *control variates*. The function $\hat{g}(u)$ is called the control variate which is typically *given* by a user. In our method, we find $\hat{g}(u)$ by (least-squares) *regression*.

Control variates are *unbiased* regardless of $\hat{g}(u)$. Since $G$ is deterministic, the expected squared error of $\langle F \rangle^*$ is given by the difference estimator

$$\mathrm{E}\left[ \left( \langle F \rangle^* - F \right)^2 \right] = \mathrm{Var}\left[ \langle D \rangle \right] = \frac{1}{N} \mathrm{Var}\left[ \hat{f}(u) - \hat{g}(u) \right]. \qquad (16)$$

Control variates thus lead to a smaller expected square error compared to conventional MC integration *only* when $\mathrm{Var}[\hat{f}(u) - \hat{g}(u)] < \mathrm{Var}[\hat{f}(u)]$. Error reduction is thus *not guaranteed* and it is widely recognized so [Hickernell et al. 2005].

Our key contribution is to show that least-squares regression not only helps us automatically finding $\hat{g}(u)$ but also *provably* reduces the expected squared error *when a constant function is included in regression*.

As we discuss later, a constant function has been usually ignored in (adaptive) control variates. Our work is the first to show that this combination of the least-square regression and inclusion of a constant function results in the provable improvement with a proper reduction to MC integration.

Fig. 2 showcases our method for simple 1D integration problems. The labels $O1$, $O3$, and $O5$ in our method mean different orders of polynomials used for $\hat{g}$ as we explain later. The convergence plots show that, after a certain number of samples, our method is no worse than MC integration regardless of the order of polynomials. It is also worth noting that regression might *not* well approximate the integrand (e.g., step and complex cases). The method of control variates alone with such a bad approximation could *increase* the error. Even in such cases, our method still outperforms MC integration once regression converges after some numbers of samples. When the integrand can be modeled exactly by $\hat{g}$, as shown in the last row where the integrand is a 5th-order polynomial, our method dramatically reduces numerical error compared to MC integration. We now explain how our estimator $\langle F \rangle^*$ achieves this reduction of errors by showing its connections to control variates and MC integration.

### 4.1 Least-squares regression of $\hat{g}(u)$

We model $\hat{g}(u, \theta)$ as a parametric function with the parameters $\theta = (c_0, \cdots, c_M)$. We define the *residual* for $\theta$ as

$$R(\theta) = \int_U \left( \hat{f}(u) - \hat{g}(u, \theta) \right)^2 du. \qquad (17)$$

Let us then consider regression of $\hat{g}(u)$ to $\hat{f}(u)$ such that the residual $R(\theta)$ is minimized. Since analytical solution to the integral in $R(\theta)$ is usually not available, we first consider using $N$ random samples $u_i$ to approximate $R(\theta)$

$$R(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \left( \hat{f}(u_i) - \hat{g}(u_i, \theta) \right)^2 = \langle R \rangle(\theta), \qquad (18)$$

which is equivalent to MC integration of the function $(\hat{f}(u) - \hat{g}(u, \theta))^2$ using the estimator $\langle R \rangle(\theta)$. This unbiased approximation thus converges to $R(\theta)$ as we increase the number of samples. This expression of $\langle R \rangle(\theta)$ can also be seen as the squared $\mathcal{L}_2$ norm of the differences between $\hat{f}(u_i)$ and $\hat{g}(u_i, \theta)$ on the samples $u_i$. The solution to this regression can be obtained via *least-squares regression* of $\hat{g}(u, \theta)$ to $N$ pairs of $(u_i, \hat{f}(u_i))$.
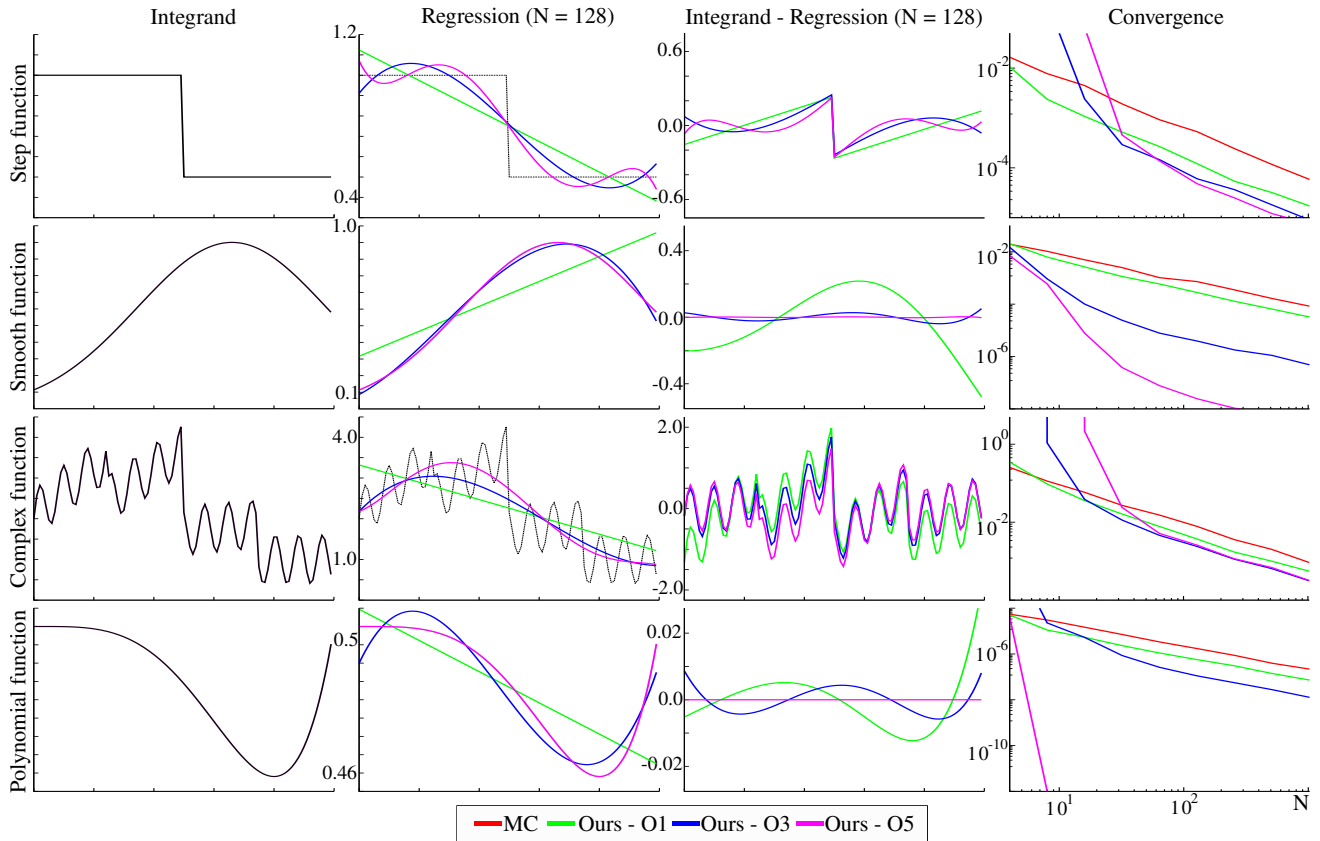
Fig. 2. Regression-based Monte Carlo integration in 1D: a step function, a shifted Gaussian, a high-frequency function, and a polynomial. Our method is no worse than MC integration. It is not required that the model function well approximates the integrand for our method to work (e.g., the step function and high-frequency function case where our model function is very different from the integrand). When the model function is a good fit (e.g., the shifted Gaussian, polynomial function case), the error can be reduced dramatically.

## 4.2 Connection to control variates

The original definition of control variates considers a given function $h(u)$ and a scalar parameter $\alpha$ such that

$$F = \int_U \hat{f}(u)\mathrm{d}u = \alpha H + \int_U (\hat{f}(u) - \alpha h(u))\mathrm{d}u \qquad (19)$$

where $H = \int_U h(u)\mathrm{d}u$. Given this definition, it is known that

$$\alpha = \mathrm{Cov}\left[\hat{f}, h\right] / \mathrm{Var}\left[h\right] \qquad (20)$$

minimizes the variance of $\langle F \rangle^*$. This approach can be seen as a special case of our formulation where $\theta = \alpha$ and $\hat{g}(u, \theta) = \alpha h(u)$.

