Eurographics Symposium on Rendering 2026
I. Gkioulekas and A. Jarabo
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 45 (2026), Number 4

Difference-aware Filtering for Event Camera Simulation

Juhyeon Kim Wojciech Jarosz Adithya Pediredla

Dartmouth College

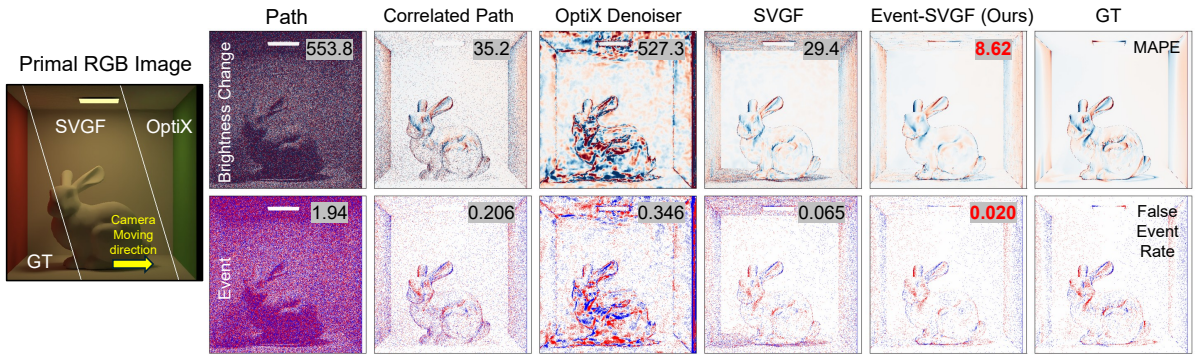

Figure 1: *Comparison of event rendering results for the* CORNELL-BUNNY *scene, where the camera moves to the right, under a roughly equal-time budget (∼5 ms). The figure visualizes brightness (log-intensity) changes and resulting event images, with MAPE and false event rates shown in the upper-right corner of each result. Although existing denoising methods produce visually clean primal RGB images, temporal differencing amplifies residual temporal instability and generates numerous false events. The proposed Event-SVGF reduces spurious events while preserving accurate event responses through correlated sampling and difference-aware denoising.*

Abstract

*We present EventSVGF, an event camera rendering framework based on spatiotemporal variance-guided filtering (SVGF) [SKW*17], designed to achieve high temporal accuracy especially in high-frequency regions, which is critical for faithful event simulation. Unlike conventional rendering, event cameras measure temporal changes in brightness (log-intensity), requiring accurate estimation of per-pixel, frame-to-frame differences. However, naively computing temporal differences from primal-domain RGB images leads to severe noise, as existing denoising methods are designed for primal signals rather than their differences. Our key contribution is a method that directly denoises the pixel-wise temporal difference signal using correlated sampling, formulated as a difference-aware extension of the SVGF pipeline, termed EventSVGF. EventSVGF incorporates a novel edge-stopping function, an adapted temporal accumulation scheme, and an albedo demodulation strategy, all tailored for accurate event camera simulation. Our method achieves stable results at low sampling rates (2 spp), whereas existing approaches typically require significantly higher sampling budgets (32–512 spp). We demonstrate EventSVGF on dynamic scenes, showing improved accuracy and high-frequency temporal stability in event simulation compared to prior works.*

CCS Concepts

• ***Computing methodologies*** → ***Ray tracing;*** • ***Hardware*** → *Sensors and actuators;*

1. Introduction

An *event camera* is an emerging imaging modality inspired by neuromorphic engineering principles. Unlike traditional frame-based cameras that capture images synchronously at fixed time intervals, event cameras operate asynchronously at the pixel level. Each pixel independently responds to changes in brightness (log-intensity) and emits events whenever sufficient temporal contrast is detected, typically using dynamic vision sensors (DVS) [CVD*24; GDO*20].

Event cameras offer several advantages over conventional cameras. They achieve a wide dynamic range (often exceeding 120 dB), enabling robust operation under extreme lighting conditions such as high-contrast or low-light environments. Their microsecond-level temporal resolution allows accurate capture of fast motion without motion blur, and the asynchronous sensing mechanism reduces redundant data acquisition, resulting in lower power consumption. Due to these strengths, event cameras have been widely adopted in various imaging and vision applications where conventional

cameras often struggle, including object tracking and recognition [MFPA18; RZL*18], depth reconstruction [FLB22; KLD16], edge detection [MHH*20; VGB16], motion deblurring [JZZ*20; XYW*21], low-light imaging [LPZ*24; LYL*23], and simultaneous localization and mapping (SLAM) [HZZT23; MGN*22].

Efficient and accurate simulation of event cameras is crucial for imaging research, as it enables rapid prototyping and evaluation of algorithms without requiring expensive or specialized hardware. Many existing event camera simulators generate event sequences from real-world RGB video frames [BA17; HLD21; KND12]. While some methods generate event sequences from rendered synthetic 3D scenes, they typically rely on noiseless input images [GGHS20; RGS18]. Such inputs are often obtained via rasterization, which is efficient and noise-free but not physically accurate. A physically accurate alternative is Monte Carlo (MC) path tracing; however, it requires a large number of samples to produce noiseless inputs for reliable event simulations.

A key challenge in MC estimation is achieving *high-frequency temporal stability*, whose lack is the main reason for numerous false events (Fig. 1). In practice, even more samples are required to obtain noise-free event simulations than to produce perceptually noise-free primal RGB images. This is because events are triggered by temporal differences in brightness, which are often much smaller in magnitude than the original signal and are therefore highly sensitive to MC noise [MYMK24]. Consequently, developing physically accurate yet efficient rendering that operates with a small number of samples is essential for event camera simulation.

Several recent works have begun to consider low-spp MC rendering as input to event simulation and mitigate noise using adaptive denoising [TYKM23] or sampling [MYMK24]. However, such approaches are primarily designed for offline rendering and still require a relatively large number of samples (32–256 spp per frame), limiting their applicability to large-scale dataset generation and scalable applications. Furthermore, these methods do not exploit inter-frame correlations in path sampling, leaving significant potential for additional variance reduction.

