Spatio-Temporal Control Variates with ReSTIR for Real-Time Rendering

ZHONG SHI, BNRist, MOE-Key Laboratory of Pervasive Computing, Department of CS&T, Tsinghua University, China
CUNHAO WU, BNRist, MOE-Key Laboratory of Pervasive Computing, Department of CS&T, Tsinghua University, China
LIFAN WU, NVIDIA, USA
KUN XU*, BNRist, MOE-Key Laboratory of Pervasive Computing, Department of CS&T, Tsinghua University, China

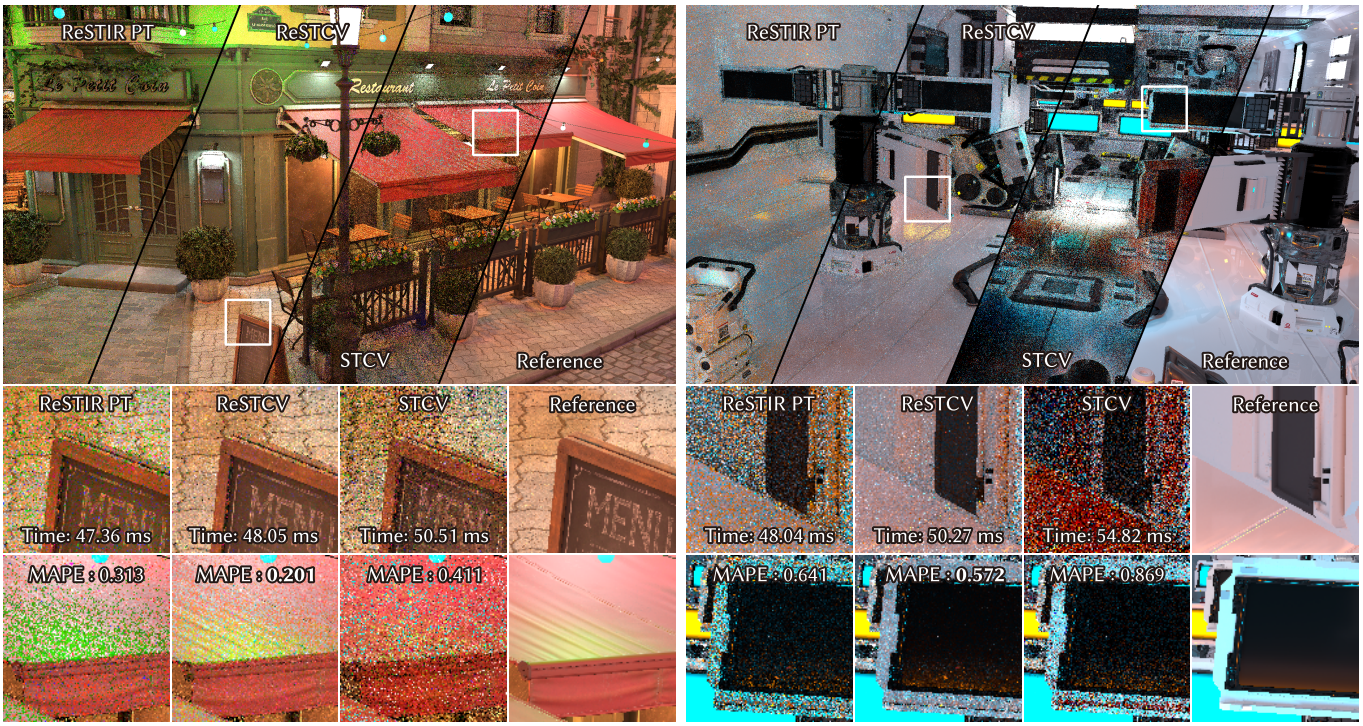


Fig. 1. We introduce ReSTCV, a novel framework that integrates Spatio-Temporal Control Variates (STCV) into ReSTIR for high-quality real-time path tracing. Standard ReSTIR PT often suffers from noticeable color noise, as it estimates pixel color from a single representative sample stored in a reservoir. Our STCV formulation mitigates this by accumulating path contributions across space and time as control variates, and constructing new estimators that leverage spatio-temporal coherence. By combining STCV with efficient ReSTIR-based estimation, our full ReSTCV algorithm significantly reduces variance and suppresses color noise, producing clean results that closely match the reference images.

Real-time path tracing demands high visual quality under extremely tight sampling budgets, often relying on reservoir-based spatio-temporal importance resampling (ReSTIR) to maximize sample quality. However, ReSTIR typically estimates the pixel integral using a single representative sample selected via a scalar target function (e.g., luminance). This inevitably leads to color noise in scenes with complex chromatic lighting or materials. In this work, we present *Reservoir-based Spatio-Temporal Control Variates* (ReSTCV), a novel framework that addresses this problem by integrating *Spatio-Temporal Control Variates* (STCV) into ReSTIR. We revisit image-space control variates—originally an offline technique—and adapt them for real-time rendering by spatio-temporal sample reuse. This unified approach combines the benefits of both techniques, enabling us to suppress color noise while maintaining the efficiency of ReSTIR. Our method introduces minimal computational overhead and requires only minor modifications to existing ReSTIR pipelines. We demonstrate that ReSTCV produces significantly cleaner images with stable colors across a variety of dynamic scenes, marking the first practical application of spatio-temporal control variates in real-time path tracing.

CCS Concepts: • **Computing methodologies** → **Ray tracing**.

Additional Key Words and Phrases: real-time rendering, control variates, resampled importance sampling, ReSTIR

*Kun Xu is the corresponding author.

Authors' Contact Information: Zhong Shi, BNRist, MOE-Key Laboratory of Pervasive Computing, Department of CS&T, Tsinghua University, Beijing, China, shizhong24@mails.tsinghua.edu.cn; Cunhao Wu, BNRist, MOE-Key Laboratory of Pervasive Computing, Department of CS&T, Tsinghua University, Beijing, China, wuch24@mails.tsinghua.edu.cn; Lifan Wu, NVIDIA, Redmond, USA, lifanw@nvidia.com; Kun Xu, BNRist, MOE-Key Laboratory of Pervasive Computing, Department of CS&T, Tsinghua University, Beijing, China, xukun@tsinghua.edu.cn.

SIGGRAPH Conference Papers '26, Los Angeles, CA, USA

ACM ISBN 979-8-4007-2554-8/26/07
https://doi.org/10.1145/3799902.3811113

ACM Reference Format:
Zhong Shi, Cunhao Wu, Lifan Wu, and Kun Xu. 2026. Spatio-Temporal Control Variates with ReSTIR for Real-Time Rendering. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '26), July 19–23, 2026, Los Angeles, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3799902.3811113

1 Introduction

Real-time path tracing has become increasingly practical with modern GPU hardware, enabling physically based lighting effects such as soft shadows and global illumination at real-time or interactive frame rates. However, strict performance constraints impose extremely limited sampling budgets, often allowing only a few path samples per pixel per frame. Under these constraints, effectively reusing path samples is critical for achieving high image quality. Consequently, modern real-time rendering systems rely heavily on spatio-temporal reuse, exploiting correlations across pixels and frames to amortize expensive light transport computation (e.g., in temporal anti-aliasing [Yang et al. 2020] and spatio-temporal denoising [Schied et al. 2017]). Designing principled path reuse strategies that reduce variance without introducing bias or excessive overhead remains a central challenge.

Reservoir-based spatio-temporal importance resampling (ReSTIR) [Bitterli et al. 2020; Lin et al. 2022] has emerged as a powerful framework for sample reuse in real-time path tracing. By repeatedly resampling candidates across space and time, ReSTIR efficiently reallocates samples toward important path contributions without introducing bias. However, ReSTIR's streaming formulation still shades each pixel from a single representative sample. When scalar path luminance is used as the target function for resampling, ReSTIR fits the distribution well, but the final vector-valued pixel estimate can remain dominated by the selected candidate. This creates a gap between the scalar intensity distribution and the per-channel color distribution, retaining per-channel variance and often appearing as visible color noise in scenes with chromatic light variation or complex surface materials. Moreover, the spatial and temporal correlations introduced by sample reuse can amplify this effect, producing visible "color-clumping" artifacts that persist even when overall variance is reduced (see §4).

