Primary-Space Adaptive Control Variates using Piecewise-Polynomial Approximations

MIGUEL CRESPO, Universidad de Zaragoza - I3A
ADRIAN JARABO, Universidad de Zaragoza - I3A, and Centro Universitario de la Defensa Zaragoza
ADOLFO MUÑOZ, Universidad de Zaragoza - I3A

Fig. 1. Bistro: Unbiased rendering of a complex scene with global illumination (22 indirect bounces, resulting in a 48-dimensional integration domain). Traditional Monte Carlo-based rendering results in high variance even with importance sampling techniques. In contrast, our technique combines multiple importance sampling with an adaptive piecewise-polynomial control variate (4D in this example): Our control variate closely approximates the low-frequency regions of the signal, while leaving the high-frequency details on the residual, which is estimated using Monte Carlo integration. This results in lower variance with faster convergence. Except for the reference, the images were generated using 512 samples per pixel.

We present an unbiased numerical integration algorithm that handles both low-frequency regions and high-frequency details of multidimensional integrals. It combines quadrature and Monte Carlo integration by using a quadrature-based approximation as a control variate of the signal. We adaptively build the control variate constructed as a piecewise polynomial, which can be analytically integrated, and accurately reconstructs the low-frequency regions of the integrand. We then recover the high-frequency details missed by the control variate by using Monte Carlo integration of the residual. Our work leverages importance sampling techniques by working in primary space, allowing the combination of multiple mappings; this enables multiple importance sampling in quadrature-based integration. Our algorithm is generic, and can be applied to any complex multidimensional integral. We demonstrate its effectiveness with four applications with low dimensionality: transmittance estimation in heterogeneous participating media, low-order scattering in homogeneous media, direct illumination computation, and rendering of distribution effects. Finally, we show how our technique is extensible to integrands of higher dimensionality by computing the control variate on Monte Carlo estimates of the high-dimensional signal, and accounting for such additional dimensionality on the residual as well. In all cases, we show accurate results and faster convergence compared to previous approaches.

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

Additional Key Words and Phrases: Numerical Integration, Adaptive Quadrature, Rendering, Control Variates, Piecewise Polynomial

Authors' addresses: Miguel Crespo, mcrescas@gmail.com, Universidad de Zaragoza - I3A; Adrian Jarabo, ajarabo@unizar.es, Universidad de Zaragoza - I3A, and Centro Universitario de la Defensa Zaragoza; Adolfo Muñoz, adolfo@unizar.es, Universidad de Zaragoza - I3A.

1 INTRODUCTION

Numerical integration forms the basis of rendering algorithms, as light arriving to a sensor (pixel) is formulated as an integral. Given the specific nature of this integrand, Monte Carlo (MC) [Cook et al. 1984] is the most commonly applied numerical integration method. However, while general and robust, MC might converge slowly to the desired solution, introducing significant variance that leads to high-frequency noise even in smooth regions.

Several methods have been proposed to successfully reduce such variance, including (multiple) importance sampling [Veach 1997],

low-discrepancy sequences [Owen 2013], or MC variants based on Markov chains [Šik and Krivanek 2018]. However, variance is still a visible artifact in low-frequency areas, where stochastic methods suffer the most. In contrast, deterministic integration methods and in particular quadrature integration [Burden and Faires 2005] excel at such smooth integrals, providing significantly faster convergence rates for relatively smooth low-dimensional integrands. Unfortunately, these methods introduce bias in the results, and perform poorly in discontinuities and high-frequency details.

In this work we present a new unbiased numerical integration technique for low-dimensional integrals, capable of accurately handling both low-frequency and high-frequency areas of the signal. Our technique combines quadrature- and Monte Carlo-based methods, which allows us to leverage the strengths of both techniques. We first adaptively build a low-dimensional multivariate polynomial approximation of the signal using nested (adaptive) quadrature rules [Burden and Faires 2005]. Then, we use this approximation as a control variate, and compute the residual using Monte Carlo integration. Intuitively, the control variate accurately approximates the low-frequency low-dimensional content, while Monte Carlo integration recovers the residual high-frequency details.

Our technique performs the integration in primary space, which allows us to take advantage of any importance sampling technique for error reduction in both the polynomial approximation and the residual estimation. Moreover, we demonstrate that several sampling (i.e., warping) techniques can be combined in quadrature via multiple importance sampling (MIS) [Veach and Guibas 1995], which generalizes the potential of MIS for error reduction to quadrature-based integration. In addition, our control variate is computed adaptively by using an accurate error estimation, allowing for importance sampling of the residual.

Our integration technique is generic, not necessarily tied to rendering, agnostic to the integrand, and can be combined with any importance sampling technique. We demonstrate its performance in four rendering applications with different dimensionality, with results showing reduced variance and faster convergence in multidimensional integrals with low-dimensionality, and better results for the same number of samples than competing methods. Finally, we demonstrate that our technique is competitive in higher-dimensional light transport integrals by building low-dimensional quadrature-based control variates using Monte Carlo estimates of the function.

In summary, our work presents the following **contributions**:

- A new unbiased integration technique for low-dimensional integrals that combines the strengths of MC and quadrature methods. Our technique is adaptive, leverages any importance sampling strategy for variance reduction, and amortizes samples between different pixels (or frames).
- A generalization of multiple importance sampling to quadrature-based integration, which we leverage in our integration technique.
- Several practical rendering applications of our technique, including transmittance estimation in heterogeneous media, low-order scattering in homogeneous media, direct illumination computation, and rendering of distribution effects.

Limitations: Our technique has some limitations: First and foremost, given the curse of dimensionality in quadrature-based methods, the control variate is only generated in low-dimensional subdomains of the integrand. However, as we show in our applications, there is a large number of subproblems in rendering that can benefit from our technique. Additionally, we show that additional dimensions (e.g., high-order light bounces) can be included in our framework, taking advantage of the variance reduction in lower dimensions while enabling integrals of higher dimensionality. Our technique also introduces an overhead with respect to plain Monte Carlo, which is nevertheless amortized by the variance reduction achieved with our technique, and becomes negligible compared to costly integrand evaluations (such as rendering complex scenes). A third limitation is that, given the nature of our control variate, our technique is offline, and it does not refine the control variate when additional samples are introduced. Finally, similarly to other adaptive methods, our technique can suffer from areas with uneven convergence, which might lead to blocky artifacts at low sampling rates.

2 RELATED WORK

Numerical integration in rendering. Monte Carlo integration is the standard for simulating light transport [Cook et al. 1984; Veach 1997]. To reduce variance, several importance sampling strategies have been developed, from strategies targeting low-dimensional subproblems (e.g., area light sampling [Guillén et al. 2017; Ureña et al. 2013] or low-order volume scattering [Kulla and Fajardo 2012; Novák et al. 2012]) to high-dimensional path-guiding methods [Müller et al. 2017; Müller et al. 2019; Vorba et al. 2014; Zheng and Zwicker 2019]. Our work is complementary to those, and can leverage any importance sampling strategy (even multiple) by working in primary sample space. Other works aim to reduce variance by carefully positioning samples adaptively to the signal and using advanced techniques for reconstruction from those samples [Zwicker et al. 2015]. Several approaches exist either by partitioning the sample space [Hachisuka et al. 2008; Kajiya 1986], on-the-fly frequency analysis of the signal [Belcour et al. 2013; Durand et al. 2005], gradient information [Jarosz et al. 2008; Marco et al. 2018; Ramamoorthi et al. 2007; Ward et al. 1988], or machine learning [Gharbi et al. 2019]. Our technique also positions samples adaptively for constructing the control variate based on multivariate nested quadrature rules. Gradient-based techniques [Hua et al. 2019; Kettunen et al. 2015] reconstruct an unbiased final image by computing its gradients via Monte Carlo estimation, followed by a Poisson reconstruction. On the other hand, our work focuses on unbiased integration; potentially it could work on the gradient domain to leverage the good properties of gradient-based methods. Finally, denoising techniques trade off variance for bias, and remove noise from the final image using sophisticated filters with adaptive kernel bandwidths [Rousselle et al. 2012], local regression to low-order functions [Bitterli et al. 2016], or machine learning [Bako et al. 2017]. Our technique works

in sample space and focuses on unbiased integration of light transport sub-problems. Potentially, it could be followed by a denoising pass to remove the remaining variance.

Quadrature rules. There has been a lot of research involving quadrature rules [Stroud and Secrest 1966; Ziegel 1987] and in developing adaptive schemes to increase their accuracy [Berntsen et al. 1991; Genz and Malik 1980]. In computer graphics, quadrature integration is somewhat less explored. A notable widespread exception is the integration from distant light through spherical harmonics [Ramamoorthi and Hanrahan 2001, 2002]. Brouillat et al. [2009] and Marques et al. [2013] proposed to use Bayesian quadrature to integrate the incident illumination. In the context of rendering participating media, rectangle quadrature rules have been used for ray marching [Perlin and Hoffert 1989] or volumetric photon mapping [Jensen and Christensen 1998]. Later, Muñoz proposed using higher-order quadrature rules [Muñoz 2014], while Johnson et al. [2011] used Gaussian quadrature to accelerate the photon beams algorithm. All these works are case-specific for low-dimensional integrals, and introduce bias in the solution. Our work proposes an *unbiased* and *generic* (not tied to any specific problem) numerical integration method by devising quadrature integration as a control variate. Moreover, we demonstrate how multiple importance sampling can be applied in the context of quadrature integration.

Control variates. Control variates have remained relatively unexplored in rendering compared to other variance reduction techniques like importance sampling. Lafortune and Willems proposed using an ambient term [Lafortune and Willems 1994], and a directional piecewise approximation of indirect radiance [Lafortune and Willems 1995] as control variate for diffuse illumination. Fan et al. [2006] presents an estimator based on control variates that varies over the scene depending on surface properties and lighting conditions, unlike previous work that only uses one generic estimator for all the scenes. Clarberg and Akenine-Moeller [2008] used an approximation of the visibility function as control variate for computing illumination from environment maps. Rousselle et al. [2016] explored two sophisticated applications of control variates in rendering: re-rendering when changing material properties, and a gradient-domain rendering reconstruction strategy. In both cases the control variate is constructed in image space, while our approach can explore any required dimensions of light transport, as illustrated in several applications. Keller [2001] proposed using Multilevel Monte Carlo [Heinrich 2001] for rendering, leveraging low-resolution renderings as a control variate of higher-resolution ones. Our approach shares a similar idea, but uses adaptive quadrature to build the control variate, and works over arbitrary sub-domains of the light transport integral. Recently, Kondapaneni et al. [2019] showed that optimal weights for multiple importance sampling can be interpreted as carefully-chosen control variates.

Spherical harmonics-based control variates have been applied to integrate environment lighting with anisotropic geometry with tangent environment maps [Mehta et al. 2012], and have been applied with polygonally-clipped incident lighting such as area lights where the control variate accounts for the higher bandwidth lighting [Belcour et al. 2018]. Vévoda et al. [2018] used control variates to obtain an unbiased approximation of the incident direct illumination computed using a Bayesian regression model. In contrast, our method is agnostic to the signal being integrated and the control variate handles multidimensional integrals because it is obtained with a multidimensional nested quadrature rule, therefore it accounts for more phenomena besides incident lighting. Close to our work, Müller et al. [2020] learn a parametric control variate for incident light in a multidimensional domain using neural networks. While their approach is very powerful, it requires sophisticated learning architectures. In contrast our work uses relatively simple polynomials based on well-studied nested quadrature rules.