To address these challenges, we propose *Event-SVGF*, an event-camera-aware adaptation of spatiotemporal variance-guided filtering (SVGF) [SKW*17; SPD18], a real-time denoising technique for path tracing. Directly applying SVGF (or other primal image denoising methods) still suffers from temporal instability, leading to false events (Fig. 1). Instead, our key idea is to denoise temporal differences directly using correlated sampling, enabling *high-frequency temporal stability*, which is critical for accurate event camera simulation. This approach is inspired by temporal gradient-domain path tracing [MKD*16] and is conceptually related to temporal gradient denoising in A-SVGF [SPD18]. However, we specifically adopt a non–motion-compensated formulation, which is required for event camera simulation. To this end, we introduce a *difference-aware* filtering scheme that enables effective bilateral filtering directly on frame-difference signals. Our method remains physically based while operating at a low sampling budget (requiring only 2 spp), providing a practical and efficient simulation tool for a wide range of event camera imaging applications.

Project Page: https://juhyeonkim95.github.io/project-pages/event_svgf.

2. Previous work

Event Camera Rendering. Most existing event camera simulators generate events from precomputed sequences of noiseless intensity images, either captured or synthetically rendered [CVV*25; SDK*24]. Early approaches such as VSBE [KND12] and PIX2NVS [BA17] trigger events from frame-to-frame log-intensity changes, while ESIM [RGS18] integrates rendering with adaptive temporal sampling and was later extended with frame interpolation in rpg_vid2e [GGHS20]. The v2e framework [HLD21] further improves realism by modeling sensor imperfections such as hot pixels and temporal noise. However, these methods assume noise-free intensity frames and are therefore not directly applicable to noisy MC rendering outputs.

More recent work incorporates event generation into physically based MC rendering pipelines. Tsuji et al. [TYKM23] improve temporal stability by selectively updating denoising parameters, while Manabe et al. [MYMK24] formulate event triggering as a statistical decision problem, enabling adaptive sampling and early termination. Li et al. [LBL*25] further propose a lightweight neural network that generates event streams directly from noisy MC inputs. Although these approaches couple event generation with MC light transport, they still require relatively high sample counts.

Gradient-Domain Rendering. Gradient-domain rendering is closely related to addressing temporal stability in event camera simulation. These methods sample both the primal image and the *difference image*, and reconstruct the final result via a screened Poisson equation. To efficiently estimate differences, these methods use *correlated sampling* through shift mappings (e.g., reusing vertices or random seeds), reducing variance compared to independent evaluations. Early work focused on spatial gradients using Metropolis Light Transport [LKL*13] or path tracing [KMA*15; MKA*15]. Manzi et al. [MKD*16] extended this idea to the temporal domain (T-GDPT), estimating temporal gradients across frames with shared random seeds. Conceptually, T-GDPT is closely related to improving temporal stability in event camera simulation, which is critical for preventing false events. However, T-GDPT still requires a relatively large number of samples to obtain low-noise temporal gradients, whereas our goal is to operate in the low-spp regime. Therefore, additional denoising of the gradient signal itself becomes essential, which we address through dedicated filtering.

Monte Carlo Denoising. Numerous techniques have been proposed to mitigate MC noise in path tracing [ZJL*15]. In this work, we adopt SVGF [SKW*17] as our baseline denoiser for both primal and difference estimates, as it is designed to operate at 1 spp, aligning with our low-spp setting. SVGF performs spatio-temporal denoising by combining temporal accumulation with a hierarchical *à-trous* wavelet filter, using edge-aware weights that preserve structural details while reducing noise. While the original SVGF filters only the primal image, its extension, A-SVGF [SPD18], also denoises temporal gradients using the same wavelet framework and leverages them for adaptive temporal filtering. Although A-SVGF is also closely related to our target task, it requires adaptation to handle per-pixel log-intensity differences across frames, which are essential for event camera simulation.

3. Preliminaries

3.1. Event Camera Preliminaries

For event cameras, each pixel responds to changes in the log of intensity $L = \log I$, which is called *brightness*. While several papers adopt this simple logarithmic response model [MYMK24; RGS18; TYKM23], it is often preferred to use a lin-log model, as DVS tends to operate linearly for small intensities due to dark current, which causes a bias voltage [ALM*24; BA17; HLD21; JMR*21]. In this paper, we adopt the following continuous model:

$$L = \log(I + I_\epsilon), \tag{1}$$

where I_ϵ is the intensity bias. Note that for $I \ll I_\epsilon$, L operates linearly, while for $I \gg I_\epsilon$, L operates logarithmically. Once the temporal brightness change ΔL crosses a threshold $\pm C$, an event is triggered with its polarity information, asynchronously across all pixels, as illustrated in Fig. 2. C varies by sensor, but typical value is $10-50\%$ change of the intensity [GDO*20].

3.2. SVGF Preliminaries

The key idea of SVGF is to exploit temporal accumulation to reduce variance over time, while applying an edge-aware spatial filter that preserves geometric and shading discontinuities.

Temporal Filtering Given a noisy input signal C_t at frame t, SVGF computes a temporally accumulated signal C'_t using reprojected samples (by motion vector) from the previous frame C'_{t-1}:

$$C'_t = \alpha C_t + (1-\alpha) C'_{t-1}, \tag{2}$$

a formulation widely used in real-time rendering and temporal antialiasing [YLS20]. Here, α controls the mixing ratio with the historical value, which is typically set between $0.05-0.2$. The reprojected coordinate can be fractional, bilinear filtering is commonly adopted within consistent geometry (depth and normal) pixels.

Spatial Filtering The spatial filtering stage iteratively applies an edge-aware kernel over a local neighborhood Ω. The spatially filtered value c at iteration $i+1$ at pixel p is given by

$$c^{(i+1)}(p) = \frac{\sum_{q\in\Omega} h(q)\, w^{(i)}(p,q)\, c^{(i)}(q)}{\sum_{q\in\Omega} h(q)\, w^{(i)}(p,q)}, \tag{3}$$

where $h(q)$ denotes the filter kernel (i.e., à-trous wavelet weights), and $w^{(i)}(p,q)$ is an edge-stopping weight that prevents blurring across discontinuities. The weight is defined as a product of feature-based terms:

$$w^{(i)}(p,q) = w_z(p,q)\, w_n(p,q)\, w_l^{(i)}(p,q), \tag{4}$$

which encode similarity in depth, normal, and luminance, respectively (details are in Appendix A). In A-SVGF, the authors additionally perform denoising of temporal gradients using the same filtering pipeline, which we discuss more in a later section.

