

Adaptive Multiple Control Variates for Many-Light Rendering

Xiaofeng Xu[✉] and Lu Wang[†]

Shandong University, China

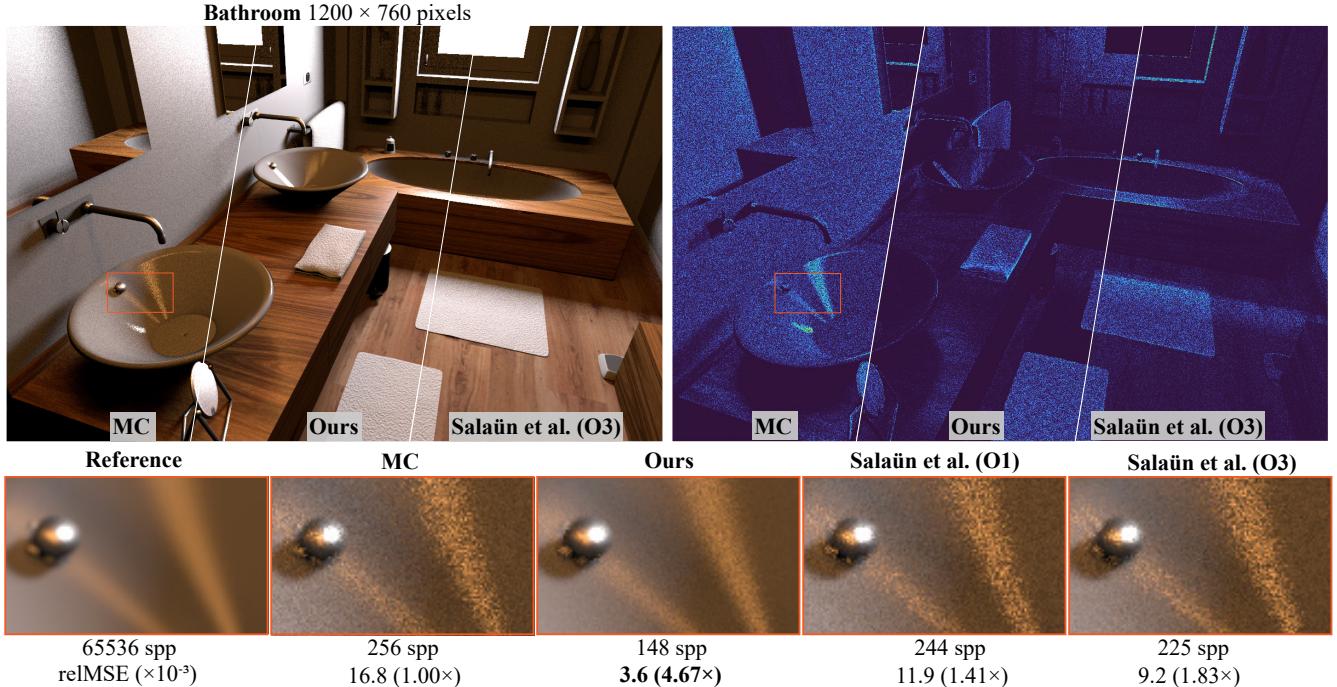


Figure 1: Equal-time comparison among our method with polynomials of order 1, conventional Monte Carlo (MC) integration, and the regression-based MC integration of Salaün et al. [SGH* 22] with polynomials of order 1 (O1) and 3 (O3) on the bathroom scene. We report the number of samples per pixel (SPP) computed by each method and the relative mean squared error (relMSE) compared to the reference image, with speed-up values relative to MC in parentheses. Our method exhibits the lowest variance both visually and quantitatively. For a more detailed comparison, please refer to the HTML viewer provided in the supplemental.

Abstract

Monte Carlo integration estimates the path integral in light transport by randomly sampling light paths and averaging their contributions. However, in scenes with many lights, the resulting estimates suffer from noise and slow convergence due to high-frequency discontinuities introduced by complex light visibility, scattering functions, and emissive properties. To mitigate these challenges, control variates have been employed to approximate the integrand and reduce variance. While previous approaches have shown promise in direct illumination application, they struggle to efficiently handle the discontinuities inherent in many-light environments, especially when relying on a single control variate. In this work, we introduce an adaptive method that generates multiple control variates tailored to the spatial distribution and number of lights in the scene. Drawing inspiration from hierarchical light clustering methods like Lightcuts, our approach dynamically determines the number of control variates. We validate our method on the direct illumination problem in scenes with many lights, demonstrating that our adaptive multiple control variates not only outperform single control variate strategy but also achieve a modest improvement over current state-of-the-art many-light sampling techniques.

CCS Concepts

- Computing methodologies → Rendering;

[†] luwang_hcivr@sdu.edu.cn

1. Introduction

The rendering equation [Kaj86] and the path integral formulation [Vea97] provide the foundational mathematical framework for physically based rendering, where light transport is formulated as an integration over all possible light paths between the camera and lights. In scenes with complex lighting distributions, especially those involving numerous lights, the integrand that determines the contribution of a point usually exhibits high-frequency discontinuities because it depends on the visibility of each light to the point, the distribution of the scattering function at the point, and the light's power and emissive characteristics. These discontinuities lead to intricate integrals that challenge traditional Monte Carlo (MC) integration, which suffer from slow and uneven convergence even with variance reduction techniques like (multiple) importance sampling [Vea97], low-discrepancy sampling [Owe13], Metropolis sampling [MRR*53], and some other adaptive sampling strategies [PBPP11, LPG13].

Recent studies [CJMn21, SGH*22] have explored the use of control variates (CVs) to accelerate the convergence of MC integration, and both approaches have achieved promising results in direct illumination application. To summarize, Crespo et al. [CJMn21] employed *nested quadrature rules* to partition the primary sample space into adaptively distributed subregions, generating a piecewise polynomial control variate. However, their heuristic method for defining these subregions may miss important regions of the integrand, leading to suboptimal approximations. Salaün et al. [SGH*22] proposed a *regression-based approach* to construct polynomial (or other basis functions) control variates for each pixel and provided a rigorous theoretical derivation to validate its effectiveness. Nevertheless, polynomial basis (as well as other basis functions) struggle to approximate functions with numerous discontinuities. As illustrated in Figure 2, their method becomes less effective as the number of lights increases, and eventually performing similarly to the standard MC estimator.

To address the limitations of the above CVs methods in many-light rendering, we have a key insight that the discontinuities in the integrand are intrinsically tied to the number and spatial distribution of lights in the scene – properties that can be efficiently captured through hierarchical light clustering similar to Lightcuts [WFA*05] approach. Building on this insight, we propose a novel method that adaptively generates multiple control variates guided by Lightcuts-based light clustering, where the number of CVs are determined by the distribution of lights in the scene. We validate our approach on the direct illumination problem with many lights. The results show that our method outperforms the single control variate approach and yields a modest improvement over current state-of-the-art many-light sampling methods. In summary, our work makes the following contributions:

- We introduce multiple control variates into MC integration to reduce variance.
- We propose an adaptive method for generating multiple control variates for each pixel based on Lightcuts.
- We demonstrate the effectiveness of our approach in solving the direct illumination problem for scenes with many lights.

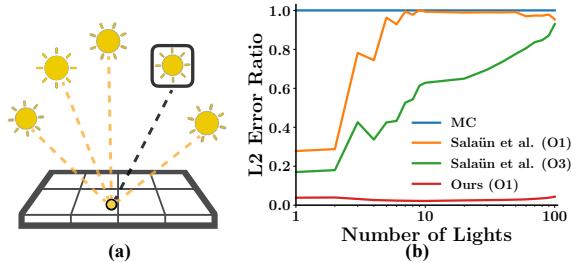


Figure 2: Comparison of integration error ratio relative to MC as a function of the number of lights in the scene, using a fixed budget of 1024 samples. Salaün et al. [SGH*22] use 1st and 3rd order polynomials, while our method uses only 1st order polynomial. (a) Schematic depiction of a diffuse surface illuminated by many small-area lights. As the number of lights increases, the discontinuities in the integrand also increase due to the random selection of lights. (b) Plot of the integration error ratio. When the number of lights is small, the method by Salaün et al. significantly outperforms MC. However, as the number of lights increases and the integrand becomes increasingly discontinuous, their advantage diminishes and their performance converges to that of the MC estimator. In contrast, our method consistently achieves substantial error reduction. Note that for this simple scene, we use one control variate per light without Lightcuts. Although the number of control variates grows with light count, the cost remains nearly constant.

2. Related Work

2.1. Control variates in rendering

Control variates [Loh95] are techniques for variance reduction that work by using MC integration to compute the difference between a target integrand f and a related function h (the control variate). This difference is then added to the analytical integral of h to approximate the integral of f .

