Supplementary Document: 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, United States of America
KUN XU*, BNRist, MOE Key Laboratory of Pervasive Computing, Department of CS&T, Tsinghua University, China

ACM Reference Format:
Zhong Shi, Cunhao Wu, Lifan Wu, and Kun Xu. 2026. Supplementary Document: 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, 3 pages. https://doi.org/10.1145/3799902.3811113

Contents

1 Variance Model for STCV

Here, we present a simple variance model of our STCV method. We assume all pixels and differences have the same variance, and all of them are independent. We denote by $\mathbb{V}[\langle F\rangle_{temporal}]^{(k)}$ and $\mathbb{V}[\langle F\rangle_{spatial}]^{(k)}$ the variances of the contribution after k iterations of temporal and spatial reuse, and by $\mathbb{V}[\langle D\rangle]$ the variance of the difference estimator.

Temporal estimators include initial samples, which introduce extra variance. We simply assume that the weighting between control variates with previous estimators and initial samples may not increase the variance. Therefore:

$$\mathbb{V}[\langle F\rangle_{temporal}]^{(k)} \approx \mathbb{V}[\langle F\rangle_{spatial}]^{(k-1)} + \mathbb{V}[\langle D\rangle]. \tag{1}$$
$$\tag{2}$$

For spatial reuse, we also assume the confidence weights of all neighbors are identical, but the center pixel may have a larger weight c, since this estimator does not require an extra difference estimator:

$$\langle F\rangle_{spatial}^{(k)} \approx \frac{1}{c+N}\left(c\langle F_i\rangle_{temporal}^{(k)} + \sum_{j=1}^{N}\left(\langle F_j\rangle_{temporal}^{(k)} + \langle D_{ij}\rangle\right)\right) \tag{3}$$

Therefore, the variance is:

$$\mathbb{V}[\langle F\rangle_{spatial}]^{(k)} \approx \frac{c^2}{(c+N)^2}\mathbb{V}[\langle F\rangle_{temporal}]^{(k)} + \frac{N}{(c+N)^2}\left(\mathbb{V}[\langle F\rangle_{temporal}]^{(k)} + \mathbb{V}[\langle D\rangle]\right). \tag{4}$$

The optimal value of c combining the center pixel estimator and the neighbor pixel estimator, which minimizes the overall variance, requires weights proportional to the variance of respective estimators. Formally:

$$c\mathbb{V}[\langle F\rangle_{temporal}]^{(k)} = \mathbb{V}[\langle F\rangle_{temporal}]^{(k)} + \mathbb{V}[\langle D\rangle]. \tag{5}$$

One may use a proper initial variance estimate to simulate this process. However, if the variance converges, the variance at iteration k for large k is a constant. We have:

$$\begin{cases} \mathbb{V}[\langle F\rangle_{spatial}] & = \frac{c^2}{(c+N)^2}(\mathbb{V}[\langle F\rangle_{spatial}] + \mathbb{V}[\langle D\rangle]) + \\ & \quad \frac{N}{(c+N)^2}\left(\mathbb{V}[\langle F\rangle_{spatial}] + 2\mathbb{V}[\langle D\rangle]\right), \\ c(\mathbb{V}[\langle F\rangle_{spatial}] + \mathbb{V}[\langle D\rangle]) & = \mathbb{V}[\langle F\rangle_{spatial}] + 2\mathbb{V}[\langle D\rangle] \end{cases} \tag{6}$$

We can explicitly solve the equation and get:

$$c = \frac{1}{2}\left(\sqrt{1+6N+N^2} - N + 1\right) \tag{7}$$

$$\mathbb{V}[\langle F\rangle_{spatial}] = \frac{1}{2N}\left(\sqrt{1+6N+N^2} - N + 1\right)\mathbb{V}[\langle D\rangle] \tag{8}$$

Accounting for neighbor rejection, we may assume $N \approx 2.4$ for PT, which gives $c = 1.6$ and $\mathbb{V}[\langle F\rangle_{spatial}] = \frac{2}{3}\mathbb{V}[\langle D\rangle]$. Therefore, we give the center pixel contribution an extra weight of $\times 1.6$ during spatial reuse. This modification gives about 5% performance improvement.

This model also indicates how the difference estimator dominates the overall variance. Since the variance of the STCV estimator and the difference estimator has a fixed ratio, improving the difference estimator is the best way to improve overall variance.