Finally, carefully chosen constant control variates have also been used for reducing variance in transmittance estimation in the presence of participating media [Kutz et al. 2017; Novák et al. 2014]. We demonstrate that our adaptive polynomial can easily be plugged into these frameworks, resulting in significant variance reduction in some cases.

3 PRELIMINARIES

3.1 Numerical integration

Any general integration problem is expressed as

$$F = \mathbb{E}[f(x)] = \int_\Omega f(x)\, d\mu(x), \tag{1}$$

where F is the integral, defined as the expected value $\mathbb{E}[\cdot]$ of the function $f(x)$ on the integration domain Ω, $x \in \Omega$ represents a differential element of the domain and $\mu(x)$ is the measure of the variable within the domain. Monte Carlo integration numerically approximates Equation (1) as

$$F \approx \langle F \rangle_N = \frac{1}{N} \sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)}, \tag{2}$$

where N is the number of samples used to estimate $\langle F \rangle_N$, x_i is a randomly sampled element of the domain, and $p(x_i)$ is the probability distribution function (pdf), that describes the probability of selecting x_i as the ith sample. Choosing a good pdf that approximates the integrand is key to reduce the variance of $\langle F \rangle_N$; this is often called *importance sampling*.

3.2 Primary space

The integration domain Ω can be difficult to treat (presents manifolds or high-order complex structures). However, by considering the sampling strategy as a change of variable, where pdf $p(x_i)$ in Equation (2) is its Jacobian [Muñoz 2014], it is possible to transform the integral domain Ω into a primary space Ω_U of random numbers, defined as the unit hypercube $\Omega_U = \bigcup_{D=1}^{\infty} [0..1]^D$ [Kelemen et al. 2002]. The domains Ω and Ω_U are related by the mapping $x = P^{-1}(\bar{u})$, where $P^{-1}(\bar{u})$ is the inverse of the cumulative function of $p(x_i)$. By applying the change of variables defined by mapping $P^{-1}(\cdot)$, and given that $d\bar{u} = p(x)\, d\mu(x)$, we can redefine Equation (1) as

$$F = \int_{\Omega_U} \frac{f\left(P^{-1}(\bar{u})\right)}{p\left(P^{-1}(\bar{u})\right)} d\bar{u}. \tag{3}$$

Multiple mappings in primary space. Equation (3) assumes a single mapping $P^{-1} : \Omega_U \mapsto \Omega$. However, multiple mappings can be used in practice, and their choice (i.e., the technique used to sample x) can dramatically affect the variance of the estimate $\langle F \rangle_N$. Multiple importance sampling (MIS) [Veach and Guibas 1995] allows to optimally combine multiple mappings by weighting the contribution of each sample x_i depending on the technique used to generate it. We can generalize Equation (3) to an arbitrary number of mappings T:

$$F = \int_{\Omega_U} \sum_{t=1}^{T} W_t \left(P_t^{-1}(\bar{u}) \right) \frac{f\left(P_t^{-1}(\bar{u})\right)}{p_t\left(P_t^{-1}(\bar{u})\right)} \, d\bar{u}, \tag{4}$$

where $P_t^{-1}(\bar{u})$ and $p_t(\bar{x})$ are the mapping technique t and its associated pdf, and $W_t(\bar{x})$ is the weight of technique t to $\bar{x}$. This weight needs to hold $\sum_{t=1}^{T} W_t(\bar{x}) = 1$ whenever $f(\bar{x}) \neq 0$, and $W_t(\bar{x}) = 0$ whenever $p_t(\bar{x}) = 0$.

3.3 Control variates

Another strategy for variance reduction is through *control variates*. Assuming a function $h(x)$ of known expected value $H = \mathbb{E}[h(x)] = \int_\Omega h(x) \, d\mu(x)$, we can then reformulate Equation (1) as

$$F = \int_\Omega f(x) - \alpha h(x) \, d\mu(x) + \alpha H, \tag{5}$$

where $f(x) - \alpha h(x)$ is the residual with respect to the control variate and the strength of the control variate $h(x)$ is controlled by the parameter α. Then, we can compute the Monte Carlo estimate $\langle F \rangle_N$ for N samples by numerically integrating its residual as

$$\langle F \rangle_N = \frac{1}{N} \sum_{i=1}^{N} f(x_i) - \alpha h(x_i) + \alpha H. \tag{6}$$

By minimizing the variance of Equation (6), we obtain that the optimal choice for α as (see [Robert and Casella 2004, Section 4.4.2])

$$\alpha = \mathrm{Cov}[f(x), h(x)] / \mathrm{Var}[h(x)], \tag{7}$$

which leads to a variance on the estimate of Equation (6)

$$\mathrm{Var}[\langle F \rangle] = \mathrm{Var}[f(x)] \left(1 - \mathrm{Corr}[f(x), h(x)]^2\right). \tag{8}$$

To further reduce variance, control variates can be combined with importance sampling by choosing an appropriate sampling routine with pdf $p(x)$. This transforms Equation (6) into [Owen 2013, Section 9.10]

$$\langle F \rangle_N = \frac{1}{N} \sum_{i=1}^{N} \frac{f(x_i) - \alpha h(x_i)}{p(x_i)} + \alpha H. \tag{9}$$

Unfortunately, in this case we cannot obtain a closed-form optimal α, and need to compute it by least squares regression of the variance of the estimate in Equation (9). For simplicity on the implementation, we opt for computing α following Equation (7). As demonstrated by Owen and Zhou [2000] this does not change the asymptotic convergence, and in our tests we observed that it provides a good tradeoff between computational cost and variance reduction.

4 ADAPTIVE POLYNOMIAL CONTROL VARIATES

To leverage the variance reduction of both control variates and importance sampling, we build a control variate that approximates the integrand in primary space. By plugging Equation (3) into Equation (9) we get

$$\langle F \rangle_N = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{\frac{f\left(P^{-1}(\bar{u}_i)\right)}{p\left(P^{-1}(\bar{u}_i)\right)} - \alpha h(\bar{u}_i)}{p_h(\bar{u}_i)} \right) + \alpha H, \tag{10}$$

where the new pdf $p_h(\bar{u})$ should be as proportional to the residual as possible. Since obtaining a global optimal $h(\bar{u})$ is unlikely, we instead define a piecewise control variate along the whole domain Ω_U. For that, we draw inspiration from quadrature-based integration [Burden and Faires 2005]. Quadrature integration approximates the expected value F of the function $f(x)$ by means of a linear combination of samples in $f(x)$, weighted by carefully chosen weights—the *quadrature rules*—as

$$F \approx \sum_{i=1}^{N_h} w_i f(x_i), \tag{11}$$

where N_h is the number of samples x_i, with associated weights w_i. The samples and corresponding weights depend on the chosen quadrature rule. Several quadrature rules exist: The simplest ones (Newton-Cotes rules) approximate the function $f(x)$ by using a piecewise-polynomial approximation, by subdividing the space in deterministic evenly-distributed regions. These techniques can be made adaptive via *nested quadrature rules* [Press et al. 2007].

While quadrature rules are biased, their convergence depends on the nature of the signal, and are strongly affected by the curse of dimensionality. However, polynomial approximations similar to Newton-Cotes rules satisfy many properties that make them interesting for using them as the control variate $h(x)$: The evaluation is efficient, the integral is analytical, the construction is lightweight and adaptive, they can approximate any function $f(x)$ up to a certain degree of accuracy, and they provide an estimate of the error that can be used as $p_h(\bar{u}_i)$ for importance sampling the residual. Our method is illustrated in Figure 2.

In the following, we first describe the (multidimensional) polynomial approximation of $f(x)$, and its adaptive generalization. Then, we describe how we use $h(x)$ as a control variate to solve Equation (9), that we will later include into primary space as in Equation (10). Finally, we analyze the convergence of our technique as a function of the dimensionality of the signal.

4.1 Piecewise-polynomial control variate

Let us assume for now that $f(x) \in \mathbb{R}$, with $x \in \Omega = \mathbb{R}$ (we generalize to $\mathbb{R}^D$ later in the subsection). Based on Newton-Cotes composite rules we build our control variate $h(x)$ as a piecewise approximation of the signal. We divide the integration Ω domain into M smaller disjoint subdomains $\Omega_r = [a_r, b_r]$, so that $\bigcup_{s=1}^{M} \Omega_r = \Omega$ and $\Omega_r \cap \Omega_s = \emptyset, \forall r \neq s$.

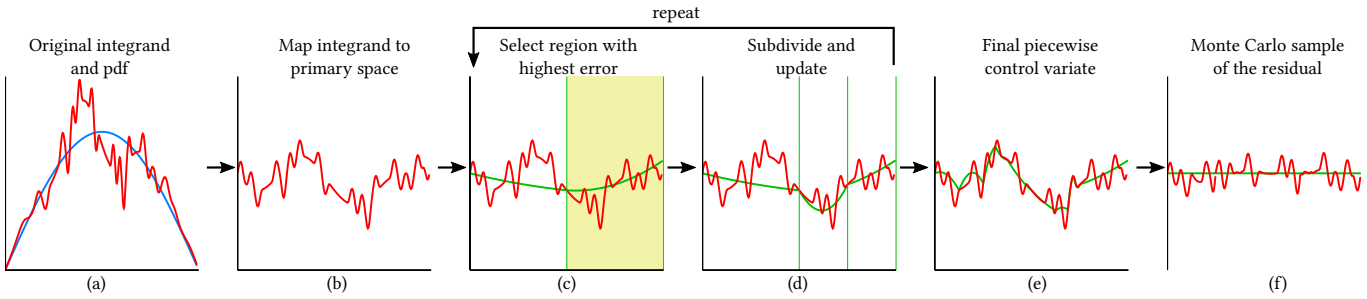

Fig. 2. This figure illustrates our approach in a one-dimensional integral. The algorithm starts (a) from the integrand (in red) and a pdf (in blue). The pdf provides a mapping to primary space (b). Then the piecewise control variate (in green) is calculated by iteratively selecting the highest error region (c) and splitting it into two subregions (d) for a specified number of iterations. Once the control variate is obtained (e), the final integral is obtained by sampling the residual difference between the primary space integrand and the control variate (f).

For each disjoint subdomain Ω_r, we approximate $f(x)$, with $x \in \Omega_r$, as a polynomial

$$f(x) \approx h_r(x) = \sum_{i=0}^{n} c_{r,i} x^i, \tag{12}$$

where n is the order of the polynomial defined in Ω_r (order two in our case) and $c_{r,i}$ are its coefficients. The coefficients $c_{r,i}$ are calculated by interpolating from a set of uniformly distributed samples $f(x_{r,i})$, where $(x_{r,i})_{i\in[0,n]} \in \Omega_r$, with $x_{r,0} = a_r$, $x_{r,i+1} = x_{r,i} + h_r$ and $h_r = (b_r - a_r)/n$. We interpolate through a precomputed linear system of equations over a monomial basis by inverting the Vandermonde matrix that defines such a system of equations. This approach naturally extends to higher-order rules and multiple dimensions.