In recent years, several studies have explored the application of control variates in solving integral problems in rendering. Lafortune and Willems proposed the use of a constant ambient term [LW94] along with a directional, piecewise approximation of indirect radiance [LW95] as control variates for computing indirect diffuse illumination. Szécsi et al. [SSSK04] enhanced variance reduction by combining correlated sampling and importance sampling in the computation of direct light source and environment mapping. Clarberg and Akenine-Möller [CAM08] further leveraged the spatiotemporal correlation of the visibility function by using a visibility cache as control variates for direct illumination in environment maps. Vévoda et al. [VKK18] employed the estimated contributions from light clusters as control variates to reduce variance in the computation of direct illumination in scenes with many lights. Pantaleoni [Pan20] proposed a path tracing-based progressive hierarchical solver that approximates the light field information in a scene by constructing finite element structures at each vertex of the path. These approximations are used both to guide sampling and as control variates for computing the outgoing radiance. Müller et al. [MRKN20] proposed a neural network-based parametric control variate, utilizing the learning capability of neural networks to optimize the construction of control variates. Hua et al. [HGS23] revisited the theoretical formulations of optimal control variate

[FCH^{*}06] and optimal multiple importance sampling [KVG^{*}19]. Then they devised local, spatially shared, low-dimensional control variates that are practical for common rendering scenarios. Rousselle et al. [RJN16] applied control variates in the contexts of re-rendering and gradient-domain rendering. In addition, Xu et al. [XLG^{*}24] proposed a residual path integral formulation that incorporates control variates and applied it to the re-rendering of object movement. Novak et al. [NSJ14] introduced residual ratio tracking for estimating attenuation, demonstrating that control variates can effectively reduce variance when evaluating transmittance in participating media [KHLN17, GMH^{*}19, KDPN21].

Crespo et al. [CJMn21] addressed the challenge of handling both low-frequency regions and high-frequency details in multi-dimensional integrals by proposing an adaptive piecewise polynomial control variates technique in the primary sample space. Their method uses *nested quadrature rules* to partition the space into adaptively distributed subregions and generate polynomial approximations. However, their subregion definition may sometimes miss important parts of the integrand, leading to inaccurate integral approximations. Salaün et al. [SGH^{*}22] performed least-squares regression on samples in the primary sample space to fit a polynomial function (or other basis functions) approximating the contribution of each sample to pixel radiance values. We observed that their method struggles with integrals featuring many discontinuities, as polynomial basis (as well as other basis functions) are not well suited to approximate such functions. In contrast, our approach, which employs multiple control variates, effectively addresses these limitations. More recently, we note that the Clustered Control Variates proposed by Zhu et al. [ZHB^{*}24] for rendering cloth appearance model shares similarity with our approach. They used K-means to segment texture map into clusters with similar visibility and then calculated the average visibility for each cluster as a control variate. Additionally, their method used a fixed number of control variates with preset weights, in contrast to our adaptive strategy.

2.2. Direct illumination computation with many lights

Direct illumination computation with next event estimation is a fundamental component of modern path tracing renderers [CEK18]. However, directly sampling every light in a scene becomes computationally infeasible as the number of lights exceeds a few dozen. To address this challenge, various strategies have been developed to manage the complexity of lighting calculations efficiently.

Earlier works focused on reducing the number of visibility tests [KJ94, War94]. Shirley et al. [SWZ96] introduced an octree-based approach to classify lights as either important or unimportant based on their expected contribution. Paquette et al. [PPD98] and Walter et al. [WFA^{*}05] proposed clustering lights into hierarchical structures and using adaptive tree cuts to approximate direct illumination. These methods improve scalability at the cost of some bias. Many subsequent approaches [WABG06, WKB12] have refined these ideas. Estevez and Kulla [CEK18] introduced improved criteria for constructing light hierarchies and adaptive tree-splitting strategy for traversal. Liu et al. [LXY19] further enhanced efficiency by incorporating BRDF evaluation at shading points during light tree traversal.

Other approaches focused on adaptively constructing probability density functions (PDFs) for direct light sampling. Donikian et al. [DWB^{*}06] introduced aggregate PDFs over fixed image blocks for progressive rendering, incrementally updating them across multiple passes until convergence. Vévoda et al. [VVK18] incorporated visibility information into light sampling by clustering shading points and applying Bayesian online regression to estimate the optimal light selection probabilities. Rath et al. [RGH^{*}20] extended this method by marginalizing the BSDF, improving performance for scenes with glossy surfaces and small lights. Wang et al. [WWLC21] proposed a progressive light sampling method that adaptively adjust both light clustering and sampling distributions based on collected samples during rendering. All these methods require a prebuilt hierarchical light clustering structure that leverages spatial coherence by grouping nearby lights with similar visibility, direction, and distance from the shading points. Similarly, in scenes with a large number of lights, we construct such a hierarchical structure to efficiently organize them

Additionally, the problem of rendering with many lights has been explored in the context of virtual point light (VPL) rendering [DKH^{*}14], as well as in various real-time [BWP^{*}20] and interactive [OA11, TH16] direct illumination algorithms. However, our work focuses specifically on direct illumination in offline rendering.

3. Theoretical Background

Direct illumination estimation with many lights involves evaluating integrals (as will be detailed in Section 5.1). In this section, we begin by introducing the general integration problems.

3.1. Monte Carlo integration

Many problems in light transport simulation reduce to computing integrals of the form

$$F = \int_{\Omega} f(x) dx. \quad (1)$$

where $f(x)$ is the measurement contribution function of a light transport path x [Vea97], and Ω is the path space. Since this integral cannot be solved analytically, MC integration estimates it by

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

where samples x_i are drawn from a probability density function (PDF) $p(x)$ and n is the number of samples.

Primary sample space. Following [CJMn21, SGH^{*}22], we map the path space Ω to the unit hypercube U via the transformation $x = \Phi(u)$. This primary sample space formulation is used throughout the paper.

3.2. Control variates

The control variates (CVs) method reduces variance by introducing a correlated function $\hat{h}(u)$ with a known integral $H = \int_U \hat{h}(u) du$. The original integral can be rewritten as

$$F = \int_U f(u) du = \int_U f(u) - \alpha \hat{h}(u) du + \alpha H, \quad (3)$$

where α controls the contribution of the control variate. The MC estimator using control variates for n samples becomes

$$\langle F \rangle_*^n = \frac{1}{n} \sum_{i=1}^n \hat{f}(u_i) - \alpha \hat{h}(u_i) + \alpha H. \quad (4)$$

The optimal $\alpha = \rho[\hat{f}(u), \hat{h}(u)] / \text{Var}[\hat{h}(u)]$, where $\rho[\hat{f}(u), \hat{h}(u)]$ is the correlation coefficient between $\hat{f}(u)$ and $\hat{h}(u)$. With this choice, the variance of the CVs estimator becomes

$$\text{Var}[\langle F \rangle_*] = \text{Var}[\langle F \rangle] \left(1 - \rho^2[\hat{f}(u), \hat{h}(u)] \right). \quad (5)$$

3.3. Regression-based Monte Carlo integration

Salaün et al. [SGH*22] proposed regression-based MC integration, which employs least-squares regression to obtain $\hat{h}(u)$. The upper part of Figure 3 shows their method applied to a one-dimensional function. In the following, we summarize their technique.

3.3.1. Least-squares regression of $\hat{h}(u)$

Salaün et al. [SGH*22] modeled $\hat{h}(u, \theta)$ as a parametric function with parameters $\theta = (c_0, c_1 \dots, c_M)$, and defined the *residual* for a given θ as

$$R(\theta) = \int_U (\hat{f}(u) - \hat{h}(u, \theta))^2 du. \quad (6)$$

The goal is to find the optimal parameters that minimize $R(\theta)$. To solve this integral, they used n random samples u_i to estimate it.

$$R(\theta) \approx \frac{1}{n} \sum_{i=1}^n (\hat{f}(u_i) - \hat{h}(u_i, \theta))^2 = \langle R \rangle(\theta). \quad (7)$$

Thus, the solution can be obtained by applying least-squares regression of $\hat{h}(u, \theta)$ to n pairs of $(u_i, \hat{f}(u_i))$.

3.3.2. Polynomial-based estimators

Among several basis function choices, Salaün et al. [SGH*22] found polynomial basis functions to be effective in many cases. A polynomial function $\hat{h}(u)$ can be written as

$$\hat{h}(u) = \sum_{j=0}^M c_j \phi_j(u), \quad (8)$$

where $\{\phi_j(u)\}$ are monomials and M is the polynomial order. The integral H can be computed analytically once the coefficients c_j are determined. Salaün et al. [SGH*22] proposed two regression methods, a *direct matrix* method that solves a linear system, and a *stochastic gradient descent* method that iteratively updates the parameter.

Discussion. Regression-based MC integration [SGH*22] provides a general framework for constructing control variates, with theoretical guarantees that the resulting estimators achieve variance equal to or lower than standard MC integration. However, since the regression is only approximate for finite sample sizes, the theoretical variance reduction is not always achieved in practice. As demonstrated in our experiments (Figure 6 and Figure 7), when the scene contains many lights, the regression-based method may perform worse compared to standard MC integration.

