

Real-time Subsurface Control Variates: Temporally Stable Adaptive Sampling

TIANTIAN XIE and MARC OLANO, University of Maryland, Baltimore County, USA



Fig. 1. Dynamic subsurface scene just after light has been turned off. Our method has consistently lower sample count (d) than SPVG [Xie et al. 2020] (c) at this frame. It leads to lower sampling pass time in dynamic lighting from 12.9 ms to 5.2 ms at 3360×1440 ($\times 2.5$), while maintaining good quality (47.5 dB) vs SPVG (48.6 dB). Separable (b) runs fastest for the whole subsurface pass at 4.0 ms, however, with visible banding artifacts.

Real-time adaptive sampling is a new technique recently proposed for efficient importance sampling in real-time Monte Carlo sampling in subsurface scattering. It adaptively places samples based on variance tracking to help escape the uncanny valley of subsurface rendering. However, the occasional performance drop due to temporal lighting dynamics (e.g., guns or lights turning on and off) could hinder adoption in games or other applications where smooth high frame rate is preferred. In this paper we propose a novel usage of Control Variates (CV) in the sample domain instead of shading domain to maintain a consistent low pass time. Our algorithm seamlessly reduces to diffuse with zero scattering samples for sub-pixel scattering. We propose a novel joint-optimization algorithm for sample count and CV coefficient estimation. The main enabler is our novel time-variant covariance updating method that helps remove the effect of recent temporal dynamics from variance tracking. Since bandwidth is critical in real-time rendering, a solution without adding any extra textures is also provided.

CCS Concepts: • Computing methodologies → Rendering; Reflectance modeling; • Theory of computation → Online algorithms; Stochastic approximation.

Additional Key Words and Phrases: Real-time adaptive sampling, subsurface scattering, control variates

Authors' address: Tiantian Xie, xtiant1@umbc.edu; Marc Olano, olano@umbc.edu, University of Maryland, Baltimore County, 1000 Hilltop Cir, Baltimore, Maryland, USA, 21250.

© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM.

This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, <https://doi.org/10.1145/3451265>.

ACM Reference Format:

Tiantian Xie and Marc Olano. 2021. Real-time Subsurface Control Variates: Temporally Stable Adaptive Sampling. *Proc. ACM Comput. Graph. Interact. Tech.* 4, 1 (May 2021), 18 pages. <https://doi.org/10.1145/3451265>

1 INTRODUCTION

Real-time adaptive sampling is a recent technique for subsurface scattering with importance sampling of Burley's normalized diffusion profile [Xie et al. 2020]. It has been adopted in real-time rendering engines (e.g., Unreal Engine 4). Its low time and space complexity $O(1)$ increases the sample efficiency of Monte Carlo algorithms for real-time rendering. The basic idea is to use sample histories in shading domain to estimate the spatial Monte Carlo sampling variance and adjust the sample count to minimize the variance to a target variance level in real time.

However, if high temporal variance is present after history projection, it will also increase the sample count and pass time to achieve convergence. Real-time applications like games are intrinsically dynamic, including i) gunshots/lightning creating rapid incoming radiance changes, ii) dynamic light particles changing intensities, and iii) moving objects creating dynamic shadowing.

In this paper, we propose a novel technique, *real-time subsurface control variates*, to address the issue. Control variates is a variance reduction technique for Monte Carlo sampling. If we can find a good approximation of the sampling function, with the optimal CV coefficient, the variance after MC sampling can be reduced given the same number of samples. In other words, given a target variance, the sample count can be minimized. Note that the application of CV in this paper is a little different from the typical use. We only reduce the monitored change in variance caused by the lighting changes, but keep the same variance as the MC estimator when the lighting is static because the monitored variance controls the sample count estimation [Xie et al. 2020]. In this paper, we focus on real-time subsurface scattering. To have real-time subsurface scattering anytime and anywhere efficiently using control variates, the following four challenges are addressed:

1) *Minimize subsurface scattering region.* Since diffuse is the subsurface scattering within a pixel, the actual scattering should only consider distant scattering further than one pixel. By separating the distant subsurface scattering component out from diffuse, it provides an opportunity to handle all scattering in the same framework efficiently. In this paper, we provided such a unified framework suitable for both density-function-based and artist-friendly control to support down to zero subsurface samples per pixel (spp) without explicitly switching to a diffuse material.

2) *Evaluate the time-variant covariance matrix.* Calculating the optimal CV coefficient online requires a covariance matrix that captures the recent spatial correlations between the sampling function and the control for time-variant scenarios like dynamic lighting, yet without temporal variance. We propose a novel covariance estimation based on the CV residual, *exponential moving covariance matrix*. We compare this to the widely adopted exponential weighted moving average (EWMA) covariance estimator [Guldmann et al. 1995].

3) *Compute the optimal CV coefficient online.* We provide online joint estimation algorithms to find the sample count and the CV coefficient numerically. Since the estimated coefficient is dependent on the control variable with unknown distribution, the rendering result based on the control variates might be biased [Lavenberg et al. 1982]. To avoid this issue, we shift the application of CV from shading domain to sample domain, where the CV guides sample count estimation. In this way, we also shift potential bias from shading domain to sample domain. Since our online covariance computation assures overestimation, the final shading result is still unbiased. Nevertheless, the rendering time for subsurface scattering is reduced up to $\times 3.11$ during dynamic lighting tests.

4) *Lightweight CV coefficient.* Since online optimal CV coefficient estimation still requires an additional texture and real-time computation, this might violate the memory budget for time critical

applications. Under such scenarios, we also provide a lightweight offline approximation for the optimal CV coefficient without adding any additional textures.

The main contribution is summarized as below:

- A unified subsurface scattering model for efficient subsurface scattering anywhere.
- Derived exponential moving covariance matrix as an extension to the largely adopted EWMA covariance estimator.
- An online sample count estimation algorithm jointly estimated with optimal CV coefficient estimation to remove temporal influence.
- A lightweight CV without adding any extra textures.

2 RELATED WORK

Real-time Subsurface scattering. The major enabler for real-time subsurface scattering is the dipole diffuse reflectance profile [Jensen et al. 2001]. There are several fast approximations for real-time applications using diffusion in texture space [d’Eon and Luebke 2007] or screen space [Jimenez et al. 2009, 2015], or pre-integration [Penner and Borshukov 2011]. More recently, Burley fit Monte-Carlo data to produce a simple analytic importance sampling function [Burley 2015; Christensen and Burley 2015], which has led to more recent work in screen space with pre-calculated sampling [Golubev 2018] or adaptive sampling [Xie et al. 2020]. In this paper, we propose a unified model for importance sampling function based subsurface scattering by separating diffuse and distant scattering. It reduces to 0 distant scattering spp for sub-pixel scattering distances.

Adaptive sampling. Adaptive sampling is a sampling technique to allocate more samples in regions with high variation [Pharr et al. 2016]. It enables faster converging than uniform sampling [Dammertz et al. 2010; Moon et al. 2014]. However, the major drawback is if the number of the initial pilot samples used to measure variation is small, the result would be biased [Kirk and Arvo 1991]. This is also the dilemma that prevents the adoption of adaptive sampling in real-time rendering - too many pilots result in less performance, and too few results in bias. *Single pass variance guiding* (SPVG) [Xie et al. 2020] introduced adaptive subsurface sampling for real-time rendering by reusing bandwidth-friendly history measures stored in a single texture. However, the performance is hindered when there are dynamic lighting changes, which contribute to the variance, causing an overestimate in the number of samples required. Our paper addresses this issue, with control variates mitigating the temporal variance effect on sample count estimation.