Since the polynomial can be integrated analytically, we can obtain weights $w_{r,i}$ that define the order-n quadrature rule through interpolation by substitution as

$$\int_{\Omega_r} f(x)dx \approx H_r = \sum_{i=0}^{n} w_{r,i} f(x_{r,i}), \tag{13}$$

which is a standard approach for deriving the weights within quadrature rules. In general, for low-order known quadrature rules (such as Simpson's rule, used in this paper) there is no need to derive such weights because they can be found in the corresponding literature. We can compute the integrand for the full domain Ω as the sum of the integrals for all regions as $H = \sum_r H_r$.

Generalizing to $\mathbb{R}^D$. For the multidimensional case, where $\Omega_r \in \mathbb{R}^D = \{[a_{r,1}, b_{r,1}] \cdots [a_{r,D}, b_{r,D}]\}$, we generalize Equation (12) for $x \in \mathbb{R}^D$ and $x = \{x_1 \cdots x_D\}$, as

$$h_r(x) = \sum_{i_1=0}^{n} \cdots \sum_{i_D=0}^{n} c_{r,\{i_1 \cdots i_D\}} \prod_{j=1}^{D} x_j^{i_j}, \tag{14}$$

where $c_{r,\{i_1 \cdots i_D\}}$ is the polynomial coefficient. We calculate the coefficients using the same approach as for a single dimension by interpolating from a multidimensional grid using a linear system over a multivariate monomial basis. For integration, we apply Fubini's theorem, and build the multidimensional rules as

$$\int_{\Omega_r} h_r(x)dx = \sum_{d=1}^{D} \sum_{i=0}^{n} w_{r,\{d,i\}} f(x_{r,i}), \tag{15}$$

where the weighs $w_{r,\{d,i\}}$ are obtained from the product of the one-dimensional rule's weights, and x_i form a D-dimensional grid of sampled points in Ω_r.

Multiple mappings. We can leverage the variance reduction provided by using multiple importance sampling (MIS) in Monte Carlo integration [Veach and Guibas 1995] by combining multiple mappings to reduce the error when computing H. Assuming the integration domain is the primary space (i.e., $\Omega_r \in \Omega_U$), we introduce $h(x)$ in Equation (4) and move the sum out of the integral as

$$\begin{aligned} H_r &= \sum_{t=1}^{T} \int_{\Omega_r} W_t\left(P_t^{-1}(x)\right) h(x)\, dx \\ &= \sum_{i=0}^{n} w_{r,i} \sum_{t=1}^{T} W_t\left(P_t^{-1}(x_{r,i})\right) \frac{f\left(P_t^{-1}(x_{r,i})\right)}{p_t\left(P_t^{-1}(x_{r,i})\right)}. \end{aligned} \tag{16}$$

4.2 Adaptive approximation

So far, we have not assumed any specific distribution of the regions $\{r\}$ within the domain Ω. Such distribution might be uniform (equally partitioning the domain), but this could be suboptimal. Ideally, we would like to have a finer sampling rate in regions where our order-n polynomial fails at approximating $f(x)$, while leaving a coarser sampling in areas with lower error.

In this context *nested quadrature rules* provide the tool for adaptive numerical approximation. The key idea is to use two quadrature rules of different order for approximating the same integral. The higher-order rule is used as an oracle of the integrated signal F_r for each region r, and the low-order rule as the approximation of the integrand F_r. The difference between both rules is the estimate of the error $\hat{E}_r$. This estimation of the error is then used to select the region to subdivide.

More formally, let the two estimates H_r^{h} and H_r^{l} computed using quadrature rules of order n_{h} and n_{l} respectively, with $n_{\mathrm{h}} > n_{\mathrm{l}}$, be

$$H_r^{\mathrm{h}} = \sum_{i=0}^{n_{\mathrm{h}}} w_{r,i}^{\mathrm{h}} f\left(x_i^{\mathrm{h}}\right) \quad \text{and} \quad H_r^{\mathrm{l}} = \sum_{i=0}^{n_{\mathrm{l}}} w_{r,i}^{\mathrm{l}} f\left(x_i^{\mathrm{l}}\right), \tag{17}$$

where x_i^{h} and x_i^{l} are the samples for each rule, and $w_{r,i}^{\mathrm{h}}$ and $w_{r,i}^{\mathrm{l}}$ their corresponding weights. For the rules to be nested, it is required that $\{x_i^{\mathrm{l}}\} \subset \{x_i^{\mathrm{h}}\}$, which allows reusing samples when computing both rules. Then, the estimate of the error is $\hat{E}_r = |H_r^{\mathrm{h}} - H_r^{\mathrm{l}}|$. In this work we use the Simpson-Trapezoidal nested rule ($n_{\mathrm{h}} = 2$ and $n_{\mathrm{l}} = 1$).

Subdivision strategy. Most nested quadrature rules use a tolerance parameter to subdivide until the error is below a threshold. In our context, we cannot use this approach since we would like to specify a samples budget. Our algorithm iteratively subdivides the region r with highest $\hat{E}_r$, until we reach the input budget of samples N_h. To efficiently obtain the region with maximum error, we store the regions at a given step in a heap structure, which is updated on each iteration. Starting from a single initial subdivision that covers the whole integration range, we split, for each required subdivision, the top of the heap using binary splitting along the dimension of highest error. Each splitting requires a fixed number of samples according to the involved quadrature rules and the dimensionality of the problem, so the sample count N_h is linear with the number of regions M. With quadrature rules with evenly distributed samples, a subset of the samples of each subregion comes from the previously subdivided one, leading to deterministic sample count as

$$N_h = (n_{\mathrm{h}} + 1)^D + (M - 1)\ n_{\mathrm{h}}(n_{\mathrm{h}} + 1)^{D-1}. \tag{18}$$

Note that depending on the (deterministic) positions of samples $\{x_i^{\mathrm{h}}\}$, high-frequency features might be missed by the error estimation. This can lead to regions with an inaccurate polynomial approximation $h_r(x)$ that are kept stagnant (i.e., never subdivided). To avoid this pitfall, we add a term to the error that accounts for the size of the region, so larger inaccurate regions can also be subdivided. As the error estimation must be calculated per dimension d (to split the dimension of highest error) the final form of $\hat{E}_{r,d}$ is

$$\hat{E}_{r,d} = \left|H_r^{\mathrm{h,d}} - H_r^{\mathrm{l,d}}\right| + \left(b_{r,d} - a_{r,d}\right)\epsilon, \tag{19}$$

where $H_r^{\mathrm{l,d}}$ is the integral of the control variate $h_r(x)$ using the higher order rule h for all the dimensions except for dimension d (in which we apply the lower rule l), $a_{r,d}$ and $b_{r,d}$ are the lower and upper limits of the integration domain Ω_r for dimension d, and ϵ is a positive constant. Intuitively, ϵ is related to the uniformity of the subdivisions: Larger values lead to a more uniform region size distribution, while smaller values will lead to subdivisions proportional to the estimated error. We empirically set $\epsilon = 10^{-5}$.

Figure 3 shows our polynomial approximation (the control variate) and the residual for two two-dimensional functions: The control variate accurately captures the low frequency regions of the function, while the high frequency details remain in the residual.

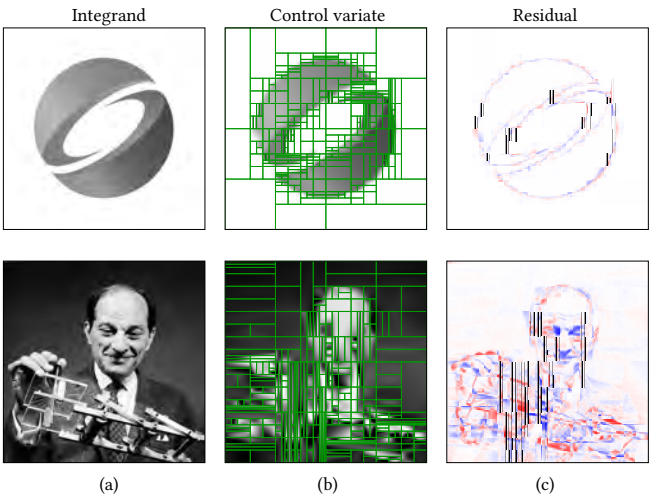

Fig. 3. Integration of two two-dimensional functions (a), its piecewise-polynomial approximation used as control variate (b, boundaries of each region in green), and the corresponding residual (c, where red and blue are the positive and negative residual, respectively).

Control variate for subdomains and bucketing. While the control variate $h(x)$ is defined for the integration domain Ω, it can also be applied to any subdomain $\Omega_b \subset \Omega$. While the integral for the whole domain Ω is $H = \sum_r H_r$, the integral of the subdomain is

$$\int_{\Omega_b} h(x)\, dx = \sum_r \int_{\Omega_r \cap \Omega_b} h_r(x)\, dx. \tag{20}$$

This is especially useful when bucketing (discretizing) the same integrand into a set of buckets (e.g., the pixels of an image or video). In these cases, the same control variate $h(x)$ can be applied for computing all buckets, effectively amortizing the construction of the control variate along multiple buckets. In Sections 6 to 8 we apply this strategy in image space where each pixel is an independent bucket but the control variate is shared among all pixels. Furthermore, in Section 7 we compare this bucketing strategy against computing the control variate per pixel, showing faster convergence and higher pixel coherence when bucketing.

4.3 Residual integration

So far we have described our adaptive construction of the piecewise-polynomial approximation of the integral in the primary domain. Now we describe how we compute the estimate in Equation (10). In order to reduce variance of the estimate, we would like to draw samples with a pdf $p_h(\bar{u})$ that is approximately proportional to the residual, so that $p_h(\bar{u}) \propto \frac{f(P^{-1}(\bar{u}))}{p(P^{-1}(\bar{u}))} - \alpha h(\bar{u})$. Assuming that the error guiding the construction of our control variate $\hat{E}_{r,d}$ (Equation (19)) is a good estimate of the residual, and that the regions r subdividing the primary domain have roughly a similar error, we can uniformly sample a region with probability M^{-1}, and then sample uniformly within the chosen region. The resulting pdf is $p_h(\bar{u}) = \frac{1}{M|\Omega_r(\bar{u})|}$, where $|\Omega_r(\bar{u})|$ is the hypervolume of the selected region r, and M is the number of regions. Note that this pdf is applied only for integrating the residual in primary space, on top of any other importance sampling