4. Proposed Method

In this work, we focus on efficiently estimating temporal brightness *changes* ΔL between consecutive frames rather than the absolute brightness L. Although these two quantities appear closely related, they emphasize fundamentally different frequency components.

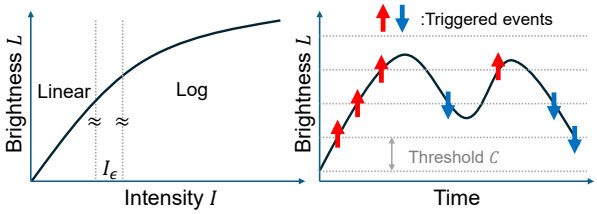

Figure 2: ***Left:*** *Event camera's lin-log response model to the pixel intensity.* ***Right:*** *An example of event trigger by brightness change over time, where arrow color and direction denote polarity.*

4.1. Why Temporal Differences (ΔL) Are Important

Event-based cameras capture high-frequency temporal variations, while they do not capture low-frequency intensity components. As shown in prior works [SBM18; SGT*26], reconstructing intensity by integrating events amplifies low-frequency drift, resulting in poor recovery of absolute brightness. This suggests that low-frequency intensity is inherently ill-conditioned in event-based representations, whereas temporal differences are well-constrained and can be estimated more reliably. Consequently, unlike conventional rendering which focuses on reconstructing absolute intensity (L or I), it is more important to accurately estimate temporal differences (ΔL) which directly govern event generation and determine the *high-frequency stability* of the resulting signal.

To quantify this notion of high-frequency stability, consider the probability of generating false events due to the temporal noise. Theoretically, this is given by (see Appendix B):

$$(\text{Probability of False Event}) \propto \frac{\sigma_{\Delta L}}{C}, \tag{5}$$

where $\sigma_{\Delta L}$ denotes the standard deviation of noise in ΔL, and C is the event threshold. The key observation is that false event generation depends on the variability of ΔL rather than the variability of L itself. For example, two signals with identical σ_L may exhibit different levels of temporal oscillation; the one with larger $\sigma_{\Delta L}$, which oscillates more, will produce more frequent false events. Another observation is that, even when $\sigma_{\Delta L} < C$, reducing $\sigma_{\Delta L}$ still remains desirable, as it suppresses noise-induced event generation. Therefore, we aim to minimize $\sigma_{\Delta L}$, which improves high-frequency stability and reduces false event generation due to temporal MC noise. Actual event generation from evaluated ΔL is discussed in Sec. 4.6.

4.2. Evaluating the Noiseless Temporal Difference

We begin by considering the per-pixel intensity difference between consecutive frames before delving into brightness differences:

$$\Delta I_t = I_t - I_{t-1}, \tag{6}$$

and our goal is to obtain a low-noise estimate of ΔI_t. A naive approach renders each frame independently with many samples and subtracts the results. However, independent sampling causes noise to accumulate in the difference, resulting in requiring twice the sample budget for convergence [MYMK24; SPD18].

Another reasonable approach is to apply existing primal RGB image denoising pipelines:

$$D[I_t] - D[I_{t-1}], \tag{7}$$

where $D[\cdot]$ denotes a denoising operator (e.g., SVGF). While temporal reprojection, adopted in many denoising algorithms, introduces some correlation between frames (cf. Eq. (2)) and reduces noise, these methods still incrementally incorporate new uncorrelated samples at each frame. As a result, residual temporal noise remains, leading to per-pixel temporal instability that is particularly pronounced when explicitly computing differences ΔI_t.

4.3. Temporally Correlated Sampling

A more principled way to reduce the variance of ΔI_t is to introduce explicit correlation by generating samples in *pairs*, commonly referred to as *shift mapping* in gradient-domain rendering. Following A-SVGF and T-GDPT, we adopt random seed reuse (i.e., primary-sample-space correlation) between consecutive frames. We operate at full resolution, as the difference signal itself is the primary quantity of interest.

Concretely, each frame t is rendered twice: pass 1 uses the *previous* frame's seed r_{t-1}, and pass 2 uses the current seed r_t. Because pass 1 of frame t and pass 2 of frame $t-1$ share the seed r_{t-1}, their difference difference

$$I_{t,\text{diff}} = I_t^1 - I_{t-1}^2 \tag{8}$$

is a low-variance estimate of the temporal change. The same two passes also yield a primal estimate

$$I_{t,\text{primal}} = (I_t^1 + I_t^2)/2, \tag{9}$$

which can serve as a denominator, correction term, or complementary filter input in later stages.

Nevertheless, correlated sampling alone is insufficient at the low sample counts we target. We thus denoise the difference signal $D[\Delta I_t]$, in a manner similar in spirit to A-SVGF [SPD18], but with several adaptations tailored to event camera rendering.

4.4. Non-Motion-Aligned Temporal Difference

A key distinction from prior work on temporal differences is that we must consider the *non-motion-aligned* temporal difference:

$$\text{(non-motion-aligned)} \quad \Delta I_t[s] = I_t[s] - I_{t-1}[s], \tag{10}$$

$$\text{(motion-aligned)} \quad \Delta I_t[s] = I_t[s] - I_{t-1}[s + \Delta_t s], \tag{11}$$

where s denotes a pixel and $\Delta_t s$ is the motion vector from frame t to $t-1$. Motion-aligned version reduces variance by comparing the same surface point across frames and is therefore widely used in standard rendering. However, event cameras *do not* perform such compensation – they respond to brightness changes observed at a fixed pixel location. Consequently, our goal is to estimate the non-motion-aligned temporal difference directly. While one could reconstruct the non-motion-aligned difference from a motion-aligned difference with an additional spatial correction term, doing so complicates the implementation and merely transfers the mismatch to the correction term (See Appendix C). We therefore directly denoise the non-aligned temporal difference throughout our pipeline.

Note that this distinction applies only to temporal-difference estimation. Motion vectors are still used later for temporal accumulation (Sec. 4.4.2), where they reproject already-denoised quantities without altering the target non-motion-aligned temporal difference.