Meanwhile, image-space control variates [Rousselle et al. 2016] offer a principled variance reduction technique by reusing correlated estimates from neighboring pixels or prior frames. By estimating the differences of path integrals between nearby pixels rather than computing their values directly, control variates can significantly reduce variance where strong spatial or temporal coherence exists. While effective in offline re-rendering and gradient-domain rendering, their application in real-time rendering has been under-explored. Traditional image-space control variate techniques rely on accurate auxiliary estimates and low-variance difference estimators; these assumptions are difficult to fulfill given limited sample budgets in real-time rendering.

In this work, we introduce ReSTCV, a *reservoir-based spatio-temporal control variates* framework tailored for real-time path tracing. Our method addresses two key challenges. First, it exploits spatio-temporal reuse to construct auxiliary estimates from neighboring pixels and previous frames, yielding stable control variates even under strict sampling budgets. Second, it leverages ReSTIR's reuse structure to estimate pixel differences efficiently.

By integrating control variates with ReSTIR, we retain ReSTIR's efficient reuse structure and sample distribution while replacing single-sample shading with accumulated path contributions from control variates. This substantially strengthens the final estimator, reducing variance that in ReSTIR often appears as color noise while preserving unbiasedness. Importantly, our method introduces negligible computational overhead compared to standard ReSTIR, requiring only a few localized code changes (see highlighted pseudocode in Listing 1).

We demonstrate that our method consistently reduces variance—leading to significantly less noticeable color noise—across a wide range of dynamic scenes and lighting conditions, as shown in Fig. 1. To the best of our knowledge, our work presents the first practical realization of image-space control variates in real-time path tracing, enabling principled variance reduction beyond ReSTIR alone. Our main contributions are:

- *Spatio-Temporal Control Variates* (STCV), a novel formulation that extends image-space control variates to the spatio-temporal domain for real-time path tracing (§5.1).
- A ReSTIR-based estimator that leverages reservoir samples to efficiently estimate pixel differences with significantly reduced variance (§5.2).
- *ReSTCV*, a practical algorithm that integrates STCV and ReSTIR into a unified framework with minimal overhead (§5.3).

2 Related Work

Control variates for rendering. Control variates are a fundamental variance reduction technique in Monte Carlo integration. They operate by subtracting a correlated auxiliary function with a known integral from the original integrand to reduce variance. Effective auxiliary functions can be constructed analytically from lighting distributions [Belcour et al. 2018; Szécsi et al. 2004] or estimated empirically from existing samples via regression [Crespo et al. 2021; Salaün et al. 2022; Vévoda et al. 2018]. More recently, neural networks have been employed to learn accurate auxiliary functions online [Müller et al. 2020]. When multiple control variates are available, their contributions can be optimally weighted, resulting in a formulation equivalent to multiple importance sampling with relaxed constraints [Hua et al. 2023; Kondapaneni et al. 2019]. Ratio control variates [Lu et al. 2025] divide the target integrand by the auxiliary function instead of subtracting it, providing better variance reduction for vector-valued functions.

Image-space control variates (ICV) [Rousselle et al. 2016] generalize this concept by exploiting spatial coherence in the image domain. By treating correlated estimates from neighboring pixels as control variates, ICV significantly reduces variance in applications such as re-rendering and gradient-domain rendering. Recently, Nicolet et al. [2023] propose recursive control variates for inverse rendering, which leverages rendering results from previous optimization steps. Xu et al. [2024] apply ICV to edited dynamic scenes and provide more accurate difference estimation by explicitly modeling residual

transport between frames. Yang and Moon [2025] extend ICV to utilize path tracing via common random numbers. Our method also builds upon the ICV framework but adapts it for spatio-temporal reuse in the context of real-time rendering, specifically addressing the challenges of low sample counts.

Gradient-domain rendering. Gradient-domain rendering [Hua et al. 2019] reduces variance by estimating finite differences (gradients) between correlated pixel integrals and reconstructing the final image via a Poisson solver. Gradient-domain metropolis light transport (GMLT) [Lehtinen et al. 2013] pioneered this approach using image-space gradients for rendering, while Kettunen et al. [2015] later adapted it to path tracing. Temporal gradient-domain path tracing [Manzi et al. 2016] further extends this to the temporal domain to enforce temporal coherence and reduce flickering in animations. Subsequent research has improved sampling efficiency through path reuse [Bauszat et al. 2017], adaptive sampling [Liang et al. 2024], better reconstruction methods [Josse et al. 2025; Tong and Hachisuka 2024], or reconstruction quality via unbiased generalized formulations [Yan et al. 2025]. Our approach shares the core insight of estimating pixel differences to reduce variance. However, rather than relying on standard gradient sampling, we leverage ReSTIR to efficiently compute low-variance difference estimates suitable for real-time rendering.

RIS and ReSTIR. ReSTIR [Bitterli et al. 2020; Wyman et al. 2023] enables efficient real-time direct illumination by resampling light candidates across space and time using weighted reservoirs, a technique rooted in resampled importance sampling (RIS) [Talbot 2005]. ReSTIR GI [Ouyang et al. 2021] extends this framework to handle global illumination. Generalized resampled importance sampling (GRIS) [Lin et al. 2022] formalizes these ideas and unifies the theoretical foundation of ReSTIR-style estimators.

A fundamental characteristic of ReSTIR is its reuse of raw path samples rather than pixel colors (i.e., final shading results by accumulating path contributions). To improve the efficiency of sample reuse, Wyman and Panteleev [2022] propose decoupling shading from resampling by reusing candidate samples from intermediate passes (rather than the final candidate only) for shading. Other approaches aggregate shading results across pixels [Lee and Chang 2025; Tokuyoshi 2024], though often at the cost of introducing bias.

3 Preliminaries

3.1 Image-space Control Variates

Based on the path integral formulation [Veach 1997], the color of pixel i is expressed as an integral of the path contribution function $f_i(\cdot)$ over the path space Ω_i:

$$F_i = \int_{\Omega_i} f_i(x)\,\mathrm{d}x, \tag{1}$$

where x is a light path and Ω_i contains all light paths connecting to pixel i. We can estimate this integral using Monte Carlo integration with a single path sample X_i drawn according to a given probability density function p_i as $\langle F_i\rangle = f_i(X_i)/p_i(X_i)$.

Control variates can reduce the variance of this estimator by introducing a correlated auxiliary function h_i whose integral over Ω_i, i.e., $H_i = \int_{\Omega_i} h_i(x)\,\mathrm{d}x$, is known. Consequently, the original estimator becomes

$$\langle F_i\rangle = \alpha_i H_i + \langle F_i - \alpha_i H_i\rangle, \tag{2}$$

where α_i is a coefficient determined to reflect the correlation between the original path contribution f_i and the auxiliary function h_i, and $\langle F_i - \alpha_i H_i\rangle$ is a new estimator indicating their difference. Assuming that f_i and h_i are correlated, this difference estimator should have lower variance than the original estimator $\langle F_i\rangle$.

Image-space control variates (ICV) [Rousselle et al. 2016] leverage the observation that neighboring pixels can serve as effective auxiliary functions due to local similarity in pixel color estimates. By generalizing control variates to functions whose integrals are unknown but for which low-variance estimates are available, image-space control variates construct estimators from a set of neighboring pixels $\mathbb{N}_i$ (which includes the pixel i itself):

$$\langle F_i\rangle = \frac{\sum_{j\in\mathbb{N}_i} q_j \langle F_i\rangle_{\leftarrow j}}{\sum_{j\in\mathbb{N}_i} q_j}, \text{ where } \langle F_i\rangle_{\leftarrow j} = \alpha_{ij}\langle F_j\rangle + \langle F_i - \alpha_{ij}F_j\rangle, \tag{3}$$

and q_j denotes the weights for combining estimates from corresponding neighboring pixels. The overall variance can be effectively reduced due to the appropriate combination of multiple estimators and the iterative aggregation from spatial neighbors.