4. Multiple Control Variates for Monte Carlo Integration

Building upon the regression-based MC integration framework [SGH*22], we propose a theoretical extension that introduces multiple control variates instead of relying on a single one. This extension forms the foundation of our approach and aims to further reduce variance while improving robustness. In this section, we explain how multiple control variates can be constructed and discuss the theoretical advantages of using them. The second row of Figure 3 illustrates our method applied to a one-dimensional function. We begin by describing how to obtain the multiple control variates.

4.1. Stratification

Stratification divides the integration domain Ω into k disjoint sub-domains $\Omega_1, \dots, \Omega_k$ such that $\bigcup_{i=1}^k \Omega_i = \Omega$. Each subdomain Ω_i which called a *stratum*, corresponds to a specific discrete selection, such as selecting a particular light in a scene with many lights.

For each stratum Ω_i , we define a transformed integrand \hat{f}_i via a mapping to the unit hypercube U , and draw n_i samples to compute the local estimate as

$$\langle F_i \rangle^{n_i} = \frac{1}{n_i} \sum_{j=1}^{n_i} \hat{f}_i(u_j). \quad (9)$$

Then the overall stratified estimator is $\langle F' \rangle = \sum_{i=1}^k v_i \langle F_i \rangle^{n_i}$, where v_i denotes the fractional volume of Ω_i (i.e., $v_i \in (0, 1]$). The variance of this estimator is

$$\text{Var}[\langle F' \rangle] = \text{Var} \left[\sum_{i=1}^k v_i \langle F_i \rangle^{n_i} \right] = \sum_{i=1}^k \frac{v_i^2 \sigma_i^2}{n_i}. \quad (10)$$

where σ_i^2 denotes the variance of \hat{f}_i within Ω_i . Assuming the number of samples in each stratum is proportional to its volume, i.e., $n_i = v_i n$ with n is the total number of samples, the variance simplifies to

$$\text{Var}[\langle F' \rangle] = \frac{1}{n} \sum_{i=1}^k v_i \sigma_i^2. \quad (11)$$

For comparison, the variance of the unstratified estimator is

$$\text{Var}[\langle F \rangle] = \frac{1}{n} \left[\sum_{i=1}^k v_i \sigma_i^2 + \sum_{i=1}^k v_i (\mu_i - I)^2 \right], \quad (12)$$

where μ_i is the mean value of \hat{f}_i in Ω_i , and I is the global mean of \hat{f} over Ω . This decomposition shows that stratification never increases variance, i.e., $\text{Var}[\langle F' \rangle] \leq \text{Var}[\langle F \rangle]$. While stratification is a well-established variance reduction technique in MC integration (see, e.g., Veach's thesis [Vea97]), our main contribution lies in how we incorporate multiple control variates within these strata.

4.2. Multiple control variates

Next, we determine a control variate $\hat{h}_i(u)$ for each stratum and compute its analytic integral $H_i = \int_U \hat{h}_i(u) du$. For each stratum, we decompose the original integral as

$$F_i = \int_U \hat{f}_i(u) - \alpha_i \hat{h}_i(u) du + \alpha_i H_i. \quad (13)$$

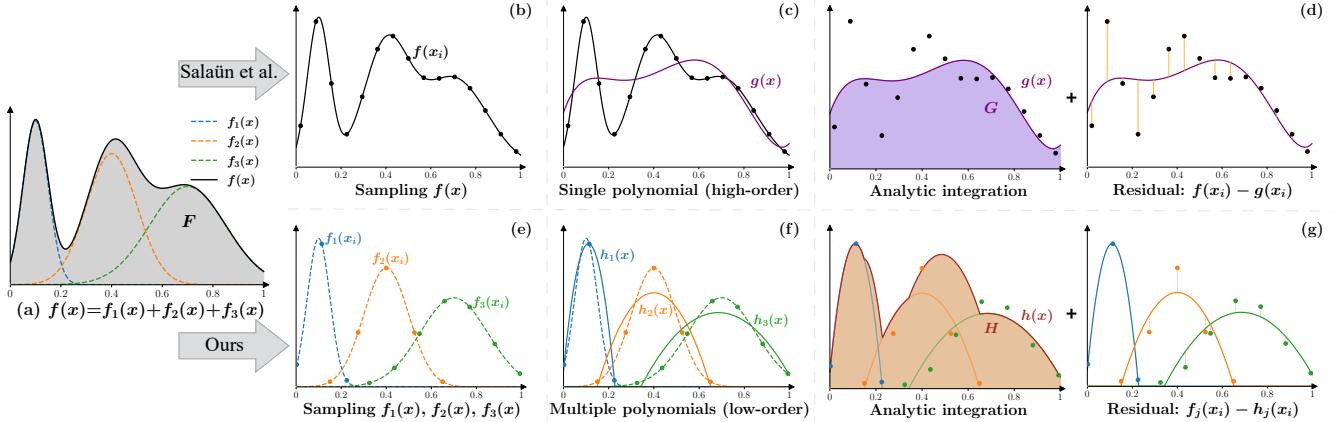


Figure 3: The figure compares our method with that of Salaiün et al. [SGH*22] for a one-dimensional integral. Given an integrand $f(x)$ (a), Salaiün et al.’s method first (b) samples $f(x)$ and then (c) fits a polynomial function to these samples. Their estimator subsequently (d) leverages control variates by adding the analytical integral of the model function to the MC estimate of the residual. In contrast, our method first (e) partitions $f(x)$ into multiple sub-integrands $f_1(x)$, $f_2(x)$, $f_3(x)$ via stratification (Section 4.1) and samples each separately with varying sample counts. We then (f) fit a polynomial to each set of samples and (g) combine their analytical integrals with MC estimates of their respective residuals. Since each sub-function is simpler and modeled with fewer samples, our method achieves greater modeling power without incurring additional regression overhead.

The corresponding CVs estimate is then given by

$$\langle F_i \rangle_*^{n_i} = \frac{1}{n_i} \sum_{i=1}^{n_i} \hat{f}_i(u_i) - \alpha_i \hat{h}_i(u_i) + \alpha_i H_i, \quad (14)$$

with $\alpha_i = \rho[\hat{f}_i(u), \hat{h}_i(u)] / \text{Var}[\hat{h}_i(u)]$, where $\rho[\hat{f}_i(u), \hat{h}_i(u)]$ is the correlation coefficient between $\hat{f}_i(u)$ and $\hat{h}_i(u)$. Consequently, the variance after applying control variates in Ω_i is

$$\text{Var}[\langle F_i \rangle_*^{n_i}] = \text{Var}[\langle F_i \rangle_*^{n_i}] \left(1 - \rho^2[\hat{f}_i(u), \hat{h}_i(u)]\right). \quad (15)$$

Thus, in each stratum, the use of control variate reduces the variance by a factor of $(1 - \rho^2[\hat{f}_i(u), \hat{h}_i(u)])$. Similarly, the overall estimator across the entire domain Ω is expressed as

$$\langle F' \rangle_* = \sum_{i=1}^k v_i \langle F_i \rangle_*^{n_i}. \quad (16)$$

4.3. Variance reduction with multiple control variates

We now compare the variance of the overall estimate using multiple control variates with that of the standard MC estimator. According to Equation 16, the overall variance is given by

$$\begin{aligned} \text{Var}[\langle F' \rangle_*] &= \sum_{i=1}^k v_i^2 \text{Var}[\langle F_i \rangle_*^{n_i}] \\ &= \sum_{i=1}^k v_i^2 \text{Var}[\langle F_i \rangle_*^{n_i}] \left(1 - \rho^2[\hat{f}_i(u), \hat{h}_i(u)]\right). \end{aligned} \quad (17)$$

Since $\rho^2[\hat{f}_i(u), \hat{h}_i(u)]$ is non-negative and less than or equal to 1, it follows that

$$\text{Var}[\langle F' \rangle_*] \leq \text{Var}[\langle F' \rangle] \quad (18)$$

This expression demonstrates that employing multiple control variates reduces variance compared to not using control variates. Com-

bined with the stratification result from Section 4.1, we obtain

$$\text{Var}[\langle F' \rangle_*] \leq \text{Var}[\langle F \rangle]. \quad (19)$$

This confirms that using multiple control variates will reduce the overall variance of the MC estimator, yielding a more accurate estimate of the target function $\hat{f}(u)$. Intuitively, using several control variates—each designed to capture distinct features of the integrand—provides greater representational capacity than relying on a single control variate. Consequently, this approach has the potential to achieve superior variance reduction compared to the single control variate method. To facilitate our discussion, we define a combined control variate $\hat{g}(u)$ as the weighted sum of individual control variates $\hat{h}_i(u)$,

$$\hat{g}(u) = \sum_{i=1}^k v_i \hat{h}_i(u), \quad (20)$$

where $v_i \geq 0$ and $\sum_{i=1}^k v_i = 1$ are the weights associated with each stratum. In practice, the effectiveness of this combined control variate hinges on two critical factors, the correlation between each control variate $\hat{h}_i(u)$ and its corresponding target function $\hat{f}_i(u)$ within each stratum and the choice of weights v_i used in constructing $\hat{g}(u)$. When the individual control variates $\hat{h}_i(u)$ exhibit strong correlation with their respective integrands, and the weights v_i are carefully optimized to reflect the relative importance of each stratum, the multiple control variates method can substantially outperform the single control variate approach. In such cases, the combined surrogate function $\hat{g}(u)$ yields a closer approximation to the target function and achieves more effective variance reduction. As illustrated in Figure 3, the integration of effective stratification and adaptive construction of control variates enables the multiple control variates method to deliver significantly higher estimation accuracy than its single-control counterpart.