Control variates. Control variates is a variance reduction technique that decomposes a Monte-Carlo integration into the sum of a known integral with some scaling coefficient and a more easily estimated residual [Ripley 2009]. It has been studied and applied in many fields, like finance [Alexander 1999], operations research [Hesterberg and Nelson 1998; Nelson 1990], and computer networks [Lavenberg et al. 1982]. It requires an estimation of the optimal CV coefficient. In computer graphics, CV has been used in offline rendering applications with a pre-defined constant CV coefficient for main part separation (to separate out a simplified estimate, e.g., residual ratio tracking [Novák et al. 2014], and for multiple correlated sampling [Szécsi et al. 2004]). Finding the optimal CV coefficient has also been explored through penalized least squares [Fan et al. 2006], iterative estimators [Kondapaneni et al. 2019; Rousselle et al. 2016], and deep learning [Müller et al. 2020]. However, no existing research provides any guidance or applications for real-time rendering where the time budget per frame is only several milliseconds and we cannot afford those expensive calculations. Moreover, the goal here is slightly different - to reduce the monitored variance due to temporal changes instead of to reduce the MC estimator’s variance as those prior techniques do in static scenes. The monitored variance is then used for sample count estimation in real-time adaptive sampling. In this paper, we propose such an online CV coefficient estimation algorithm based on a

novel online covariance matrix formula. We also provide a constant lightweight approximation in the spirit of *main part separation* for easy adoption.

Online Covariance. Covariance is a fundamental concept in computational statistics and has great applications in many fields. When memory is limited, it is critical to have a single pass online algorithm. Welford [1962] proposed an online single pass algorithm to calculate the overall covariance numerically when each value is equally weighted. In a real-time time-variant system, weighted covariance that favors latest results are used to track temporal changes. Two notable examples are the prediction error covariance matrix update in the Kalman filter [Welch et al. 1995] with varying weights, which is frequently used in control systems, and the EWMA covariance estimator [Guldmann et al. 1995; Tsay 2005] in finance with constant weights. In this paper, we exploit the boundary of second case. We propose an online exponential moving covariance matrix that is mathematically derived from the weighted covariance matrix with inspiration from exponential moving variance [Finch 2009]. We also provide its relationship to the well-known EWMA covariance estimator. Moreover, we demonstrate how temporal variance can be removed during the monitoring.

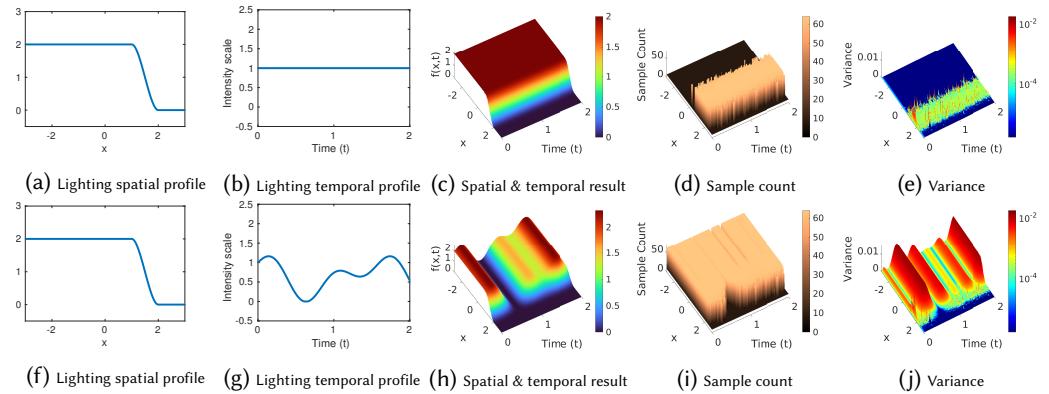


Fig. 2. Temporal instability leads to sample count over-estimation with real-time adaptive sampling. Diffuse lighting on a 1D surface is shown in (a) and (f). During subsurface scattering sampling, when lighting is temporally stable (b), the estimated sample count (d) leads to lighting (c) with a variance close to target variance 10^{-4} (e). When lighting is not stable (g), although we can get high quality lighting (h), $rmse = .0028$, the sample count (i) is over estimated because of temporal variance in (j). Temporal instability induces 242% (i) as many samples as that shown in (d) on average in this example. The induced calculation and bandwidth demands might threaten the performance with real-time adaptive sampling.

3 BACKGROUND

We briefly summarize the main concepts of subsurface scattering, real-time adaptive sampling, and *control variates*.

3.1 Subsurface scattering

In subsurface scattering, the outgoing radiance at $p \in \partial\Omega$ has

$$L_o(p, \omega_o) = \int_{\partial\Omega} \int_{S^2} L_i(q, \omega_i) S(q, \omega_i, p, \omega_o) d\omega_i dq, (q \in \partial\Omega), \quad (1)$$

where the *bidirectional scattering-subsurface reflectance distribution function* (BSSRDF), $S(\cdot)$, is simplified in real-time rendering as

$$S(q, \omega_i, p, \omega_o) = C F_t(q, \omega_i) R(r_q) F_t(p, \omega_o), \quad (2)$$

where $r_q = ||p - q||$, $R(\cdot)$ is the diffuse reflectance profile (or diffusion profile), $F_t(\cdot)$ is the directional Fresnel transmission term, and C is a constant. This simplification holds only when S is radially symmetric in a homogeneous semi-infinite planar medium. Then, the radiance function becomes:

$$L_o(p, \omega_o) = \int_{\partial\Omega} R(r_q) \cdot b(p, q, \omega_o) \cdot dq \quad (3)$$

$$b(p, q, \omega_o) = \int_{S^2} C F_t(p, \omega_o) F_t(q, \omega_i) L_i(q, \omega_i) \langle \omega_i, n_q \rangle d\omega_i, \quad (4)$$

where $b(p, q, \omega_o)$ is the lighting contribution at p from q with an outgoing direction of ω_o . It can be pre-integrated in deferred rendering. Since textures are discrete, we denote the discrete pre-integrated lighting texture as $B(p, q, \omega_o)$.

3.2 Real-time adaptive sampling

The adaptive sampling algorithm [Xie et al. 2020] has two steps: 1) history update and 2) sample count estimation. For an image plane location p , the temporal update functions at time i are:

$$\mu_i = (1 - \eta)\mu_{i-1} + \eta S(p_i) \quad (5)$$

$$\sigma_i^2 = (1 - \eta)\sigma_{i-1}^2 + \eta(1 - \eta)(S(p_i) - \mu_{i-1})^2 \quad (6)$$

$$\bar{n}_i = (1 - \eta)\bar{n}_{i-1} + \eta n_i \quad (7)$$

Eq. 5–7 update the shading mean, variance and sample count mean respectively with exponential moving coefficient η specified by the user for the corresponding texture cache location p_i . $S(\cdot)$ is the luminance shading function to compact history, $\mathcal{H} = (\mu_i, \sigma_i^2, \bar{n}_i)$, into one texture. In this paper, we mitigate temporal variance with CV in Eq. 6.

To estimate the actual samples required to reach the target quality level σ_0^2 , the estimated sample count has:

$$\hat{n}_i = \kappa \cdot \Delta(i) + E(\bar{n}_i), \kappa \in [0, 1] \quad (8)$$

$$\Delta(i) = \frac{(\sigma_{i-1}^2 - \sigma_0^2)}{\sigma_0^2} \cdot \bar{n}_{i-1} \cdot (2/\eta - 2) \quad (9)$$

$$E(\bar{n}_i) = \sigma_{i-1}^2 / \sigma_0^2 \cdot \bar{n}_{i-1} \quad (10)$$

where the control factor κ limits the correction term (Eq. 9) to favor convergence ($\kappa = 1$) or performance ($\kappa = 0$). \hat{n}_i is then clamped to prevent bias due to undersampling, and meet the time budget.