Let us analyze the error of $\langle F \rangle^*$ when $\hat{g}(u, \theta)$ is given by least-squares regression. Since the expected squared error of $\langle F \rangle^*$ is equal to the variance of $\hat{f} - \hat{g}$ (Equation (16)), we have

$$\mathrm{Var}\left[\hat{f}(u) - \hat{g}(u, \theta)\right] = \mathrm{E}\left[\left(\hat{f}(u) - \hat{g}(u, \theta)\right)^2\right] - \mathrm{E}\left[\hat{f}(u) - \hat{g}(u, \theta)\right]^2$$

$$= \int_U \left(\hat{f}(u) - \hat{g}(u, \theta)\right)^2 \mathrm{d}u - (F - G)^2 = R(\theta) - (F - G)^2. \qquad (21)$$

where we used the fact that $p(u) = 1$ in the primary sample space and substituted $R(\theta)$. The expected squared error of the estimator is thus bounded as

$$\mathrm{E}\left[\left(\langle F \rangle^* - F\right)^2\right] = \frac{1}{N}\left(R(\theta) - (F - G)^2\right) \le \frac{1}{N}R(\theta). \qquad (22)$$

We can see that, by defining the control variate $\hat{g}(u)$ via least-squares regression, one can also minimize (the bound of) the

expected squared error of the estimator. In practice, we cannot directly minimize $R(\theta)$, but we minimize its estimate $\langle R \rangle(\theta)$, so this relationship does not exactly hold for regression with a finite number of samples as one can also see in Fig. 2. Our experiments, however, show that minimizing $\langle R \rangle(\theta)$ works well in practice when $N$ is not too small.

Note also that control variates directly minimize the variance by setting $\alpha$, while we minimize only its bound by least-squares regression.

While it is tempting to consider more general regression to directly minimize the variance in our method, it is the use of least-squares regression which allowed us to reveal clear connections among least-squares regression, control variates, and MC integration as we see in the followings.

## 4.3 Reduction to Monte Carlo integration

It is now trivial to show that our approach reduces to MC integration when regression happened to return a constant function $\hat{g}(u) = c$ as its best fit. Substituting $\hat{g}(u) = c$ to Equation (15),

$$\langle F \rangle^* = c|U| + \frac{1}{N}\sum_{i=1}^{N}\left(\hat{f}(u_i) - c\right) = \frac{1}{N}\sum_{i=1}^{N}\hat{f}(u_i) = \langle F \rangle. \qquad (23)$$

It shows that, when $\hat{g}(u) = c$ is the solution to regression, our approach reduces to MC integration.

In this case, since we have $F = G$, the upper-bound in Equation (22) becomes exact and

$$\mathrm{E}\left[(\langle F \rangle^* - F)^2\right] = \mathrm{E}\left[(\langle F \rangle - F)^2\right] = R(c) \tag{24}$$

where $R(c)$ is the residual for $\hat{g}(u) = c$.

Based on this reduction to Monte Carlo integration, we can show for the first time that the expected squared error of this approach is *provably* better than or equal to MC integration. Deceivingly simple, but the important difference from prior work is that we include a constant function in least-squares regression by adding $c$ to $\hat{g}(u)$ and we minimize the residual $R(\theta)$. This allows us to have a "safe-guard" against performing worse than MC integration, when combined with least-squares regression. Since $R(\theta)$ is minimized for a model function including a constant,

$$R(\theta) \le R(c). \tag{25}$$

It thus follows that

$$\mathrm{E}\left[(\langle F \rangle^* - F)^2\right] \le \frac{1}{N}R(\theta) \le \frac{1}{N}R(c) = \mathrm{E}\left[(\langle F \rangle - F)^2\right]$$

$$\therefore \mathrm{E}\left[(\langle F \rangle^* - F)^2\right] \le \mathrm{E}\left[(\langle F \rangle - F)^2\right]. \tag{26}$$

The two errors become equal only when $R(\theta) = R(c)$, that is, our estimator exactly reduces to MC integration when a constant function is the solution to regression. This inequality also shows that, when a constant function is not the solution, it will always lead to a better or equivalent estimator $\langle F \rangle^*$ than MC integration $\langle F \rangle$. Our estimator is thus *provably better than or equal to* MC integration in terms of its expected squared errors.

In practice, since the solution to least-squares regression is not exact for any finite $N$, this theoretical property of reduction of errors is not guaranteed for any finite $N$. Nonetheless, our numerical results demonstrate that reduction of error happens and stabilize well even for a practical range of $N$.

*Discussion.* While regression is often used as an approach to find a control variate, none of the prior work shows how including a constant function in regression allows us to unify control variates and MC integration. For example, a very thorough discussion on control variates [Hickernell et al. 2005, Section 4] immediately dismisses a constant control variate as redundant since it will not reduce variance at all.

Recognizing a problem of control variates that it could perform worse than MC integration, Pajot et al. [2014] proposed switching to conventional MC when the estimated variance of adaptive control variate is determined to be larger. Our approach does not need this explicit switching, as it is automatically done as a result of least-squares with theoretical guarantee. Although our estimator looks similar to a general mixture estimator by Owen and Zhou [2000], their estimator does not revert back to MC integration. The difference becomes apparent when we fit a mixture density including a constant function. When regression returns a constant as the solution, the method of Owen and Zhou also changes the sampling density to a constant. On the other hand, our method reduces to MC integration while keeping the sampling density as the given mixture density. The two estimators are thus fundamentally different.

Some existing methods [Rousselle et al. 2016; Rubinstein and Marcus 1985] use multiple different control variate functions to take advantage of their respective strengths. Rubinstein and Marcus [1985], in particular, shows that the variance reduction depends on the *optimal* number of control variates used in the estimator. Our work is orthogonal to this work.

---

**Algorithm 1:** Direct matrix estimator.

$f \leftarrow \{\}; u \leftarrow \{\}; \Phi_0 \leftarrow 0^{M \times M}; b_0 \leftarrow 0^{M \times 1};$
**for** $i \leftarrow 1$ **to** $N$ **do**
  $u_i \leftarrow$ random();
  $f_i \leftarrow \hat{f}(u_i);$
  $(\Phi_{i+1}, b_{i+1}) \leftarrow$ update_system$(\Phi_i, b_i, u_i, f_i);$
**end**
$\theta_{\min} \leftarrow$ solve_linear_system$(\Phi_{N+1}, b_{N+1});$
**return** $G(\theta_{\min}) + \frac{1}{N}(\sum_{i=1}^{N} f_i - \hat{g}(u_i, \theta_{\min}));$

---

## 5 POLYNOMIAL-BASED ESTIMATORS

While our formulation supports arbitrary integrable functions for $\hat{g}$, we elaborate a practical implementation with a polynomial function for $\hat{g}$ to demonstrate the strength of our method. In general, a polynomial function can be written as

$$\hat{g}(u) = \sum_{q=0}^{M} c_q \phi_q(u) \tag{27}$$

with monomials $\phi$ defining a polynomial. We can analytically integrate $\hat{g}$ as the weighted sum of the integrals of monomials as

$$\int_U \hat{g}(u)\mathrm{d}u = \sum_{q=0}^{M} c_q \int_U \phi_q(u)\mathrm{d}u. \tag{28}$$

The advantage of polynomials is that it is simple and easily generalizable for different dimensions with different orders of polynomials. Note that we do not claim that polynomials are optimal as control variates. The integrand $\hat{f}$ is, in general, far more complex than a polynomial, and it will not be well approximated by a polynomial in many cases. Using polynomials in the conventional control variates might be seen as a bad approach because of this reason.

What is important here is that a polynomial is still *more complex* than a constant and includes a constant. Our formulation thus provably guarantees improvement over MC integration in this case. Therefore, even with a lower-order polynomial, it is expected that our method either outperforms or at least performs equally to MC integration. We explain two different approaches for regression.

### 5.1 Direct matrix estimator

It is well known that regression with a polynomial can be formulated as a solution to a linear system. The size of this linear system is proportional to the number of monomials $M$ and independent from the number of samples $N$, making it suitable for MC integration.