5. Direct Illumination with Many Lights based on Multiple Control Variates

In this section, we present our method for estimating direct illumination with many lights using multiple control variates. By decomposing the three-dimensional integration problem into discrete light selection and two-dimensional sampling over the light surface, we construct separate control variates for the light selection dimension. We first formulate the problem, then outline the overall solution, and finally provide implementation details.

5.1. Direct illumination estimation with many lights

Given a shading point \mathbf{x} with viewing direction ω_o and a set of lights \mathcal{L} , the outgoing radiance $L_o(\mathbf{x}, \omega_o)$ is computed by summing the contributions F from all lights:

$$L_o(\mathbf{x}, \omega_o) = \sum_{l \in \mathcal{L}} F(\mathbf{x}, \omega_o, l). \quad (21)$$

If l is a point light, then F is the product of light intensity, material (i.e., the Bidirectional Scattering Distribution Function), visibility, and a geometry term. In contrast, for an area light, F is obtained via integration over the light's surface. When the light set \mathcal{L} is large, evaluating the complete discrete sum for every shading point become impractical. Therefore, MC integration is used to estimate the contributions of all lights.

To improve the efficiency of importance sampling for this discrete sums, previous studies often cluster lights into more manageable, non-overlapping subsets $\mathcal{C}(\mathbf{x})$:

$$L(\mathbf{x}) = \sum_{c \in \mathcal{C}(\mathbf{x})} \sum_{l \in c} F(\mathbf{x}, l) = \sum_{c \in \mathcal{C}(\mathbf{x})} F_c(\mathbf{x}), \quad (22)$$

where $F_c(\mathbf{x}) = \sum_{l \in c} F(\mathbf{x}, l)$ (omitting ω_o for brevity). This double sum is then estimated via MC sampling as

$$L(\mathbf{x}) \approx \langle L(\mathbf{x}) \rangle = \frac{1}{n} \sum_{i=1}^n \frac{F(\mathbf{x}, l_i)}{p(l_i | \mathbf{x}, c_i) p(c_i | \mathbf{x})}, \quad (23)$$

where $p(c_i | \mathbf{x})$ is the probability of selecting cluster c_i for the shading point \mathbf{x} , and $p(l_i | \mathbf{x}, c_i)$ is the probability of selecting light $l_i \in c_i$ given the shading point and cluster (or probability density if l_i is an area light). This formulation corresponds to the stratification step in our multiple control variates framework (Section 4.1), where the light cluster c_i is first sampled.

5.2. Solution overview

Our goal is to reduce variance in the many-light direct illumination problem by leveraging multiple control variates. To this end, we construct a polynomial control variate for each light or light cluster to mitigate the discontinuities in the integrand caused by numerous lights. Our method processes each pixel independently and consists of the following steps:

- **Stratification & Sampling** (Fig.3.e):

- 1) For scenes with a small number of lights, we construct one control variate per light and use uniform light sampling to generate samples. Each control variate $\hat{h}_i(x)$ gathers samples

from its corresponding light, and the number of control variates equals the number of lights, with each control variate weighted by the reciprocal of the total number of lights.

- 2) For scenes with many lights, we first cluster the lights into a BVH tree (detailed in Section 5.3.1) and then select representative points (detailed in Section 5.3.2) for each pixel to generate light cuts (a set of light clusters). This identifies the light clusters (or individual lights) contributing to that pixel. For each light cluster (or light), we construct a control variate and generate samples using a Lightcuts-based light sampling method [CEK18] (detailed in Section 5.3.1). In this case, the number of control variates per pixel varies with the number of light clusters, and weights are computed based on the normalized importance of each cluster at the representative points.

- **Solving** (Fig.3.f): We employ the gradient descent method (Section 3.3.2) to solve for the coefficients of a polynomial function for each control variate, subsequently calculating the corresponding α_i and the analytic integral H_i .
- **Estimating** (Fig.3.g): We combine the analytical integrals of the polynomial control variates with the MC estimates of the differences using Equation 16, ensuring that each control variate is assigned the correct weight v_i . At this stage, the weight values for the light cluster case have not yet been determined (and will be derived in Section 5.3.2).

5.3. Implementation

For scenes with many lights, generating a separate control variate for each individual light is impractical. Therefore, we adopt a Lightcuts-based approach to adaptively determine the appropriate number of control variates. Our implementation consists of two key components: *Lightcuts-based light sampling* and *adaptive construction of multiple control variates*.

5.3.1. The Lightcuts-based light sampling

To determine the number and weights of the control variates, as well as to generate the corresponding samples for each control variate, we adopt a Lightcuts-based light sampling method. This method involves the following steps:

Light hierarchy construction. We build a hierarchical light tree by recursively splitting the set of all lights, starting from a root node, to form a structure similar to a Bounding Volume Hierarchy (BVH). We use the *Surface Area Orientation Heuristic* (SAOH) metric [CEK18] for node splitting. This metric extends the traditional *Surface Area Heuristic* (SAH) by incorporating bounding cone orientation and emitted energy, thereby grouping lights effectively based on spatial proximity, directionality, and intensity. Each node in the hierarchy stores a spatial bounding box centered at C_c , an orientation cone that encapsulates the surface normals of the contained lights, and the total energy E of these lights (Figure 4 (b)).

Light cut construction. After constructing the light tree, we traverse it from top to bottom to generate a light cut for each shading point during path tracing. As illustrated in Figure 4(a), the number of nodes in the light cut determines the number of control variates

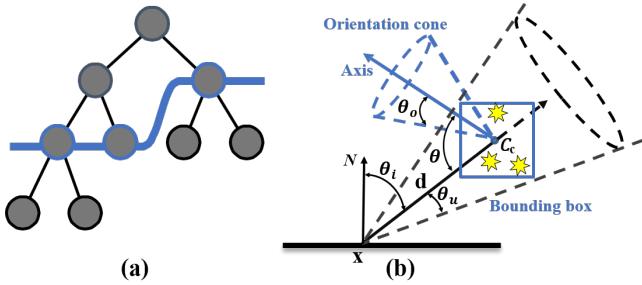


Figure 4: (a) A light tree and an example light cut. (b) Calculating the cluster importance from a shading point \mathbf{x} using the cluster bounds.

to be used. This process is analogous to the cut selection phase of the Lightcuts algorithm [WFA*05].

Light cluster sampling. Once a light cut is determined, we sample a light cluster based on its computed importance, which also informs the weighting of the corresponding control variates. This importance is derived from the cluster's spatial and orientation bounds relative to the shading point, factoring in geometric decay from inverse square distance and the cosine effect from the orientation bounds [CEK18]. As illustrated in Figure 4 (b), θ_u is the cone angle encompassing the bounding box from the shading point \mathbf{x} , θ_i is the incident angle from \mathbf{x} to the cluster center, and d is the distance of this segment. The importance of a light cluster is defined as

$$I_C(\mathbf{x}) = \frac{f_{\mathbf{x}} |\cos \theta'_i| E \cos (\theta')}{d^2}, \quad (24)$$

where $f_{\mathbf{x}}$ is an arbitrary approximation of the BSDF at \mathbf{x} , $\theta'_i = \max\{\theta_i - \theta_u, 0\}$ is the minimum incidence angle and $\theta' = \max\{\theta - \theta_o - \theta_u, 0\}$ is the minimum angle between the emitter's normal and the direction toward \mathbf{x} . We recursively sample light clusters with probabilities proportional to their importance until a leaf node containing a single light is reached.

Light sampling. After a specific light is selected, we sample its surface uniformly following standard techniques [PJH16, SWZ96] to obtain a light sample.

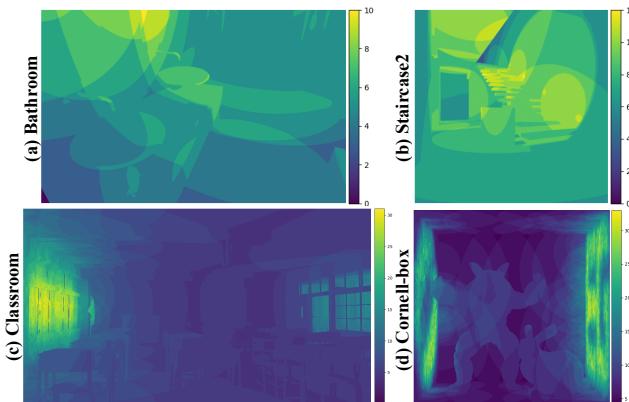


Figure 5: Visualization of the number of control variates for each pixel on four scenes: (a) Bathroom, (b) Staircase2, (c) Classroom and (d) Cornell-box. In other scenes, the number of control variates equals the number of lights.