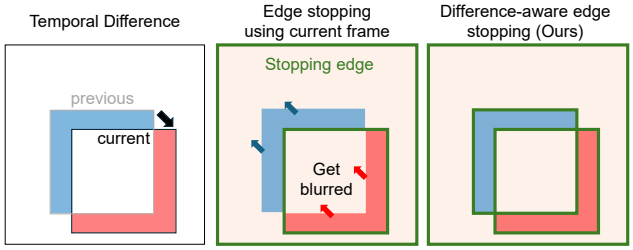

Figure 3: *Our proposed difference-aware edge-stopping filter prevents regions that have very different primary hits across frames (marked in red and blue) from being blurred with other regions that have similar current-frame information (indicated by arrows).*

4.4.1. Difference-aware Edge-Stopping Filter

Applying the SVGF edge-stopping filter (Sec. 3.2) directly to temporal differences leads to blurred edges in the non-motion-aligned case, as it relies only on features from the current frame, assuming motion-vector alignment. This is illustrated with blue and red regions in Fig. 3. To address this artifact, we introduce a difference-aware edge-stopping filter that restricts denoising to regions with consistent temporal differences.

The implementation is straightforward: the overall pipeline follows the SVGF à-trous filtering framework, but we modify the edge-stopping weights as

$$w_{\text{diff}} = w_{\text{curr}} \cdot w_{\text{prev}}, \tag{12}$$

where w_{curr} and w_{prev} are computed from features (i.e., depth, normal, luminance) of the current and previous frames, respectively. This formulation ensures that filtering is restricted to regions that are consistent across both frames. For the luminance term, instead of multiplying, we compute it directly from the difference signal. We use a single à-trous filtering pass that jointly processes both primal and difference signals, using separate channels with their corresponding edge-stopping weights.

4.4.2. Difference-aware Temporal Accumulation

Prior works typically do not apply temporal accumulation to temporal differences, as they are treated as intermediate quantities for which moderate noise is acceptable. However, we observe that denoising a single-frame difference is insufficient to achieve high-quality difference results. Therefore, we introduce temporal accumulation for the temporal difference signal as well.

We begin with the standard temporal accumulation of the primal signal Eq. (2) for frame $t, t-1$:

$$I'_t[s] = \alpha I_t[s] + (1-\alpha) I'_{t-1}[s + \Delta_t s], \tag{13}$$

$$I'_{t-1}[s] = \alpha I_{t-1}[s] + (1-\alpha) I'_{t-2}[s + \Delta_{t-1} s]. \tag{14}$$

Subtracting the two equations and rearranging yields:

$$(I'_t - I'_{t-1})[s] = \alpha (I_t - I_{t-1})[s] + (1-\alpha) \times \tag{15}$$

$$\Big[(I'_{t-1} - I'_{t-2})[s + \Delta_t s] + I'_{t-2}[s + \Delta_t s] - I'_{t-2}[s + \Delta_{t-1} s] \Big]. \tag{16}$$

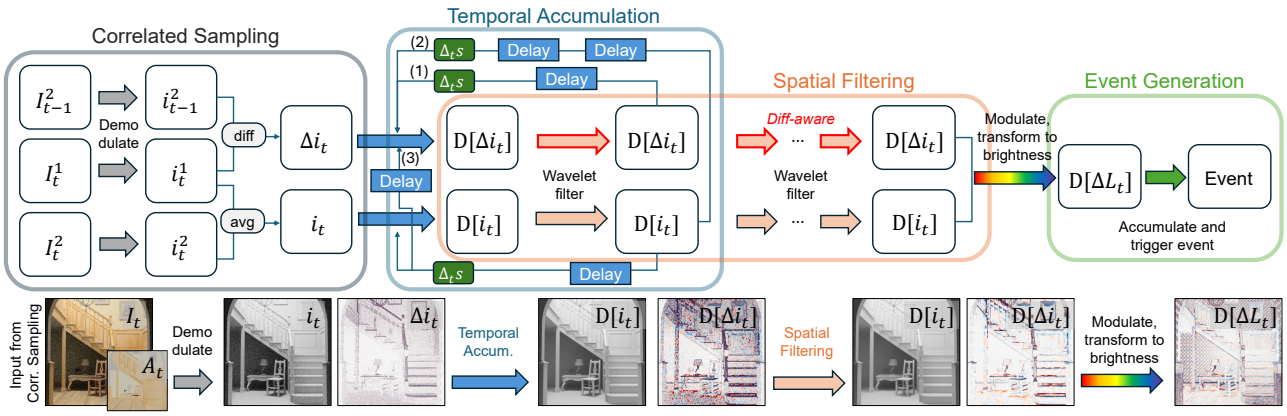

Figure 4: *Overall pipeline of the proposed Event-SVGF. (A) We first perform correlated sampling and obtain difference and primal samples; albedo demodulation is also applied here. (B) We then temporally accumulate both primal and difference samples using the motion vector $\Delta_t s$. (C) Spatial filtering is performed via an à-trous wavelet filter, where a difference-aware weight is used for denoising the difference signal. (D) Finally, we reconstruct brightness differences from the denoised primal and difference signals and generate events. The bottom row shows intermediate results from each stage of the pipeline.*

This expression resembles the standard temporal accumulation of ΔI_t, but introduces an additional correction term involving I'_{t-2}. For the first two terms inside the bracket, evaluated at $s+\Delta_t s$, we reuse the primal temporal accumulation pipeline with the corresponding textures, while the last term is obtained via a history lookup from the previous frame. Each term is visualized in Fig. 4 (temporal accumulation stage), labeled as (1), (2), and (3).

4.4.3. Difference-aware Albedo Demodulation

Rather than directly denoising the intensity I, SVGF performs albedo demodulation and denoises the illumination component:

$$I_t = A_t\, i_t + E_t, \tag{17}$$

where A_t denotes albedo, i_t denotes illumination, and E_t denotes emission at frame t. This separation prevents high-frequency texture (encoded in A_t) from being blurred during denoising.

In A-SVGF, albedo demodulation is not applied to temporal differences. However, in our setting, the temporal difference itself is the final output, making demodulation essential for preserving fine details. Under non-motion-aligned differences, demodulation is not straightforward, as I_t and I_{t-1} may correspond to different surfaces and thus different albedos. We therefore decompose the temporal difference as

$$\begin{aligned} I_t - I_{t-1} &= (A_t i_t + E_t) - (A_{t-1} i_{t-1} + E_{t-1}) && (18)\\ &= A_t(i_t - i_{t-1}) + (A_t - A_{t-1}) i_{t-1} + (E_t - E_{t-1}), && (19) \end{aligned}$$

which now depends on both primal and difference values. Here, one important consideration is that filtered texture values (e.g., obtained via trilinear filtering) should be used for the albedo values to reduce texture-induced oscillations in the $(A_t - A_{t-1})i_{t-1}$ term.