We can define a linear system by taking partial derivatives with respects to coefficients and setting all to zero

$$\frac{\partial \langle R \rangle(\theta)}{\partial c_q} = 0 \qquad \forall q \in \{0, 1, \dots, M\}. \tag{29}$$

Since $\langle R \rangle(\theta)$ is the squared $\mathcal{L}_2$ norm of the differences between $\hat{f}(u_i)$ and $\hat{g}(u_i, \theta)$, the solution to this system is guaranteed to give a global minimum of $\langle R \rangle(\theta)$. In a matrix form, this linear
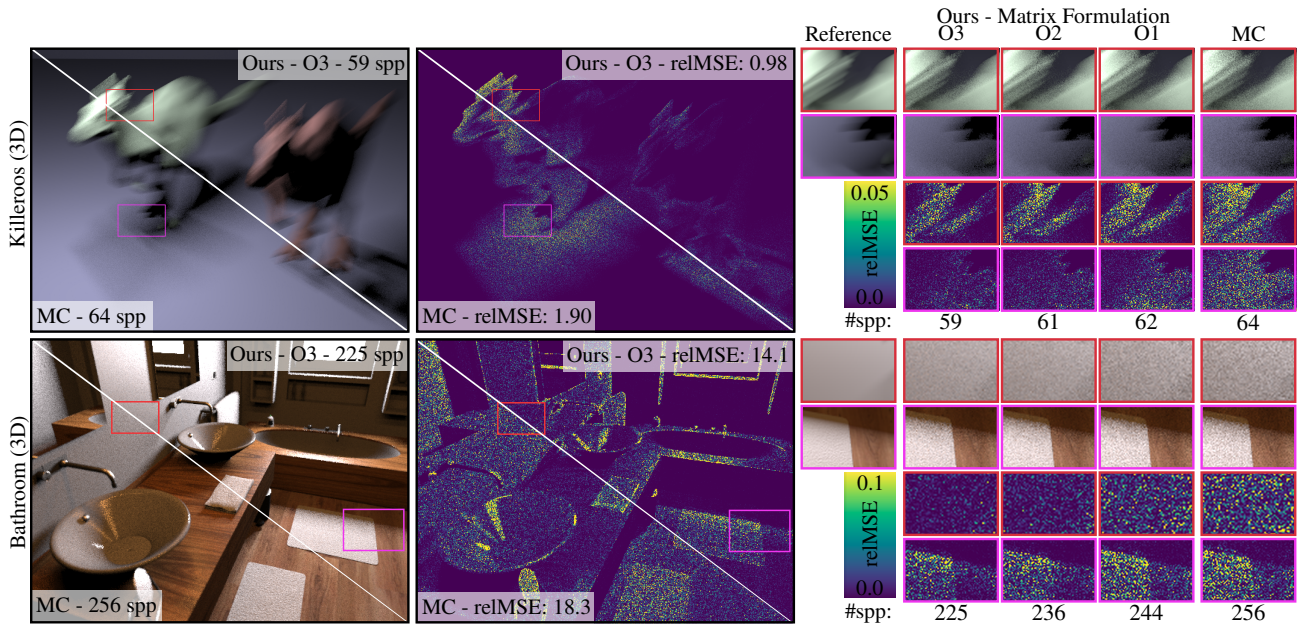
Fig. 3. Equal-time comparisons between regression-based and conventional Monte Carlo integration. We demonstrate the effectiveness of our approach with polynomials of order 1 to 3. The rendering involves 3D integrations (Killeroos and Bathroom). relMSE values are scaled by $10^3$.

system can be represented as

$$
\begin{bmatrix}
\Phi(0,0) & \cdots & \Phi(0,M) \\
\Phi(1,0) & \cdots & \Phi(1,M) \\
\vdots & & \\
\Phi(M,0) & \cdots & \Phi(M,M)
\end{bmatrix}
\begin{bmatrix}
c_0 \\
c_1 \\
\vdots \\
c_M
\end{bmatrix}
=
\begin{bmatrix}
\sum_{i=1}^{N} \hat{f}(u_i)\phi_0(u_i) \\
\sum_{i=1}^{N} \hat{f}(u_i)\phi_1(u_i) \\
\vdots \\
\sum_{i=1}^{N} \hat{f}(u_i)\phi_M(u_i)
\end{bmatrix}
\tag{30}
$$

where the function $\Phi(\cdot, \cdot)$ represents

$$
\Phi(p,q) = \sum_{i=1}^{N} \phi_p(u_i)\phi_q(u_i)
\tag{31}
$$

which is precomputed for given bases $\phi$. Updating the full matrix requires $M^2 + M$ floating point multiplication and summation if we ignore the cost of evaluating the monomials.

Since we need to evaluate the resulting $\hat{g}(u)$ at each of $u_i$ after regression, we chose to store all the $N$ samples first, solve the linear system, and then evaluate $\hat{g}(u_i)$ afterward. An iterative approach which does not require storing all the samples is possible under the matrix formulation [Nakatsukasa 2018], but we leave it as future work and use the simpler alternative to store all the samples.

Note also that the size of the matrix stays the same regardless of the number of samples $N$. We use complete orthogonal decomposition as a solver as it is fast for small matrices and does not require that the matrix is invertible (which may happen when $N$ is small). Algorithm 1 shows the details of this matrix-based approach.

## 5.2 Gradient descent estimator

Since our formulation does not assume a specific approach for regression, we can use an alternative approach based on gradient descent. Rather than setting partial derivatives to zero to define a linear system, we use them as a gradient to iteratively update

$\theta_k$ by starting from its initial guess $\theta_1$ and calculating

$$
\theta_{k+1} = \theta_k - \gamma \left( \cdots, \frac{\partial \langle R \rangle (\theta_k)}{\partial c_{q,k}}, \cdots \right)
\tag{32}
$$

where $\gamma$ is a parameter which we set to 0.01. Using *stochastic* gradient descent, we can modify the above equation by observing that $u_i$ is a random sample used for Monte Carlo estimator $\langle R \rangle (\theta)$ of $R(\theta)$ and use only *one* sample for $k = 1, \cdots, N$ as

$$
\theta_{k+1} = \theta_k - \gamma \left( \cdots, \frac{\partial r(u_k, \theta_k)}{\partial c_{q,k}} \cdots \right).
\tag{33}
$$

where we defined $r(u, \theta) = (\hat{f}(u) - \hat{g}(u, \theta))^2$ for simplicity. This process can be repeated multiple times over the same set of samples to potentially reach a better solution for regression.

This gradient descent approach is quite general and can be used for more complex integrable functions than polynomials such as neural networks [Müller et al. 2020]. In the case of polynomials, the computation of gradients is straightforward and only requires one evaluation of $\hat{g}(u)$ for each sample $u_i$ and the evaluations of the monomials. We thus found that the gradient descent approach is generally less computationally costly than the matrix-based approach. While more sophisticated algorithms may perform better, we have found that the above simple algorithm works well in our case. Algorithm 2 shows the details.

*Discussion.* The gradient descent approach is general and updating the coefficient is less costly compared to solving a matrix. The matrix approach can be used only when the residual is quadratic to the parameters, as in the case for polynomials. In more general cases where the residual is not a quadratic function of the parameters, one can still use the gradient descent approach. We advise to use the matrix for polynomials because it directly solves for the optimal solution.

At a low sample count, however, we found that the matrix approach can sometimes be numerically unstable as the matrix could be ill-conditioned. We have observed this issue in particular

---

**Algorithm 2:** Gradient descent estimator.

---

**Parameter:** lr: learning rate, T: number of iterations
$f \leftarrow \{\}; u \leftarrow \{\};$
**for** $i \leftarrow 1$ **to** $N$ **do**
   | $u_i \leftarrow \text{random}();$
   | $f_i \leftarrow \hat{f}(u_i);$
**end**
$\theta_{\min} \leftarrow 0^{M \times 1};$
**for** $t \leftarrow 1$ **to** $T$ **do**
   | $\theta_{\min} \leftarrow \text{gradient\_descent}(\theta_{\min}, \text{lr}, u_i, f_i);$
**end**
**return** $G(\theta_{\min}) + \frac{1}{N}(\sum_{i=1}^{N} f_i - \hat{g}(u_i, \theta_{\min}));$

---
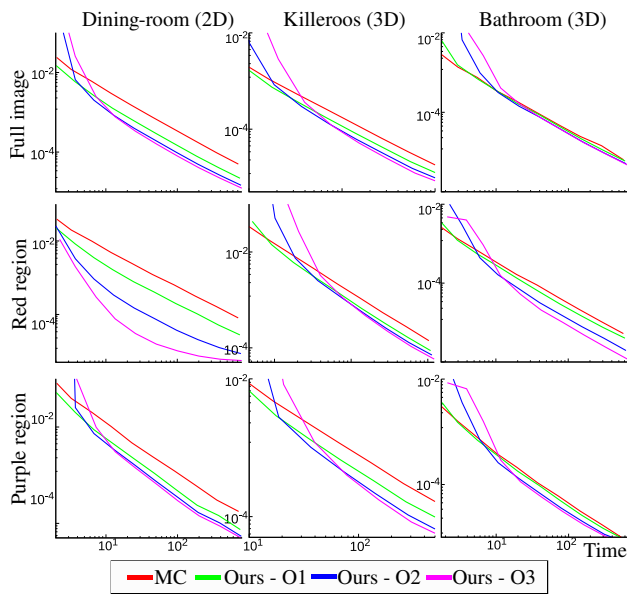


Fig. 4. Convergence plot for equal-time comparisons in Fig. 1 and Fig. 3. The metric is relMSE. Our method converges faster than Monte Carlo estimation in all but has similar performance in the BATHROOM scene.

for a higher-order polynomial in a high dimensional space, as the number of entries in the matrix will be large. In this case, the gradient descent approach might be better suited.

Our regression is by no means perfect, and using more efficient regression techniques and implementation could lead to further performance improvement.