3.2 Difference Estimation

Estimating the difference $\langle F_i - \alpha_{ij}F_j\rangle$ is well studied in gradient-domain rendering [Kettunen et al. 2015; Lehtinen et al. 2013; Manzi et al. 2016]. It relies on a bijective shift mapping $T_{i\to j}$ between domains (or their subsets) Ω_i and Ω_j. By evaluating corresponding samples connected through the shift mapping, the variance of the difference estimator can be reduced. Using multiple importance sampling (MIS), the difference estimator can be formulated as

$$\begin{aligned}\langle F_i - \alpha_{ij}F_j\rangle = &\frac{\omega_{ij}(X_i)}{p_i(X_i)}\left(f_i(X_i) - \alpha_{ij}f_j(T_{i\to j}(X_i))\right)\\ &- \frac{\omega_{ji}(X_j)}{p_j(X_j)}\left(\alpha_{ij}f_j(X_j) - f_i(T_{j\to i}(X_j))\right),\end{aligned} \tag{4}$$

where $X_i \in \Omega_i$ and $X_j \in \Omega_j$ are path samples drawn according to the probability density functions p_i and p_j, respectively. We use the traditional balance heuristic for the MIS weights:

$$\omega_{ij}(x) = \frac{p_i(x)}{p_i(x) + p_j(T_{i\to j}(x))|\partial T_{i\to j}/\partial x|}, \tag{5}$$

where $|\partial T_{i\to j}/\partial x|$ is the Jacobian determinant of the shift mapping. Note that the MIS weights are critical to ensure unbiasedness, since samples from the target domain $T_{i\to j}(\Omega_i)$ of the shift mapping may not fully cover Ω_j [Kettunen et al. 2015].

3.3 ReSTIR

Reservoir-based spatio-temporal importance resampling (ReSTIR) [Bitterli et al. 2020] is an emerging technique in real-time path tracing, enabling high-quality rendering at real-time or interactive frame rates. It builds on reservoir sampling [Chao 1982] to efficiently reuse samples across space and time, significantly reducing the number of samples needed per pixel while maintaining low variance. In the following, we briefly review the basic concepts of ReSTIR.

Please refer to the ReSTIR course notes [Wyman et al. 2023] for more details.

For each pixel i, its associated reservoir maintains a representative sample Y_i and its unbiased contribution weight (UCW) W_{Y_i}, satisfying $\mathbb{E}[f_i(Y_i)W_{Y_i}] = F_i$. Note that the UCW W_{Y_i} is analogous to the reciprocal PDF $\frac{1}{p_i(Y_i)}$ in standard Monte Carlo integration. Additionally, each reservoir also tracks a confidence weight M_i, which is primarily used for modulating the MIS weight, reducing the influence of low-confidence reservoirs to mitigate variance and bias in the final estimate.

At a high level, the ReSTIR algorithm operates in the following stages for each frame:

(1) **Initial sampling**: Generates several independent samples $\{X_{i,k}\}$ for each pixel i and performs resampled importance sampling (RIS) [Talbot 2005], which becomes the initial state of a reservoir.
(2) **Temporal reuse**: Combines the current pixel's reservoir with the reservoir from the corresponding reprojected pixel in the previous frame (usually based on its motion vector). The reservoir is updated via a *resampling* operation, which stochastically selects a new representative sample from the candidate pool represented by the two reservoirs.
(3) **Spatial reuse**: Performs a similar *resampling* operation, updating the current pixel's reservoir with the reservoirs from a set of randomly selected neighboring pixels as candidates.
(4) **Shading**: Computes the final pixel color using the representative sample stored in the reservoir as $\langle F_i \rangle = f_i(Y_i)W_{Y_i}$.

Mathematically, the *resampling* operation involved in spatio-temporal reuse is formulated using generalized resampled importance sampling (GRIS) [Lin et al. 2022]. Assuming that we update the reservoir of pixel i by merging the neighboring reservoirs $\{Y_j\}$, first we need to apply the shift mappings $T_{j\to i}$ to map these samples from different domains to the canonical domain Ω_i. Then, we compute the resampling weight w_j for each shifted sample $Y'_j = T_{j\to i}(Y_j)$. Lastly, we select a new representative sample Z from the candidates with probability proportional to the resampling weights, store it in the reservoir of pixel i, and update its UCW using $W_Z = \sum_j w_j / \hat{p}(Z)$.

4 Motivation

ReSTIR [Bitterli et al. 2020] provides an efficient mechanism for reusing samples across pixels and frames via importance resampling [Lin et al. 2022; Talbot 2005]. However, the final shading stage still relies on a single representative sample rather than a spatio-temporally accumulated pixel color estimate. ReSTIR effectively aggregates multiple samples into one representative sample; as the iteration increases, the distribution of the representative sample is near the normalized target function, i.e., $Y \sim \frac{\hat{p}(Y)}{\int \hat{p}(x)\,\mathrm{d}x}$, which gives the final color estimate as $\langle F \rangle = f(Y)W_Y$. In practice, the target function $\hat{p}$ is typically the scalar luminance of the path contribution. Therefore, the reservoir may be viewed as having two parts: the accumulated UCW W_Y fits the target luminance intensity I with $\langle I \rangle = \hat{p}(Y)W_Y$, while the representative sample Y controls the spectrum through $\frac{f(Y)}{\hat{p}(Y)}$. Consequently, the final color can remain highly sensitive to the specific choice of Y, particularly when the spectral content (encoded in the vector-valued f) varies strongly across samples. This creates a gap between the scalar luminance estimation and the per-channel color distribution: even when the resampling process fits the target luminance correctly, a single retained sample may still be a poor proxy for the pixel's expected color, increasing per-channel variance and leading to high-frequency color noise.

Fig. 2 illustrates this with a white diffuse plane illuminated by monochromatic red and green lights. Although ReSTIR correctly balances sampling probabilities between the two sources, the final 1-spp estimate is inevitably either purely red or purely green, rather than the correct yellow mixture, leading to significant color noise (see Fig. 2b).

Simply expanding the scalar distribution or UCW to the full spectrum distribution is difficult within the ReSTIR framework, since they serve as a probability distribution, and estimating each channel separately requires 3 times more sampling budget, which is prohibitive in real-time rendering. One mitigation strategy is to leverage additional samples evaluated during reuse, such as reusing discarded candidates for shading [Wyman and Panteleev 2022]. While decoupling shading from resampling helps, the pool of available samples remains small. In practice, these extra samples are often not enough to recover the correct color mixture reliably (see Fig. 2c).

Conversely, techniques like denoising [Bako and Vogels 2017; Schied et al. 2017] directly reuse accumulated contributions from neighbors. This effectively accumulates path contributions, reducing noise even with low sample counts, but often introduces bias. Image-space control variates (ICV) [Rousselle et al. 2016] offer a principled, unbiased alternative. By using neighboring pixel estimates as control variates, ICV estimates the difference between pixels—which typically has low variance—allowing efficient accumulation across the image (see Fig. 2d). However, standard ICV and its subsequent variants [Yang and Moon 2025] are typically designed for offline rendering, which relies on multiple rounds of iterative estimation and high initial sample counts for obtaining reliable auxiliary and difference estimates, making them impractical for real-time rendering.

These observations motivate a hybrid approach combining the strengths of ReSTIR and ICV. In this work, we generalize ICV to the spatio-temporal domain, enabling stable aggregation across pixels and frames without running expensive iterative estimation. We also leverage ReSTIR to efficiently estimate auxiliary functions and pixel differences, reducing variance without additional path tracing costs. By integrating both techniques into a unified framework, we achieve effective variance reduction at real-time frame rates (see Fig. 2e).

5 Our Method

In this section, we present our spatio-temporal control variates framework and its integration with ReSTIR. We start by extending image-space control variates [Rousselle et al. 2016] to the spatio-temporal domain, introducing the Spatio-Temporal Control Variates (STCV) framework that progressively accumulates pixel color estimates across space and time (§5.1). To efficiently compute the new STCV estimators, we show how ReSTIR can be generalized to estimate pixel differences (§5.2). We then describe a practical algorithm,

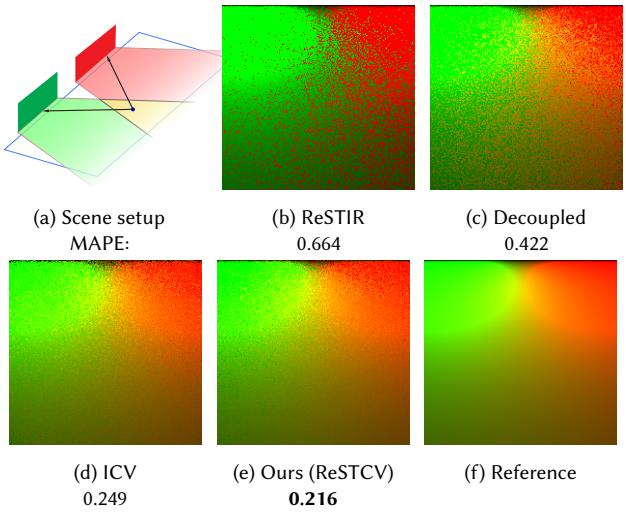


Fig. 2. An example comparing existing methods with ours at 1 spp. (a) **Scene setup**: a white diffuse plane illuminated by red and green lights. Each light sample is either green or red, but the final color in the center region should appear yellow. (b) **ReSTIR** selects a single representative sample per pixel, producing severe color noise at a low sample count. (c) **Decoupled shading** reuses discarded samples during resampling for shading, which marginally reduces color noise. (d) **Image-space control variates** reuse accumulated path contributions and estimate color more accurately but suffer from noisy base estimates due to the low sample count (as shown in the top region near the lights). (e) **Our method (ReSTCV)** integrates ICV with ReSTIR and reuses more samples effectively, leading to significantly reduced noise. (f) **Reference** image computed by path tracing with a high sample count.