4.5. Reformulation of Brightness Change

Using the components introduced above, we now address the main objective: estimating the brightness change ΔL between two consecutive frames,

$$\Delta L_t = L_t - L_{t-1} = \log(I_\epsilon + I_t) - \log(I_\epsilon + I_{t-1}), \tag{20}$$

from Eq. (1). A direct application of SVGF to this log-expression is undesirable for two reasons: (1) filtering is typically designed for linear quantities, and (2) the logarithmic form prevents straightforward albedo demodulation. To address this, we reformulate the brightness change in terms of primal and difference components:

$$\begin{aligned} \Delta L_t &= \log(I_\epsilon + A_t i_t + E_t) - \log(I_\epsilon + A_{t-1} i_{t-1} + E_{t-1}) \\ &= \log\left(1 + \frac{(A_t i_t - A_{t-1} i_{t-1}) + (E_t - E_{t-1})}{I_\epsilon + A_{t-1} i_{t-1} + E_{t-1}}\right) \\ &= \log\left(1 + \frac{A_t(i_t - i_{t-1}) + (A_t - A_{t-1}) i_{t-1} + (E_t - E_{t-1})}{I_\epsilon + A_{t-1} i_{t-1} + E_{t-1}}\right). \end{aligned}$$

Therefore, we obtain the final denoised estimator $D[\Delta L_t]$ as

$$\boxed{\log\left(1 + \frac{A_t\, D[\Delta i_t] + (A_t - A_{t-1})\, D[i_{t-1}] + (E_t - E_{t-1})}{I_\epsilon + A_{t-1}\, D[i_{t-1}] + E_{t-1}}\right)}, \tag{21}$$

which depends on both denoised difference and primal images. The overall pipeline for obtaining denoised ΔL_t is illustrated in Fig. 4.

4.6. Event Generation from ΔL

Up to this point, our framework focuses on estimating the temporal brightness difference ΔL_t. To generate events, this estimate must be converted into threshold crossings between consecutive frames.

Direct Accumulation The most direct formulation accumulates brightness changes over time to reconstruct the brightness signal:

$$L_t = L_0 + \sum_{\tau=1}^{t} \Delta L_\tau, \tag{22}$$

where L_0 is the initial offset brightness value. An event is generated whenever the accumulated brightness deviates from the reference

brightness L_{ref} by more than the threshold $\pm C$, after which L_{ref} is updated accordingly.

Since ΔL_t is estimated through a denoising process, direct accumulation may introduce low-frequency drift in L_t. However, importantly, this is not unique to our framework: real event cameras also exhibit low-frequency drift and are generally more reliable at capturing high-frequency brightness changes than absolute brightness values, as discussed in Sec. 4.1. Consequently, event-camera-based reconstruction commonly combines *complementary* signals that are reliable in different frequency bands: temporal differences from event cameras provide accurate high-frequency information, while primal intensity estimates from standard RGB cameras constrain the low-frequency content. Predict–correct recursion [SBM18] and Kalman filtering [WNSM21] have been proposed for causal brightness reconstruction. Screened Poisson reconstruction in gradient-domain rendering [KMA*15; MKD*16] can also be viewed as a complementary filtering technique, but it provides a global solution rather than a causal one. While the design of such complementary reconstruction filters is beyond the scope of this work, our method provides low-noise estimates of ΔL_t, making it well suited for downstream reconstruction and event-generation pipelines.

Probabilistic Generation We also consider a simplified stochastic event-generation model that avoids explicit accumulation. Since the initial offset brightness L_0 is generally unknown in actual DVS [GDO*20], we model the threshold phase as uniformly distributed over $[0, C)$. Under this assumption, events can be generated independently at each frame with probabilities proportional to the absolute value of the estimated temporal difference [CS14]. This formulation can also be interpreted as a mean-field approximation of the accumulation-based model: it preserves the same marginal event statistics and expected event counts while ignoring temporal correlations across frames. We adopt this stochastic event-generation model unless otherwise noted.

5. Results

We implemented the proposed rendering pipeline using the Falcor framework [KCK*22] and conducted experiments on a desktop equipped with a Ryzen Pro 3955WX CPU and an RTX 3090 GPU. Instead of using full RGB channels, we operate on a single luminance channel; thus, all intensity values refer to luminance.

5.1. Rendering Comparison with Baseline Methods

We first present rendered difference images for intensity (ΔI), brightness (ΔL), and the resulting events generated by several methods. These include naive path tracing (with and without correlated sampling via random-seed reuse) and primal-domain denoising algorithms such as OptiX and SVGF, both available in the Falcor framework. The path tracing methods use 4 spp, while the denoising-based methods use a budget of 2 spp, resulting in comparable runtimes. All images are rendered at a resolution of 1024×1024, with the camera moving forward. We used $\alpha = 0.1$ and $I_\epsilon = 10^{-8}$. For all other SVGF parameters, we used the default values provided by the Falcor framework. Ground-truth (GT) images were generated using correlated sampling with a large number of samples. The results are shown in Fig. 6. Quantitative errors

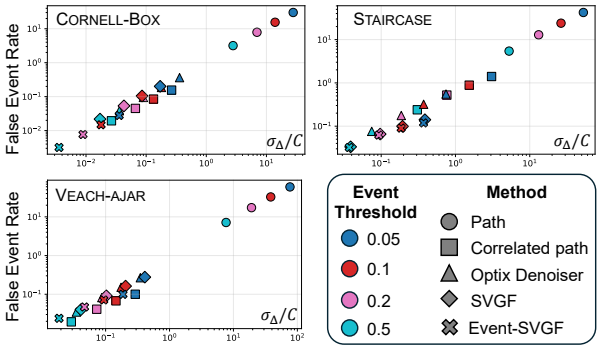

Figure 5: *False event occurrence for different event thresholds C and methods with varying $\sigma_{\Delta L}$. A clear linear relationship is observed between the false event rate and $\sigma_{\Delta L}/C$.*

(MAPE, Mean Absolute Percentage Error) are displayed in the top-left corner of each subfigure, while rendering times are shown in the bottom-right corner. For textured scenes, we also show the demodulated Δi in the upper-right corner for SVGF-based methods. For the "Event" row, we report the average false event rate, defined as the number of events generated where the GT contains no event.