## 6 EXPERIMENTS

Our implementation uses PBRT v3 [Pharr et al. 2016] as a rendering system and Eigen [Guennebaud et al. 2010] as a solver for linear systems. All the results were generated on AMD EPYC 7702 64-Core or Intel Core i9-8950HK Processor using eight threads for each. For comparison, we compare our method against conventional MC integration and Crespo et al. [2021] using the relative mean squared error (relMSE):

$$1/N \sum_{i=1}^{N} (I_i - R_i)^2 / (R_i^2 + 0.01),$$

where $I$ and $R$ represent the rendered and the reference image, respectively. The reference image $R$ for each scene is computed with 65,536 samples per pixel.

To handle a vector (i.e., RGB) integrand in rendering, we first estimate the RGB estimate $F_{rgb}$ by averaging RGB samples just as

$F_{rgb} \approx \langle F_{rgb} \rangle = 1/N \sum_{i=1}^{N} \hat{f}_{rgb}(u_i)$ where $\hat{f}_{rgb}(u_i)$ is the integrand that returns an RGB sample. We then perform regression on the luminance value of the samples, resulting in the luminance valued $G$ as the analytical integral. The reconstruction of the RGB value based on our estimator is written as $\langle F^*_{rgb} \rangle = (\langle F_{rgb} \rangle / y(\langle F_{rgb} \rangle)) \langle F \rangle^*$ where $y$ is the luminance function and $\langle F \rangle^*$ estimate the luminance value of $F_{rgb}$ based on our method. In other words, we correct the noisy luminance value of $F_{rgb}$ by our estimate.

### 6.1 Low-dimensional comparisons

We show in Fig. 1 and Fig. 3 an equal-time comparison between the conventional MC integration and our approach using polynomials of order 1 to 3 on three different scenes. The DINING ROOM scene (Fig. 1) is composed of a single triangle light source for direct illumination, making it a 2-dimensional integration problem. Our method shows significant variance reduction compared to MC integration as predicted. Even in the penumbra region where the reduction of error is small, our method is no worse than MC integration. The KILLEROOS scene (top row, Fig. 3) is composed of a single spherical light and motion blur, which is a 3-dimensional integration problem. Despite the fact that motion blur introduces more discontinuities in the integrand—which makes it difficult to approximate by polynomials of any order— our method can still reduce the variance in most of the regions. This result demonstrates that it is possible to achieve variance reduction with model functions that only roughly approximate the underlying integrand.

The BATHROOM scene (bottom row, Fig. 3) has multiple mesh light sources picked randomly using a random number, making it also a 3-dimensional integration problem. Moreover, each light in this scene is a mesh light with several triangles making discontinuities in the 2D area light parameterization. For this scene, the improvement in our method is limited but no worse than MC integration.

Fig. 4 shows the error convergence plots for the scenes described above. As predicted by our theory, our method consistently outperforms MC integration once the solution to regression is almost converged. At a fewer sample count than 10 spp, our method with higher order polynomials (order two and three) may perform worse than Monte Carlo integration due to the error in regression. This is because the matrix tends to be ill-conditioned due to noise in the sampling. Even though a linear model function (O1) is only slightly more complex than a constant model function (MC), our method with O1 significantly outperforms MC integration. We also observe that the maximum possible reduction in variance is limited by the complexity of the model function. There is thus a trade-off between the maximum reduction of variance and its stability at a few sample count, when one chooses a model function. The order or number of basis also seems to impact the ovearhead.

### 6.2 Combination with multiple importance sampling

In Fig. 5, we show that our method can be used in conjunction with MIS [Veach and Guibas 1995]. From different sampling strategies available during Multiple importance sampling (MIS), we simply perform the regression on weighted samples from each sampling strategy.

Fig. 5 shows the results using 64 samples per pixel of our method with two different importance sampling techniques: BRDF and light sampling. We compare the balance heuristic [Veach 1997] and the optimal weights [Kondapaneni et al. 2019].
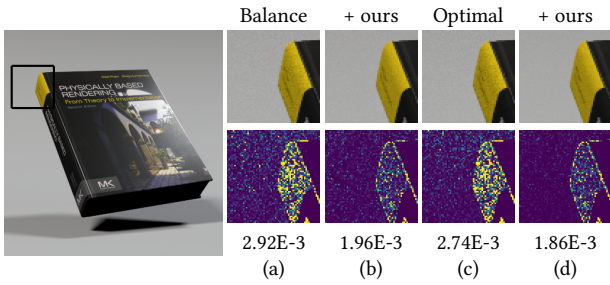
Fig. 5. Equal sample comparison of two MIS techniques (BRDF+light sampling) is performed using balance heuristics [Veach 1997] and optimal weights [Kondapaneni et al. 2019]. The combination with our method uses an order-2 polynomial basis. (a-d) Error maps and the relMSE values marked at the bottom shows improvement brought by our method.
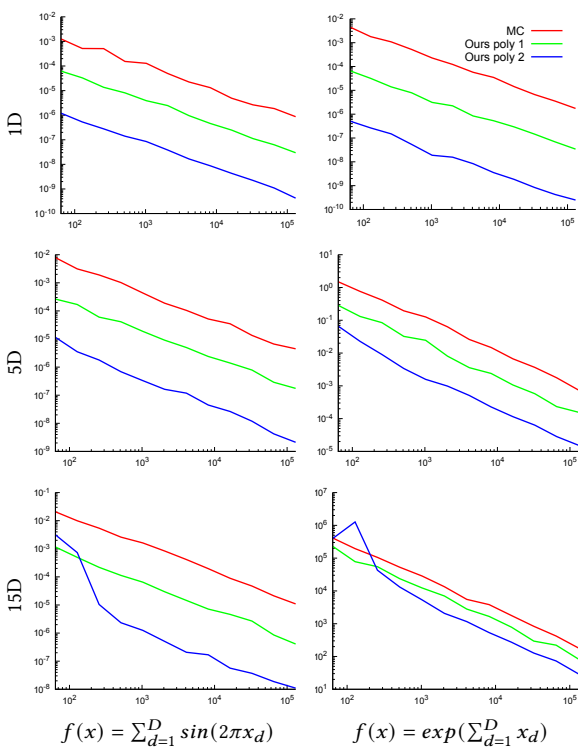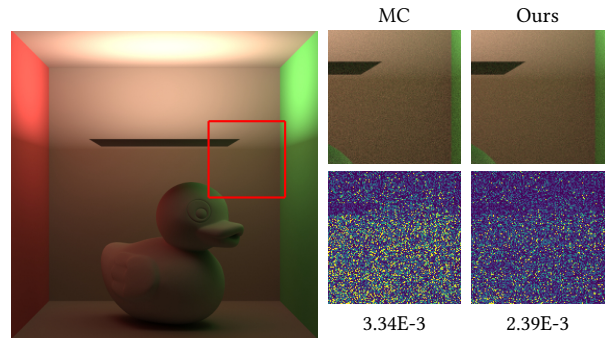


Fig. 7. Equal sample comparison between classical Monte Carlo and our method using an O2 polynomial basis. In this DUCK-CORNELL scene, only the upper part is directly illuminated by the light source. The lower part has indirect-illumination. The error map (relMSE) shows clear improvements using our method. relMSE values marked at the bottom.
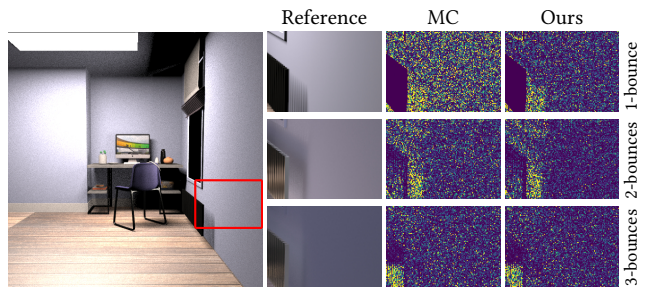


Fig. 8. We study the impact of our method on the function complexity by increasing the path length: 1, 2 and 3 bounces between the camera and the light source. The dimensionality of the respective path length is 3D, 5D and 7D. At equal sample count, the improvements of our method are directly affected by the complexity of the integrand.



Fig. 6. Convergence plots comparing our method using polynomial bases with a classical Monte Carlo estimator at different dimensionality. We observe that performance stays unchanged if the function complexity does not depend on the integration dimensionality (left column). Otherwise, our method experiences a slight decrease of improvement with the dimensionality but is still clearly advantageous compared to the Monte Carlo estimator (right column).

## 6.3 High-dimensional comparisons

Our method directly extends to higher dimensions.

We start the numerical analysis with simple analytic integrands. In Fig. 6, we show the convergence curves over 1D, 5D and 15D space for two integrands: (a) sum of sinusoids and (b) a multi-dimensional exponential function. In both cases, our method shows significant improvements in variance reduction. Both O1 and O2 polynomials gives significant improvements. However, at low sample count (up to about 100 samples) in 15D, O2 polynomial performs poorly. This is because with increasing polynomial order, in higher dimensions, more samples are needed to better approximate the control variate.