5.3.2. Adaptive construction of multiple control variates

This subsection details the adaptive construction of control variates for each pixel, including how we determine their number and weights v_i . This process corresponds to the *stratification* step in Section 4.1. As illustrated in Figure 5, the number of control variates varies adaptively across different scenes. We begin by selecting representative points within each pixel to guide the construction.

Selection of representative points. A straightforward strategy is to use the pixel center as the representative point. From this point, we emit a camera ray, compute the corresponding shading point, and traverse the light tree to obtain a light cut. In this setting, the number of control variates equals the number of nodes in the light cut, with weights v_i computed from the normalized importance of the corresponding light clusters. As shown in Figure 9, using only the pixel center as the representative point suffices for most regions but causes aliasing near geometric edges.

To mitigate aliasing, we employ supersampling by subdividing each pixel into sub-pixels and selecting the center of each as a representative point. For each sub-pixel, we repeat the process of emitting camera rays, determining shading points, and traversing the light tree to generate its own light cut. In our implementation, we use 16 representative points per pixel. And this setup offers a good balance between improved image quality and computational cost

Merging light clusters and computing weights. Each representative point yields a distinct set of light clusters from its corresponding light cut. Assume that there are M representative points, to construct the control variates for a pixel, we traverse all clusters using an *unordered_map* keyed by cluster ID to de-duplicate them into N clusters. The final control variate weight for the i -th cluster is computed as follows:

- **Normalization:** For each representative point m , the normalized weight for the i -th cluster is computed by

$$v_{i,m} = \frac{w_{i,m}}{\sum_{j=1}^{N_m} w_{j,m}}, \quad (25)$$

where $w_{i,m}$ is the importance (Equation 24) of the i -th light cluster at point m , and N_m is the total number of light clusters associated with that point.

- **Aggregation:** The final control variate weight V_i is obtained by averaging the normalized weights over all representative points:

$$V_i = \frac{1}{M} \sum_{m=1}^M v_{i,m} \quad (26)$$

Solving multiple control variates. For each light cluster, we construct a first-order polynomial control variate and generate samples using the Lightcuts-based light sampling method (Section 5.3.1). Each control variate $\hat{h}_i(x)$ collects samples from its corresponding light cluster. Since the integrand for each light cluster is relatively simple and the sample size is small, we use the *stochastic gradient descent* method (set the learning rate to 0.01) with a single iteration to solve for the polynomial coefficients, thereby obtaining the corresponding α_i and the analytical integral H_i . The resulting coefficients are stored in a single-column matrix, while the multiple polynomials for each pixel are organized into a vector.

Table 1: We present the number of area lights, average cut size and the number of our multiple control variates for each scene.

Scene	# lights	avg. cut size	# control variates
Staircase1	2	/	2
Veach-mis	5	/	5
Staircase2	21	7.9	0 - 12
Bathroom	32	5.1	0 - 10
Classroom	162	7.6	0 - 31
Cornell-box	1494	10.28	4 - 33

6. Results

We implemented our method in the PBRT renderer [PJH16], building on the regression-based MC integration approach proposed by Salaün et al. [SGH*22]. All results were generated on a 3.20 GHz Intel i9 CPU (24 cores) with 64GB of RAM. To assess performance, we compared our method with conventional MC integration (uniform light sampling), regression-based MC integration (using a single control variate) by Salaün et al. [SGH*22], optimal multiple importance sampling by Kondapaneni et al. [KVG*19] and adaptive tree splitting for many-light sampling by Estevez et al. [CEK18], using relative mean squared error (relMSE) as the evaluation metric. The reference image for each scene was computed with 65,536 samples per pixel. We evaluated our method for direct illumination in different scenes, and Table 1 presents the relevant statistics for these scenes.

To handle multiple color channels, we first estimate the RGB value F_{rgb} by averaging RGB samples as $F_{rgb} \approx \langle F_{rgb} \rangle = 1/n \sum_{i=1}^n \hat{f}_{rgb}(u_i)$, where \hat{f}_{rgb} represents the integrand that returns an RGB sample. Next, we perform regression on the luminance value of the samples to obtain the luminance value H as the analytical integral. The final RGB reconstruction based on our estimator is given by $\langle F'_{rgb} \rangle_* = (\langle F_{rgb} \rangle / y(\langle F_{rgb} \rangle)) \langle F' \rangle_*$, where y is the luminance function and $\langle F' \rangle_*$ is the luminance estimate obtained using our method.

6.1. Comparison to Salaün et al. [SGH*22]

Figure 1, Figure 6 and Figure 7 compare our method using first-order polynomial with conventional MC integration and regression-based MC integration by Salaün et al. [SGH*22] using first-order (O1) and third-order (O3) polynomials across five different scenes, all rendered with equal time budgets. These scenes contain both diffuse and glossy materials.

For scenes with a small number of lights, e.g., VEAH-MIS (top row, Figure 6), our method constructs one control variate for each light, while Salaün et al.’s method employs only a single control variate. In this case, our approach provides a significant quality improvement without additional time overhead.

For scenes with many lights, BATHROOM (Figure 1), STAIRCASE2 (bottom row, Figure 6), CLASSROOM and CORNELL-BOX (Figure 7), our method adaptively determines the number of control variates based on light cuts and employs the Lightcuts-based light sampling technique. To ensure an equal time comparison with [SGH*22], we use a fewer SPP due to the extra computational cost of light cuts construction. Despite this, our method still

shows a noticeable reduction in noise compared to Salaün et al.’s approach for both O1 and O3. Moreover, Salaün et al.’s method becomes less efficient than conventional MC integration in scenes with many lights (e.g., CLASSROOM and CORNELL-BOX) because the increasing discontinuities in the integrand hinder effective regression on a finite number of samples. In contrast, our multiple control variates approach remains robust under such conditions, as shown in Figure 2, delivering significant quality gains as light count and complexity grow.

Figure 8 illustrates error convergence plots for the scenes described above. In scenes with fewer than 100 lights (e.g., BATHROOM and STAIRCASE2), the method of Salaün et al. outperforms MC estimation. However, as the number of lights increases substantially (e.g., CLASSROOM and CORNELL-BOX), their approach fails to surpass MC estimation. In contrast, our method consistently achieves faster convergence than both MC and Salaün et al.’s method.

6.2. Comparison to Kondapaneni et al. [KVG*19]

Kondapaneni et al. [KVG*19] derived an estimator in the form of control variates by optimizing the multiple importance sampling (MIS) weighting function and applied it to direct illumination. In Figure 10, the light and BSDF sampling samples are first combined using balance heuristics weights to estimate the illumination from individual light (a), then combined using Kondapaneni et al.’s optimal weights (b), and finally our method constructs one control variate for each light (c). In this setting, our method achieves a slight improvement over Kondapaneni et al.’s approach.

Figure 11 illustrates the STAIRCASE1 scene, which contains two lights. This example clearly demonstrates how both our method and Kondapaneni et al.’s method improve estimation quality. Uniform light sampling in MC estimation introduces noise due to the random selection of lights. The *Trained* technique in PBRT [PJH16] partitions the scene into a regular grid, estimates the unoccluded contributions of all lights in each cell, and utilizes these estimates as light selection probabilities. While this approach performs nearly optimally on unoccluded surfaces, it introduces significant noise in shadowed areas. Kondapaneni et al. addressed this issue by optimally combining the *Trained* and *Uniform* techniques, resulting in improved estimation quality. In contrast, our method employs multiple control variates in MC integration, constructing one control variate for each light in the scene, thereby improving the quality of MC estimation. Our approach is complementary to that of Kondapaneni et al., as we focus on improving the MC estimation process through control variates, whereas they concentrate on optimizing light sampling.