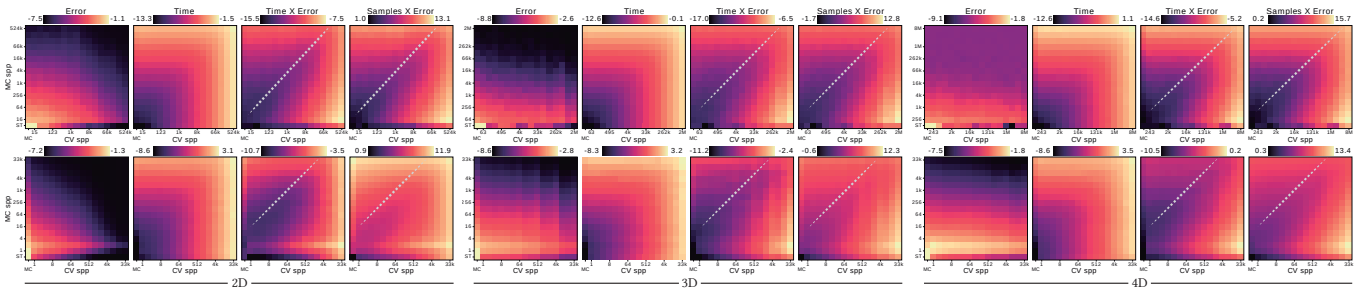

Fig. 4. Average error, cost, and efficiency maps (brighter means higher in logarithmic scale) for a set of integrals with increasing dimensionality, as a function of the number of samples allocated to building the control variate and to integrate the residual (horizontal and vertical axes, respectively). The leftmost column in each map represents Monte Carlo integration, while the bottom row in each map represents nested quadrature (Simpson-Trapezoidal). **Top row:** integration over the full integral domain. **Bottom row:** integration into 1000 buckets (pixels), with sample count representing samples per bucket. The white lines show the optimal ratio between the number of samples allocated to compute the control variate and the residual.

strategy used for the corresponding application. When bucketing (see last paragraph of the previous section) we first stratify per bucket (pixel), search all regions of the control variate falling in the bucket, and then uniformly sample each region within the bucket using $p_h(\bar{u})$. We select a per-bucket $\alpha = \text{Cov}[\langle F\rangle, \langle H\rangle]/\text{Var}[H]$ (Equation (7)), where we estimate $\text{Cov}[\langle F\rangle, \langle H\rangle]$ and $\text{Var}[H]$ from the set of random samples falling within the bucket. Note that computing α from the same set of samples used to estimate $\langle F\rangle$ results into a negligible but non-zero bias on the final result; this bias vanishes as we increase the number of samples [Owen 2013, Section 8.9].

4.4 Analysis

Here we analyze the performance of our technique as a function of the samples used for building the control variate (built using a Simpson-Trapezoidal nested rule), and for computing the residual. We integrate a number of functions of increasing dimensionality (from 2D to 4D), and include the boundary cases, i.e., Monte Carlo and quadrature for comparison. Analysis for each individual function can be found in the supplemental (Section S.1).

Figure 4 shows the average error, cost, and the product between cost and error for each function's dimensionality when integrating over the full domain (top), and projecting the integral into buckets (bottom). The horizontal and vertical axes represent the number of samples for generating the control variate and for computing the residual, respectively. As expected, the increased dimensionality slows down the convergence rate of the control variate, while the residual converges with the usual rate in Monte Carlo integration. In terms of cost, the samples generating the control variate are more expensive than Monte Carlo samples, specifically when integrating the full domain (top row). However, this cost is amortized when subdividing the integration domain into buckets (bottom row).

Pure Simpson-Trapezoidal quadrature integration (bottom row in each map, marked as *ST*) seems to converge relatively fast, but its convergence is irregular and it introduces bias that translates into perceivable artifacts. These artifacts, as well as higher-order nested rules, are explored in the supplemental (Section S.4).

By computing the efficiency of the integration (as a function of the time and error, and the number of samples and error), we found that there is an optimal trade-off between the samples allocated to the control variate and to the residual. Such optimal trade-off is, on average, one sample for the control variate out of three for full integrals and one sample out of 16 when amortizing among different buckets (white dashed line in the efficiency maps). These ratios are used for all the results in this paper.

4.5 Implementation

We implement our adaptive control variate as a generic template in C++. It is agnostic to the nature of the function being integrated, and easy to integrate into other systems. We plug it in Mitsuba [Jakob 2010], which provides the function to be integrated.

We compute the results on an Intel Xeon Gold 6400 3.7 GHz CPU workstation with 256 GB of RAM. We measure the error using the root mean square error (RMSE).

We build the control variate using a Simpson-Trapezoidal nested rule. This results in an order-two polynomial approximation of the signal. For each iteration, we deterministically draw three samples per dimension. For bucketing we use a box filter as the reconstruction kernel. Including other kernels with analytical integration is left as future work. For the residual, we randomly sample the regions as described in Section 4.3 using a 64-bit Mersenne twister random number generator.

Based on our analysis in Section 4.4, in all our results we allocate 1/3 (full integrals) and 1/16 (amortized samples by bucketing) of the total samples to build the control variate, while the rest are used to compute the residual. Detailed cost breakdown for all our results can be found in the supplemental (Section S.3).

5 APPLICATION 1 : ADAPTIVE RESIDUAL RATIO TRACKING

Here we apply our technique to the computation of transmittance in heterogeneous participating media. As light travels from position $\mathbf{x}_0$ to $\mathbf{x}_1$ through a participating medium, it is attenuated following the one-dimensional integral $T(\mathbf{x}_0, \mathbf{x}_1)$:

$$T(\mathbf{x}_0, \mathbf{x}_1) = \exp(-\tau) = \exp\left(-\int_0^t \mu(\mathbf{x}_s) ds\right), \tag{21}$$

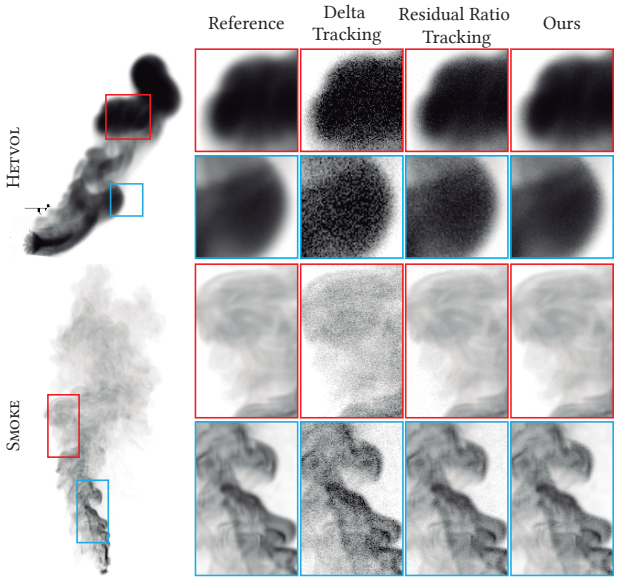

Fig. 5. Renderings of two purely absorbing media, with high (first row, Hetvol) and low (second row, Smoke) densities, computed using delta tracking [Woodcock et al. 1965], residual ratio tracking [Novák et al. 2014], and our adaptive residual ratio tracking (full image). The three methods have approximately the same number of media queries.

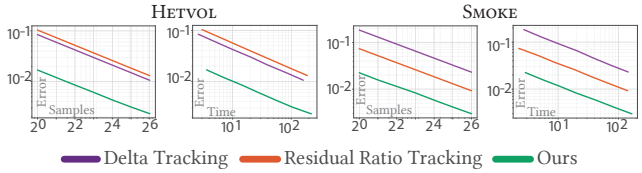

Fig. 6. Convergence for the scenes in Figure 5 as a function of media queries (left and middle right) and core time in seconds (middle left and right) for delta tracking, residual ratio tracking using the average extinction as control extinction, and our adaptive control variate. We compute the error metric over the entire image.

with $t = |\mathbf{x}_1 - \mathbf{x}_0|$, $\mu(\mathbf{x})$ the extinction coefficient at $\mathbf{x}$, $\mathbf{x}_s = \mathbf{x}_0 + s\,\omega$, and $\omega = \frac{\mathbf{x}_1 - \mathbf{x}_0}{t}$.

Several unbiased Monte Carlo-based methods have been proposed to numerically solve Equation (21), based on the key idea of introducing null virtual particles to fill the medium, resulting in a constant *virtual* extinction (the majorant $\bar{\mu}$, see [Novák et al. 2018] for an in-depth overview on the topic), at the cost of introducing variance. Residual ratio tracking [Novák et al. 2014] reduces variance by introducing a *control extinction* μ_c, transforming Equation (21) to

$$T(\mathbf{x}_0, \mathbf{x}_1) = \exp\left(-\int_0^t \mu(\mathbf{x}_s) - \mu_c(\mathbf{x}_s) ds + \mu_c\, t\right). \quad (22)$$

Note that the estimate of τ in Equation (22) is essentially Equation (5) with $\alpha = 1$. Unfortunately, this approach uses a constant μ_c, which works well if the signal varies slightly around $\mu(\mathbf{x}_s)$, but that might increase variance if μ_c diverges from the actual extinction. While in practice this is partially solved using a piecewise constant (or linear) estimate of μ_c, it requires precomputing a supervoxel hierarchy which limits its applicability to voxelized media, while still requiring heuristics to solve special cases. Instead, we propose to use our adaptive polynomial approximation as the control extinction $\mu_c(\mathbf{x}_s)$.

We analyze the performance of our technique against residual ratio tracking with constant precalculated μ_c (set to $\mu_c = \int_0^t \mu(\mathbf{x}_s) ds$, which is the optimal parameter according to the authors) and delta tracking [Woodcock et al. 1965]. In all cases we use the same tight majorant $\bar{\mu} = \max_{\mathbf{x}}(\mu(\mathbf{x}))$. We build our control variate by performing three iterations, which results in a small overhead (just nine additional medium queries).

Figure 5 shows a comparison between the three techniques at an equal number of media queries, for two absorbing heterogeneous media with high (Hetvol, top) and low density (Smoke, bottom). Without introducing a spatially-varying control extinction μ_c (using, e.g, supervoxels), residual ratio tracking introduces noise in regions where the extinction deviates significantly from μ_c, resulting in higher variance than delta tracking. While this could be alleviated by subdividing the space in subvolumes with tighter majorants and control extinctions, these would also benefit our method.

Figure 6 shows the convergence of the three methods. As expected, the performance of residual ratio tracking and our method are related to the quality of the approximation. When residual ratio tracking performs well, our technique in general performs similarly. However, when a constant control fails at representing the medium extinction (e.g., in cases with non-uniform densities), our technique adapts to the signal without introducing a significant overhead. We refer the reader to the supplemental (Section S.2) for more examples.

6 APPLICATION 2 : LOW-ORDER SCATTERING

We apply our technique to computing one- and two-bounces scattering in homogeneous media from a point light source (1D integral) and a collimated beam (2D integral), respectively. In both cases, we want to compute the radiance at point x_o from direction ω as

$$L(x, \omega) = \int_0^t T(x, x_s)\sigma_s L_i(x_s, \omega)\, ds, \quad (23)$$

where t is distance of intersection of the ray, $x_s = x - \omega\, t$, $T(x, x_s) = e^{-\sigma_t \|x_s - x\|}$ is the transmittance between x and x_s, σ_t and σ_s are the extinction and scattering coefficients, and $L_i(x_s, \omega)$ is the in-scattered radiance. For light incoming from a point source then

$$L_i(x_s, \omega) = \frac{\Phi_l}{\|x_s - x_l\|^2} V(x_l, x_s) T(x_l, x_s) \rho(x_l \to x_s \to x_o), \quad (24)$$

where x_l and Φ_l are the light's position and intensity, $V(x_l, x_s)$ is the binary visibility term, and $\rho(x_l \to x_s \to x)$ is the phase function at x_s.