The naive path tracer without correlation produces extremely noisy difference images and leads to incorrect event generation across the entire image. Introducing correlated sampling significantly improves the results in terms of temporal differences; however, it suffers from fireflies and especially fails to accurately capture indirect illumination (e.g., it fails to reproduce shadow region in the CORNELL-BOX or STAIRCASE scene), which requires a higher sample count for convergence. Primal-domain denoisers such as OptiX and SVGF perform well on the reconstructed images themselves, but they produce inaccurate difference signals characterized by ripple-shaped artifacts. These artifacts lead to false event generation and are particularly pronounced in the OptiX denoiser. In contrast, our proposed EventSVGF achieves the best performance in rendering difference images, which in turn results in more accurate event generation while suppressing false events.

5.2. False Event Occurrence Analysis

We also perform a false event occurrence analysis across different methods to validate the relationship described in Eq. (5). We first estimate the temporal noise level $\sigma_{\Delta L}$ from a sequence of 500 frames. To avoid overestimating noise due to abrupt discontinuities (which are not noise), we exclude extreme outlier values. We then evaluate the false event rates for different thresholds $C \in [0.05, 0.1, 0.2, 0.5]$ and plot their pixel-frame-wise average as a function of $\sigma_{\Delta L}/C$ for each method in Fig. 5.

Overall, we observe a clear linear relationship between the false event occurrence rate and $\sigma_{\Delta L}/C$, which supports our analysis and highlights the importance of temporal stability in event simulation. Our method consistently achieves the lowest false event rate, owing to its high-quality ΔL estimates with reduced temporal noise, resulting in lower $\sigma_{\Delta L}$.

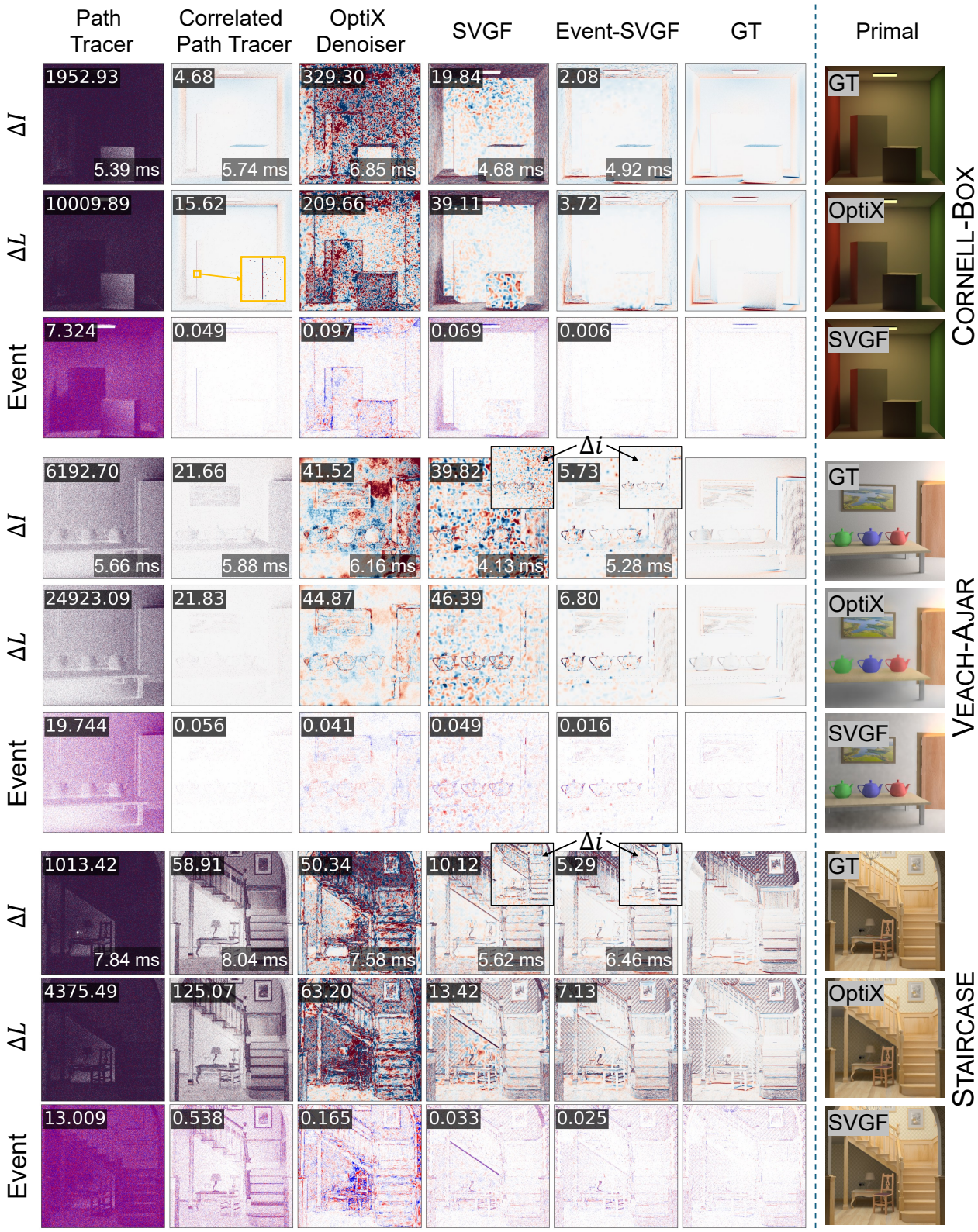

Figure 6: *We present denoised difference images produced by several methods. Compared to baseline approaches, our method achieves significantly higher quality across all difference quantities, as well as in the resulting event generation. The numbers shown in each subfigure indicate the MAPE, render time per frame, and false event rate (for "Event" rows).*