6.3. Comparison to Estevez et al. [CEK18]

In scenes with many lights, we incorporate a Lightcuts-based light sampling technique into our adaptive control variate framework. While Estevez et al. [CEK18] proposed an adaptive tree splitting method for importance sampling of many lights, our method builds on this by combining it with adaptive construction of multiple control variates. As shown in Figure 12, when comparing our method

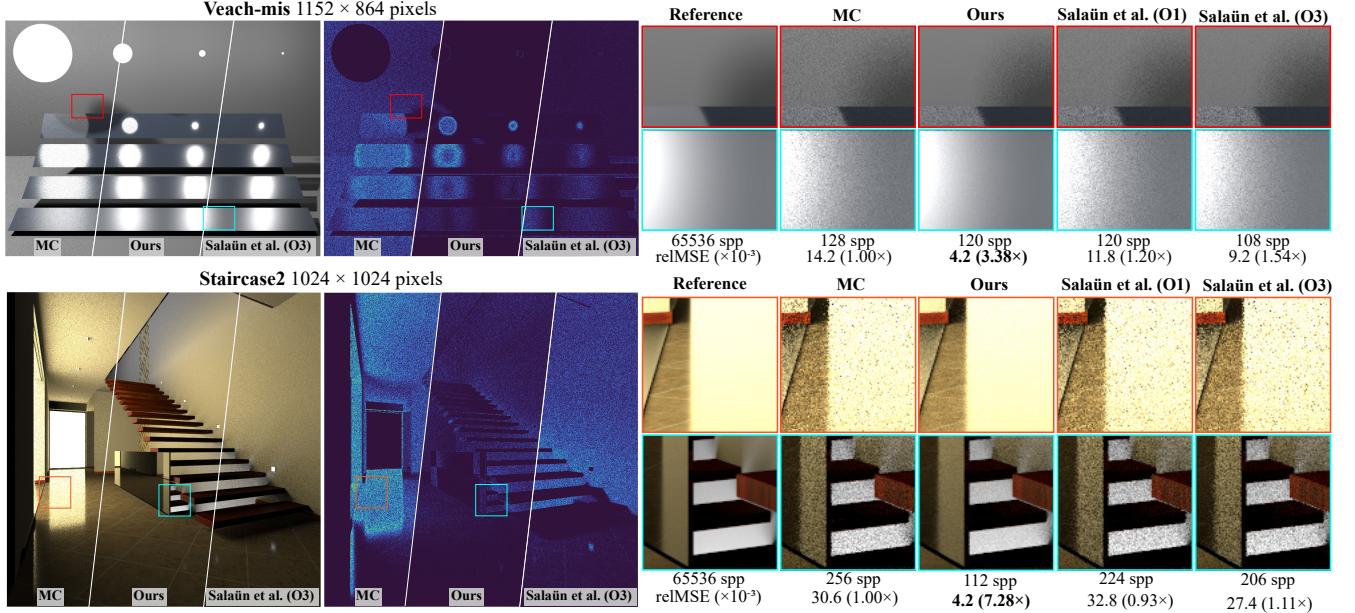


Figure 6: Equal-time comparison among our method with polynomials of order 1, conventional MC, and regression-based MC integration of Salaün et al. [SGH*22] with polynomials of order 1 (O1) and 3 (O3) on the Veach-mis and Staircase2 scene. For a more detailed comparison, please refer to the HTML viewer provided in the supplemental.

with that of Estevez et al. under equal SPP, our approach yields a modest improvement in image quality.

7. Discussion and Future Work

In this section, we discuss the limitations of our current approach and outline promising directions for future research.

Comparison with other many-light sampling methods. Existing methods [VKK18, LXY19, RGH*20, WWLC21] improve light clustering and sampling by leveraging information collected during rendering. In contrast, our approach focuses on enhancing the final MC estimation through multiple control variates. We believe that integrating our approach with these techniques could yield further improvements.

Visibility. Our current implementation does not account for visibility when computing light cluster selection probabilities, which limits its ability to handle occlusion-related discontinuities. Integrating visibility-aware strategies, such as those proposed by [VKK18] and [WWLC21], represents a promising direction to overcome this limitation.

Incremental estimators. In our implementation, we construct the sample estimator $\langle F_i \rangle_*^{n_i}$ for each control variate \hat{h}_i by performing regression using all n_i samples, and then evaluating the estimator with the same samples. It can be extended to progressive rendering where samples are added incrementally, for instance, the incremental estimate $\langle F_i \rangle_*^{n_{i+1}}$ can be defined as

$$\langle F_i \rangle_*^{n_{i+1}} = \frac{1}{n_i + 1} \left(n_i \langle F_i \rangle_*^{n_i} + (H_i^{n_i} + \hat{f}_i(u_{n_{i+1}}) - \hat{h}_i^{n_i}(u_{n_{i+1}})) \right), \quad (27)$$

where $\langle F_i \rangle_*^{n_i}$ is the current estimate, $\hat{h}_i^{n_i}$ is the regression solution

based on n_i samples, and $H_i^{n_i}$ is the corresponding analytic integral. With each new sample, the control variate is recomputed for the appropriate light cluster, and the corresponding estimator is updated accordingly.

Bias. Our current implementation uses the same sample set for both solving the regression model of the control variate and estimating the residual, which introduces bias into the estimator. This bias stems solely from the shared sample set and does not impact the validity of our theoretical analysis. Nonetheless, future work could explore decoupling the regression and estimation stages to mitigate this bias.

Environment light. Our method does not yet handle high-frequency environment lighting effectively. In particular, using a single control variate is insufficient to manage the discontinuities in the integrand produced by complex environment maps. We plan to extend our approach by combining multiple control variates with structured importance sampling of the environment map [ARB03], which may improve performance in these scenes.

Generalizability. Our work demonstrates that multiple control variates can effectively accelerate MC integration convergence for low-dimensional direct illumination problems. However, by storing statistics solely in image space, our method faces limitations in generalizing to high-dimensional integrals prevalent in many rendering tasks. Exploring scene-space control variates, as investigated in recent works [Pan20, HGS23], offers a promising avenue for future improvements.

8. Conclusion

In this work, we introduce an adaptive multiple control variates framework to improve the Monte Carlo integration for many-light

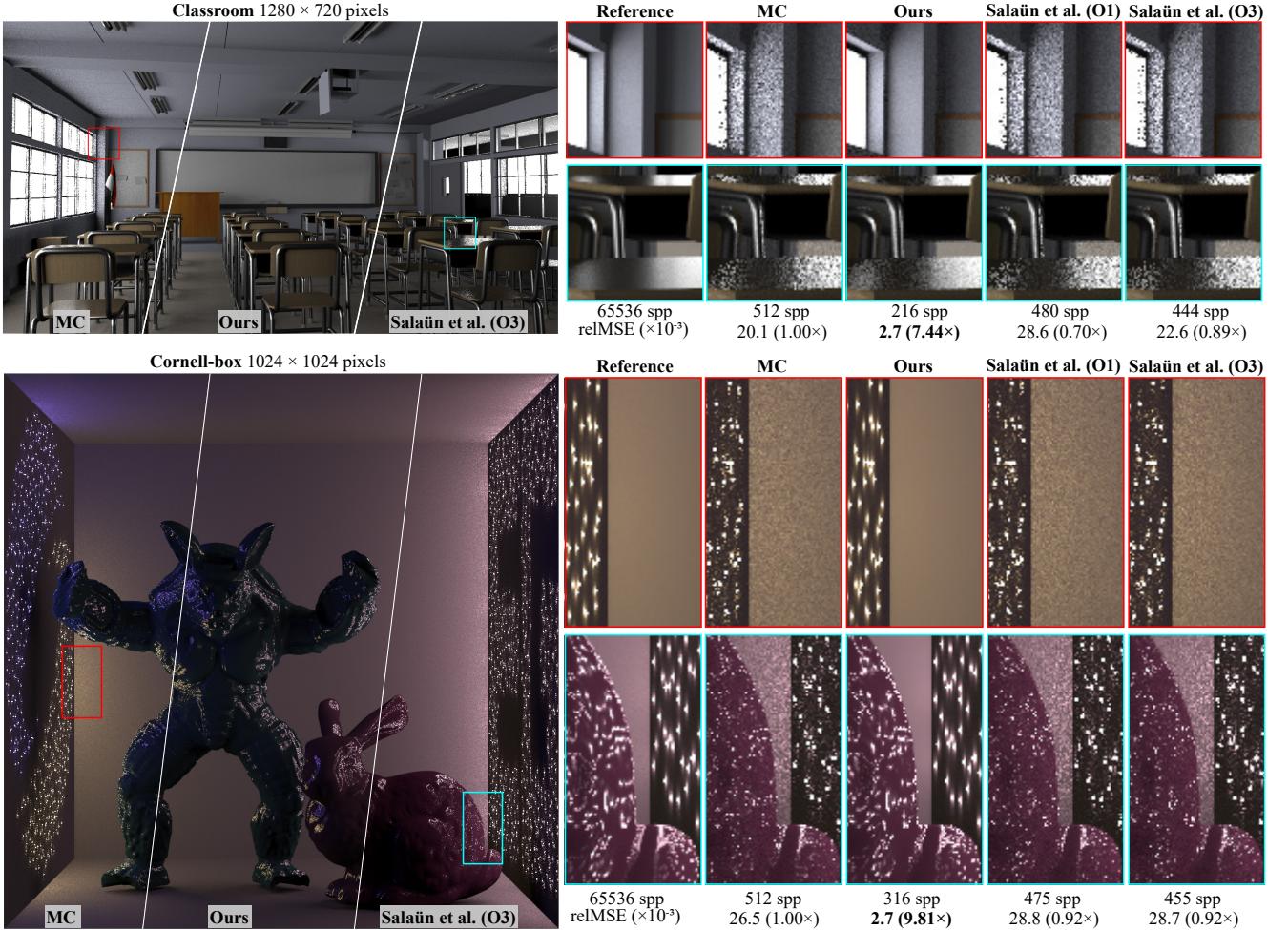


Figure 7: Equal-time comparison among our method with polynomials of order 1, conventional MC, and regression-based MC integration of Salaün et al. [SGH*22] with polynomials of order 1 (O1) and 3 (O3) on the Classroom and Cornell-box scene.

rendering. By leveraging hierarchical light clustering inspired by the Lightcuts approach, our method dynamically adjusts the number of control variates according to the spatial distribution and inherent discontinuities of lights within the scene. We provided a rigorous mathematical foundation to demonstrate that our approach reduces variance, leading to marked improvements in rendering quality for the direct illumination problem. Although our experiments focused on low-dimensional integration scenarios, the encouraging results pave the way for future research aimed at extending this adaptive strategy to high-dimensional integration and more complex rendering challenges.