3.3 Control variates

In the rendering literature, control variates (CV) has been used to increase Monte Carlo (MC) rendering efficiency. The basic idea is to modify the original function with a known integral corrected by a coefficient over a spatial domain (e.g., 2D surface or 3D Volume). In this paper, we deal with temporal change for sample estimation. Instead of working in space domain, we extend the concept to time domain to address time-variant instability. This is slightly different from prior

Table 1. Selected Symbols

Symbol	Description
\mathcal{H}	Real-time adaptive sampling history
η	Exponential moving coefficient
σ_0^2	Target variance level for adaptive sampling
n	Sample count for total scattering
m	Sample count for distant scattering
\mathcal{J}	Control Variates history
α	Control Variates coefficient
$g(t)$	Control variable at time t
$G(t)$	Integration of $g(t)$, a known constant value at time t that varies according to time.
$R(r)$	Diffuse reflectance profile
$b(p, q, \omega_o)$	Lighting contribution at point p from point q for direction ω_o
$b(t)$	Lighting contribution at point p from p at time t
$B(p, q, \omega_o)$	Discrete pre-integrated lighting texture for $b(p, q, \omega_o)$
$B(t)$	Pre-integrated lighting at texture point p at time t
γ	Distant scattering energy ratio

techniques that minimize MC estimator variance in static scenes. The modified formula is:

$$F(x, t) = \alpha(x, t) \cdot G(x, t) + \int_{y \in \mathcal{D}} f(x, y, t) - \alpha(x, t) \cdot g(x, y, t) dy. \quad (11)$$

where $G(x, t) = \int_{y \in \mathcal{D}} g(x, y, t) dy$ and the optimal CV coefficient has $\alpha = \text{Cov}(f, g) / \text{Var}(g)$ [Lavengberg et al. 1982]. In non-real-time rendering, time can be regarded as another dimension. If $x \in \mathbb{R}^n$, then $(x, t) \in \mathbb{R}^{n+1}$. However, in real-time rendering only limited history can be accessed at t for performance and storage, and no forward time evaluation is possible. In real-time adaptive sampling, only one history texture is utilized, we explore in the same spirit to make control variates possible in real-time rendering.

4 MOTIVATING EXAMPLE

In real-time adaptive sampling, there are two major contributions to the variance: spatial variance due to MC sampling, and temporal variance due to lighting change.

Spatial variance. Fig. 2(a)–(e) shows a 1D example of spatial variance with stable lighting over 2 seconds. For lighting gradient change region around $x \in [1, 2]$, the sample count is estimated to the max budget allowed at 64 spp to minimize the variance to $\sigma_0^2 = 10^{-4}$ as shown in Fig. 2(e).

Temporal variance. Dynamic lighting introduces *temporal variance*. For example, it could be caused by gunshots, dynamic lighting particles, or moving objects that occlude/dis-occlude lighting randomly for a shading point. This variance is present as we use temporal histories. It leads to sample count overestimation. We demonstrate it in the same setup in Fig. 2(f)–(j), lighting intensity changes over time is shown in Fig. 2(g). It leads to high sample count Fig. 2(i) across space to reduce the high variance Fig. 2(j) induced by temporal lighting changes. It leads to sample count increase by 242% in this example. It is critical to mitigate the effect of dynamic lighting changes for real-time performance.

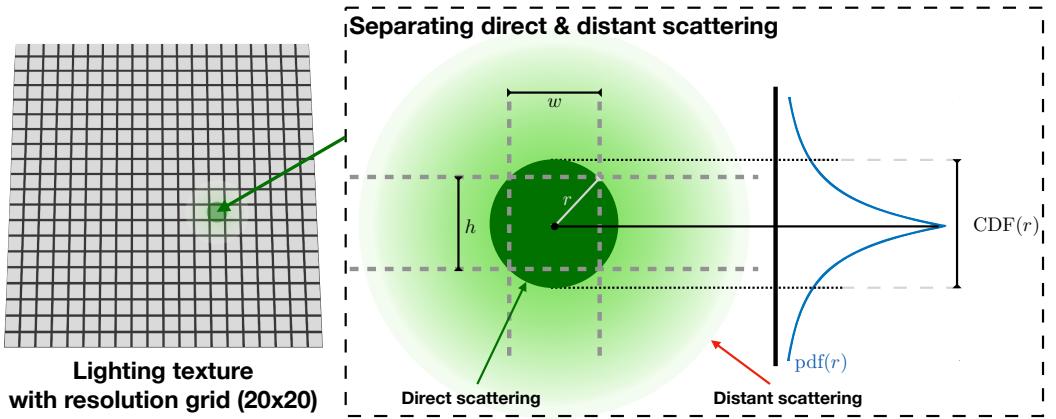


Fig. 3. Subsurface scattering with sampling resolution.

5 UNIFICATION OF SCATTERING

Scattering decomposition. Due to the resolution of $B(p, q, \omega_o)$ in practice, the sampling resolution is bounded by texel size $\mathbf{t} = (w, h)$. Denote the diffuse scattering sampling radius as $r_0 = Z\sqrt{w^2 + h^2}/2$, where Z is the depth of p , we divide the region into a direct scattering (or diffuse) region $\partial\Omega_d$, and distant scattering region $\partial\Omega_s$, $\partial\Omega = \partial\Omega_d \cup \partial\Omega_s$ as shown in Fig. 3. From Eq. 3 we have:

$$L_o(p, \omega_o) = L_d(p, \omega_o) + L_s(p, \omega_o) \quad (12)$$

$$= \int_{\partial\Omega_d} R(r_q) b(p, q, \omega_o) dq + \int_{\partial\Omega_s} R(r_q) b(p, q, \omega_o) dq \quad (13)$$

$$\approx B(p, p, \omega_o) \cdot cdf(r_0) + \int_{\partial\Omega_s} R(r_q) b(p, q, \omega_o) dq \quad (14)$$

where the first term in Eq. 14 is the analytic result after a change of variables from the Cartesian coordinate system to the polar coordinate system with the assumption that the covered region of direct scattering has constant lighting and it can be approximated by the discrete representation $B(p, p, \omega_o)$. Because we have $\int_0^{r_0} 2\pi r R(r) dr = A \cdot cdf(r_0)$ by Christensen and Burley [2015]’s work where A is the surface albedo. We ignore this constant for simplicity. It can be introduced back with a direct multiplication after subsurface scattering [Xie et al. 2020]. If we deploy importance sampling for this formula with $r_q \sim pdf(r)$, and the corresponding cumulative density function (cdf) inverse $g(\xi) = cdf^{-1}(\xi)$, the numerical approximation is

$$E(L_o(p, \omega_o)) \approx B(p, p, \omega_o) \cdot cdf(r_0) + \frac{1}{m} \sum_{i=n-m+1}^n \frac{2\pi r_{q_i} R(r_{q_i}) B(p, q_i, \omega_o)}{pdf(r_{q_i})} (1 - cdf(r_0)). \quad (15)$$

where $r_{q_i} = g((1 - \xi_i) \cdot cdf(r_0) + \xi_i)$, n is the sample count for subsurface scattering, and $m = n(1 - cdf(r_0))$ is the sample count for distant scattering. In this formulation, we directly use the pre-integrated lighting for diffuse, and weight between diffuse and scattering based on cdf .

Generalization. With Eq. 15, we summarize the real-time subsurface scattering model for a given ω_o as:

$$L_o(p) = (1 - \gamma) \cdot B(p, p) + \gamma \cdot \mathcal{L}_s(p, R_F, \partial\Omega_s) \quad (16)$$

where γ is the distant scattering energy ratio to blend the pre-integrated direct lighting and un-normalized distant subsurface scattering $\mathcal{L}_s(p, S_F, \partial\Omega_s)$, R_F is the corresponding subsurface scattering profile. Table 2 shows the realization of γ and R_F for different models.

Table 2. The realization of different subsurface scattering models.