*Kun Xu is the corresponding author.

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

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

ACM ISBN 979-8-4007-2554-8/2026/07
https://doi.org/10.1145/3799902.3811113

2 ReSTCV for Direct Illumination

In principle, our spatio-temporal image-space control variates framework can be applied to direct illumination by replacing the full path contribution F_i with the direct illumination contribution $\mathbf{D}_i$, and by using light samples generated by ReSTIR DI. However, practical ReSTIR DI implementations are more optimized, which complicates a direct application of existing control variate formulations. In particular, ReSTIR DI commonly relies on simplified scalar target distributions during resampling to reduce computation. As a result, accurate full-spectrum direct illumination values are often unavailable during the resampling stage, which poses challenges for constructing effective auxiliary functions for control variates.

2.1 Revisiting ReSTIR DI

According to the rendering equation [Kajiya 1986], the direct illumination contribution at a shading point x can be written as:

$$\mathbf{D}_i(x, \omega_o, y) = \mathbf{B}(x, \omega_o, \omega_i) V(x, y) \mathbf{L}(y, \omega_i), \text{ where } \omega_i = y - x, \quad (9)$$

where $\mathbf{B}(x, \omega_o, \omega_i)$ is the BSDF evaluated at x, $V(x, y)$ denotes visibility between the shading point and the light sample y, and $\mathbf{L}(y, \omega_i)$ is the exitant radiance from the light. Both $\mathbf{B}$ and $\mathbf{L}$ are evaluated over the full spectrum.

In contrast, ReSTIR DI may employ a simplified proxy target distribution $\widehat{D}_i$ for resampling:

$$\widehat{D}_i(x, \omega_o, y) = (\widehat{B}_d + \widehat{B}_s)(x, \omega_o, \omega_i) V(x, y) \widehat{L}(y, \omega_i) \quad (10)$$

where $\widehat{B}_d$ and $\widehat{B}_s$ are scalar proxy BSDF terms for diffuse and specular reflection, respectively, and $\widehat{L}$ is a scalar light radiance proxy. Transmission terms are omitted despite zero contribution to direct illumination. Since $\widehat{D}_i$ is used only for importance resampling rather than final shading, such approximations trade accuracy for efficiency.

Furthermore, visibility evaluation requires additional ray tracing and is therefore sometimes omitted or deferred during resampling. In this work, we assume an unbiased ReSTIR DI variant with initial visibility reuse. The overall ReSTIR DI pipeline can be summarized as follows:

- **Initialization:** Generate candidate light samples using NEE and BSDF sampling, and resample them using RIS with a target distribution that ignores visibility.
- **Initial candidate evaluation:** Evaluate visibility for the selected initial candidate. Occluded samples receive zero contribution and are therefore excluded from further reuse.
- **Spatio-temporal reuse:** Reuse samples across neighboring pixels and frames using pairwise MIS, following the previous ReSTIR formulation.

2.2 Control Variates for Direct Illumination

These optimizations introduce several difficulties for control variates. First, the proxy target distribution $\widehat{D}_i$ is scalar and cannot directly serve as an auxiliary function for full-spectrum shading. Second, ignoring visibility during reuse of control variates breaks unbiasedness. Finally, the shift mappings used in ReSTIR DI are typically simple (e.g., reusing light UV coordinates), which can lead to high-variance difference estimators when applied naively.

To address these issues, we revise the role of spatio-temporal control variates (STCV). Instead of directly estimating $\mathbf{D}_i$ via control variates, we first construct a proxy auxiliary estimate $\dot{\mathbf{D}}_i$ using information already available from the ReSTIR DI resampling passes. We estimate $\dot{\mathbf{D}}_i$ during spatio-temporal reuse as F_i in ReSTIR PT. To obtain direct illumination $\mathbf{D}_i$, the final estimator is then formed using a standard control variate decomposition:

$$\langle \mathbf{D}_i \rangle = \langle \dot{\mathbf{D}}_i \rangle + \langle \mathbf{D}_i - \dot{\mathbf{D}}_i \rangle. \quad (11)$$

The difference term is estimated with the final candidate sample every frame independently, so we estimate the same accurate direct illumination as ReSTIR. The proxy $\dot{\mathbf{D}}_i$ is designed to closely match $\mathbf{D}_i$ while remaining stable under ReSTIR shift mappings. Specifically, we reuse most computations from the ReSTIR target function $\widehat{D}_i$ and define:

$$\dot{\mathbf{D}}_i(x, \omega_o, y) = \frac{\rho_i}{\widehat{\rho}_i} \widehat{B}_d(x, \omega_o, \omega_i) V(x, y) \mathbf{L}(y, \omega_i) \quad (12)$$

where ρ_i and $\widehat{\rho}_i$ denote the albedos of the true and proxy BRDFs, respectively.