In the case of the light source being a collimated beam defined by position x_l and direction ω_l, then $L_i(x_s, \omega)$ becomes an additional

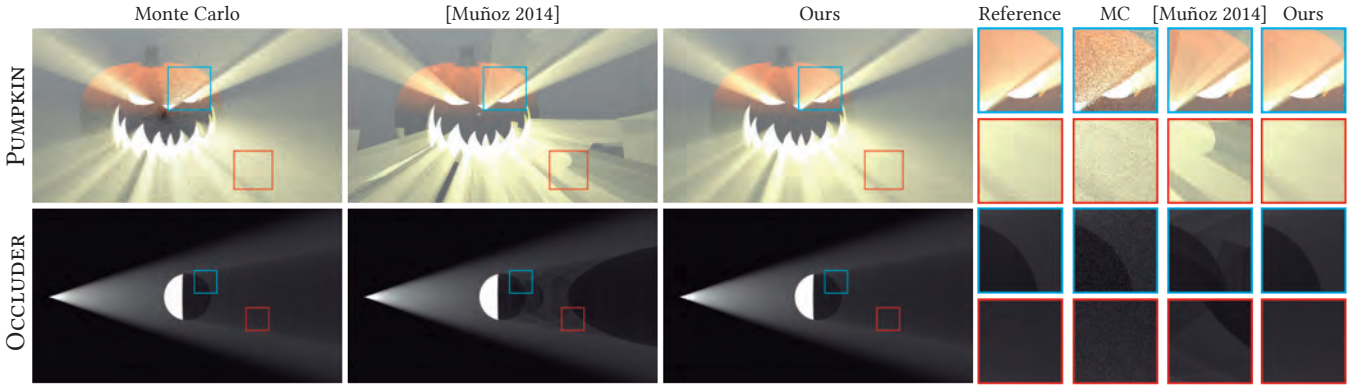

Fig. 7. Equal-samples (64 spp) comparison between Monte Carlo, Simpson-Trapezoid quadrature [Muñoz 2014] and our technique for computing single scattering from a point light source in isotropic homogenous media. Our technique yields more accurate results and recovers both the smooth global structure of light transport and the high frequency details of the scene, while remaining unbiased.

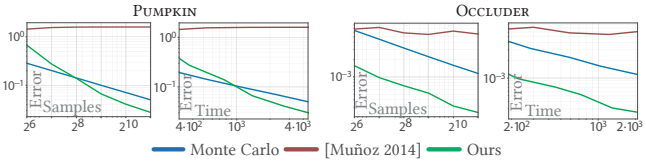

Fig. 8. Convergence for the scenes in Figure 7 for Monte Carlo integration, Simpson-Trapezoid quadrature [Muñoz 2014], and our technique, as a function of the number of samples and core time in seconds.

1D integral [Novák et al. 2012] as

$$L_i(x_s, \omega) = \int_0^{t'} \frac{\Phi_l}{\|x_s - x_l\|^2} V(x_s, x_{s'}) T(x_l, x_u) T(x_s, x_{s'}) \sigma_s(x_{s'}) \rho(x_l \to x_{s'} \to x_s) \rho(x_u \to x_s \to x) ds', \quad (25)$$

where t' is distance of intersection of the light beam, with $x_{s'} = x_l + \omega_l s'$. We amortize the cost of generating the control variate along pixels by bucketing an additional integration domain (image plane). This results into two integration problems of three (point light) and four dimensions (collimated beam).

Figure 7 shows the results for single scattering in isotropic homogeneous media. We compare against pure Monte Carlo, as well as the quadrature-based integration proposed by Muñoz [2014] for single scattering. In all cases, we use equiangular sampling for mapping to primary space [Kulla and Fajardo 2012].

Muñoz's approach [2014] integrates only along a single dimension (the ray distance), missing some high-frequency details (e.g., discontinuities). In contrast, our control variate adapts over a three-dimensional space (screen space and ray distance), which allows finding potential discontinuities along these multiple dimensions. Moreover, these discontinuities and high-frequency content are handled on the residual by Monte Carlo integration. As shown in Figure 8, the ability to handle both low- and high-frequency parts of the integrand results in better convergence than both alternative limit cases.

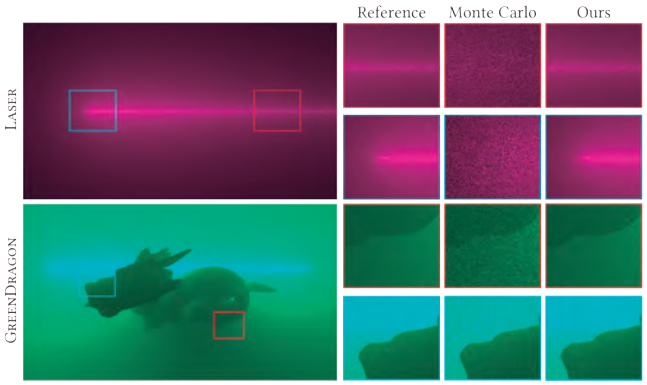

Fig. 9. Equal-samples (64 spp) comparison between Monte Carlo and our technique for computing two-bounce scattering from a collimated beam in isotropic homogenous media. While pure Monte Carlo generates high-frequency noise, our approach excels at the smooth regions, accurately handling the sharp details.

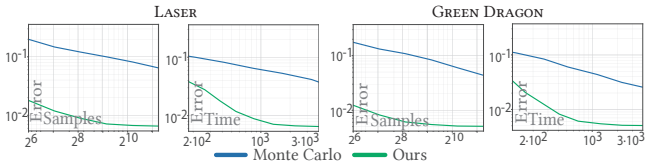

Fig. 10. Convergence for the scenes in Figure 9 for Monte Carlo integration and our technique, as a function of the number of samples and core time in seconds.

Similar trends can be found for the case of two-bounce scattering, as shown in Figures 9 and 10. In this case, we use the two-dimensional mapping proposed by Novak et al. [2012]. Again, our control variate is able to recover most of the low frequencies common in scattering media, while the details are handled by means of Monte Carlo integration of the residual.

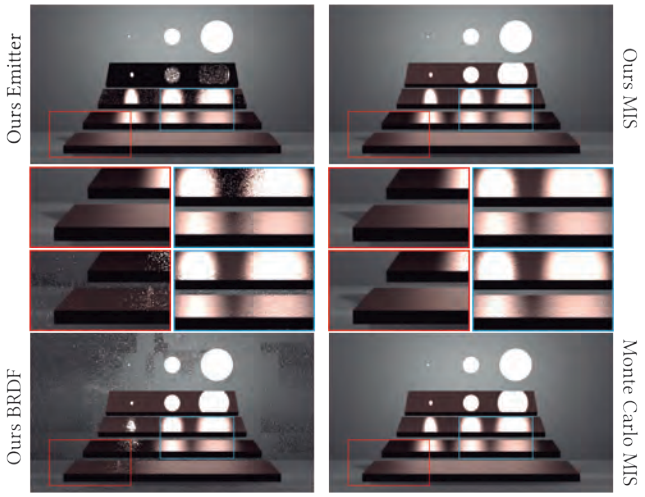

Fig. 11. MIS Test: Comparison of our approach with individual specific mappings to primary space (left column) only sampling the emitter (top) or the BRDF (bottom). Right column shows results with both combined mappings (MIS) with our technique (top) and Monte Carlo (bottom). Notice how the ability to exploit multiple mappings better fits our control variate to the integrand.

7 APPLICATION 3 : DIRECT ILLUMINATION

Here we compute direct illumination at a point x as seen from a direction ω. We solve the rendering equation, as an integral over all points x_l on the surface of the emitters A:

$$L(x,\omega) = \int_A \Phi(x_l \to x) B(x_l \to x \to \omega) V(x \leftrightarrow x_l) G(x \leftrightarrow x_l)\, dx_l, \tag{26}$$

where $\Phi(x_l \to x)$ is the radiance emitted at y towards x, $B(x_l \to x \to \omega)$ is the bidirectional reflectance distribution function (BRDF) at x, and $G(x \leftrightarrow y)$ is the geometric attenuation.

As discussed in Section 3.2, our technique leverages the use of multiple mappings in primary space (Equation (4)) in our adaptive polynomial control variate. We solve Equation (26) by combining BRDF and emitter sampling techniques using the power heuristic [Veach and Guibas 1995]; we illustrate the effect of each technique in Figure 11. Note that other sophisticated sampling methods could be applied on top of our technique.

Figure 12 shows a visual comparison of several scenes with different types of emitters and materials. We compare the performance of computing the control variate per pixel ("Ours 2D", resulting in a 2D integral per pixel) and building a single control variate on the full image ("Ours 4D", resulting in a single bucketed 4D integral). Both approaches result in less noise than Monte Carlo for the same number of samples. In addition, bucketing the full image space ("Ours 4D") results in both less error and structure on the noise. Figure 13 shows the convergence for the three scenes: In all cases there is a similar trend, with a faster convergence of our technique, specially when bucketing the full 4D integration domain.

Note that, depending on the smoothness and complexity of the signal, our technique may need more samples until the control variate starts improving convergence.

8 APPLICATION 4 : DISTRIBUTION EFFECTS

As a final application, we use our technique for rendering distribution effects such as motion blur or depth of field [Cook et al. 1984], which increase the dimensionality of the light transport problem in one and two dimensions (time and aperture, respectively). We assume a constant shutter time for motion blur, and a thin lens model for depth of field. In all cases, we amortize samples along pixels, increasing the dimensionality of our control variate with the additional dimensions of the sensor.

We compare our method against Monte Carlo integration and Hachisuka et al.'s multidimensional adaptive technique [2008] in four different test scene setups (Figure 14): Pool (3D) includes motion blur, Chess (4D) includes depth of field, Helicopter (5D) features both motion blur and area lighting (Section 7), and Volley Balls (6D) includes both depth of field and area lighting. Our approach generates low noise results even in challenging scenarios such as rotational motion blur (Helicopter). In contrast, Hachisuka et al.'s method [2008], being biased, overblurs the result due its reconstruction kernel (e.g., the focused ball in Volley Balls or the glossy reflections in Helicopter), although it produces noiseless results in smoother areas.

Figure 15 shows the convergence of our method, compared to Monte Carlo and Hachisuka's. Our method converges faster than Monte Carlo in all the scenes. However, the additional cost of building and evaluating the control variate might result in a time penalty in scenes with simple relative cheap sampling evaluation (e.g., scenes with simple geometry like Helicopter or Volley Balls). Still, note that the slope of convergence shows that our approach pays off in the long run. We refer to Section S.3 of the supplemental material for the per-scene per-stage temporal cost breakdown. Our method also converges faster than the method by Hachisuka et al. [2008], with better or on-par performance with respect to samples per pixel, and outperforming it in terms of computational cost. Finally, the method by Hachisuka et al. requires to store all samples, resulting in a heavy memory overhead (×140 on average compared with Monte Carlo). In contrast, our method introduces a significantly smaller memory footprint (×3 on average) because it only stores the samples for the control variate (1/16 of the total). The individual memory usage per scene can be found in the supplemental (Section S.3).

9 BEYOND LOW DIMENSIONALITY

Since our control variate is based on quadrature rules, it is limited by the curse of dimensionality. This unfortunately limits its applicability to relatively low-order integration domains. However, general integration problems in rendering are of arbitrary dimensionality. In this section we analyze the performance of our control variate in high-dimensional problems by building a low-dimensional control variate on top of an estimate of the high-dimensional integral.