	γ	R_F	Reference
Normalized Burley	$1 - cdf_b(r_0)$	R_b	[Burley 2015]
Dipole	$1 - cdf_d(r_0)$	$R_{d,ss+ms}$	[Jensen et al. 2001]
Artist Friendly Dipole	c	$R_{d,ms} - R_{G_0}$	[Jimenez et al. 2015]

Specialization. The generalization provides us a way to reason existing subsurface scattering model and whether it is applied in the same way for real-time rendering. For example, the artist friendly separable subsurface scattering [Jimenez et al. 2015] uses a constant blending factor c (strength) and a modified profile $\tilde{R}_d = R_{d,ms} - R_{G_0}$ where R_{G_0} is the Gaussian approximations with the minimal variance. The result is consistent with offline dipole models, only when:

$$c = \frac{(1 - cdf_d(r_0)) \cdot (B(p, p) - \mathcal{L}_s(p, R_{d,ss+ms}, \partial\Omega_s))}{B(p, p) - \mathcal{L}_s(p, \tilde{R}_d, \partial\Omega_s)} \quad (17)$$

Unified representation. Since γ is dependent on parameters of the CDF (e.g., diffuse mean free path for Burley), resolution and depth, we are able to enable 0 spp for distant scattering when most scattering is less or equal to one pixel as

$$\hat{m}_{(i)} = \begin{cases} 0 & \gamma < \epsilon_u \\ \hat{n}_i \cdot \gamma & \text{otherwise} \end{cases} \quad (18)$$

where ϵ_u is a small constant to determine when distant subsurface scattering is not performed. Fig. 4 shows an example of scattering regions with different ϵ_u . Most of the walls do not need distant scattering when $\epsilon_u = 0.05$. Note that the estimator becomes biased due to energy loss when $\epsilon_u > 0$. As with other uses of biased estimators in rendering, this can remove unnecessary samples for distant scattering and variance tracking, but needs to be used carefully.

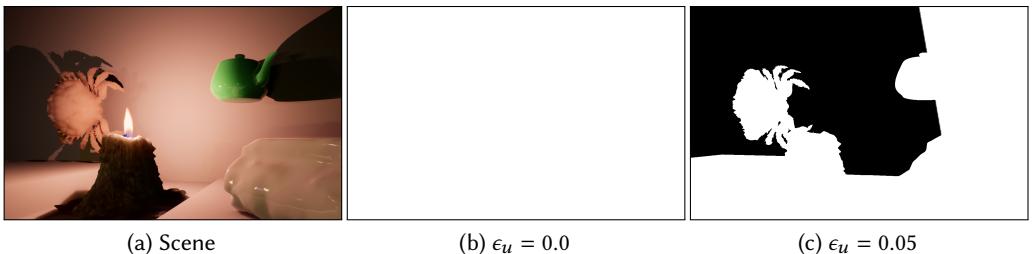


Fig. 4. Direct/diffuse region (black) and direct+distant (white) (b,c) for scene (a). The vertical line on the wall (c) is the boundary where only 5% of scattering energy is from distant scattering.

With this formulation, instead of estimating \hat{n}_i first and then calculating the sample count m_i for distant scattering, we can directly estimate \hat{m}_i with Eq. 8 with the history tuple $\mathcal{H}_i = (\mu_i, \sigma_i^2, \bar{m}_i)$. Then the target quality σ_0^2 is set for distant scattering. This switch also implies that the variance contribution of the direct scattering due to temporal change (e.g., jittering and lighting change) has been removed. It will not affect the sample count estimation.

6 CONTROL VARIATES

We use CV to reduce the influence of temporal variance. Starting from Eq. 11 and 14, we introduce the control variable at time t for position p as $g(q, t)$ where p is neglected for simplicity. Then we have time dimension in distant subsurface scattering as

$$L_s(t) = \alpha(t) \cdot G(t) + \underbrace{\int_{\partial\Omega_s} R(r_q) \beta(q, t) - \alpha(t) \cdot g(q, t) dq}_{Res(t)} \quad (19)$$

Denote $f(q, t) = R(r_q) \beta(q, t)$, and suppose we have a realization of $g(q, t)$ whose known integral is $G(t)$, which captures most of the temporal varying component. The optimal coefficient can be estimated with $\alpha^* = Cov(f, g) / Var(g)$ after sampling using the m_t samples. However, we might not have enough samples in one frame to get a good estimation. Instead, we use the covariance of the two batch means to estimate the CV coefficient and use history to improve the estimation, because we have $Cov(X, Y) = nCov(\bar{X}, \bar{Y})$ (see supplementary). It leads to $\alpha^* = Cov(\bar{X}, \bar{Y}) / Var(\bar{Y})$. This enables us to estimate the coefficient with temporal observations over time with online covariance estimation instead of just in a single frame. Then, the actual sample count could be estimated based on the residual component $Res(t)$ with minimal temporal variance.

6.1 Theory

Before running into the detail of the online updating algorithm, we provide a simplified theory to guide the algorithm design. We assume that the temporal change component can be independently separated out from the function to integrate $f(q, t)$ and the control variable $g(q, t)$.

Namely, we have three random variables T , F , and G . TF is the function to estimate, TG is the control variable. Then the general optimal CV coefficient is

$$\alpha^* = \frac{Cov(TF, TG)}{Var(TG)}. \quad (20)$$

When T is independent from F and G , the variance for $\langle TF \rangle$ and the residual variance are

$$Var(\langle TF \rangle) = Var(T(F - \alpha G)) + Var(\alpha TG) + 2Cov(T(F - \alpha G), \alpha TG) \quad (21)$$

$$Var(T(F - \alpha G)) = E(T)^2 Var(F - \alpha G) + Var(T) Var(F - \alpha G) + Var(T) E(F - \alpha G)^2 \quad (22)$$

where $\langle TF \rangle$ is an unbiased estimator. The goal in this paper is to reduce the variance contribution of T to $V(\langle TF \rangle)$ during variance tracking. For distant subsurface scattering as illustrated in Eq. 19 when the lighting intensity $T = I(t)$ is independent from the scattering function F and the control variable G , we have

$$F = \frac{L_s(t_r)}{I(t_r)} = \int_{\partial\Omega_s} R(r_q) \frac{\beta(q, t_r)}{I(t_r)} dq, \quad G = \frac{G(t_r)}{I(t_r)} = \int_{\partial\Omega_s} \frac{g(q, t_r)}{I(t_r)} dq, \quad (23)$$

$$TF = L_s(t) = I(t) \frac{L_s(t_r)}{I(t_r)}, \quad TG = G(t) = I(t) \frac{G(t_r)}{I(t_r)}, \quad (24)$$

where $I(t_r)$ is a reference intensity $I(t_r) \neq 0$ at time t_r . This assumption holds when the intensity of all lights (e.g., point light, directional light and spotlight) are controlled by a single intensity parameter $I(t)$. Because we have a linear relationship between the intensity of incoming lights and the exitance radiance as shown in Eq. 3 and Eq. 4. Although $g(q, t)$ is unknown, it does not affect the reasoning in this section. Please refer to Section 6.2 for a concrete realization.

6.1.1 In-frame standard control variable. In standard CV, the control variable requires Monte Carlo sampling and $E(TG)$ is expected to be known. Since T is independent, we have $E(TG) = E(T)E(G)$. However, $E(T)$ is unknown. Even though our proposed algorithm can estimate $E(T)$ to some extent, the MC sampling of G brings in variance, which makes the estimation more vulnerable. Fig. 5 illustrates the basic idea by using our CV coefficient updating algorithm. To introduce temporal dynamics, the intensity follows a sine function. It demonstrates an ability to adjust α under both static and dynamic lighting (see supplementary document).

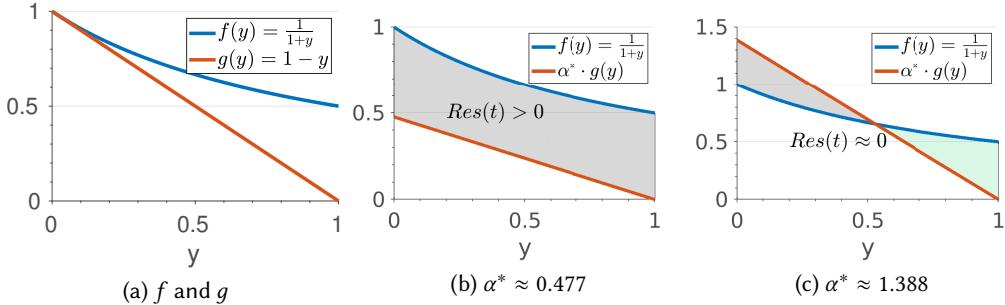


Fig. 5. Illustration of our novel application of CV. For function f and g (a), a standard CV coefficient estimation results in (b) with a residual vulnerable to temporal variance. However, our method can find the coefficient that leads to (b) during time-invariant scenarios, but to (c) during time-variant scenarios, where the residual is near zero. The residual variance is less vulnerable to temporal change based on Eq. 22.