Acknowledgements

We thank the reviewers for the valuable comments. We thank the following for scenes used in our experiments: nacimus (Bathroom), Veach (Veach-mis), NewSee2l035 (Staircase2), Jay-Artist (Living-room2), NovaZeeke (Classroom), Wig42 (Staircase1). This work has been partially supported by the National Natural Science Foundation of China (No. 62272275),

the Taishan Scholars Program (No. tsqn202312231), Qilu University of Technology (Shandong Academy of Sciences) Faculty of Computer Science and Technology Pairing Program (No. 2024JDJH13).

References

- [ARB03] AGARWAL S., RAMAMOORTHI R., BELONGIE S., JENSEN H. W.: Structured importance sampling of environment maps. In *ACM SIGGRAPH 2003 Papers*. 2003, pp. 605–612. 9
- [BWP*20] BITTERLI B., WYMAN C., PHARR M., SHIRLEY P., LEFOHN A., JAROSZ W.: Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 148–1. 3
- [CAM08] CLARBERG P., AKENINE-MÖLLER T.: Exploiting visibility correlation in direct illumination. In *Computer Graphics Forum* (2008), vol. 27, Wiley Online Library, pp. 1125–1136. 2
- [CEK18] CONTY ESTEVEZ A., KULLA C.: Importance sampling of many lights with adaptive tree splitting. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 1, 2 (2018), 1–17. 3, 6, 7, 8, 12

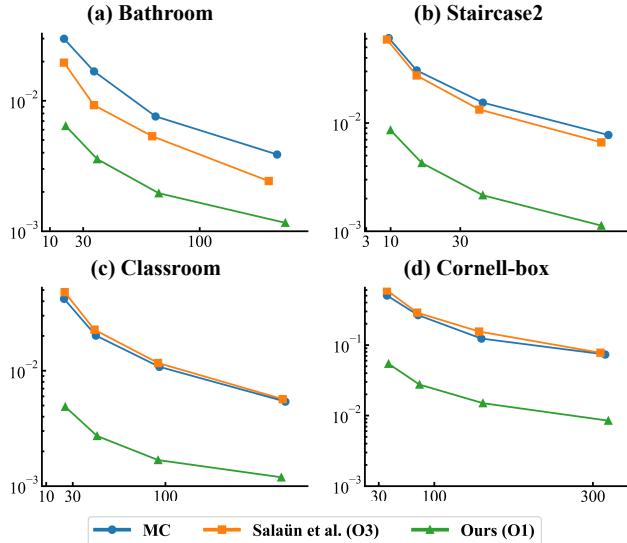


Figure 8: Convergence plot for equal-time comparisons across four scenes: (a) Bathroom, (b) Staircase2, (c) Classroom and (d) Cornell-box. The metric is relMSE and horizontal axis is time (s). In all scenes, our method shows faster convergence compared to both MC estimation and the method proposed by Salaün et al. [SGH*22].

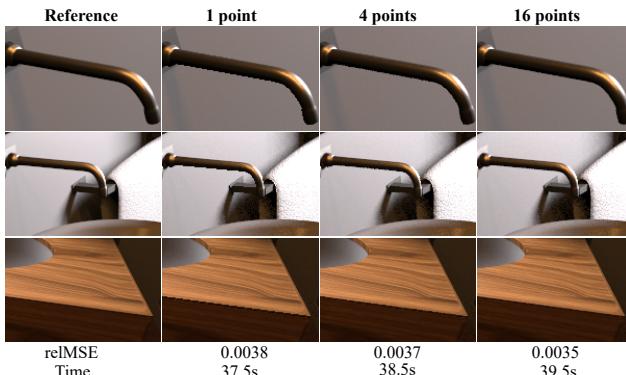


Figure 9: Experimental results for increasing the number of representative points per pixel in the Bathroom scene. As the number of represented points increases, the degree of aliasing decreases and the time cost increases.

[CJMn21] CRESPO M., JARABO A., MUÑOZ A.: Primary-space adaptive control variates using piecewise-polynomial approximations. *ACM Transactions on Graphics* 40, 3 (July 2021). URL: <https://doi.org/10.1145/3450627>, doi:10.1145/3450627. 2, 3

[DKH*14] DACHSBACHER C., KŘIVÁNEK J., HAŠAN M., ARBREE A., WALTER B., NOVÁK J.: Scalable realistic rendering with many-light methods. In *Computer Graphics Forum* (2014), vol. 33, Wiley Online Library, pp. 88–104. 3

[DWB*06] DONIKIAN M., WALTER B., BALA K., FERNANDEZ S., GREENBERG D. P.: Accurate direct illumination using iterative adaptive sampling. *IEEE Transactions on Visualization and Computer Graphics* 12, 3 (2006), 353–364. 3

[FCH*06] FAN S., CHENNEY S., HU B., TSUI K.-W., LAI Y.-C.: Optimizing control variate estimators for rendering. In *Computer Graphics*

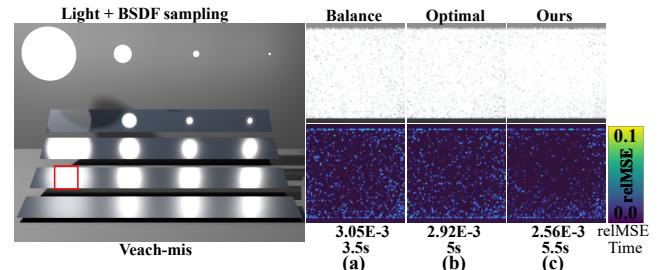


Figure 10: Equal-sample comparison of our method (c) with the balance heuristics (a) and optimal MIS weights [KVG*19] (b) in the classic light vs. BSDF sampling scenario in the Veach-mis scene. The relMSE values are computed after 24 samples per light per technique per pixel.

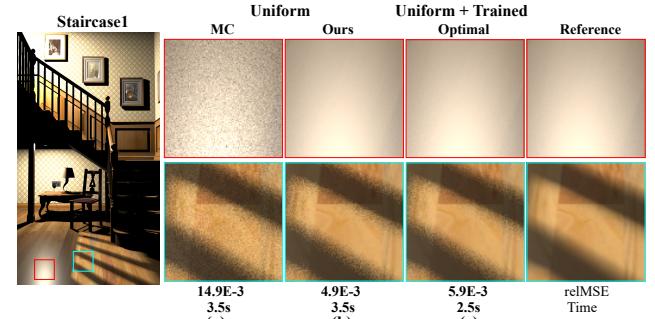


Figure 11: Equal-sample comparison of direct illumination estimated by MC (uniform light and BSDF sampling with the balance heuristics, 40 SPP) (a), our method applied to MC (b) and an MIS combination of two light sampling techniques (Trained and Uniform, 20 SPP per technique) with optimal weights [KVG*19] (c). Note that the result in (c) does not use BSDF sampling, and therefore requires slightly less time.