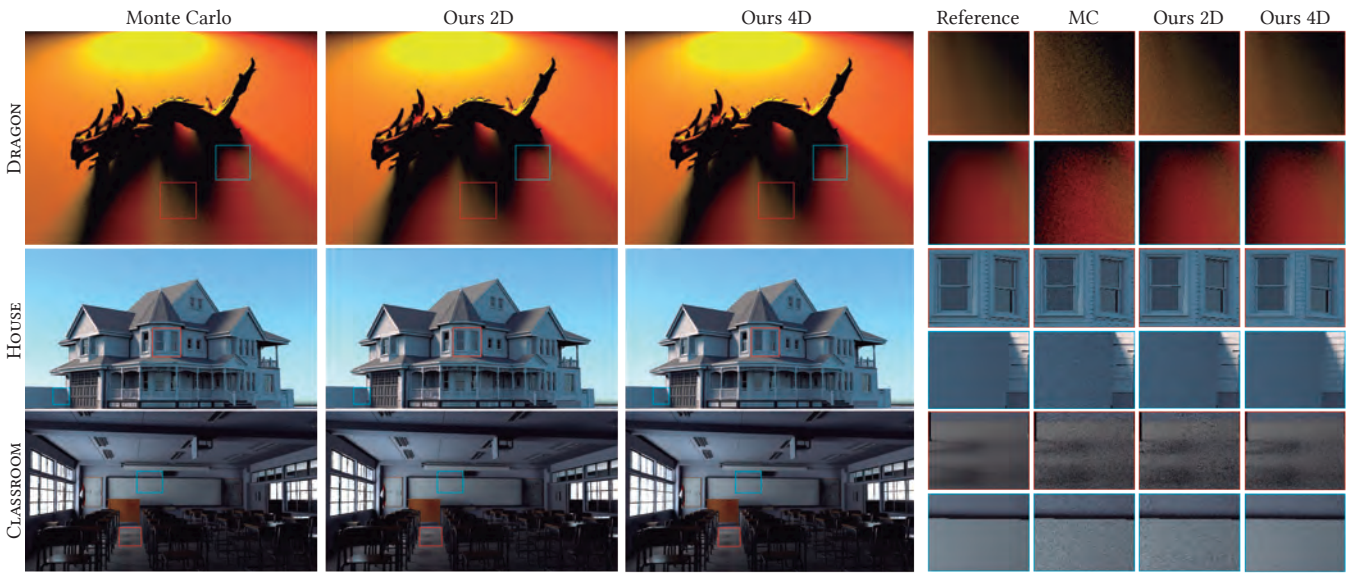

Fig. 12. Comparison of the different approaches of our technique against Monte Carlo integration for the same number of evaluations of direct illumination. In all cases, Monte Carlo produces noisier images even with MIS. In contrast, our technique leverages MIS adapting the control variate to the integrand, yielding better results both per pixel ("Ours 2D") and for the whole image space ("Ours 4D"). Furthermore, amortizing the control variate among the whole image space reduces noise in low frequency areas, removes structured noise, and serves as antialiasing. All results are calculated using 155 spp.

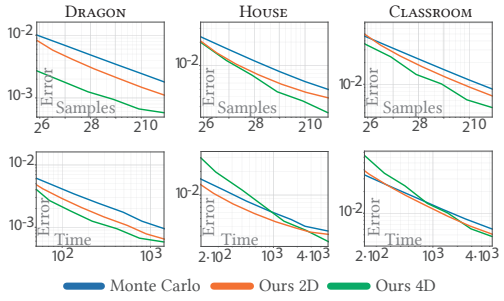

Fig. 13. Convergence of the scenes in Figure 12 as a function of the number of samples and core time in seconds for Monte Carlo, our technique applied per pixel ("Ours 2D"), and our technique extended to the full image space and bucketed per pixel ("Ours 4D"). Note that extending our control variate to 4D results in faster convergence.

Let us rewrite Equation (3) as two nested integrals on orthogonal subdomains

$$F = \int_{\Omega_U^L} \underbrace{\int_{\Omega_U^*} g(\{\bar{u}|\hat{u}\}) d\hat{u}}_{g(\bar{u})} d\bar{u}, \tag{27}$$

where $g(\bar{u}) = \frac{f(P^{-1}(\bar{u}))}{p(P^{-1}(\bar{u}))}$, the integration domain $\Omega_U \in \mathbb{R}^D$ is $\Omega_U = \Omega_U^L \cup \Omega_U^*$ with $\Omega_U^L \in \mathbb{R}^L$ and $\Omega_U^* \in \mathbb{R}^{D-L}$, and $\{\bar{u}|\hat{u}\} \in \mathbb{R}^D$ is the concatenation of the variables $\bar{u}$ and $\hat{u}$. To construct our control variate on Ω_U^L we need to evaluate the integrand function $g(\bar{u})$ over the set of samples $\bar{u} \in \Omega_U^L$. Unfortunately, this requires solving a $(D-L)$-dimensional integral, which is unlikely to have analytical form. In order to do so, we rely on simple Monte Carlo integration of this high-dimensional domain, so that $g(\bar{u}) \approx \frac{1}{N^*} \sum_i g(\{\bar{u}|\hat{u}_i\})$. Note that this has two main drawbacks: It only leverages the variance reduction of our control variate for the first L dimensions of the integral, and the control variate is built itself on non-perfect samples of the integral, which might result in an inaccurate control variate. In fact, the Monte Carlo estimate could introduce variance on top of the error driving the construction of the control variate (Equation (19)): While in our experiments we have found that this additional variance has a small effect on the final result (we refer to Section S.6 in the supplemental), even at relatively low N^*, exploring a variance-aware version of Equation (19) to account for uncertainty of the control variate when computing α is an interesting avenue of future work.

Figures 1 and 16 illustrate the results of this approach with high-order indirect illumination, while our control variate is only four-dimensional, accounting for image space and direct illumination (Cornell Box I, Cornell Box II, and Bistro) and image space and depth of field (Chess GI). We use four Monte Carlo samples to compute $g(\bar{u})$ when building the control variate (i.e. $N^* = 4$). By building the control variate by accounting for higher dimensions of the integral, we can leverage its low-dimensional structure and explore high-dimensional integrals. This behavior can be seen in Cornell Box I, where direct light does not reach the ceiling but the low-dimensional representation of the control variate is able to account for an estimate of the indirect illumination. This is similar to the depth of field example (Chess GI), where both direct and indirect illumination are used to compute the control variate for the integral along the aperture. As shown in Figure 17, leveraging the

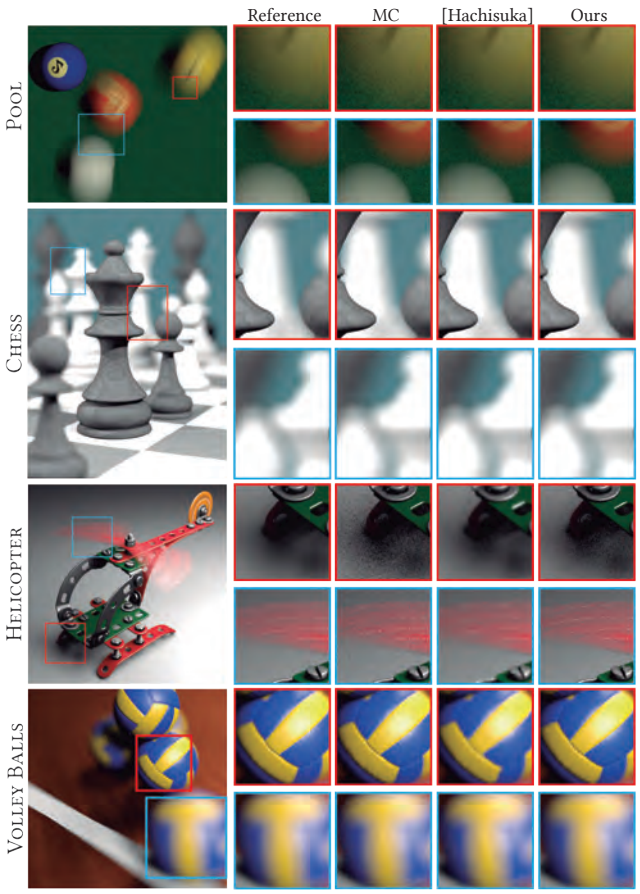

Fig. 14. Comparison between our approach (left column), Monte Carlo, and previous related work [Hachisuka et al. 2008] in four different scenes (with increasing dimensionality), at equal number of samples (64 spp). The scenes feature several distribution effects including motion blur, depth-of-field and soft shadows. In all cases, Monte Carlo produces renders with high variance, while Hachisuka et al.'s approach achieves good results in smooth domains, but tends to overblur the sharp regions of the scene. In contrast, our unbiased method outperforms previous work keeping the high contrast areas sharp.

low-dimensional projection of high-dimensional integrals allows us to have faster convergence than Monte Carlo, despite not explicitly accounting for these higher-order dimensions.

9.1 Bucketing in higher dimensionality (video)

Finally, we show that bucketing (Section 4.2) is not limited to image space (pixels), but can be generalized to higher-dimensional functions. We add the temporal dimension by rendering a video amortizing the samples of the control variate for all pixels and frames. Figure 18 shows a set of frames of a video rendered with a moving area light source of the Violin scene, while the supplementary material shows the full sequence, plus other videos from other applications including single scattering (Pumpkin) and varying distribution effects (Chess). As expected, our integration

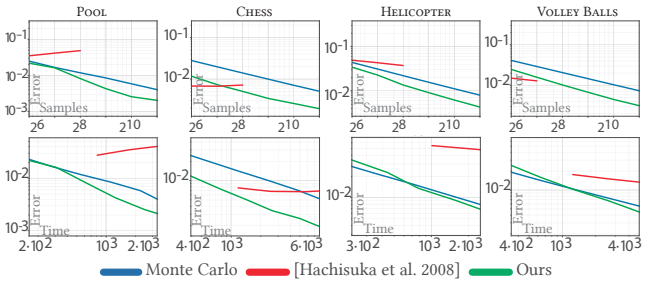

Fig. 15. Convergence for the results in Figure 14 as a function of the number of samples and core time in seconds, for pure Monte Carlo, Hachisuka et al.'s [2008], and our method. The incomplete graphs of Hachisuka et al.'s technique are due to impractical memory consumption for high sample count.

technique produces less noise than Monte Carlo, significantly reducing flickering (temporal noise) by exploiting temporal consistency.

10 DISCUSSION

In this paper we have presented a novel Monte Carlo-based integration technique that takes advantage of variance reduction through both adaptive control variates and importance sampling. We combine both by working in primary sample space, which seamlessly allows for the use of any sample distribution. We design our control variates as a multidimensional adaptive piecewise-polynomial approximation of the signal, inspired by nested quadrature rules. This allows us to accurately reconstruct low frequencies of the integral using the control variate, and to leverage Monte Carlo integration of the residual for handling high frequencies. The combination of both allows for faster convergence than previous approaches, while remaining unbiased.