Even worse, as TG and TF need to be sampled in a correlated way. This often means the bandwidth demand doubles in real-time rendering. Monte Carlo sampling leads to further cache incoherence. This is against the idea of real-time adaptive sampling, where bandwidth demand is minimized by reducing sample count with extremely low overhead. Because of this, applying standard CV to maintain stability for real-time adaptive sampling does not seem to be an attractive feature even when we have demonstrated some capability in Fig. 5.

6.1.2 In-frame constant control variable. To deal with the bandwidth demand hazard, a known in-frame constant can be used as the control variable. Namely, $E(G)$ is constant with $V(G) = 0$. This only adds a low overhead as one texture fetch. Under this condition, however, we do have a valid best CV coefficient derived from Eq. 20 as:

$$\alpha^* = \frac{E(F)}{E(G)} = \frac{E(TF)}{E(TG)}. \quad (25)$$

With this formula, the optimal CV coefficient for the example in Fig. 5 has $\alpha^* = \frac{\int_0^1 \frac{1}{1+y} dy}{\int_0^1 1-y dy} = \ln(4) \approx 1.386$, and $E(F - \alpha^* G) = 0$. Then Eq. 21 and Eq. 22 simplify to

$$Var(\langle TF \rangle) = Var(T(F - \alpha^* G)) + Var(T)E(F)^2 \quad (26)$$

$$Var(T(F - \alpha^* G)) = (Var(T) + E(T)^2)Var(F) \quad (27)$$

This analytic CV coefficient is the reason why the CV coefficient in Fig. 5 under dynamic lighting is approximately 1.386. When $(F - \alpha G) \rightarrow 0$, the right term diminishes in both Eq. 21 and Eq. 22, leaving only one controllable variance $Var(F)$ in the residual variance that should be considered for variance tracking. More specifically for subsurface scattering with Eq. 23 and Eq. 24, this optimal

CV coefficient is

$$\alpha^*(t) = \frac{E(L_s(t_r))}{E(G(t_r))} = \frac{E(I(t)L_s(t_r))}{E(I(t)G(t_r))} = \frac{E(L_s(t))}{G(t)} \quad (28)$$

where $G(t)$ is constant at frame t and $I(t) \neq 0$. Then, our online algorithm for subsurface scattering is designed with the following guidance:

- (1) Use analytic optimal CV from Eq. 25 for regions where intensity dynamics T is independent from F and G .
- (2) Adaptively switch to use online estimation of Eq. 20 to estimate the CV coefficient when the assumption does not hold.

6.2 Online solution

In this section, a novel online covariance estimation method is first introduced. We then provide our online CV coefficient and sample count joint estimation algorithm. To provide a concrete example for α^* estimation that separates out time-variant signal, we select one reasonable realization of f, g as

$$f(q, t) = R(r_q)\beta(q, t), \quad \bar{f} \approx \gamma\mathcal{L}_s(t) \quad (29)$$

$$g(q, t) = R(r_q)\beta(t), \quad \bar{g} = G(t) = \gamma B(t) \quad (30)$$

Note that \bar{f} and \bar{g} are the MC sampling result of the integration at time t . They are not necessarily equal to the analytic integration if the control variable requires Monte Carlo sampling. However, we use in-frame constant control variable to maintain temporal stability and deal with the bandwidth hazard. Therefore, $\bar{g} = G(t)$. $\beta(t)$ is the lighting contribution at p itself at time t . It leads to a constant integration of $\gamma B(t)$, where $B(t)$ is a texture fetch directly at texel p in the pre-integrated lighting texture. Since the algorithm will run for each p , we ignore p for simplicity (some important symbols are listed in Table 1). Note that this fetch can also be queried from different level of details to add in the correlation of temporal intensity change of surrounding lighting for further optimization. Exploring this is outside the scope of this paper. With this realization in large flat lighting region, Eq. 28 leads to

$$\alpha^* = 1, \quad (31)$$

the *main part separation*, and we have zero variance during dynamic lighting.

6.2.1 Exponential moving covariance (EMC). For two random variable X, Y that are incrementally observed according to time as $\{x_0, x_1, \dots, x_t\}$ and $\{y_0, y_1, \dots, y_t\}$ with a constant weight of η , the exponential moving covariance between them at time t ($t \geq 1$) is:

$$\begin{aligned} Cov_t(X, Y) = & (1 - \eta)Cov_{t-1}(X, Y) + \\ & \eta(1 - \eta)(x_t - \mu_{t-1})(y_t - v_{t-1}) \end{aligned} \quad (32)$$

where x_t, y_t are the current observation, μ_{t-1}, v_{t-1} are the corresponding exponential moving average at $t - 1$, and $Cov_0(X, Y) = 0$. Since there is no reference in the scientific literature, we provide a detailed proof in the supplementary material. The new covariance formulation enables direct calculation of CV coefficient as $\alpha_t(X, Y) = Cov_t(X, Y)/Var_t(Y)$. Then, we can derive a more generalized matrix form.

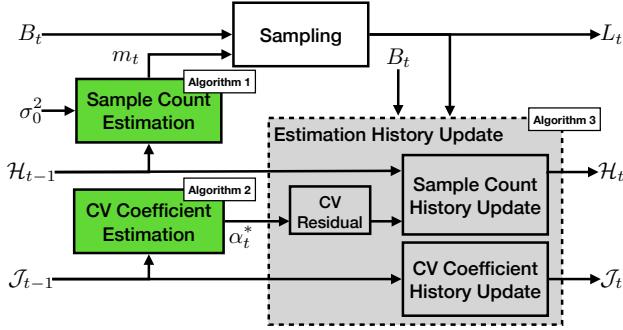


Fig. 6. The online control variates based adaptive sampling diagram at frame time t . With sample count history \mathcal{H}_{t-1} and CV history \mathcal{J}_{t-1} , we estimate the sample count required at time t based on the control variates residual, trying to meet the target variance level σ_0^2 .

6.2.2 Exponential moving covariance matrix (EMCM). If we have a time series vector $\mathbf{Z}_t \in \mathbb{R}^{n \times 1}$ with the exponential moving average at time t as $\zeta_t \in \mathbb{R}^{n \times 1}$, then the covariance matrix $\Sigma_t (t \geq 1)$ is

$$\Sigma_t = (1 - \eta)\Sigma_{t-1} + \eta(1 - \eta)(\mathbf{Z}_t - \zeta_{t-1})(\mathbf{Z}_t - \zeta_{t-1})^T \quad (33)$$

where $\Sigma_0 = \mathbf{0}^{n \times n}$. This is different from EWMA covariance estimator used in stock analysis[Guldmann et al. 1995] as

$$\tilde{\Sigma}_t = (1 - \eta)\tilde{\Sigma}_{t-1} + \eta(\mathbf{Z}_t - \zeta_{t-1})(\mathbf{Z}_t - \zeta_{t-1})^T \quad (34)$$

The estimator uses the previous history at $t - 1$ to predict the value at t , solving for the estimated covariance between X, Y :

$$\widetilde{\text{Cov}}_t(X, Y) \approx E_t((X - \mu_{t-1})(Y - v_{t-1})) \quad (35)$$

where $E_t(\cdot)$ calculates the exponential weighted average at t . While the equation we resolve is to calculate the covariance as

$$\text{Cov}_t(X, Y) = E_t((X - \mu_t)(Y - v_t)) \quad (36)$$

(see supplementary). For rapid changing frames with dynamic lighting, it is preferable to have a direct calculation instead of prediction to get the CV coefficient. Then, the coefficient matrix is $\mathbf{A}_t = \text{diag}(\sum_1^n (\mathbf{e}_i^T \Sigma_t \mathbf{e}_i)^{-1} \Sigma_t)$ where \mathbf{e}_i is the i th matrix basis, and $\text{diag}(\cdot)$ creates the diagonal matrix from a vector.