We apply our method to a path tracing rendering. Fig. 7 shows the results of our method for a DUCK-CORNELL scene rendered in path tracing with 3 bounces. This corresponds to a 15D problem in our algorithm: a 3D random number for each next-event estimation and a 2D random number for the next-direction sampling at each bounce. Improvements are visible in both directly- and indirectly-illumnied regions. However, the improvements in the indirectly-lit regions are less due to the strong variations of the function. These variations come from the light making several bounces, which makes the control variate function less accurate than in the case of direct illumination. One can of course consider having a more complex model function, but it will put an additional computational overhead on our method.

Our method improves the quality of the result for both weighting schemes compared to Monte Carlo counterpart. We also found that the combination of our method with optimal weight [Kondapaneni et al. 2019] keeps the strength of these two techniques, leading to the best variance reduction. The error reduction brought by the optimal weights allows a closer regression to the integrand leading to a greater error reduction. This result indicates that our method can take advantage of orthogonal variance reduction methods that simplify the shape of the integrand in primary sample space.
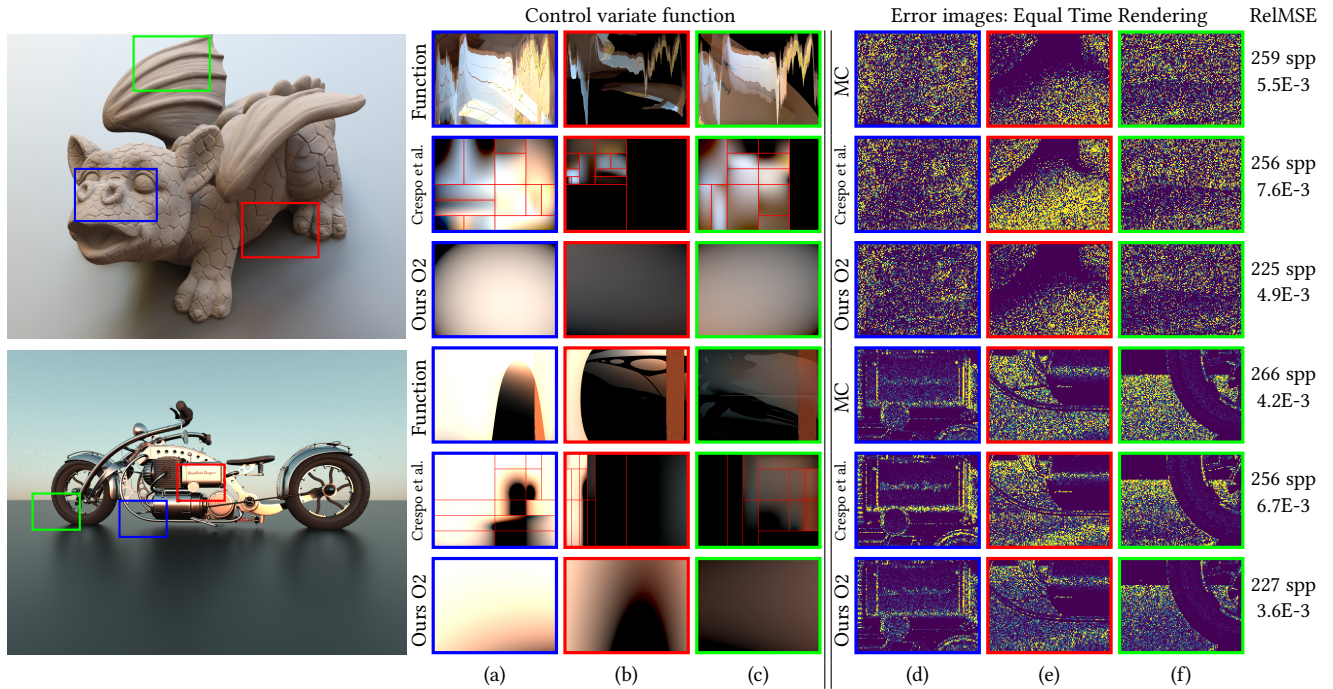
Fig. 9. Equal-time comparison between our method and Crespo et al. [2021] where each pixel is considered independently. Columns (a)-(c) show the control variate function for a pixel centered at each inset. Columns (d)-(f) show the error images for each inset. Following Crespo et al., we ensure 30% of the samples are used for building the control variate. For both the DRAGON and the CHOPPER-TITAN scene, our method show relatively less error ((d)-(f)). Crespo et al. is efficient in some cases. For DRAGON scene, column (e), Crespo et al. gives worse error than the classical MC. The relMSE value of our method is lower than Crespo et al.. For better visualization, please see the HTML viewer in the supplemental.
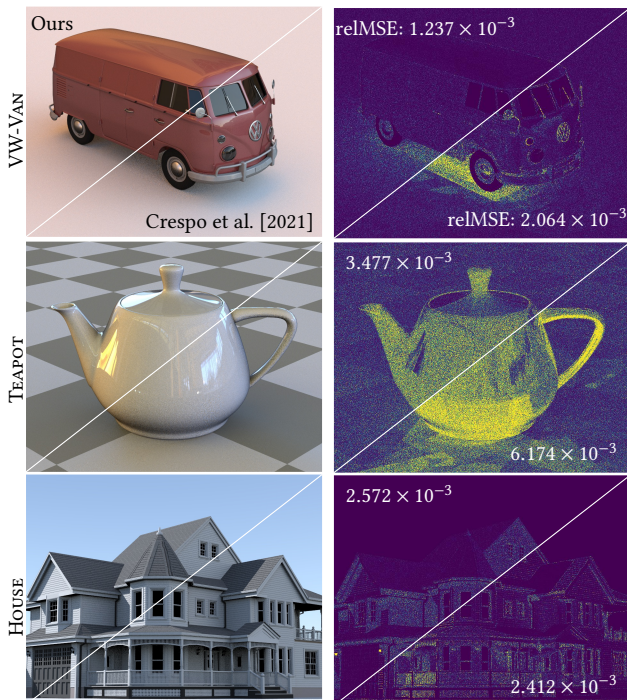


Fig. 10. Equal-sample comparison of our method using an $O2$ polynomial function basis with Crespo et al. [2021] method.

Fig. 8 shows the improvement due to our method on illumination coming from different number of bounces. Our method provides the largest improvement on direct illumination (the

top row), and the improvement generally decreases with more bounces (the middle and the bottom rows). As we discussed above, it is not directly due to the increased number of dimensions for higher number of bounces. We speculate that it is rather specific to the fact that the integrand in the primary sample space becomes more and more complex for higher number of bounces. Similar observations have been made in prior work that attempt to approximate the integrand in the primary sample space [Crespo et al. 2021; Guo et al. 2018; Zheng and Zwicker 2019], so we believe that it is not specific to our regression approach. We note that our method is still not performing worse than MC integration even for higher number of bounces.

## 6.4 Comparison to Crespo et al. [2021]

Crespo et al. [2021] proposed two adaptive control variate techniques using quadratures: one based on a quadrature per pixel and one with a global quadrature for the whole image. As we also treat each pixel independently, we have chosen to compare our method to the former method, which has been shown to be competitive. Specifically, Crespo et al. [2021] construct an adaptive piecewise-polynomial control variate (per pixel) in the primary sample space by recursively splitting the space based on an error heuristic built upon nested quadrature rules and regions' volume. We leave the possibility of extending our approach over a block of pixels as future work.

Fig. 9 provides an equal-time comparison of Crespo et al.'s approach with our $O2$ polynomial basis method for a 2D direct illumination integration in two scenes. The DRAGON scene consist of a simple object lit by a high-frequency environment map. The CHOPPER-TITAN scene contains glossy complex objects lit by a low-frequency environment map. For both scenes, we show the
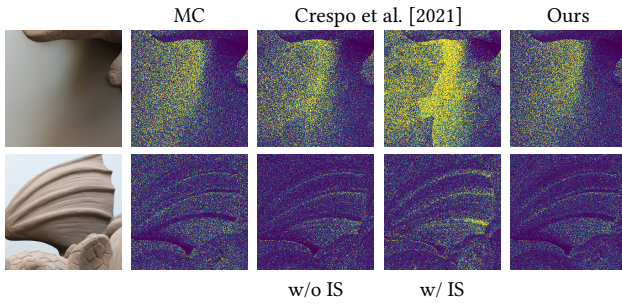
Fig. 11. Importance sampling (IS) performed in Crespo et al.'s approach can increase the variance wrt classical MC due to bad IS strategy.