We have demonstrated the applicability of our technique in four different complementary rendering applications: transmittance estimation in heterogeneous participating media, low-order scattering in homogeneous media, direct illumination computation, and rendering of distribution effects. All of them show fast convergence, accurate results, and reasonably low memory requirements. Note that our technique is generic, not tied to any specific integrand, and could be used in other problems involving numerical computations of multidimensional integrals with complex structure. We provide the source code, aiming to inspire other applications of our method.

The presented integration technique works in primary space, and is orthogonal to specific importance sampling strategies. Therefore, it can be used in combination with other works that introduce sophisticated sampling strategies [Vévoda et al. 2018; Vorba et al. 2014; West et al. 2020]. Furthermore, another avenue of future work could be to combine our work with modern denoising techniques [Bako et al. 2017; Gharbi et al. 2019], which can be used to remove the high-frequency noise coming from the integration of the residual. A preliminary test in this direction can be found in Section S.5 in the supplemental.

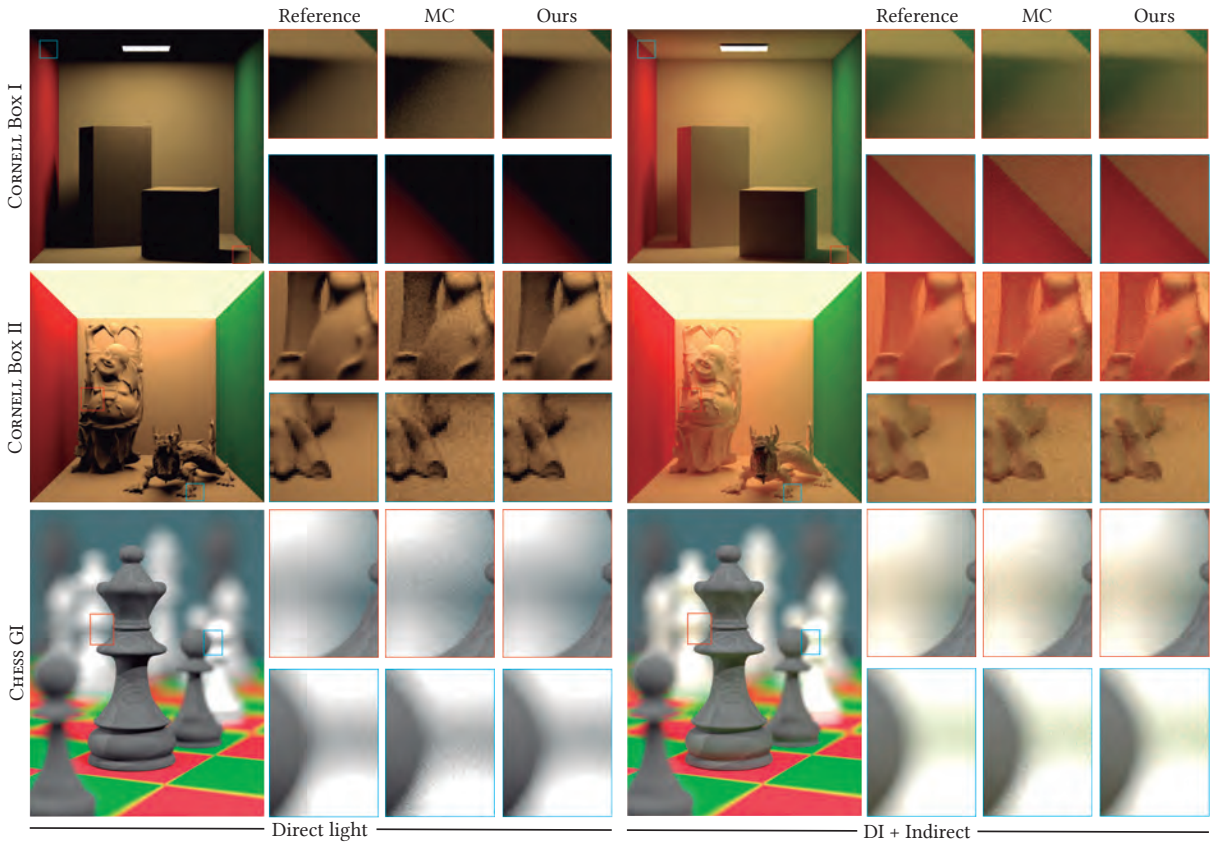

Fig. 16. Comparison between Monte Carlo and our approach (left images) when dealing with high-dimensional integrals. Note that even when our control variate is four dimensional, with our approach we can handle fifteen indirect bounces without incurring in the curse of dimensionality (results are computed using an average of 256 spp).

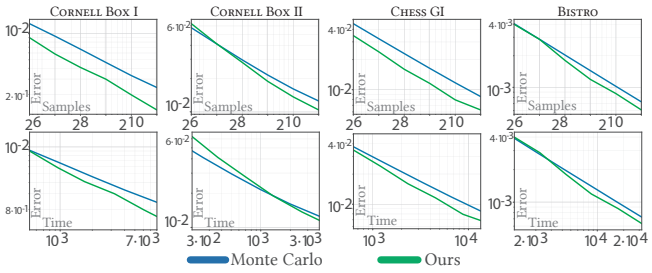

Fig. 17. Convergence curves of the scenes in Figure 16, as a function of then number of samples and core time in seconds, for both Monte Carlo and our approach.

The main limitation of our technique comes from the *curse of dimensionality*: The generation of our control variate is based on nested quadrature rules, which scale poorly when the number of dimensions is very high. While our approach allows the sampling rate to be linear with respect to iterations, it is still exponential with the dimensionality. Therefore, our control variate is fixed to a finite dimensionality (we tested up to six dimensions in the control variate in Volley Balls scene), which contrasts with the infinite dimensionality of the path integral. However, in Section 9 we have demonstrated that our technique can be applied to integrals of arbitrary dimensionality by using Monte Carlo estimates to project high-dimensional functions on our low-dimensional piecewise-polynomial control variate. As we have shown in our examples, this still allows for faster convergence than traditional Monte Carlo. Finally, our method may suffer from uneven convergence among different rectangular areas of the image, produced by uneven distribution of regions in image space, which only becomes perceivable for low sampling rates. This limitation is shared with other adaptive techniques in image space. Extended results of the convergence in those areas can be found in Section S.7 of the supplemental.

Future work. To generate the control variate, we use the Simpson-Trapezoidal nested rule. Higher order rules (Boole-Simpson) were tested but they introduced additional costs and resulted in unwanted oscillations (Runge phenomenon) as illustrated in Section S.4 of the supplemental material. More sophisticated nested rules (e.g., Clenshaw-Curtis or Gauss-Kronrod) were considered, but the regular distribution of samples of

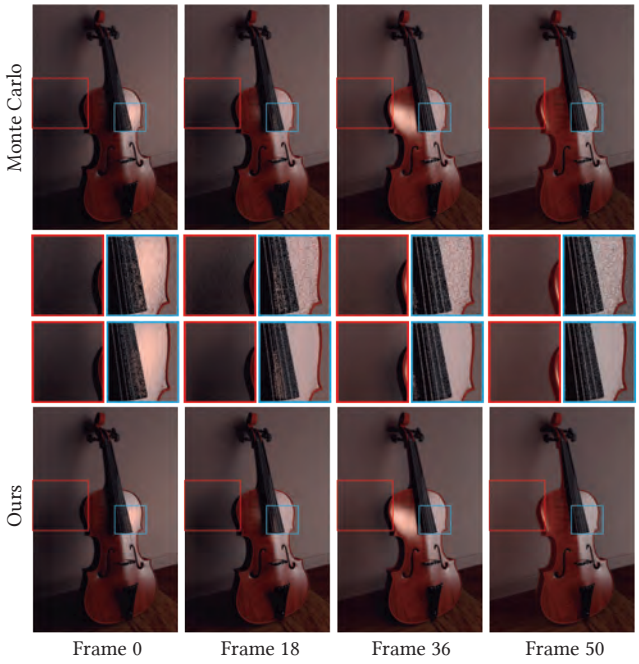

Fig. 18. Violin: Selected frames of the same sequence rendered independently with Monte Carlo versus rendered with our method. All the videos contain 60 frames with 16 spp for each frame. Note how distributing samples in time, as our adaptive stage does, helps to reduce variance in the final video. Full sequences can be seen in the supplementary video.

Newton-Cotes formulas allowed for a high rate of sample reuse. Still, experimenting with other nested rules as control variates is an interesting path for future work. In addition, exploring how to fit polynomial rules from unstructured samples could lead to an online refinement of our control variate. Finally, some of our findings might inspire further research. We have presented how to include multiple importance sampling within quadrature rules, through multiple mappings to primary space (Sections 3.2 and 7). We have also introduced a strategy to combine two different variance reduction approaches (control variates and multiple importance sampling); exploring alternative combinations is an exciting avenue for future work.

ACKNOWLEDGMENTS

We thank Ibón Guillén for comments and discussion throughout the project; Manuel Lagunas for help with figures; Diego Gutierrez for advice on the reviews; Felix Bernal for helping with the Single Scattering implementation; all the members of the Graphics & Imaging Lab that helped with proof-reading; and the reviewers for the in-depth reviews. The Pool and Chess are by Hachisuka et al.; Cornell Box, House, Classroom and MIS Test are by Benedikt Bitterli; Violin was modeled by Tahseen; Helicopter was modeled by Mond; Volley Balls models by Shri; Dragon and Budha are from the Stanford 3D Scanning Repository; Bistro was modelled by Amazon Lumberyard. Lightfields used in Figure 4 are courtesy of Jarabo et al [2014]. This project has been funded by the European Research Council (ERC) under the EU's Horizon 2020 research and innovation programme (project CHAMELEON, grant No 682080), DARPA (project REVEAL, HR0011-16-C-0025), the Spanish Ministry of Economy and Competitiveness (project TIN2016-78753-P) and the Spanish Ministry of Science and Innovation (project PID2019-105004GB-I00).

REFERENCES

Steve Bako, Thijs Vogels, Brian McWilliams, Mark Meyer, Jan Novák, Alex Harvill, Pradeep Sen, Tony Derose, and Fabrice Rousselle. 2017. Kernel-predicting Convolutional Networks for Denoising Monte Carlo Renderings. *ACM Transactions on Graphics* 36, 4 (2017), 97.

Laurent Belcour, Cyril Soler, Kartic Subr, Nicolas Holzschuch, and Fredo Durand. 2013. 5D Covariance Tracing for Efficient Defocus and Motion Blur. *ACM Transactions on Graphics* 32, 3 (2013), 31.

Laurent Belcour, Guofu Xie, Christophe Hery, Mark Meyer, Wojciech Jarosz, and Derek Nowrouzezahrai. 2018. Integrating Clipped Spherical Harmonics Expansions. *ACM Transactions on Graphics* 37, 2 (2018).

Jarle Berntsen, Terje O Espelid, and Alan Genz. 1991. An Adaptive Algorithm for the Approximate Calculation of Multiple Integrals. *ACM Trans. Math. Software* 17, 4 (1991), 437–451.

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. In *Computer Graphics Forum*, Vol. 35. Wiley Online Library, 107–117.