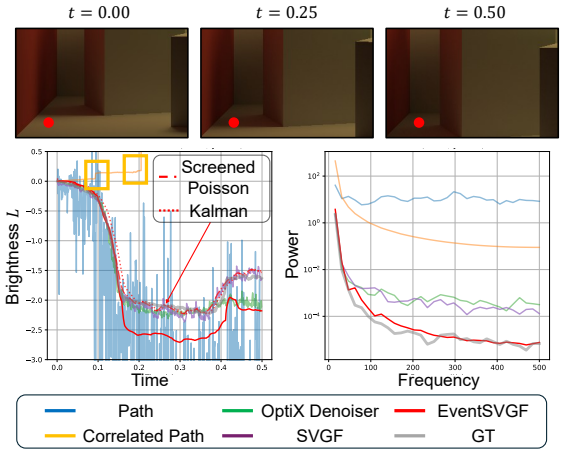

Figure 7: *We show the time- and frequency-domain brightness signal profiles at the specific pixel denoted by a red dot. We also show the reconstructed signal with complementary filters.*

5.3. Temporal and Frequency-Domain Noise Analysis

We further analyze the brightness signal in both the temporal and frequency domains (Fig. 7). We first plot the brightness L over time (accumulated from ΔL using Eq. (22)) at a specific pixel location denoted by a red dot. The uncorrelated path tracer exhibits severe temporal noise, which is significantly reduced when using correlated sampling; however, occasional sharp transitions remain due to fireflies in ΔL. Primal image denoising methods provide improved temporal stability, but some noise still exists. While this noise appears smaller than the event threshold, the probability of false event occurrences scales linearly with its magnitude, and therefore must be minimized. Our method clearly achieves the highest temporal stability, although it exhibits long-term drift due to accumulated bias in the gradient. As discussed in Sec. 4.1, this issue is not a critical problem for event camera simulation, which primarily depends on high-frequency changes.

We also perform a Fast Fourier Transform (FFT) over the simulated frames to characterize the frequency-domain noise profile. The proposed method clearly demonstrates superior stability, especially in the high-frequency range, compared to other methods (it is slightly worse at very low frequencies due to drift issues), which is essential for accurate event camera simulation.

Reconstruction using Complementary Filtering. To improve the low-frequency fidelity of our simulator, the difference signal can be combined with a primal signal using complementary filtering, as described in Sec. 4.6. We present results using both screened Poisson reconstruction and Kalman filtering, shown as dash-dotted and dotted lines, respectively, in the temporal plot of Fig. 7. Both approaches improve low-frequency behavior compared to using the difference signal alone, at the cost of additional filtering.

5.4. Comparison with Adaptive Filtering [TYKM23]

We also compare our method with a prior event camera simulation approach based on adaptive filtering [TYKM23]. Since their official implementation operates as a post-processing step on full video sequences and cannot be directly integrated into the Falcor pipeline, we perform this comparison in an offline manner.

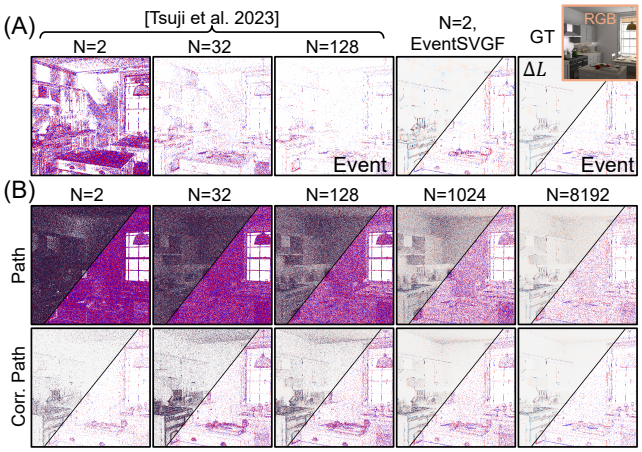

Figure 8: *(A) Comparison with the adaptive filtering method of Tsuji et al. [TYKM23] using different sample counts (N) in the* KITCHEN *scene. Their method clearly outperforms naive path tracing, but still requires a large number of samples to converge. (B) We also show results obtained with path tracing (w/o denoising) both without and with correlated sampling.*

The results are shown in Fig. 8-(A) with varying numbers of samples for the KITCHEN scene, where the camera is moving forward. Their method provides clear improvements over a naive path tracer (first row of Fig. 8-(B)) at the same number of spp. However, under a low-sample budget (2 spp), it exhibits significant noise. Using 32 spp, as suggested in their work, still does not yield satisfactory results for event simulation. Only at higher sample counts ($N = 128$) do we obtain near noise-free event simulation. In contrast, our method achieves faithful result with only 2 spp, owing to correlated sampling and a tailored difference denoising strategy.

5.5. Ablation Study

To evaluate the effectiveness of each component in the proposed EventSVGF pipeline, we perform an ablation study on the TEXTURED-CORNELL-BOX scene in Fig. 9 and show ΔI images.

First, we disable correlated sampling and use independent random seeds across frames. As shown in Fig. 9-(A), this introduces ripple-like artifacts similar to those observed in conventional SVGF and significantly increases noise. Next, we consider a variant that denoises the difference signal using only information from the current frame (i.e., using only w_{curr} in Eq. (12)). As shown in Fig. 9-(B), this results in noticeable edge blurring, consistent with the analysis in Fig. 3. Incorporating information from the previous frame helps preserve discontinuities and mitigates this issue. We also evaluate a variant without temporal accumulation of the difference signal (Fig. 9-(C)). Compared to the full method, the result is substantially noisier, indicating that temporal filtering is important for stabilizing the difference estimate when it is used directly as the output. Finally, we remove albedo demodulation in Fig. 9-(D). Without demodulation, the denoising process tends to blur high-

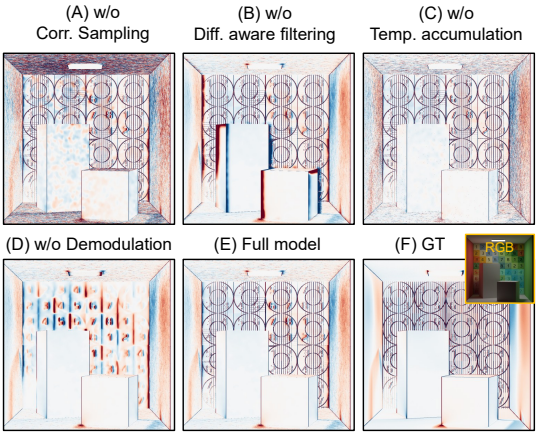

Figure 9: *Ablation study of the proposed method. We visualize denoised ΔI: (A) without correlated sampling, causing artifacts and increased noise; (B) without difference-aware filtering, resulting in edge blurring; (C) without temporal accumulation, producing a noisier estimate; (D) without albedo demodulation, resulting in texture blurring; (E) the full method; and (F) the ground truth. The camera moves to the right.*

frequency texture details. Overall, the full method (Fig. 9-(E)), which combines all of the above components, produces the highest visual quality and most closely matches the ground truth (Fig. 9-(F)). Slight texture blurring can still be observed in the center-right region, likely due to imperfect albedo demodulation, which could be further improved in future work.