Forum (2006), vol. 25, Wiley Online Library, pp. 351–357. 3

[GMH*19] GEORGIEV I., MISSO Z., HACHISUKA T., NOWROUZEZAHRAI D., KŘIVÁNEK J., JAROSZ W.: Integral formulations of volumetric transmittance. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 1–17. 3

[HGS23] HUA Q., GRITTMANN P., SLUSALLEK P.: Revisiting controlled mixture sampling for rendering applications. *ACM Transactions on Graphics (TOG)* 42, 4 (2023), 1–13. 2, 9

[Kaj86] KAJIYA J. T.: The rendering equation. *SIGGRAPH Comput. Graph.* 20, 4 (aug 1986), 143–150. URL: <https://doi.org/10.1145/15886.15902>, doi:10.1145/15886.15902. 2

[KDPN21] KETTUNEN M., D'EON E., PANTALEONI J., NOVÁK J.: An unbiased ray-marching transmittance estimator. *ACM Trans. Graph.* 40, 4 (July 2021). URL: <https://doi.org/10.1145/3450626.3459937>, doi:10.1145/3450626.3459937. 3

[KHLN17] KUTZ P., HABEL R., LI Y. K., NOVÁK J.: Spectral and decomposition tracking for rendering heterogeneous volumes. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–16. 3

[KJ94] KOK A. J., JANSEN F. W.: Source selection for the direct lighting computation in global illumination. In *Photorealistic Rendering in Computer Graphics: Proceedings of the Second Eurographics Workshop on Rendering* (1994), Springer, pp. 75–82. 3

[KVG*19] KONDAPANENI I., VÉVODA P., GRITTMANN P., SKŘIVAN T., SLUSALLEK P., KŘIVÁNEK J.: Optimal multiple importance sam-

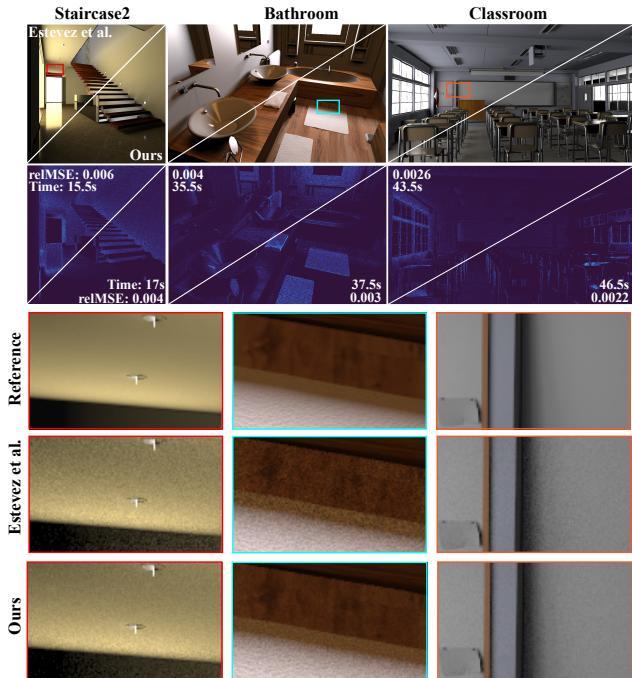


Figure 12: Equal-sample comparison of Estevez et al. [CEK18] and our method. We report the time and the relMSE value compared to the reference image for the whole image.

- pling. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–14. 3, 8, 11
- [Loh95] LOH W. W.: *On the method of control variates*. Stanford University, 1995. 2
- [LPG13] LU H., PACANOWSKI R., GRANIER X.: Second-Order Approximation for Variance Reduction in Multiple Importance Sampling. *Computer Graphics Forum* (2013). doi:10.1111/cgf.12220. 2
- [LW94] LAFORTUNE E. P., WILLEMS Y. D.: The ambient term as a variance reducing technique for monte carlo ray tracing. In *Proceedings of the 5th Eurographics Workshop on Rendering* (Darmstadt, Germany, 1994), pp. 168–176. 2
- [LW95] LAFORTUNE E. P., WILLEMS Y. D.: A 5d tree to reduce the variance of monte carlo ray tracing. In *Rendering Techniques' 95: Proceedings of the Eurographics Workshop in Dublin, Ireland, June 12–14, 1995* 6 (1995), Springer, pp. 11–20. 2
- [LXY19] LIU Y., XU K., YAN L.-Q.: Adaptive brdf-oriented multiple importance sampling of many lights. In *Computer Graphics Forum* (2019), vol. 38, Wiley Online Library, pp. 123–133. 3, 9
- [MRKN20] MÜLLER T., ROUSSELLE F., KELLER A., NOVÁK J.: Neural control variates. *ACM Transactions on Graphics (TOG)* 39, 6 (2020), 1–19. 2
- [MRR*53] METROPOLIS N., ROSENBLUTH A. W., ROSENBLUTH M. N., TELLER A. H., TELLER E.: Equation of state calculations by fast computing machines. *The journal of chemical physics* 21, 6 (1953), 1087–1092. 2
- [NSJ14] NOVÁK J., SELLE A., JAROSZ W.: Residual ratio tracking for estimating attenuation in participating media. *ACM Trans. Graph.* 33, 6 (2014), 179–1. 3
- [OA11] OLSSON O., ASSARSSON U.: Tiled shading. *Journal of Graphics, GPU, and Game Tools* 15, 4 (2011), 235–251. 3
- [Owe13] OWEN A. B.: *Monte Carlo theory, methods and examples*. <https://artowen.su.domains/mc/>, 2013. 2

- [Pan20] PANTALEONI J.: Online path sampling control with progressive spatio-temporal filtering. *SN Computer Science* 1, 5 (2020), 279. 2, 9
- [PBPP11] PAJOT A., BARTHE L., PAULIN M., POULIN P.: Representativity for robust and adaptive multiple importance sampling. *IEEE Transactions on Visualization and Computer Graphics* 17, 8 (aug 2011), 1108–1121. URL: <https://doi.org/10.1109/TVCG.2010.230>. 2
- [PJH16] PHARR M., JAKOB W., HUMPHREYS G.: *Physically Based Rendering: From Theory to Implementation* (3rd ed.), 3rd ed. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, Nov. 2016. 7, 8
- [PPD98] PAQUETTE E., POULIN P., DRETTAKIS G.: A light hierarchy for fast rendering of scenes with many lights. In *Computer Graphics Forum* (1998), vol. 17, Wiley Online Library, pp. 63–74. 3
- [RGH*20] RATH A., GRITTMANN P., HERHOLZ S., VÉVODA P., SLUSALLEK P., KŘIVÁNEK J.: Variance-aware path guiding. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2020)* 39, 4 (July 2020), 151:1–151:12. doi:10.1145/3386569.3392441. 3, 9
- [RJN16] ROUSSELLE F., JAROSZ W., NOVÁK J.: Image-space control variates for rendering. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 35, 6 (Dec. 2016), 169:1–169:12. doi:10.1145/3009906.3009933
- [SGH*22] SALA "UN C., GRUSON A., HUA B.-S., HACHISUKA T., SINGH G.: Regression-based monte carlo integration. *ACM Trans. Graph.* 41, 4 (2022). doi:10.1145/3528223.3530095. 1, 2, 3, 4, 5, 8, 9, 10, 11
- [SSSK04] SZÉCSI L., SBERT M., SZIRMAY-KALOS L.: Combined correlated and importance sampling in direct light source computation and environment mapping. In *Computer Graphics Forum* (2004), vol. 23, Wiley Online Library, pp. 585–593. 2
- [SWZ96] SHIRLEY P., WANG C., ZIMMERMAN K.: Monte carlo techniques for direct lighting calculations. *ACM Transactions on Graphics (TOG)* 15, 1 (1996), 1–36. 3, 7
- [TH16] TOKUYOSHI Y., HARADA T.: Stochastic light culling. *J Comput. Graph. Tech.* 5, 1 (2016). 3
- [Vea97] VEACH E.: *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford University, 1997. 2, 3, 4
- [VKK18] VÉVODA P., KONDAPANENI I., KŘIVÁNEK J.: Bayesian online regression for adaptive direct illumination sampling. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–12. 2, 3, 9
- [WABG06] WALTER B., ARBREE A., BALA K., GREENBERG D. P.: Multidimensional lightcuts. In *ACM SIGGRAPH 2006 Papers*. 2006, pp. 1081–1088. 3
- [War94] WARD G. J.: Adaptive shadow testing for ray tracing. In *Photorealistic Rendering in Computer Graphics: Proceedings of the Second Eurographics Workshop on Rendering* (1994), Springer, pp. 11–20. 3
- [WFA*05] WALTER B., FERNANDEZ S., ARBREE A., BALA K., DONIKIAN M., GREENBERG D. P.: Lightcuts: a scalable approach to illumination. In *ACM SIGGRAPH 2005 Papers*. 2005, pp. 1098–1107. 2, 3, 7
- [WKB12] WALTER B., KHUNGURN P., BALA K.: Bidirectional lightcuts. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 1–11. 3
- [WWLC21] WANG Y.-C., WU Y.-T., LI T.-M., CHUANG Y.-Y.: Learning to cluster for rendering with many lights. *ACM Transactions on Graphics (TOG)* 40, 6 (2021), 1–10. 3, 9
- [XLG*24] XU B., LI T.-M., GEORGIEV I., HEDSTROM T., RAMAMOORTHI R.: Residual path integrals for re-rendering. In *Computer Graphics Forum* (2024), vol. 43, Wiley Online Library, p. e15152. 3
- [ZHB*24] ZHU J., HERY C., BODE L., ALIAGA C., JARABO A., YAN L.-Q., CHIANG M. J.-Y.: A realistic multi-scale surface-based cloth appearance model. SIGGRAPH '24, Association for Computing Machinery. URL: <https://doi.org/10.1145/3641519.3657426>. doi:10.1145/3641519.3657426. 3