Jonathan Brouillat, Christian Bouville, Brad Loos, Charles Hansen, and Kadi Bouatouch. 2009. A Bayesian Monte Carlo Approach to Global Illumination. In *Computer Graphics Forum*, Vol. 28. Wiley Online Library, 2315–2329.

RL Burden and J Douglas Faires. 2005. Numerical Analysis 8th Ed. *Thomson Brooks/Cole* (2005).

Petrik Clarberg and Tomas Akenine-Möller. 2008. Exploiting Visibility Correlation in Direct Illumination. In *Computer Graphics Forum*, Vol. 27. Wiley Online Library, 1125–1136.

Robert L Cook, Thomas Porter, and Loren Carpenter. 1984. Distributed Ray Tracing. In *Proceedings of SIGGRAPH*. 137–145.

Frédo Durand, Nicolas Holzschuch, Cyril Soler, Eric Chan, and François X Sillion. 2005. A Frequency Analysis of Light Transport. *ACM Transactions on Graphics* 24, 3 (2005), 1115–1126.

Shaohua Fan, Stephen Chenney, Bo Hu, Kam-Wah Tsui, and Yu-chi Lai. 2006. Optimizing Control Variate Estimators for Rendering. In *Computer Graphics Forum*, Vol. 25. Wiley Online Library, 351–357.

Alan Genz and Aftab Ahmad Malik. 1980. Remarks on Algorithm 006: An Adaptive Algorithm for Numerical Integration over an N-dimensional Rectangular Region. *Journal of Computational and Applied Mathematics* 6, 4 (1980), 295 – 302.

Michaël Gharbi, Tzu-Mao Li, Miika Aittala, Jaakko Lehtinen, and Frédo Durand. 2019. Sample-based Monte Carlo Denoising using a Kernel-splatting Network. *ACM Transactions on Graphics* 38, 4 (2019), 1–12.

Ibón Guillén, Carlos Ureña, Alan King, Marcos Fajardo, Iliyan Georgiev, Jorge López-Moreno, and Adrian Jarabo. 2017. Area-preserving Parameterizations for Spherical Ellipses. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 179–187.

Toshiya Hachisuka, Wojciech Jarosz, Richard Peter Weistroffer, Kevin Dale, Greg Humphreys, Matthias Zwicker, and Henrik Wann Jensen. 2008. Multidimensional Adaptive Sampling and Reconstruction for Ray Tracing. In *ACM Transactions on Graphics*, Vol. 27. ACM, 33.

Stefan Heinrich. 2001. Multilevel Monte Carlo Methods. In *International Conference on Large-Scale Scientific Computing*. Springer, 58–67.

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.

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

Adrian Jarabo, Belen Masia, Adrien Bousseau, Fabio Pellacini, and Diego Gutierrez. 2014. How Do People Edit Light Fields? *ACM Transactions on Graphics (SIGGRAPH 2014)* 33, 4 (2014).

Wojciech Jarosz, Craig Donner, Matthias Zwicker, and Henrik Wann Jensen. 2008. Radiance Caching for Participating Media. *ACM Transactions on Graphics* 27, 1 (2008), 1–11.

Henrik Wann Jensen and Per H Christensen. 1998. Efficient Simulation of Light Transport in Scenes with Participating Media using Photon Maps. In *Proceedings of SIGGRAPH*. 311–320.

Jared M Johnson, Dylan Lacewell, Andrew Selle, and Wojciech Jarosz. 2011. Gaussian Quadrature for Photon Beams in Tangled. In *SIGGRAPH 2011 Talks*. ACM, 54.

James T Kajiya. 1986. The Rendering Equation. In *Computer Graphics (Proceedings of SIGGRAPH)*, Vol. 20. ACM, 143–150.

Csaba Kelemen, László Szirmay-Kalos, György Antal, and Ferenc Csonka. 2002. A Simple and Robust Mutation Strategy for the Metropolis Light Transport Algorithm. In *Computer Graphics Forum*, Vol. 21. Wiley Online Library, 531–540.

Alexander Keller. 2001. Hierarchical Monte Carlo Image Synthesis. *Mathematics and Computers in Simulation* 55, 1-3 (2001), 79–92.

Markus Kettunen, Marco Manzi, Miika Aittala, Jaakko Lehtinen, Frédo Durand, and Matthias Zwicker. 2015. Gradient-domain Path Tracing. *ACM Transactions on Graphics* 34, 4 (2015).

Ivo Kondapaneni, Petr Vévoda, Pascal Grittmann, Tomaš Skřivan, Philipp Slusallek, and Jaroslav Křivánek. 2019. Optimal Multiple Importance Sampling. *ACM Transactions on Graphics* 38, 4 (2019).

Christopher Kulla and Marcos Fajardo. 2012. Importance Sampling Techniques for Path Tracing in Participating Media. In *Computer graphics forum*, Vol. 31. Wiley Online Library, 1519–1528.

Peter Kutz, Ralf Habel, Yining Karl Li, and Jan Novák. 2017. Spectral and Decomposition Tracking for Rendering Heterogeneous Volumes. *ACM Transactions on Graphics* 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. In *Photorealistic Rendering Techniques*. Springer, 168–176.

Eric P Lafortune and Yves D Willems. 1995. A 5D Tree to Reduce the Variance of Monte Carlo Ray Tracing. In *Rendering Techniques 95*. Springer, 11–20.

Julio Marco, Adrian Jarabo, Wojciech Jarosz, and Diego Gutierrez. 2018. Second-order Occlusion-aware Volumetric Radiance Caching. *ACM Transactions on Graphics* 37, 2 (2018), 1–14.

Ricardo Marques, Christian Bouville, Mickaël Ribardière, Luís Paulo Santos, and Kadi Bouatouch. 2013. A Spherical Gaussian Framework for Bayesian Monte Carlo Rendering of Glossy Surfaces. *IEEE Transactions on Visualization and Computer Graphics* 19, 10 (2013), 1619–1632.

Soham Uday Mehta, Ravi Ramamoorthi, Mark Meyer, and Christophe Hery. 2012. Analytic Tangent Irradiance Environment Maps for Anisotropic Surfaces. In *Computer Graphics Forum*, Vol. 31. Wiley Online Library, 1501–1508.

Thomas Müller, Markus Gross, and Jan Novák. 2017. Practical Path Guiding for Efficient Light-Transport Simulation. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 91–100.

Thomas Müller, Brian Mcwilliams, Fabrice Rousselle, Markus Gross, and Jan Novák. 2019. Neural Importance Sampling. *ACM Transactions on Graphics* 38, 5 (2019).

Adolfo Muñoz. 2014. Higher Order Ray Marching. In *Computer Graphics Forum*, Vol. 33. Wiley Online Library, 167–176.

Jan Novák, Iliyan Georgiev, Johannes Hanika, and Wojciech Jarosz. 2018. Monte Carlo Methods for Volumetric Light Transport Simulation. In *Computer Graphics Forum*, Vol. 37. Wiley Online Library, 551–576.

Jan Novák, Derek Nowrouzezahrai, Carsten Dachsbacher, and Wojciech Jarosz. 2012. Virtual Ray Lights for Rendering Scenes with Participating Media. *ACM Transactions on Graphics* 31, 4 (2012), 60:1–60:11.

Jan Novák, Andrew Selle, and Wojciech Jarosz. 2014. Residual Ratio Tracking for Estimating Attenuation in Participating Media. *ACM Transactions on Graphics* 33, 6 (2014).

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.*

Ken Perlin and Eric M Hoffert. 1989. Hypertexture. In *Computer Graphics (Proceedings of SIGGRAPH)*, Vol. 23. ACM, 253–262.

William H Press, Saul A Teukolsky, William T Vetterling, and Brian P Flannery. 2007. *Numerical Recipes 3rd Edition: The Art of Scientific Computing.* Cambridge university press.

Ravi Ramamoorthi and Pat Hanrahan. 2001. An Efficient Representation for Irradiance Environment Maps. In *Proceedings of SIGGRAPH.* 497–500.

Ravi Ramamoorthi and Pat Hanrahan. 2002. Frequency Space Environment Map Rendering. *ACM Transactions on Graphics* 21, 3 (2002), 517526.

Ravi Ramamoorthi, Dhruv Mahajan, and Peter Belhumeur. 2007. A First-Order Analysis of Lighting, Shading, and Shadows. *ACM Transactions on Graphics* 26, 1 (2007), 2.

Christian P Robert and George Casella. 2004. Monte Carlo Statistical Methods. (2004).

Fabrice Rousselle, Wojciech Jarosz, and Jan Novák. 2016. Image-space Control Variates for Rendering. *ACM Transactions on Graphics* 35, 6 (2016), 169.

Fabrice Rousselle, Claude Knaus, and Matthias Zwicker. 2012. Adaptive Rendering with Non-local Means Filtering. *ACM Transactions on Graphics* 31, 6 (2012), 195.

Martin Šik and Jaroslav Krivanek. 2018. Survey of Markov Chain Monte Carlo Methods in Light Transport Simulation. *IEEE Transactions on Visualization and Computer Graphics* (2018).

Arthur H Stroud and Don Secrest. 1966. Gaussian Quadrature Formulas. (1966).

Carlos Ureña, Marcos Fajardo, and Alan King. 2013. An Area-preserving Parametrization for Spherical Rectangles. In *Computer Graphics Forum*, Vol. 32. Wiley Online Library, 59–66.

Eric Veach. 1997. *Robust Monte Carlo Methods for Light Transport Simulation.* Vol. 1610. Stanford University PhD thesis.

Eric Veach and Leonidas J Guibas. 1995. Optimally Combining Sampling Techniques for Monte Carlo Rendering. In *Proceedings of SIGGRAPH' 95.* ACM, 419–428.

Petr Vévoda, Ivo Kondapaneni, and Jaroslav Křivánek. 2018. Bayesian Online Regression for Adaptive Direct Illumination Sampling. *ACM Transactions on Graphics* 37, 4 (2018), 125.

Jiří Vorba, Ondřej Karlík, Martin Šik, Tobias Ritschel, and Jaroslav Křivánek. 2014. On-line Learning of Parametric Mixture Models for Light Transport Simulation. *ACM Transactions on Graphics* 33, 4 (2014), 101.

Gregory J Ward, Francis M Rubinstein, and Robert D Clear. 1988. A Ray Tracing Solution for Diffuse Interreflection. In *Proceedings of SIGGRAPH.*

Rex West, Iliyan Georgiev, Adrien Gruson, and Toshiya Hachisuka. 2020. Continuous Multiple Importance Sampling. *ACM Transactions on Graphics* 39, 4 (2020).

E. Woodcock, T. Murphi, P. Hemmings, and S. Longworth. 1965. Techniques Used in the GEM Code for Monte Carlo Neutronics Calculations in Reactors and Other Systems of Complex Geometry. In *Proceeding Conference Applications of Computing Methods to Reactors, ANL-7050.*

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.

Eric Ziegel. 1987. Numerical Recipes: The Art of Scientific Computing.

Matthias Zwicker, Wojciech Jarosz, Jaakko Lehtinen, Bochang Moon, Ravi Ramamoorthi, Fabrice Rousselle, Pradeep Sen, Cyril Soler, and S-E Yoon. 2015. Recent Advances in Adaptive Sampling and Reconstruction for Monte Carlo Rendering. In *Computer Graphics Forum*, Vol. 34. 667–681.