6.2.3 Coefficient boundary. Since the numerical estimation might lead to instability (e.g., oscillating larger or being NaN according to time), we bound the range of CV coefficient for subsurface scattering with two considerations:

- (1) Both diffuse, B , and distant scattering results are non-negative, the maximum residual cannot be larger than the distant scattering result \bar{f} , thus $\text{Res}_{\max} \leq \bar{f}$.
- (2) Residual can be negative, however, we anticipate that the minimal residual can be raised to non-negative by $G(t)$ as $\text{Res}_{\min} + \bar{g} \geq 0$. Thus, we have $\text{Res}_{\min} \geq -\bar{g}$.

With $\text{Res} = \bar{f} - \alpha \cdot \bar{g}$ we have the bound $D \in [0, (\bar{f} + \bar{g})/\bar{g}]$. If α is clamped by D , we denote it as $\alpha|_D$. Since $\text{Cov}_t(\bar{f}, \bar{g})$ and $\text{Var}_t(\bar{g})$ can be zero for hard shadows, to deal with this issue, we added a small constant factor ϵ_α as $\alpha_t = \frac{\text{Cov}_t(\bar{f}, \bar{g}) + \epsilon_\alpha}{\text{Var}_t(\bar{g}) + \epsilon_\alpha}$. In this way, when both variables become zero, it could simplify to *main part separation* ($\alpha_t = 1$).

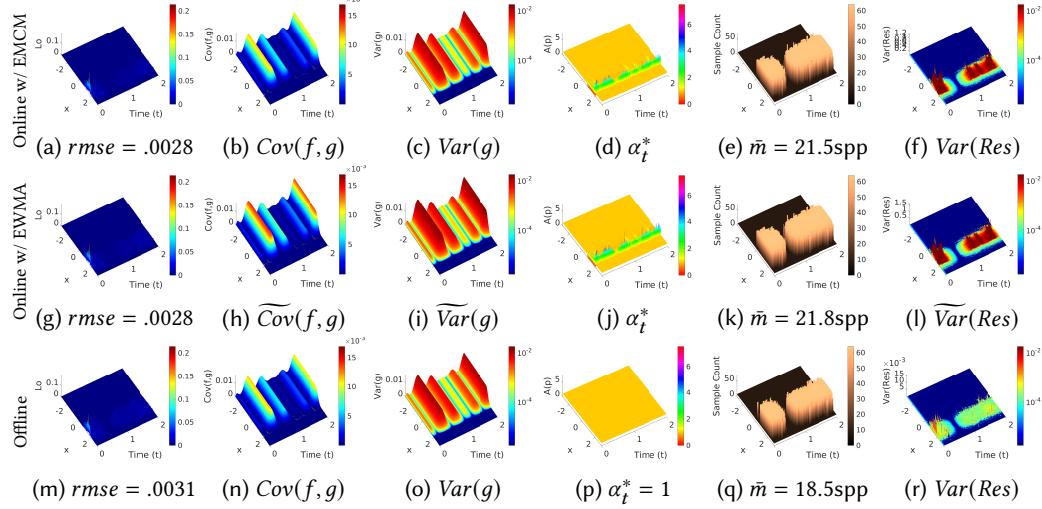


Fig. 7. Sample count m_t estimation with online and offline CV coefficients α_t^* estimation.

6.2.4 Online joint estimation algorithm. With the ability to calculate covariance matrix online, the online algorithm to estimate sample count m_t as well as the optimal CV coefficient α_t is provided in this section. Fig. 6 shows an overview of the online control variates based adaptive sampling diagram. It composes of three main parts: i) Sample count estimation, ii) CV coefficient estimation, and iii) Estimation history update.

Algorithm 1 Sample Count Estimation

Require: $\mathcal{H}_{t-1}, \gamma, \epsilon_u, \beta_{min}, \beta_{max}$

- 1: **if** $\gamma < \epsilon_u$ **then**
- 2: $m_t = 0$
- 3: **else**
- 4: Update \hat{m}_t with Eq. 8
- 5: $m_t = \hat{m}_t|_{[\beta_{min}, \beta_{max}]}$
- 6: **end if**
- 7: **return** m_t

Sample count estimation. As shown in Algorithm 1, if most contributions come from direct scattering (Eq. 18), there is no need to perform distant scattering (Line 1-2). Otherwise, it will use the real-time adaptive sampling algorithm to estimate sample count in frame t (Line 4). To have adequate observations and also consider the computing capability, the estimated sample count is restricted within $[\beta_{min}, \beta_{max}]$ (Line 5).

Algorithm 2 CV Coefficient α_t^* Estimation

Require: $\Sigma_{t-1}, \epsilon_\alpha, D$

- 1: $\alpha_{t-1} = \frac{\Sigma_{t-1} \cdot y + \epsilon_\alpha}{\Sigma_{t-1} \cdot y \cdot y + \epsilon_\alpha}|_D$
- 2: **return** α_{t-1}

CV coefficient estimation. With the proposed exponential moving covariance matrix, $\Sigma_t = \begin{pmatrix} Var_t(\bar{f}) & Cov_t(\bar{f}, \bar{g}) \\ Cov_t(\bar{g}, \bar{f}) & Var_t(\bar{g}) \end{pmatrix}$. We can easily calculate the CV coefficient at $t - 1$ as $\alpha_{t-1} = (\Sigma_{t-1}.yy + \epsilon_\alpha) / (\Sigma_{t-1}.yy + \epsilon_\alpha)$ (Line 1 in Algorithm 2) and make it bounded by D to deal with instability. At last, the coefficient solution is approximated by the coefficient at $t - 1$ as α_{t-1} (Line 2). Note that if we use EWMA covariance estimator (Eq. 34), the coefficient would be approximated by covariance estimator at $t - 1$ using history from $t - 2$. If \bar{g} is an in-frame standard control variable, another potential solution for CV coefficient estimation is to calculate per-frame CV coefficient directly and use exponential moving average (EMA) for a good estimation as Fig. 5(b). However, it cannot lead to Fig. 5(c) to remove the temporal variance.

Algorithm 3 Estimation History Update

Require: $\mathcal{H}_{t-1}, m_t, \mathcal{J}_{t-1} = (\Sigma_{t-1}, \zeta_{n-1}), \alpha_t^*, t$

- 1: // Sample count history update
 - 2: $\mathcal{S}(t) = Res(t)$
 - 3: Update $\mathcal{H}_t = (\mu_t, \sigma_t^2, \bar{m}_t)$ based on Eq. 5-7
 - 4: // CV coefficient history update
 - 5: $Z_t = [\bar{f}, \bar{g}]$
 - 6: Update Σ_t with Eq. 33
 - 7: $\zeta_t = (1 - \eta)\zeta_{t-1} + \eta Z_t$
 - 8: $\mathcal{J}_t = (\Sigma_t, \zeta_t)$
 - 9: **return** $(\mathcal{H}_t, \mathcal{J}_t)$
-

Estimation history update. After sampling, we need to update the history buffer for both sample estimation and VC coefficient estimation: 1) *Sample count estimation*. After applying the control variates, the value we monitor is $Res(t)$ instead of $L_o(t)$. Because $L_o(t)$ contains both spatial and temporal variance, what represents the spatial variance most is $Res(t)$. So $Res(t)$ is set as the shading result for history update (Line 1). For storage efficiency, only luminance is used for $Res(t)$. 2) *CV coefficient estimation*. The exponential moving covariance matrix and the exponential moving average for \bar{f} and \bar{g} are updated between line 5-7. Note that the exponential moving coefficient for CV can be different from adaptive sampling to reduce the variance caused by α_t^* estimation. There are two considerations during implementation:

- (1) *Number of textures*. Line 8 indicates that we need to store 9 parameters, which would require three textures of *floatRGBA16*. But Σ_t is symmetric, and $\Sigma_t.xx$ is not used. Thus only 7 parameters (two textures) are actually needed to keep the history.
- (2) *Sampling value*. Since rendering solves $L_o(p, t)$, to efficiently compute the monitoring value $\mathcal{S}(p_t)$ and Z_t , we can derive another formulation without calculating intermediate values as

$$\bar{f} = L_o(t) - (1 - \gamma) \cdot B(t) \quad (37)$$

$$Res(t) = \bar{f} - \alpha_t^* \cdot G(t) \quad (38)$$

Fig. 7(a)-(l) shows an example of applying the algorithm with EWMA covariance estimator and EMCM. The sample count and quality are close. However, we have a slightly smaller sample count with EMCM. Moreover, the approximation of EWMA brought higher variance and covariance changes shown between Fig. 7 (c) and (i), (b) and (h). Therefore, we select EMCM. Note that Eq. 38 uses $G(t)$ instead of \bar{g} (no matter whether the standard or constant control variable is used) to remove temporal variance while still keeping spatial variance to avoid under-sampling in both static and dynamic lighting scenarios (see supplementary).