Reservoir-based Spatio-Temporal Control Variates (ReSTCV), that integrates STCV and ReSTIR into a unified framework with minimal computational overhead (§5.3).

5.1 Spatio-Temporal Control Variates

The original image-space control variates formulation primarily focuses on offline rendering for static scenes or re-rendering for slightly edited scenes (e.g., changing material parameters). To adapt this technique to real-time rendering with extremely low sample counts per frame, we introduce *Spatio-Temporal Control Variates* (STCV) by accumulating control variate estimates both spatially and temporally.

5.1.1 Initial sampling. For each pixel i, we first sample a path X_i according to the probability density function p_i, and compute the initial color estimation for pixel i, denoted as $\langle F_i \rangle_{\text{init}}$.

5.1.2 Temporal control variates. To obtain a better estimate of pixel colors, we use motion vectors to find the reprojected pixel j in the previous frame, and then reuse its color estimation $\langle F_j \rangle$ as a control variate, combining it with the initial estimation $\langle F_i \rangle_{\text{init}}$ as

$$\langle F_i \rangle = \frac{q_j \langle F_i \rangle_{\leftarrow j} + q_{\text{init}} \langle F_i \rangle_{\text{init}}}{q_j + q_{\text{init}}}, \tag{6}$$

where q_j and q_{init} denote the compositing weights, and the "from-j" estimator that leverages sample reuse and control variates is defined as

$$\langle F_i \rangle_{\leftarrow j} = \alpha_{ij} \langle F_j \rangle + \langle F_i - \alpha_{ij} F_j \rangle. \tag{7}$$

Note that we will skip this step if we fail to reproject the pixel to the previous frame.

5.1.3 Spatial control variates. Next, we apply spatial reuse of estimators using nearby pixels as control variates:

$$\langle F_i \rangle = \frac{\sum_{j \in \mathbb{N}_i} q_j \langle F_i \rangle_{\leftarrow j}}{\sum_{j \in \mathbb{N}_i} q_j}, \tag{8}$$

where $\langle F_i \rangle_{\leftarrow j}$ is the "from-j" estimator defined in Eq. 7, q_j denotes its corresponding compositing weight, and $\mathbb{N}_i$ is the union of pixel i itself and a set of randomly selected neighboring pixels.

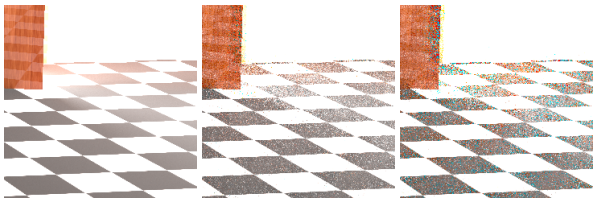
5.1.4 Compositing weights. As discussed in the ICV work [Rousselle et al. 2016], the optimal compositing weights in Eqs. 6 and 8 should minimize the variance of the composite estimator $\langle F_i \rangle$. This requires access to the variance and covariance of all estimators involved, which is expensive to estimate and thus impractical for real-time rendering. Instead, we propose a simple heuristic that weights each estimator using its confidence weight, which is inspired by ReSTIR techniques [Bitterli et al. 2020; Lin et al. 2022] and will be discussed in more detail in §5.3.

5.1.5 Coefficient α_{ij}. The optimal control variate coefficient α_{ij} for the CV estimator $\langle F_i \rangle_{\leftarrow j}$ is: $\alpha_{ij} = \text{Cov}(\langle F_i \rangle, \langle F_j \rangle) \,/\, \text{Var}(\langle F_j \rangle)$ [Rousselle et al. 2016]. Since it relies on the variance and covariance of the base estimators, which is challenging to estimate accurately and efficiently in real-time rendering, we use the following heuristic to determine the coefficient α_{ij}:

$$\alpha_{ij} = \min(\rho_i / \rho_j, 2.0), \tag{9}$$

where ρ_i and ρ_j are the average material reflectances at the primary hit points of light paths X_i and X_j (connecting through pixel i and pixel j), respectively. Although a constant coefficient already provides good results, we found this heuristic effective because it serves as a good approximation to the optimal coefficient when neighboring pixels share the same incident radiance distribution but have different BRDFs (especially albedo variations), resembling albedo decomposition in denoising. Specifically, the average reflectance ρ_i is computed by integrating the corresponding BRDF values over the hemisphere, which we approximate using closed-form expressions for microfacet BRDF models [Haines and Akenine-Möller 2019]. We clamp the reflectance ratio within 2.0 to avoid instability due to bad shift mappings. Note that the coefficient α_{ij} is vector-valued.

5.1.6 Direct estimation. These estimators in Eqs. 6–8 under our spatio-temporal control variates formulation can be directly computed using standard Monte Carlo integration as shown in Eqs. 4. However, under the low sample counts of real-time rendering (usually 1 spp per frame), the difference term may still be very noisy even when the accumulated control variate estimate becomes stable. To address this issue, we further introduce a ReSTIR-based approach that uses reservoir samples to evaluate the difference term more effectively, as presented in the next subsection.



(a) Reference
MAPE:

(b) Reflectance ratio
0.243

(c) Constant
0.344

Fig. 3. Illustration of the coefficient α_{ij} used in ReSTCV on VeachAjar (door and floor crop). Setting α_{ij} to the reflectance ratio according to Eq. 9 reduces color noise compared with the constant choice $\alpha_{ij} = 1$. Scene asset: VeachAjar (Benedikt Bitterli, CC0 1.0).

5.2 Efficient Difference Estimation using ReSTIR

In our spatio-temporal control variates formulation, the pixel color estimate is a linear combination of "from-j" estimators[1], whose dominant source of variance is the difference estimator $\langle F_i - \alpha_{ij} F_j \rangle$ (see the analysis in § S.1 of the supplementary document). In principle, the optimal sampling distribution for this estimator should be constructed from the difference integrand itself rather than from ReSTIR's luminance target. Nevertheless, the ReSTIR-induced distribution is already much better than naive random sampling because it provides a good proxy for important path regions, and high-intensity contributions often correlate with large differences. We therefore retain the original ReSTIR structure and reuse its reservoirs directly, avoiding a substantially more complex scheme that would require dedicated candidate samples or reservoirs for every difference-estimation pair together with shift mapping for the difference term. We now show how to extend ReSTIR to efficiently estimate $\langle F_i - \alpha_{ij} F_j \rangle$.

As previously discussed in §3.3, each pixel i maintains a reservoir (Y_i, W_{Y_i}, M_i), which is a tuple consisting of a representative sample Y_i, its unbiased contribution weight W_{Y_i}, and its confidence weight M_i. Given a pair of pixels i and j, as well as their corresponding reservoirs, we modify the standard difference estimator (Eq. 4) by reusing the reservoir information:

$$\begin{aligned}\langle F_i - \alpha F_j \rangle = \hat{\omega}_{ij}(Y_i) W_{Y_i} (f_i(Y_i) - \alpha_{ij} f_j(T_{i\to j}(Y_i))) \\ - \hat{\omega}_{ji}(Y_j) W_{Y_j} (\alpha_{ij} f_j(Y_j) - f_i(T_{j\to i}(Y_j))),\end{aligned} \tag{10}$$

where the MIS weight is defined as

$$\hat{\omega}_{ij}(x) = \frac{M_i \hat{p}_i(x)}{M_i \hat{p}_i(x) + M_j \hat{p}_j(T_{i\to j}(x)) |\partial T_{i\to j}/\partial x|}. \tag{11}$$

Compared to the standard difference estimator (Eqs. 4 and 5), our ReSTIR-based estimator has the following differences:

(1) We reuse the selected samples Y_i and Y_j stored in the reservoirs, instead of sampling new paths. The shift mappings are computed using the reservoir samples Y_i and Y_j.
(2) We use the unbiased contribution weights W_{Y_i} and W_{Y_j} instead of the reciprocal PDFs.
(3) For MIS weights, we use the product of the confidence weight M and the target distribution function $\hat{p}$, following the robust MIS weights used in GRIS [Lin et al. 2022].