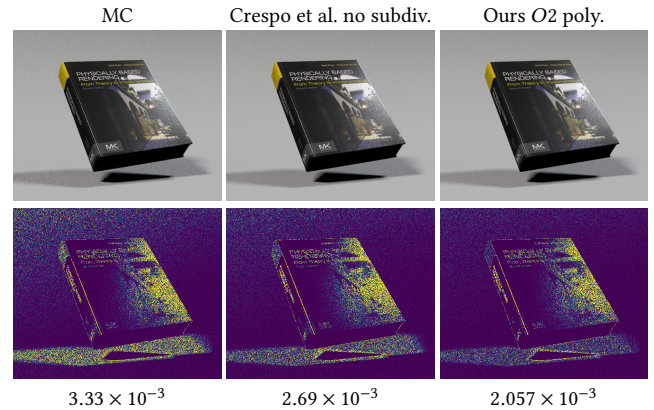| $3.33 \times 10^{-3}$ | $2.69 \times 10^{-3}$ | $2.057 \times 10^{-3}$ |

Fig. 12. Without recursive space subdivision, Crespo et al. appears to be similar to our approach, but it is not. Our approach still performs better. All methods are rendered with 64 spp. relMSE values are at the bottom.
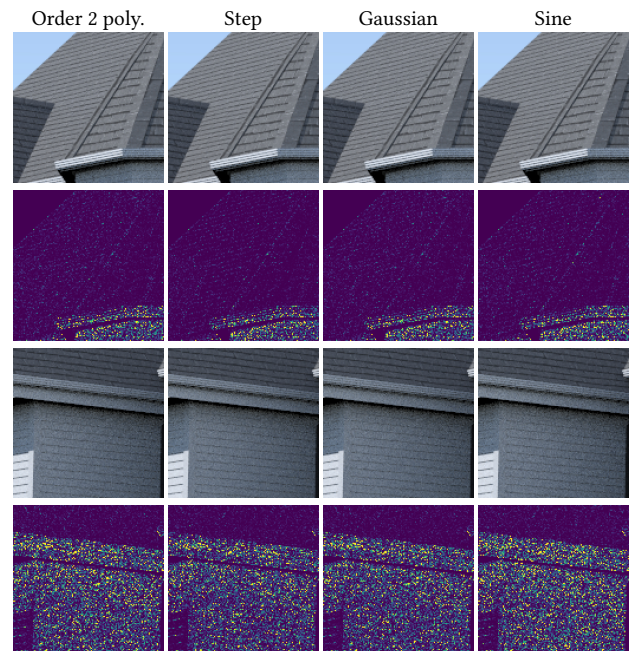


Fig. 13. We compare our method with different basis functions at equal sample count. Please refer to the supplemental material for more scenes and bases.

reference control variate function, the control variate functions constructed by Crespo et al. and our method (columns (a)-(c)). We also shows on columns (d)-(f) the error images of MC and these different technique.

For functions shown in column (a), Crespo et al.'s method can produce very accurate function approximations, resulting in error reduction compared to MC method (d). However, in columns (b, c), their adaptive control variate construction misses some critical function regions, resulting in a poor approximation. Moreover, stratifying the samples across regions with importance sampling increases this problem, resulting in higher error than MC (columns e,f). In comparison, our method uses a crude approximation of control variate via a polynomial function but consistently outperforms the classical MC and is more robust compared to Crespo et al.'s method.

Fig. 10 shows an equal-sample comparison between Crespo et al.'s method and ours. For the vw-van and the teapot scenes, our method significantly outperforms Crespo et al.'s in difficult regions such as the bottom of the van and the reflections on the bottom side of the teapot. However, for the house scene, Crespo et al.'s method performs better especially on the house's facades. Despite such, it is noteworthy that the error reduction by our method is more uniform across the scenes. Crespo et al.'s method, on the other hand, shows significant improvement upon classical MC in some regions, e.g., directly illuminated regions of the vw-van and the teapot, but performs worse than MC in some other regions.

Fig. 11 demonstrates the impact of importance sampling in Crespo et al.'s method. We found that their importance sampling strategy can increase variance, which can be explained by closely investigating their adaptive subdivision scheme. In the first row of Fig. 9, Crespo et al.'s control variate function misses important integrand features. They used this approximation for importance sampling as well which may increase variance.

Fig. 12 shows another experiment with no adaptive subdivision in the method of Crespo et al.. Without adaptive subdivision, their method falls back to a control variate estimator using a simple quadrature for a polynomial model function. This setup looks deceivingly equivalent to ours, but they are different. Crespo et al.'s method finds a polynomial via a quadrature rule with a fixed set of points within the domain for their control variates. Consequently, their control variate cannot take advantage of any more samples than being required for the quadrature rule without splitting. Our method, on the other hand, can take any number of samples to improve the least-squares fit without any splitting. The least-squares regression further makes sure to reduce the

variance bound as our formulation shows. Both the error map and the relMSE values reported in Fig. 12 confirm our observation.

In summary, due to the lack of robustness during the quadrature construction caused by their fixed sample locations and error heuristics, Crespo et al.'s method does not consistently outperform the classical MC overall. In the case of inaccurate subdivisions of the primary sample space, their method might result in inefficient importance sampling, leading to higher errors than MC. By contrast, our method is provably never worse than the MC estimator despite that ours could result in rather coarse approximations of the function, limiting the error reduction in some regions. As future work, Crespo et al.'s adaptive model for $\hat{g}$ can potentially be useful in our formulation, but developing a practical least-squares regression algorithm for piecewise polynomials remains challenging.

## 6.5 Analysis

*Choice of basis functions.* We perform an experiment using different basis functions in the regression. The results are shown in Fig. 13. We compare the performance of order 2 polynomial, step functions, Gaussian mixture and Sine functions. All these basis have been chosen to contain the same number of parameters for a fair comparison. The quality of the results obtained slightly varies according to the bases used and the scenes chosen. Other results for additional bases and scenes is available in the supplemental materials.

In general, while it is challenging to define a function basis robust for all situations, our experiments show that polynomial basis works relatively well in many cases. As long as a constant function is included in least-squares regression, our method would be no worse than MC integration, no matter, what basis functions are chosen.

*Impact of discontinuities on polynomial basis.* Polynomial bases have difficulty in representing functions with many discontinuities. In Fig. 14, we study the evolution of the integration error for a fixed sample budget of 1024 samples when integrating the contribution of multiple light sources over a diffuse plane (Fig. 14.a). In this case, the number of discontinuities inside the 3D integrand increases with the number of lights due to the random decision on the light selection (Fig. 14.b). We study the impact of using O1, O2 and O3 polynomials with our method against the conventional Monte Carlo estimator. The plot in Fig. 14(c) demonstrates that at a low number of lights, our method significantly improves over the conventional Monte Carlo estimation. However, this improvement decreases with the increase in the number of discontinuities. With a large number of lights, the error produced by our method becomes equivalent to that of the Monte Carlo estimator, confirming that the polynomial basis behaves equivalently to a constant model function, i. e., the conventional approach.

*Computation time.* We empirically found that the computation overhead of our method becomes significant for higher-order polynomial regression using the direct matrix approach. For example, our method with low-order polynomials has negligible overhead over MC integration (Fig. 3) but a polynomial of order 5 can cause 68% overhead in terms of running time. Similarly, higher-dimensional integration also requires significant overhead over MC integration. However, we focus in demonstrating

the versatility of our approach. Improving the regression computation is out of the scope for this work. Following above observation, we also investigate another approach for regression, such as gradient descent, which can be a better option for higher-order polynomials. The gradient descent approach scales well thanks to its lower computational complexity ($O(M)$ vs $O(M^2)$)) where $M$ is the number of monomials.

*Matrix vs. Gradient descent solver.* Fig. 15 shows the comparison between the gradient descent and the direct-matrix solver. The results are demonstrated with 32spp and 256spp on a BATH-ROOM scene taken from PBRT. We use three different polynomials of order 2, 5 and 7. We compare the direct matrix approach based on complete orthogonal decomposition and stochastic gradient descent estimator (SGD) to solve the polynomial regression. At a low sample count, SGD performs better than the direct matrix approach as the matrix tends to be ill-conditioned due to noise in sampling. At a high sample count, the direct matrix approach outperforms SGD due to its ability to find a global minimum. For a high-order polynomial (order 7), the direct matrix approach becomes computationally expensive whereas SGD runs at at least 2× faster speed while giving similar relMSE.

## 7 DISCUSSION

*Incremental estimators.* In our implementation, we assume that the $N$ sample estimator $\langle F \rangle^*$ is constructed by first finding the solution to regression $\hat{g}$ using all the $N$ samples, and then evaluating $\langle F \rangle^*$ with the same $N$ samples. Therefore, when $N$ is changed to $N + 1$, we need to solve regression again with $N + 1$ samples first and then re-evaluate $\langle F \rangle^*$ with the new $\hat{g}$ using $N + 1$ samples. In some applications, such as progressive rendering, being able to incrementally update both $\langle F \rangle^*$ and $\hat{g}$ as we add a new sample is desirable. Thanks to the use of control variates, our formulation allows such incremental update as follows.