6.3 Offline CV coefficient estimation

From the online solution, we find an approximation, $\alpha_t^* = 1$, leading to an even more efficient implementation for real-time applications.

Fig. 7(d) shows that for most regions in time, even with time-variant lighting condition, α_t^* remains 1. Fig. 7(m)–(r) show the corresponding states if we set $\alpha_t^* = 1$. The quality is a little worse (.0031>.0028) with a reduced average sample count (-3 spp), but we can save one texture and the corresponding calculation.

Our online solution requires two textures. When the memory capacity or bandwidth budgets are tight, one more texture per pass might be too demanding.

Table 3. The mean metrics across all frames and the max performance ratio for adaptive sampling (AS), adaptive sampling with Online CV coefficient (+OCV), and with constant CV coefficient (+CCV).

	Sample Count (spp)				Sampling Pass Time (ms)				PSNR (dB)			
	AS	+OCV	+CCV	Max ratio	AS	+OCV	+CCV	Max ratio	AS	+OCV	+CCV	Max Ratio
Flash lighting	61.97	29.58	26.88	x2.75	1.47	0.80	0.75	x2.79	51.19	49.55	49.43	x0.94
Regular lighting	26.51	18.32	17.82	x4.00	3.39	2.55	2.58	x3.11	47.85	47.57	47.55	x1.01
Dynamic scene	22.01	14.70	14.13	x2.02	2.02	1.54	1.52	x1.98	50.62	49.11	49.11	x0.91

7 IMPLEMENTATION AND RESULT

We implemented our algorithm in Unreal Engine 4 (UE4) in the subsurface-scattering postprocess pass. The target quality has $\sigma_0^2 = 0.0001$, $\eta = 0.2$, $\eta_{cv} = 10^{-6}$, $\kappa = 0.2$, $\epsilon_u = 0.01$ and $\epsilon_\alpha = 10^{-6}$, the sample count is clamped between 8 and 64 spp. η_{cv} is the exponential moving coefficient for CV history update. We created three scenes to capture typical dynamic lighting changes to show how our proposed algorithm works in whole scene subsurface scattering:

- (1) Flashing lighting. Flashing light is used to simulate gunshots or lightning on human face, using a directional light with intensity $I(t) = \sin(2\pi ft) + 1$ with $f = 5\text{Hz}$.
- (2) Regular lighting. We simulate regular lighting changes with a light switching on and off, and flickering candle lights. The directional light is turned on/off every 3 seconds.
- (3) Dynamic scene. Dynamic moving boids are added to the scene creating dynamic shadowing when they move around a candle.

We choose to use three metrics for the evaluation.

- (1) Sample count (spp). The per-frame average samples per pixel. It is calculated on subsurface regions only.
- (2) Sampling pass time (ms). The sampling pass time is only measured for the MC sampling process. A 3rd-order median filter is applied to remove unstable measurements.
- (3) PSNR (dB). The PSNR is measured on the luminance of the subsurface scattering region. The ground truth image is captured with 1024 spp per frame.

To make the results reproducible, we use fixed random seed and UE4 *Sequencer* to render a standard dynamic range (SDR) 60 frame-per-second (FPS) avi, with sample count and PSNR measured per frame. To get the sampling pass time, the built-in csv log is used during sequencer recording. The performance is measured with a resolution of 2560x1440p on an NVIDIA RTX 2080Ti.

Fig. 8 shows the performance and quality test results for the selected three scenes over 300 frames. For each test scene, we show the scene, two captures to demonstrate temporal dynamics, the sample count texture for adaptive sampling, the texture with online CV, the sample count, sampling pass time, and PSNR. Table 3 shows the corresponding average over time. Since it is very

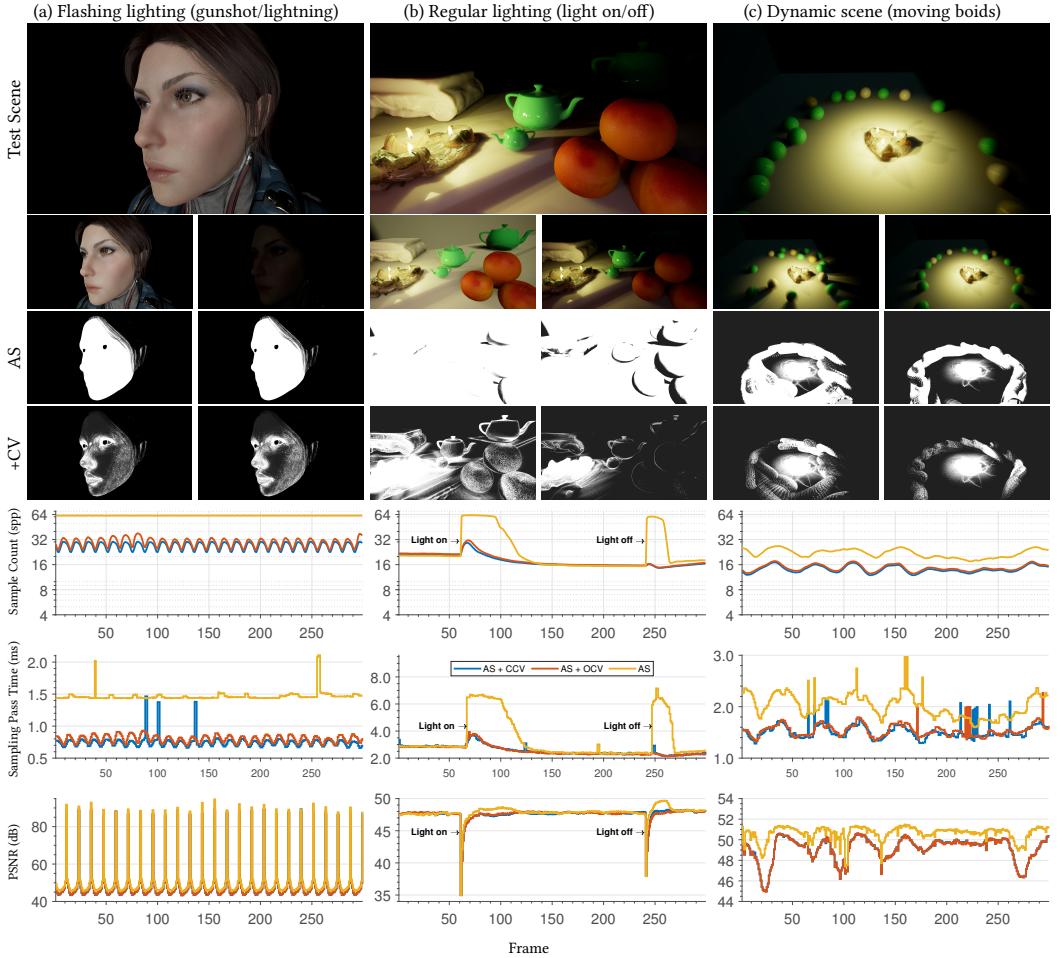


Fig. 8. Real-time CV performance and quality test at resolution 2560x1440 for the three test scenes: (a) Flashing lighting with additional snapshots at frame 101 and 108, (b) Regular lighting at 71 and 251, and (c) Dynamic scene at 25 and 200.

important to always maintain high performance in real-time rendering, the max performance ratios before and after adding CV are also shown.