Since our MIS weight $\hat{\omega}_{ij}$ satisfies the unbiased condition, our new ReSTIR-based estimator remains unbiased, which is verified in §6.1. In practice, its variance is substantially lower than that of the standard difference estimator because the reused reservoir samples are drawn from a much better distribution.

[1]For temporal control variates (Eq. 6), the initial estimate $\langle F_i \rangle_{\text{init}}$ can essentially be interpreted as $\langle F_i \rangle_{\leftarrow i}$.

```
class Reservoir:
    Y = None # Selected sample Y
    W_Y = 0  # Unbiased contribution weight of Y
    W = 0    # Sum of weights
    M = 0    # Effective sample count
    F = 0    # Pixel color estimate as control variates
    def Update(Y_j, w_j, M_j): # RIS
        W += w_j
        M += M_j
        if rand() < (w_j / W):
            Y = Y_j

def ReusePass(i: int, pixels: list[int]) -> Reservoir:
    r = Reservoir()
    for j in pixels:
        Z_i = ShiftMapping(r[j].Y, i)
        # Compute the inverse shift mapping from i to j
        Z_j = ShiftMapping(r[i].Y, j)
        # Compute the pixel difference (Eq. 10)
        diff = EstimateDifference(r[i], r[j], Z_i, Z_j)
        F_from_j = alpha_ij * r[j].F + diff # Eq. 7
        r.F += F_from_j * r[j].M # Eqs. 6,8
        # Compute the resampling weight
        w = ResamplingWeight(r[i], r[j], Z_i, Z_j)
        # Update the reservoir
        r.Update(Z_i, w, r[j].M)
    if r.Y is not None:
        r.W_Y = r.W / p_hat(r.Y)
    r.F /= r.M
    return r

# Run the following program per frame for each pixel i
def ReSTCV(i: int)-> float3:
    # 1. Initial sampling
    X_i, w_i, f_i, p_i = SamplePath(i)
    r[i] = Reservoir(X_i, w_i, 1, f_i/p_i)
    # 2: Temporal reuse
    j = CorrespondingPixelInPreviousFrame(i)
    r[i] = ReusePass(i, [i,j]) if j is not None
    # 3: Spatial reuse
    neighbors = GetNeighborPixels(i) # Returns a list
    r[i] = ReusePass(i, [i] + neighbors)
    # 4: Final shading
    return r[i].F
    return shading(r[i].Y) * r[i].W_Y
```

Listing 1: Pseudocode of our ReSTCV algorithm. The highlighted lines show the modifications compared to the original ReSTIR: blue lines are added while red ones are removed.

5.3 Practical Algorithm

We now present a practical algorithm, *Reservoir-based Spatio-Temporal Control Variates* (ReSTCV), that integrates our spatio-temporal control variates and ReSTIR. Our algorithm is illustrated in Listing 1, which modifies the original ReSTIR with a few lines of code (highlighted in blue and red). The modifications can be summarized as follows:

(1) For each reservoir, we store an additional value F_i, indicating the color estimate of pixel i (Line 6). This value is updated during the spatio-temporal reuse steps according to Eqs. 6 and 8 (Lines 22 and 29).
(2) At every reuse step, we estimate the pixel difference using our new ReSTIR-based estimator (Lines 19–22). This adds negligible computational overhead since we reuse the reservoir samples and shift mappings, which are already computed for the original ReSTIR.
(3) We no longer compute the final shading using the reservoir sample, but instead directly use the stored pixel color estimate F_i (Lines 44–45).

The implementation of our method incurs minimal computational overhead while delivering significant variance reduction, especially in reducing color noise. This variance reduction comes from two sources. First, ReSTIR's sample reuse lowers the variance of pixel-difference estimation. Second, when ReSTIR fails to reuse samples (e.g., due to invalid shift mappings), the variance remains controlled by the correlation between the accumulated control-variate estimate and the ReSTIR-based single-sample contribution.

We now discuss some implementation details of our method.

Sample rejection. Similar to the original ReSTIR, when merging reservoirs during spatial or temporal reuse, we will reject bad sample candidates when their geometry properties (e.g., normal and depth) deviate too much from the target sample.

Compositing weights. We use the following heuristic to compute the compositing weights q used in Eqs. 6 and 8: The weight q_j for the estimator $\langle F_i \rangle_{\leftarrow j}$ from pixel j is simply the confidence weight M_j stored in the reservoir, and we set $q_{\text{init}} = 1$ for the initial estimator $\langle F_i \rangle_{\text{init}}$. Following GRIS [Lin et al. 2022], the confidence weight M_j is capped at 20 to balance noise and correlation.

Direct illumination. Following the original ReSTIR PT implementation [Lin et al. 2022], direct illumination (DI) and indirect illumination are handled separately. ReSTIR DI uses a proxy target distribution during resampling, which does not provide the accurate direct illumination required by control variates. Additionally, the shift mappings employed in ReSTIR DI [Bitterli et al. 2020] are typically simple (e.g., reusing the UV coordinates of light samples), leading to high variance in difference estimation when the material is highly glossy. To address these issues, we construct an intermediate auxiliary function $\dot{D}_i$, which computes the DI contribution with the proxy target distribution and spectral illumination, and omits the glossy component of the BRDF term. Then, we use ReSTCV to estimate $\langle \dot{D}_i \rangle$, which has much lower variance without the high-frequency glossy component. The final DI estimate is computed using another level of control variates as $\langle D_i \rangle = \langle \dot{D}_i \rangle + \langle D_i - \dot{D}_i \rangle$. This approach preserves the efficiency of ReSTIR DI while reducing noise through spatio-temporal control variates. For more details, please refer to § S.2 of the supplementary document.

Codebase. We build our ReSTCV algorithm[2] upon the public implementation of ReSTIR PT [Lin et al. 2022] in the Falcor real-time rendering framework [Kallweit et al. 2022]. Unless otherwise specified, we use the same parameter settings as ReSTIR PT.

[2]Code repository available at: https://github.com/Hercier/ReSTCV.

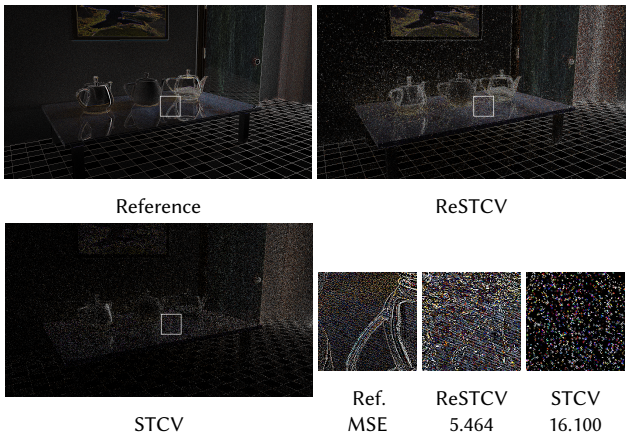


Fig. 4. Validation of the difference estimator on `VeachAjar`. ReSTCV better matches the high-spp reference and lowers the MSE over the whole image. `VeachAjar` (Benedikt Bitterli, CC0 1.0).

6 Results

We evaluate our method on various dynamic scenes adapted from the Computer Graphics Archive [McGuire 2017] and the Open Research Content Archive (ORCA) [Games 2017; Lumberyard 2017; Winkelmann 2019]. We render each scene in real-time at 1920×1080 on a workstation equipped with an RTX 5080 GPU and a Ryzen 9 9950X CPU. Please refer to the supplementary video for animations.

We compare our STCV and ReSTCV algorithms with standard path tracing and ReSTIR PT [Lin et al. 2022]. For direct illumination, we use 32 light samples for next event estimation and 5 neighboring pixels for spatial reuse; for indirect illumination, we use 3 neighboring pixels. Unless otherwise noted, all relMSE and MAPE values are computed over the full image. Under equal-time comparisons, standard path tracing corresponds to 5 spp.