Let us denote $\langle F \rangle^*_N$ and $\hat{g}_N$ as the estimated value and the solution to regression with $N$ samples. Using an online regression algorithm, one can obtain $\hat{g}_{N+1}$ based on $\hat{g}_N$ by adding a new sample $u_{N+1}$. We define the incremental estimate $\langle F \rangle^*_{N+1}$ as

$$\langle F \rangle^*_{N+1} = \tfrac{1}{N+1} \left( N \langle F \rangle^*_N + \left( G_N + \hat{f}(u_{N+1}) - \hat{g}_N(u_{N+1}) \right) \right) \quad (34)$$

where $G_{N+1}$ is the integral of $\hat{g}_{N+1}$. This estimator remains unbiased since

$$
\begin{aligned}
& E\left[ \langle F \rangle^*_{N+1} \right] \\
&= E\left[ \frac{1}{N+1} \left( N \langle F \rangle^*_N + \left( G_N + \hat{f}(u_{N+1}) - \hat{g}_N(u_{N+1}) \right) \right) \right] \\
&= \frac{1}{N+1} \left( NF + E\left[ \left( G_N + \hat{f}(u_{N+1}) - \hat{g}_N(u_{N+1}) \right) \right] \right) \\
&= \frac{1}{N+1} \left( NF + F \right) = F. \quad (35)
\end{aligned}
$$

The expected squared error of incremental estimators is likely worse than that of non-incremental counterparts; the $N$ sample incremental estimator now combines the results of regression for all the $N - 1, N - 2, ..., 1$ samples which might be numerically unstable at the beginning. The incremental estimators are, however, often computationally less expensive for the same number of samples than the non-incremental counterparts. We leave the exact analysis of these differences as future work, but we show a preliminary result in Fig. 16 where the incremental estimate is performed with the gradient descent estimator. The numerical error of the incremental estimate is slightly higher than the non-incremental version, but it still outperforms MC integration.
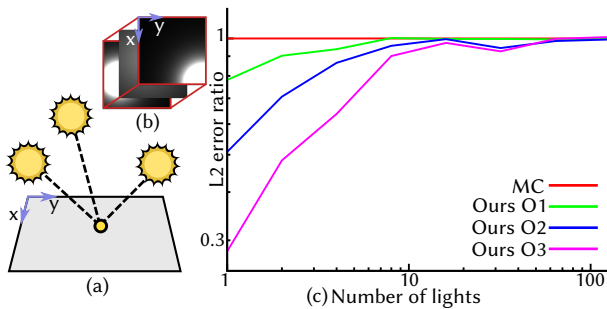


Fig. 14. Comparison of integration error ratio with Monte Carlo as a function of the number of lights in the scene. Our method is used here with polynomial basis of order 1, 2 and 3. The drawing on the left schematizes the integration problem.

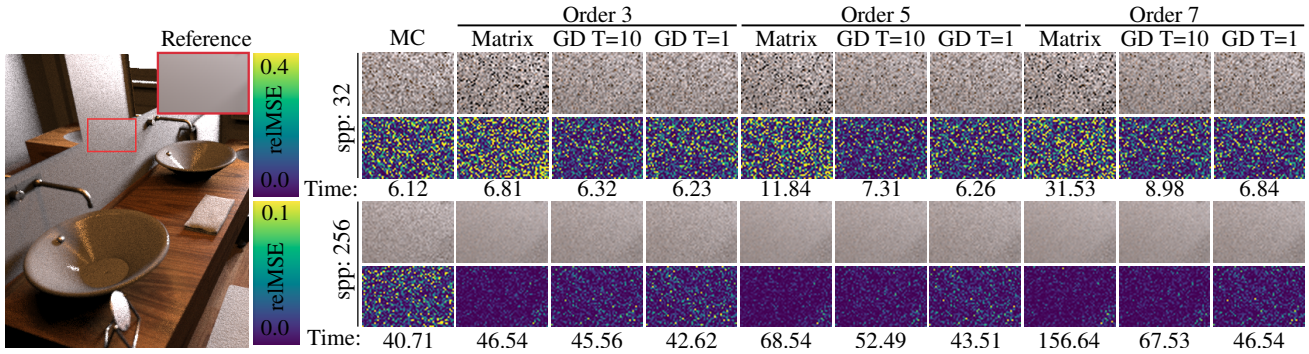| | | | Order 3 | | | Order 5 | | | Order 7 | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | MC | Matrix | GD T=10 | GD T=1 | Matrix | GD T=10 | GD T=1 | Matrix | GD T=10 | GD T=1 |
| spp: 32 | Time: | 6.12 | 6.81 | 6.32 | 6.23 | 11.84 | 7.31 | 6.26 | 31.53 | 8.98 | 6.84 |
| spp: 256 | Time: | 40.71 | 46.54 | 45.56 | 42.62 | 68.54 | 52.49 | 43.51 | 156.64 | 67.53 | 46.54 |

Fig. 15. Regression using a direct matrix approach and stochastic gradient descent (SGD). At a low-sample count, SGD performs better. At a high-sample count, the direct matrix approach outperforms SGD. At high-order polynomial (order 7), the direct matrix approach has a very high overhead, making SGD an attractive alternative. Time is reported in seconds.
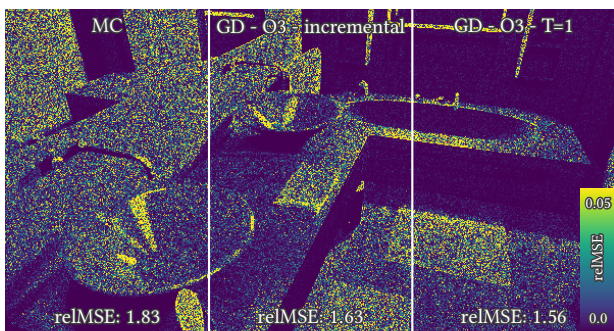


Fig. 16. Incremental regression-based Monte Carlo integration with gradient descent. We render the BATHROOM scene (256 spp) and compare with MC and non-incremental gradient descent estimator. While the incremental gradient descent has slightly higher error than its non-incremental counterpart, both achieves less error than MC integration. relMSE values are scaled by $10^{-2}$.

*Differences from MCLS.* Our regression-based Monte Carlo integration bears some similarities to Monte Carlo Least-Squares MCLS) [Nakatsukasa 2018].

Nakatsukasa reformulated Monte Carlo integration as least-squares regression followed by the analytical integration of the resulting function. MCLS also performs no worse than Monte Carlo integration as long as a constant function is included in regression. There are two major differences between MCLS and our approach. Firstly, MCLS considers a model function $g$ which integrates exactly to $F$ (i.e., $F = G$).

Such a family of model functions is far more restrictive than our model function which is allowed to integrate to *any* value. Secondly, Nakatsukasa claims that MCLS is derived under an entirely different framework than control variates, thus its connection to control variates is unclear. In contrast, we show how control variates is closely related to the combination of Monte Carlo integration and least-squares regression.

When we use a model function $\hat{g}$ such that $G = F$ (i.e., keeps the integral value), then our approach in fact reduces to MCLS. MCLS is thus can be seen as a special case of our formulation.

*Russian roulette.* In our current implementation, we manually set all path lengths to be the same. This is because, for regression it is necessary to fix the dimensionality of the search space. Russian roulette, however may result in paths of different length/dimensionality.

One simple way to combine Russian roulette with our approach would be to perform regression up to a certain prescribed dimension and only invoke Russian roulette for paths that have dimensionality higher than the prescribed one. The high-dimensional sample components can then be projected back into the lower-dimensional space.

## 8 LIMITATIONS AND FUTURE WORK

While our method often just works as a drop-in improvement over Monte Carlo integration in many cases, there are several limitations and future work.

Regression in our method incurs additional computation over MC integration. Regression may or may not be worth the additional cost depending on the amount of error reduction in the end. In general, regression becomes computationally more costly for a more complex model function, but a complex function typically leads to better error reduction. There is a trade-off among the reduction of error, the complexity of a model function, and the computational overhead of the regression process. Finding an optimal balance among those can potentially be achieved in an adaptive manner. Our formulation does not specify how regression should be solved in practice, so there is a large room of design choices in its implementation.

Since our formulation is based on the primary sample space, applying our method to a higher number of bounces in light transport simulation is challenging. The dimensionality of the primary sample space is directly proportional to the number of bounces, and regression in a high dimensional is not numerically stable. It is being recognized that light transport simulation has a much lower dimensional structure in the original path space, as was utilized by path guiding methods [Müller et al. 2017; Vorba et al. 2014]. It would be interesting to study how our regression-based formulation would benefit from such a low dimensional structure of light transport.