Max performance ratio. Fig. 8(b) shows a good example when light is switching on/off. Without control variates, the sampling pass time increases up to $\times 3.11$ at frame 252 (7.15 ms vs. 2.33 ms) compared to with-CV even when quality is not increased. With CV, the pass time is more stable and friendly to real-time applications. Note that, in the sampling pass time of Fig. 8, some sudden increase in pass time can be observed. We believe it is due to memory incoherence since the samples are based on importance sampling.

Sample count reduction. The reason we can have better performance is that CV leads to lower sample counts, thus less pass rendering time. During continuously frequent dynamic lighting change (Fig. 8(a)), temporal variance leads to high sample counts (61.97 spp) and an average of 1.47 ms for the sampling pass. CV leads to consistently lower average sample count (17.82 spp) and pass time (0.75 ms) with quality drops of 1.76 dB.

Quality effects. We observe a slight quality drop after applying CV. the quality drops the highest during high dynamic scenes (Fig. 8(c)) with fast moving subsurface objects, down $\times 0.91$ when compared with adaptive sampling alone at frame 25. However, we think the absolute PSNR is high enough for the introduced performance.

Online vs. constant CV. The average quality is expected to be better with online CV. We tend to have more samples during our 1D illustration example in Fig. 7 as the algorithm tends to perform oversampling. In Fig. 8, a similar small increase in sample count is detected during our 3D scene tests. The constant CV seems to be more efficient regarding the memory usage of the online CV in real-time rendering.

8 CONCLUSION

In this paper, we proposed a *real-time subsurface control variates* algorithm to reduce sample count demands in dynamic lighting and scenes. It minimized both regions and sample demands to perform stable subsurface scattering with a unified representation. The CV coefficient is based on our novel exponential moving covariance matrix, which should help in all research and application domains that rely on accurate online exponential moving covariance matrix. A bandwidth-friendly CV coefficient approximation is also provided for real-time rendering engine integration.

ACKNOWLEDGMENTS

This work is supported in part by Epic Games and inspired by an observation by John Hable. Additionally, we thank the rendering team at Epic Games for thoughtful discussions to motivate this work. We particularly give our special thanks to all the anonymous reviewers for their valuable feedback which improved the quality of this paper. We also acknowledge free assets from the *Unreal Engine* Marketplace, including candles and marble from *Megascans Goddess Temple*, Paragon: Lt. Belica, orange from *Megascans - Definitive Fruits*, and Dark Finger Reef Crab from Three D Scans.

REFERENCES

- Carol Alexander. 1999. Risk Management and Analysis. Volume 1: Measuring and Modelling Financial Risk.
- Brent Burley. 2015. Extending the Disney BRDF to a BSDF with integrated subsurface scattering. In *SIGGRAPH Course: Physically Based Shading in Theory and Practice*. ACM, New York, NY, 19 pages.
- Per H. Christensen and Brent Burley. 2015. *Approximate Reflectance Profiles for Efficient Subsurface Scattering*. Technical Report. Pixar.
- Holger Dammertz, Johannes Hanika, Alexander Keller, and Hendrik Lensch. 2010. A hierarchical automatic stopping condition for Monte Carlo global illumination. In *Eurographics WSCG 2010: Full Paper Proceedings*. Václav Skala-UNION Agency, 159–164.
- Eugene d’Eon and David Luebke. 2007. Advanced techniques for realistic real-time skin rendering. *GPU Gems 3*, 3 (2007), 293–347.
- Shaohua Fan, Stephen Chenney, Bo Hu, Kam-Wah Tsui, and Yu-chi Lai. 2006. Optimizing control variate estimators for rendering. *Computer Graphics Forum* 25, 3 (2006), 351–357.
- Tony Finch. 2009. Incremental calculation of weighted mean and variance. *University of Cambridge* 4, 11-5 (2009), 41–42.
- Evgenii Golubev. 2018. Efficient screen-space subsurface scattering using Burley’s normalized diffusion in real-time. In *SIGGRAPH 2018 Courses: Advances in Real-Time Rendering*.
- Till Guldmann, Peter Zangari, Jacques Longerstaey, John Matero, and Scott Howard. 1995. *RiskMetrics Technical Document*. Technical Report. Morgan Guaranty Trust Company.
- Timothy C Hesterberg and Barry L Nelson. 1998. Control variates for probability and quantile estimation. *Management Science* 44, 9 (1998), 1295–1312.
- Henrik Wann Jensen, Stephen R Marschner, Marc Levoy, and Pat Hanrahan. 2001. A practical model for subsurface light transport. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, New York, NY, 511–518.
- Jorge Jimenez, Veronica Sundstedt, and Diego Gutierrez. 2009. Screen-space perceptual rendering of human skin. *ACM Transactions on Applied Perception (TAP)* 6, 4 (2009), 23.

- Jorge Jimenez, Károly Zsolnai, Adrian Jarabo, Christian Freude, Thomas Auzinger, Xian-Chun Wu, Javier von der Pahlen, Michael Wimmer, and Diego Gutierrez. 2015. Separable Subsurface Scattering. *Computer Graphics Forum* 34, 6 (Sept. 2015), 188–197.
- David Kirk and James Arvo. 1991. Unbiased sampling techniques for image synthesis. *ACM SIGGRAPH Computer Graphics* 25, 4 (1991), 153–156.
- Ivo Kondapaneni, Petr Vévoda, Pascal Grittmann, Tomáš Skřivan, Philipp Slusallek, and Jaroslav Krivánek. 2019. Optimal multiple importance sampling. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–14.
- Stephen S Lavenberg, Thomas L Moeller, and Peter D Welch. 1982. Statistical results on control variables with application to queueing network simulation. *Operations Research* 30, 1 (1982), 182–202.
- Bochang Moon, Nathan Carr, and Sung-Eui Yoon. 2014. Adaptive rendering based on weighted local regression. *ACM Transactions on Graphics (TOG)* 33, 5 (2014), 1–14.
- Thomas Müller, Fabrice Rousselle, Alexander Keller, and Jan Novák. 2020. Neural Control Variates. *ACM Trans. Graph.* 39, 6, Article 243 (Dec. 2020), 19 pages. <https://doi.org/10.1145/3414685.3417804>
- Barry L Nelson. 1990. Control variate remedies. *Operations Research* 38, 6 (1990), 974–992.
- Jan Novák, Andrew Selle, and Wojciech Jarosz. 2014. Residual ratio tracking for estimating attenuation in participating media. *ACM Trans. Graph.* 33, 6 (2014), 179–1.
- Eric Penner and George Borshukov. 2011. Pre-integrated skin shading. *GPU Pro* 2 (2011), 41–55.
- Matt Pharr, Wenzel Jakob, and Greg Humphreys. 2016. *Physically based rendering: From theory to implementation*. Morgan Kaufmann, Cambridge, MA.
- Brian D Ripley. 2009. *Stochastic simulation*. Vol. 316. John Wiley & Sons, New Jersey.
- Fabrice Rousselle, Wojciech Jarosz, and Jan Novák. 2016. Image-space control variates for rendering. *ACM Transactions on Graphics (TOG)* 35, 6 (2016), 1–12.
- László Szécsi, Mateu Sbert, and László Szirmay-Kalos. 2004. Combined correlated and importance sampling in direct light source computation and environment mapping. *Computer Graphics Forum* 23, 3 (2004), 585–593.
- Ruey S Tsay. 2005. *Analysis of financial time series*. Vol. 543. John Wiley & Sons, New Jersey.
- Greg Welch, Gary Bishop, et al. 1995. An introduction to the Kalman filter.
- BP Welford. 1962. Note on a method for calculating corrected sums of squares and products. *Technometrics* 4, 3 (1962), 419–420.
- Tiantian Xie, Marc Olano, Brian Karis, and Krzysztof Narkowicz. 2020. Real-time subsurface scattering with single pass variance-guided adaptive importance sampling. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 3, 1 (2020), 1–21.