As shown in Fig. 5, ReSTCV provides the best overall noise reduction and color stability in our comparisons against standard PT, ReSTIR PT, and STCV alone. STCV performs well in scenes with relatively simple lighting conditions, such as `LivingRoom`. However, in scenes with complex indirect illumination (`VeachAjar`) or many lights (`ZeroDay`), STCV cannot obtain sufficiently informative difference estimates from random sampling, limiting its reuse quality. ReSTIR PT, on the other hand, suffers from noticeable color noise caused by multiple colored light sources and color bleeding from indirect illumination. ReSTCV alleviates these issues by accumulating neighboring contributions through control variates while preserving accurate pixel-difference estimation from ReSTIR samples, resulting in overall variance reduction.

Figure 6 isolates direct illumination. ReSTCV still mixes contributions from multiple colored lights effectively, while preserving better sample distribution than STCV alone. However, since the scalar auxiliary functions used in ReSTIR do not incorporate full

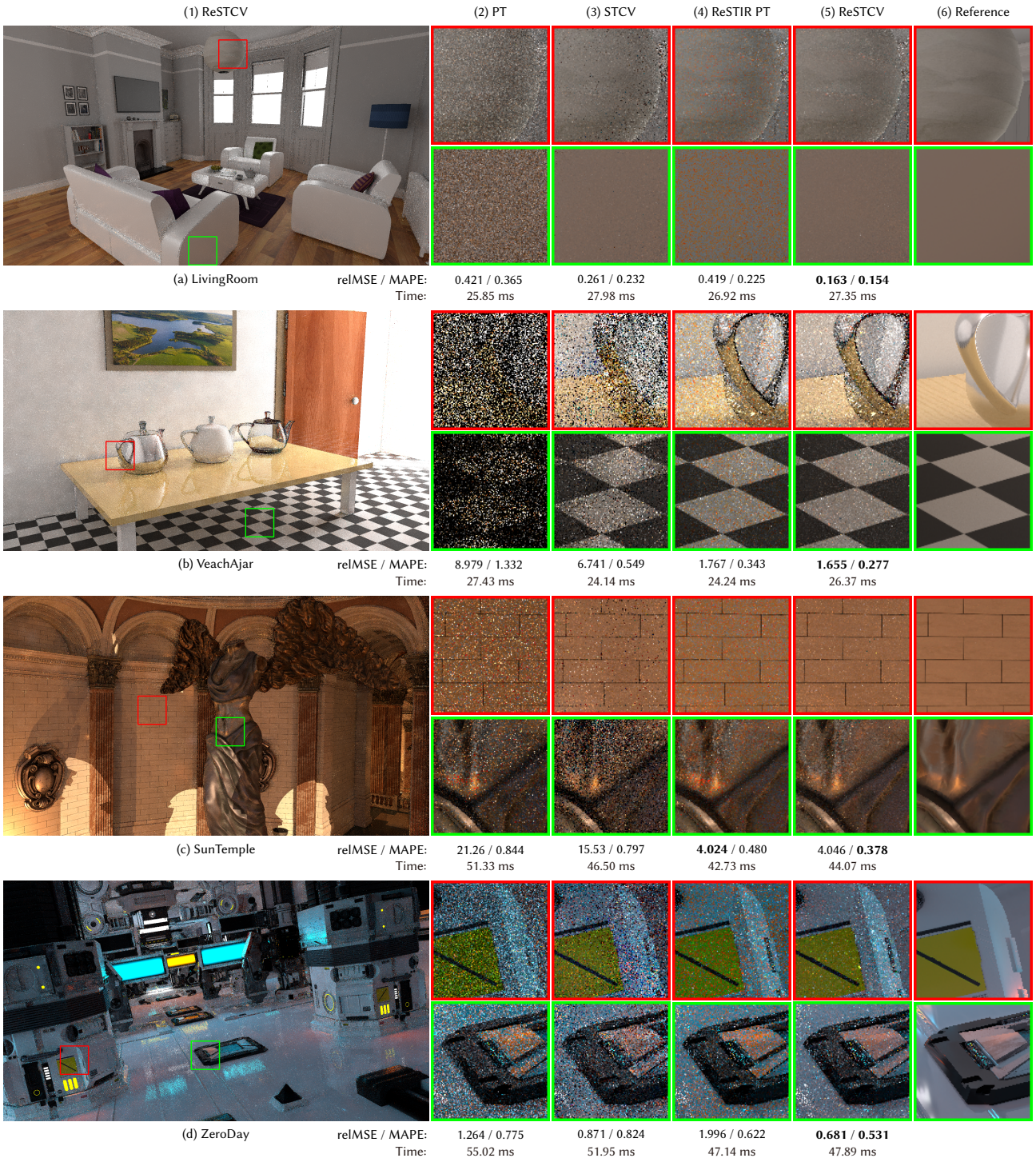


Fig. 5. Equal-time comparison with standard path tracing (PT), our STCV baseline, and ReSTIR PT [Lin et al. 2022]. ReSTIR PT reduces variance but often exhibits color noise, while STCV alone struggles with complex indirect illumination. ReSTCV combines both to reduce variance and improve color stability. `LivingRoom` (© 2012 Jay, CC BY 3.0), `VeachAjar` (Benedikt Bitterli, CC0 1.0), `SunTemple` (© Epic Games, CC BY-NC-SA 4.0), and `ZeroDay` (© Mike Winkelmann (Beeple), CC BY 4.0).

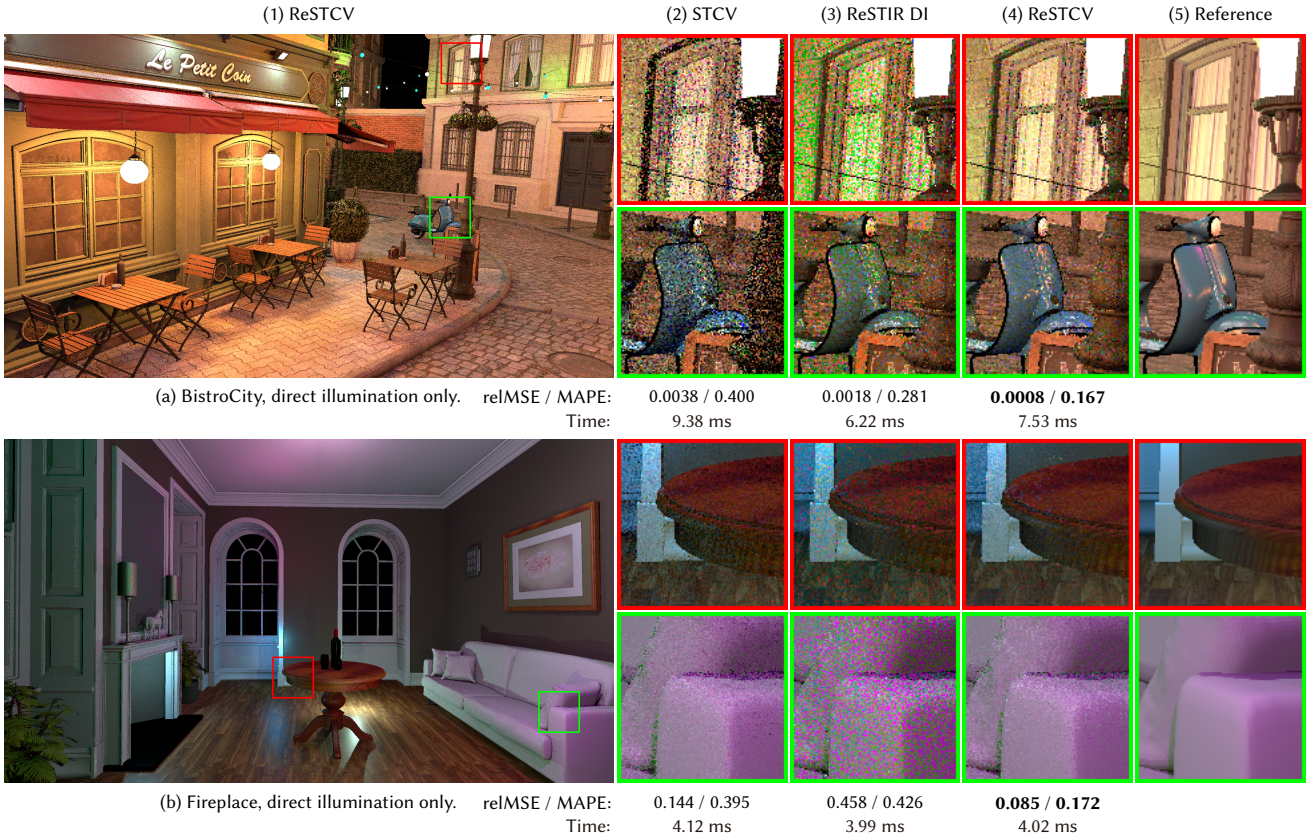


Fig. 6. Equal-time direct-illumination comparison between ReSTCV, our STCV baseline, and ReSTIR DI [Bitterli et al. 2020]. ReSTCV reduces visible color noise and improves stability in these examples. `Amazon Lumberyard Bistro` (© Amazon Lumberyard, CC BY 4.0) and Fireplace Room (© Wig42, CC BY 3.0).

spectral BRDF information, color noise still remains in certain cases, as visible in the `BistroCity` scene.