Evaluating full-spectrum light radiance $\mathbf{L}$ is typically cheap, so we use the true spectral light while retaining scalar BSDF terms. We intentionally omit the specular proxy term $\widehat{B}_s$, as specular contributions tend to produce large, unstable values in the difference estimator. To compensate for the full-spectrum BRDF, we use the ratio between the true BRDF albedo and the proxy BRDF albedo. The albedo ratio compensates for the mismatch between the scalar proxy and true spectral BRDF values.

Compared to the original ReSTIR DI target distribution, this formulation requires only an additional evaluation of full-spectrum lighting during resampling. To handle visibility consistently, we accumulate $\dot{\mathbf{D}}_i$ only for the final selected candidate after RIS, rather than for all initial candidates. For temporal reuse, we explicitly re-evaluate visibility to avoid propagating errors across frames.

3 Additional Experiments

3.1 Denoising

Color noise can be partially alleviated by applying a post-process denoising method in real-time rendering. In Fig. 2, we compare our method against ReSTIR after denoising with the NVIDIA OptiX AI denoiser [NVIDIA Corporation 2026]. Denoisers typically reduce variance by spatially averaging neighboring pixels, guided by auxiliary buffers such as depth and normals. However, when color noise originates from sampling, the denoiser struggles to distinguish true high-frequency scene content from color fluctuations caused by sampling variance. As a result, the ReSTIR images often retain residual color artifacts even after denoising. In contrast, our method reduces color noise at the sampling stage by reusing path contributions, producing a more accurate and stable signal that is easier to denoise and preserves better high-frequency details.

3.2 Design choices for STCV

Our spatio-temporal control variates (STCV) differ from the original image-space control variates [Rousselle et al. 2016] in the following two aspects. First, we randomly select spatial neighbors within a