Depending on how the regression is implemented, our method can have a very small bias. When the same set of samples are used for both regression of the control variate and estimation of the difference, our estimator can be biased. Owen and Zhou [2000] and Hickernell et al. [2005] have analyzed the amount of bias introduced when estimating the $\alpha$ parameter and evaluating the difference estimator using the same set of samples in control variates. In brief, the bias is negligible compared to the variance of the estimator and reduces at the order of $O(N^{-1})$ for $N$ samples as opposed to $O(N^{-1/2})$ of the error of Monte Carlo integration. This makes our estimator *consistent* and the bias reduces faster

than variance. One can also always remove this bias by separating a set of samples into two halves, performing regression only on the other half, and taking the average of the two difference estimators, in a manner akin to cross-validation. We, however, have not used this approach in the results in this paper since we have observed that bias is negligible. Our analysis is not influenced by this bias since this bias happens only when the same set of samples are used in its implementation of regression, while our analysis does not assume any specific implementation of regression.

## 9 ACKNOWLEDGEMENTS

## REFERENCES

Laurent Belcour, Guofu Xie, Christophe Hery, Mark Meyer, Wojciech Jarosz, and Derek Nowrouzezahrai. 2018. Integrating clipped spherical harmonics expansions. *ACM Transactions on Graphics (TOG)* 37, 2 (2018), 1–12.

Benedikt Bitterli, Fabrice Rousselle, Bochang Moon, José A. Iglesias-Guitián, David Adler, Kenny Mitchell, Wojciech Jarosz, and Jan Novák. 2016. Nonlinearly Weighted First-Order Regression for Denoising Monte Carlo Renderings. 35, 4 (June 2016), 107–117. https://doi.org/10/f842kc

Petrik Clarberg and Tomas Akenine-Möller. 2008. Exploiting Visibility Correlation in Direct Illumination. 27, 4 (2008), 1125–1136. https://doi.org/10/d2gd6k

Miguel Crespo, Adrian Jarabo, and Adolfo Muñoz. 2021. Primary-Space Adaptive Control Variates Using Piecewise-Polynomial Approximations. *ACM Trans. Graph.* 40, 3, Article 25 (jul 2021), 15 pages. https://doi.org/10.1145/3450627

Philip J Davis. 2013. *Methods of numerical integration.* Academic, New York ; London.

Shaohua Fan, Stephen Chenney, Bo Hu, Kam-Wah Tsui, and Yu-Chi Lai. 2006. Optimizing Control Variate Estimators for Rendering. (2006). https://doi.org/10/fb9fmz

Peter W. Glynn and Roberto Szechtman. 2002. Some New Perspectives on the Method of Control Variates. In *Monte Carlo and Quasi-Monte Carlo Methods 2000*, Kai-Tai Fang, Harald Niederreiter, and Fred J. Hickernell (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 27–49.

Gaël Guennebaud, Benoît Jacob, et al. 2010. Eigen v3. http://eigen.tuxfamily.org.

Jerry Guo, Pablo Bauszat, Jacco Bikker, and Elmar Eisemann. 2018. Primary sample space path guiding. In *Eurographics Symposium on Rendering*, Vol. 2018. The Eurographics Association, 73–82.

Stefan Heinrich. 2001. Multilevel Monte Carlo Methods. In *Proceedings of the Third International Conference on Large-Scale Scientific Computing-Revised Papers (LSSC '01)*. Springer-Verlag, Berlin, Heidelberg, 58–67.

Fred J. Hickernell, Christiane Lemieux, and Art B. Owen. 2005. Control Variates for Quasi-Monte Carlo. *Statist. Sci.* 20, 1 (2005), 1 – 31. https://doi.org/10.1214/088342304000000468

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

Csaba Kelemen, László Szirmay-Kalos, György Antal, and Ferenc Csonka. 2002. A Simple and Robust Mutation Strategy for the Metropolis Light Transport Algorithm. 21, 3 (Sept. 2002), 531–540. https://doi.org/10/bfrsqn

Alexander Keller. 2001. Hierarchical Monte Carlo Image Synthesis. *Math. Comput. Simul.* 55, 1–3 (Feb. 2001), 79–92. https://doi.org/10.1016/S0378-4754(00)00248-2

Ivo Kondapaneni, Petr Vévoda, Pascal Grittmann, Tomas Skrivan, Philipp Slusallek, and Jaroslav Krivanek. 2019. Optimal Multiple Importance Sampling. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2019)* 38, 4 (July 2019), 37:1–37:14. https://doi.org/10.1145/3306346.3323009

Peter Kutz, Ralf Habel, Yining Karl Li, and Jan Novák. 2017. Spectral and decomposition tracking for rendering heterogeneous volumes. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–16.

Eric P. Lafortune and Yves D. Willems. 1994. The Ambient Term as a Variance Reducing Technique for Monte Carlo Ray Tracing. 163–171.

W. W. Loh. 1995. *On the Method of Control Variates.* Ph.D. Thesis. Stanford University.

Bochang Moon, Nathan Carr, and Sung-Eui Yoon. 2014. Adaptive Rendering Based on Weighted Local Regression. 33, 5 (Sept. 2014), 170:1–170:14. https://doi.org/10/f6km7m

Bochang Moon, Jose A. Iglesias-Guitian, Sung-Eui Yoon, and Kenny Mitchell. 2015. Adaptive Rendering with Linear Predictions. 34, 4 (July 2015), 121:1–121:11. https://doi.org/10/f7m2hp

Thomas Müller, Markus Gross, and Jan Novák. 2017. Practical Path Guiding for Efficient Light-Transport Simulation. 36, 4 (June 2017), 91–100. https://doi.org/10/gbnvrs

Thomas Müller, Fabrice Rousselle, Alexander Keller, and Jan Novák. 2020. Neural Control Variates. *ACM Trans. Graph.* 39, 6, Article 243 (Nov. 2020), 19 pages. https://doi.org/10.1145/3414685.3417804

Yuji Nakatsukasa. 2018. Approximate and integrate: Variance reduction in Monte Carlo integration via function approximation. arXiv:math.NA/1806.05492

Harald Niederreiter. 1992. *Random Number Generation and Quasi-Monte Carlo Methods.* SIAM.

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.* To be published.

Anthony Pajot, Loic Barthe, and Mathias Paulin. 2014. Globally Adaptive Control Variate for Robust Numerical Integration. *SIAM Journal on Scientific Computing* 36, 4 (2014), A1708–A1730. https://doi.org/10.1137/130937846

Matt Pharr, Wenzel Jakob, and Greg Humphreys. 2016. *Physically Based Rendering: From Theory to Implementation* (3 ed.). San Francisco, CA, USA.

Fabrice Rousselle, Wojciech Jarosz, and Jan Novák. 2016. Image-Space Control Variates for Rendering. 35, 6 (Nov. 2016), 169:1–169:12. https://doi.org/10/f9cphw

Reuven Y. Rubinstein and Ruth Marcus. 1985. Efficiency of Multivariate Control Variates in Monte Carlo Simulation. (1985). https://doi.org/10.1287/opre.33.3.661

Kartic Subr. 2021. Q-NET: A Network for Low-dimensional Integrals of Neural Proxies. In *Computer Graphics Forum*, Vol. 40. Wiley Online Library, 61–71.

Lloyd N. Trefethen. 2012. *Approximation Theory and Approximation Practice (Other Titles in Applied Mathematics).* Society for Industrial and Applied Mathematics, USA.

Eric Veach. 1997. *Robust Monte Carlo Methods for Light Transport Simulation.* Ph.D. Thesis. Stanford University, United States – California.

Eric Veach and Leonidas J. Guibas. 1995. Optimally Combining Sampling Techniques for Monte Carlo Rendering, Vol. 29. 419–428. https://doi.org/10/d7b6n4

Petr Vévoda, Ivo Kondapaneni, and Jaroslav Křivánek. 2018. Bayesian online regression for adaptive direct illumination sampling. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–12.

Jiří Vorba, Ondřej Karlík, Martin Šik, Tobias Ritschel, and Jaroslav Křivánek. 2014. On-Line Learning of Parametric Mixture Models for Light Transport Simulation. 33, 4 (Aug. 2014), 101:1–101:11. https://doi.org/10/f6c2cp

Quan Zheng and Matthias Zwicker. 2019. Learning to importance sample in primary sample space. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 169–179.

Matthias Zwicker, Wojciech Jarosz, Jaakko Lehtinen, Bochang Moon, Ravi Ramamoorthi, Fabrice Rousselle, Pradeep Sen, Cyril Soler, and Sung-Eui Yoon. 2015. Recent Advances in Adaptive Sampling and Reconstruction for Monte Carlo Rendering. 34, 2 (May 2015), 667–681. https://doi.org/10/f7k6kj