In Fig. 7, we compare with a variant of ReSTIR PT equipped with decoupled shading [Wyman and Panteleev 2022]. Decoupled shading reuses the path contributions evaluated from neighboring candidate samples, which is equivalent to our method using a zero-valued neighboring integrand (both $\langle F_j \rangle$ and $f_j(x)$ are 0) with ReSTIR samples. Without spatiotemporal accumulation, it therefore exhibits higher levels of color noise. This approach does not benefit from temporal accumulation and therefore exhibits higher levels of color noise.

In the following, we validate our method through several focused validation and ablation experiments. For additional ablation studies, please refer to § S.3 of the supplementary document.

Ablation on coefficient α_{ij}. We investigate the impact of the control variate coefficient α_{ij} on the rendering quality, as shown in Fig. 3. We compare our proposed heuristic (Eq. 9), which adapts α_{ij} based on the ratio of material reflectances, with a baseline where α_{ij} is fixed to 1. In general, our heuristic (Fig. 3b) effectively suppresses color noise by adjusting α_{ij} to better correlate the estimators of the current and neighboring pixels, resulting in a cleaner image with a lower error compared to the constant baseline (Fig. 3c).

6.1 Difference-estimator validation

We directly validate the difference estimator (with the coefficient α_{ij}) on `VeachAjar` and visualize the vector-valued difference in Fig. 4 with an equal-time 1spp comparison. For convenience, we use 4-connected neighbors for both STCV and ReSTCV. Although ReSTIR samples are not explicitly optimized for difference estimation, ReSTCV still produces a more accurate difference estimate than STCV, with more valid samples contributing to the estimation.

Unbiasedness validation. We verify the unbiasedness of ReSTCV by repeatedly executing our method on static scenes and accumulating the frames. Fig. 8 shows the convergence curves of our method and ReSTIR PT using a 65536 spp reference image. The overall convergence behavior indicates that our method does not introduce bias, although the final convergence rate could be affected by residual noise in the reference.

7 Conclusion

In this work, we introduced reservoir-based spatio-temporal control variates (ReSTCV) for real-time rendering. By combining ReSTIR-style sample reuse with accumulated path contributions from control variates linked by a ReSTIR-based pixel-difference estimator,

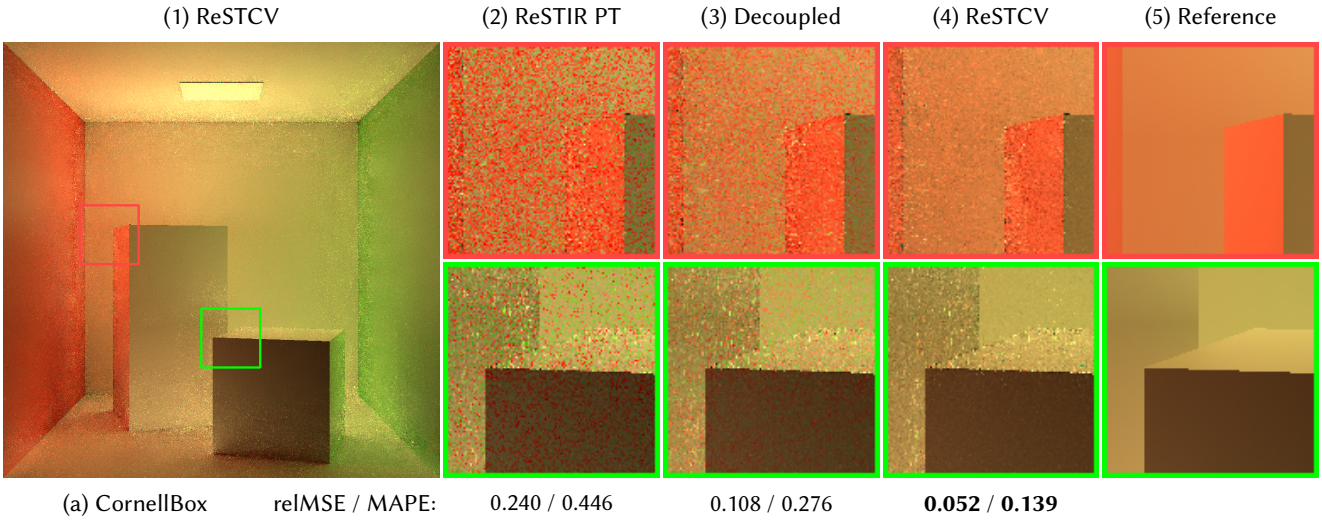


Fig. 7. Equal-time comparison of ReSTCV against ReSTIR PT [Lin et al. 2022] and decoupled shading [Wyman and Panteleev 2022] on indirect illumination only. Decoupled shading ignores temporal color history and therefore exhibits higher noise. ReSTCV accumulates color estimates over time, improving stability in these cases.

ReSTCV reduces overall variance with only minor changes to standard ReSTIR and modest overhead. Across our examples, this manifests as substantially more stable colors. Our results show consistent variance reduction over prior work across a wide range of dynamic scenes. Overall, this work shows that control variates can be effectively adapted to real-time path tracing by aligning their design with ReSTIR. We believe our work enables new possibilities for unifying rendering techniques that exploit correlated samples.

Limitations and future work. Our method focuses on reducing the gap between ReSTIR's scalar intensity distribution and the per-channel color distribution. In weakly chromatic or largely achromatic scenes, baseline ReSTIR already aligns well with intensity, so this gap can be smaller. For further variance reduction, it would be interesting to explore more optimal, variance-aware strategies for combining estimators, since our current heuristic based on sample counts and pairwise MIS does not explicitly minimize variance or account for covariance. Additionally, our current target distribution for estimating pixel differences still follows the path contribution. Ideally, ReSTIR's resampling process would directly target the distribution of pixel differences, which could further reduce variance.

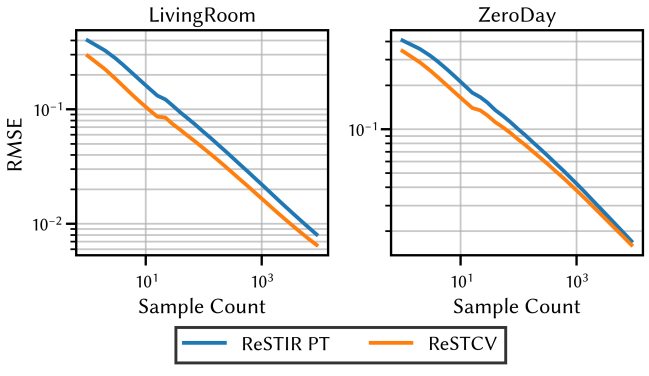


Fig. 8. Relative mean squared error curves with respect to sample count.

Acknowledgments

We thank the anonymous reviewers for their valuable comments and insightful suggestions. This work was supported by the National Natural Science Foundation of China (Project No. 62372257).

References

Steve Bako and Thijs Vogels. 2017. Kernel-predicting convolutional networks for denoising Monte Carlo renderings. (2017).

Pablo Bauszat, Victor Petitjean, and Elmar Eisemann. 2017. Gradient-domain path reusing. *ACM Trans. Graph.* 36, 6, Article 229 (Nov. 2017), 9 pages. doi:10.1145/3130800.3130886

Laurent Belcour, Guofu Xie, Christophe Hery, Mark Meyer, Wojciech Jarosz, and Derek Nowrouzezahrai. 2018. Integrating Clipped Spherical Harmonics Expansions. *ACM Trans. Graph.* 37, 2, Article 19 (March 2018), 12 pages. doi:10.1145/3015459

Benedikt Bitterli, Chris Wyman, Matt Pharr, Peter Shirley, Aaron Lefohn, and Wojciech Jarosz. 2020. Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting. *ACM Trans. Graph.* 39, 4, Article 148 (Aug. 2020), 17 pages. doi:10.1145/3386569.3392481

Min-Te Chao. 1982. A general purpose unequal probability sampling plan. *Biometrika* 69, 3 (1982), 653–656.