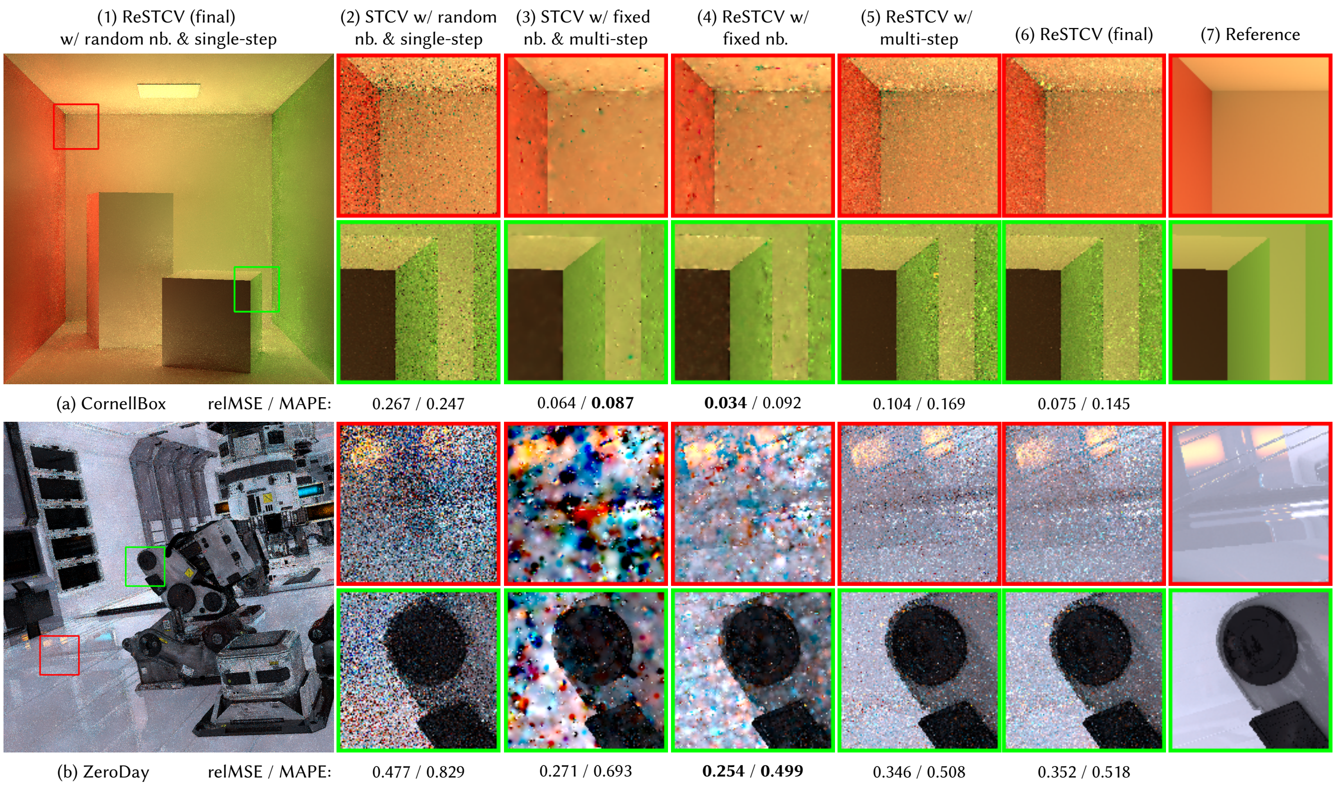

Fig. 1. Ablation study on STCV and ReSTCV using fixed/random neighbors or single/multiple updates. (ZeroDay © Mike Winkelmann (Beeple), CC BY 4.0).

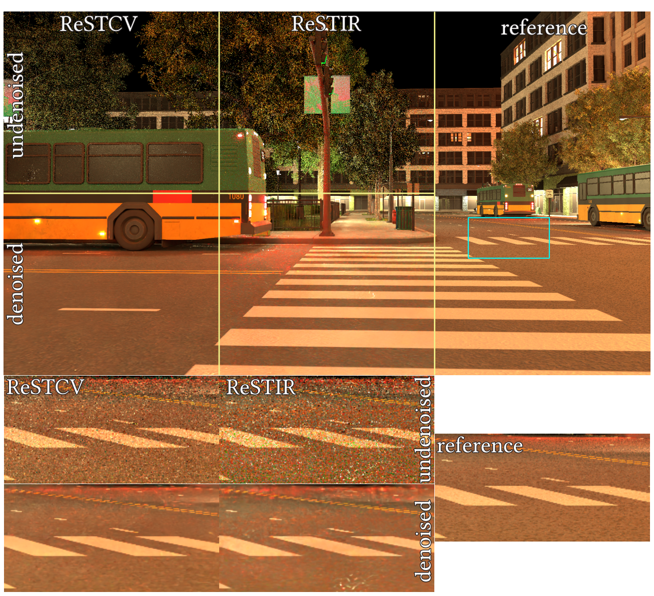

Fig. 2. Comparison between ReSTCV and ReSTIR before and after denoising. Images at the bottom are zoomed-in insets. NVIDIA (Emerald Square from ORCA (© Nicholas Hull, CC BY-NC-SA 3.0)).

radius, similar to ReSTIR's spatial reuse, while the original ICV always uses four adjacent neighbors. Second, we compute the STCV estimators using a single iteration per frame, while the original ICV updates the estimates using multiple iterations. We perform an ablation study on these two design choices, applying fixed neighbors and multiple-step updates (5 iterations) to STCV and ReSTCV, and show rendering comparisons in Fig. 1. While choosing four fixed neighbors (effectively estimating pixel gradients) often leads to smaller error numbers, this introduces severe structural visual artifacts and thus is less robust than choosing spatial neighbors randomly. Although applying multiple-step updates is critical to ICV, combining it with ReSTCV increases the computational cost (adding $0.5 \sim 1$ ms for every extra iteration) but provides diminishing returns in visual quality. Our ReSTCV method effectively amortizes multiple-step updates by leveraging spatio-temporal reuse at each frame, achieving similar visual quality with only a single-step estimation. Based on these observations, we adopt random neighbor selection and single-step estimation in our final method.

References

James T. Kajiya. 1986. The rendering equation. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '86)*. Association for Computing Machinery, New York, NY, USA, 143–150. doi:10.1145/15922.15902

NVIDIA Corporation. 2026. NVIDIA OptiX AI Denoiser. https://developer.nvidia.com/optix-denoiser. Accessed: 2026-01-21.

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