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

Epic Games. 2017. Unreal Engine Sun Temple, Open Research Content Archive (ORCA). http://developer.nvidia.com/orca/epic-games-sun-temple

Eric Haines and Tomas Akenine-Möller. 2019. *Ray Tracing Gems: High-Quality and Real-Time Rendering with DXR and Other APIs.* Apress.

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

Qingqin Hua, Pascal Grittmann, and Philipp Slusallek. 2023. Revisiting controlled mixture sampling for rendering applications. *ACM Trans. Graph.* 42, 4, Article 64 (July 2023), 13 pages. doi:10.1145/3592435

Matthieu Josse, Joey Litalien, and Adrien Gruson. 2025. Adaptive Neural Kernels for Gradient-domain Rendering. In *Proceedings of the SIGGRAPH Asia 2025 Conference*

Papers (SA Conference Papers '25). Association for Computing Machinery, New York, NY, USA, Article 67, 11 pages. doi:10.1145/3757377.3763920

Simon Kallweit, Petrik Clarberg, Craig Kolb, Tom'aš Davidovič, Kai-Hwa Yao, Theresa Foley, Yong He, Lifan Wu, Lucy Chen, Tomas Akenine-Möller, Chris Wyman, Cyril Crassin, and Nir Benty. 2022. The Falcor Rendering Framework. https://github.com/NVIDIAGameWorks/Falcor https://github.com/NVIDIAGameWorks/Falcor.

Markus Kettunen, Marco Manzi, Miika Aittala, Jaakko Lehtinen, Frédo Durand, and Matthias Zwicker. 2015. Gradient-domain path tracing. *ACM Trans. Graph.* 34, 4, Article 123 (July 2015), 13 pages. doi:10.1145/2766997

Ivo Kondapaneni, Petr Vévoda, Pascal Grittmann, Tomáš Skřivan, Philipp Slusallek, and Jaroslav Křivánek. 2019. Optimal multiple importance sampling. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–14.

Fei Lee and Chun-Fa Chang. 2025. ReSTIR PT with MCMC Decoupled Shading. In *Companion Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3D Companion '25)*. Association for Computing Machinery, New York, NY, USA, Article 13, 2 pages. doi:10.1145/3722564.3728381

Jaakko Lehtinen, Tero Karras, Samuli Laine, Miika Aittala, Frédo Durand, and Timo Aila. 2013. Gradient-domain metropolis light transport. *ACM Trans. Graph.* 32, 4, Article 95 (July 2013), 12 pages. doi:10.1145/2461912.2461943

Yuzhi Liang, Tao Liu, Yuchi Huo, Rui Wang, and Hujun Bao. 2024. Adaptive sampling and reconstruction for gradient-domain rendering. *Computational Visual Media* 10, 5 (2024), 885–902.

Daqi Lin, Markus Kettunen, Benedikt Bitterli, Jacopo Pantaleoni, Cem Yuksel, and Chris Wyman. 2022. Generalized resampled importance sampling: foundations of ReSTIR. *ACM Trans. Graph.* 41, 4, Article 75 (July 2022), 23 pages. doi:10.1145/3528223.3530158

Haolin Lu, Delio Vicini, Wesley Chang, and Tzu-Mao Li. 2025. Vector-Valued Monte Carlo Integration Using Ratio Control Variates. *Transactions on Graphics (Proceedings of SIGGRAPH)* 44, 4 (Aug. 2025).

Amazon Lumberyard. 2017. Amazon Lumberyard Bistro, Open Research Content Archive (ORCA). http://developer.nvidia.com/orca/amazon-lumberyard-bistro

Marco Manzi, Markus Kettunen, Frédo Durand, Matthias Zwicker, and Jaakko Lehtinen. 2016. Temporal gradient-domain path tracing. *ACM Trans. Graph.* 35, 6, Article 246 (Dec. 2016), 9 pages. doi:10.1145/2980179.2980256

Morgan McGuire. 2017. *Computer Graphics Archive*. https://casual-effects.com/data

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

Baptiste Nicolet, Fabrice Rousselle, Jan Novák, Alexander Keller, Wenzel Jakob, and Thomas Müller. 2023. Recursive Control Variates for Inverse Rendering. *Transactions on Graphics (Proceedings of SIGGRAPH)* 42, 4 (Aug. 2023). doi:10.1145/3592139

Yaobin Ouyang, Shiqiu Liu, Markus Kettunen, Matt Pharr, and Jacopo Pantaleoni. 2021. ReSTIR GI: Path resampling for real-time path tracing. In *Computer Graphics Forum*, Vol. 40. Wiley Online Library, 17–29.

Fabrice Rousselle, Wojciech Jarosz, and Jan Novák. 2016. Image-space control variates for rendering. *ACM Trans. Graph.* 35, 6, Article 169 (Dec. 2016), 12 pages. doi:10.1145/2980179.2982443

Corentin Salaün, Adrien Gruson, Binh-Son Hua, Toshiya Hachisuka, and Gurprit Singh. 2022. Regression-based Monte Carlo integration. *ACM Trans. Graph.* 41, 4, Article 79 (July 2022), 14 pages. doi:10.1145/3528223.3530095

Christoph Schied, Anton Kaplanyan, Chris Wyman, Anjul Patney, Chakravarty R Alla Chaitanya, John Burgess, Shiqiu Liu, Carsten Dachsbacher, Aaron Lefohn, and Marco Salvi. 2017. Spatiotemporal variance-guided filtering: real-time reconstruction for path-traced global illumination. In *Proceedings of High Performance Graphics*. 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. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2004.00790.x doi:10.1111/j.1467-8659.2004.00790.x

Justin F Talbot. 2005. *Importance resampling for global illumination*. Brigham Young University.

Yusuke Tokuyoshi. 2024. Efficient Visibility Reuse for Real-time ReSTIR. In *ACM SIGGRAPH 2024 Talks*. 1–2.

Xiaochun Tong and Toshiya Hachisuka. 2024. Efficient Image-Space Shape Splatting for Monte Carlo Rendering. *ACM Trans. Graph.* 43, 6, Article 233 (Nov. 2024), 11 pages. doi:10.1145/3687943

Eric Veach. 1997. *Robust Monte Carlo methods for light transport simulation*. Vol. 1610. Stanford University PhD thesis.

Petr Vévoda, Ivo Kondapaneni, and Jaroslav Křivánek. 2018. Bayesian online regression for adaptive direct illumination sampling. *ACM Trans. Graph.* 37, 4, Article 125 (July 2018), 12 pages. doi:10.1145/3197517.3201340

Mike Winkelmann. 2019. Zero-Day, Open Research Content Archive (ORCA). https://developer.nvidia.com/orca/beeple-zero-day

Chris Wyman, Markus Kettunen, Daqi Lin, Benedikt Bitterli, Cem Yuksel, Wojciech Jarosz, and Pawel Kozlowski. 2023. A gentle introduction to ReSTIR path reuse in real-time. In *ACM SIGGRAPH 2023 Courses*. 1–38.

Chris Wyman and Alexey Panteleev. 2022. Rearchitecting spatiotemporal resampling for production. In *Proceedings of the Conference on High-Performance Graphics (HPG '21)*. Eurographics Association, Goslar, DEU, 23–41. doi:10.2312/hpg.20211281

Bing Xu, Tzu-Mao Li, Iliyan Georgiev, Trevor Hedstrom, and Ravi Ramamoorthi. 2024. Residual path integrals for re-rendering. *Computer Graphics Forum* 43, 4 (2024), e15152. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.15152 doi:10.1111/cgf.15152

Difei Yan, Zengyu Li, Lifan Wu, and Kun Xu. 2025. Generalized Unbiased Reconstruction for Gradient-Domain Rendering. *ACM Trans. Graph.* 44, 6, Article 211 (Dec. 2025), 14 pages. doi:10.1145/3763297

Chanu Yang and Bochang Moon. 2025. Imperfect Image-Space Control Variates for Monte Carlo Rendering. *ACM Trans. Graph.* 44, 6, Article 205 (Dec. 2025), 11 pages. doi:10.1145/3763335

Lei Yang, Shiqiu Liu, and Marco Salvi. 2020. A survey of temporal antialiasing techniques. In *Computer graphics forum*, Vol. 39. Wiley Online Library, 607–621.