5.6. Further Discussion

Correlated Sampling with Random Seed Reuse. Correlated sampling via shift mapping with random seed reuse tends to produce rare but large differences, especially for diffuse materials. For example, the same BRDF sample direction evaluated at slightly different surface positions rarely hits entirely different objects between consecutive frames. These abrupt changes can be observed both spatially (Fig. 6) and temporally (Fig. 7) highlighted by the yellow boxes. At very low spp, such events are rare, resulting in a relatively clean difference image with only occasional fireflies. As spp increases, the probability of observing these abrupt changes also increases, making the difference and resulting event appear noisier. At sufficiently high spp, however, the estimator converges to the ground truth and the noise in the event image decreases again. This behavior is illustrated in Fig. 8-(B). While naive path tracing improves monotonically with increasing spp, correlated sampling exhibits the non-monotonic behavior described above. We adopt random seed reuse due to its simplicity of implementation, although other strategies (e.g., path reconnection) could be explored in future work.

Low-Intensity Light. For low-intensity environments, brightness tends to behave linearly. We do not explicitly show results for low-brightness scenes; instead, we have also visualized ΔI in Fig. 6 and Fig. 9, indicating that our method also performs well in low-intensity regions where event cameras operate in the linear regime.

Large-Motion. For large motions, the primary hit points between frames can differ significantly, reducing the effectiveness of correlated sampling. We compare different camera velocities (moving forward) in the CORNELL-DRAGON scene in Fig. 10. EventSVGF significantly outperforms SVGF for small motions, where correlated sampling remains effective. However, as the motion magnitude increases, the benefit gradually diminishes.

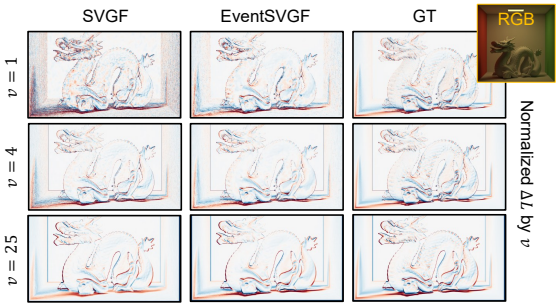

Figure 10: *Comparison under different camera velocities. The advantage of correlated sampling decreases as the velocity increases.*

SVGF Parameters. We use the default SVGF parameters throughout our experiments, although the optimal values may vary across scenes and pixels. For example, as shown in Fig. 11, increasing the illumination parameter σ_l (see Eq. (25)) reduces noise but also blurs fine details, illustrating the trade-off between denoising and detail preservation. More robust (possibly adaptive) parameter selection remains an interesting direction for future work.

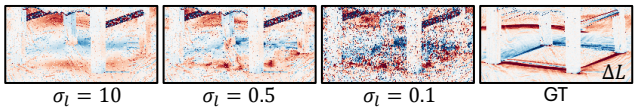

Figure 11: *Effect of the different values of σ_l. Larger values provide stronger denoising but may blur high-frequency details.*

Anti-Aliasing and Jittering. In all previous experiments, we use center sampling within each pixel. Without texture filtering, high-frequency textures can introduce significant temporal instability under center sampling, which is why we employ texture filtering (we use trilinear filtering) throughout our pipeline. In fact, we can further reduce this texture-induced instability by explicitly performing sub-pixel jittering (e.g., Halton sequence) and using the average albedo within each pixel (while still using 2 spp, where the averaging is obtained via temporal accumulation). An example on the SAN MIGUEL scene is shown in Fig. 12 for each method.

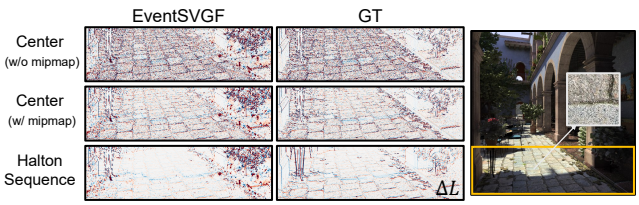

Figure 12: *Comparison of center-point sampling without texture filtering, with texture filtering (mipmap, trilinear), and with additional Halton-sequence jittering with albedo averaging.*

6. Conclusion and Future Work

In this work, we introduced *EventSVGF*, a rendering framework for event camera simulation that directly denoises temporal difference signals using a difference-aware extension of SVGF. By focusing on the high-frequency accuracy of ΔL, our method enables stable and faithful event generation, significantly reducing false events even at extremely low sampling rates. Overall, our results demonstrate that tailoring denoising and sampling strategies to the characteristics of event-based sensing leads to substantial improvements over conventional approaches. We believe this work provides a strong foundation for future research on high-quality event camera simulation and its applications across computer vision, robotics, imaging, and computer graphics.

Future Work. Despite its effectiveness, our method has several limitations that suggest directions for future work.

First, we primarily focus on accurate reconstruction of ΔL, which governs the high-frequency behavior of event cameras. However, the accumulation of small biases in ΔL can lead to long-term drift, potentially affecting low-frequency accuracy in event generation and downstream applications that require temporally consistent low-frequency information. While we demonstrated two examples using complementary filtering to mitigate this issue, more principled approaches, such as adaptive Kalman filtering with variance-aware weighting, remain a promising direction.

Furthermore, there is still substantial room for improvement in the denoising pipeline, as discussed in Sec. 5.6. Better shift-mapping strategies than simple random-seed reuse may exist, including path reconnection techniques or more general hybrid shift mappings [LKB*22]. Developing more effective sampling strategies for large motions, as well as improved texture anti-aliasing and pixel jittering methods, also represent promising future research directions.

While EventSVGF significantly reduces noise at low sample counts, it still relies on heuristic filtering components inherited from SVGF, which may not generalize optimally to all scene types and material configurations (e.g., the table legs in the VEACH-AJAR scene or the glass surfaces in the KITCHEN scene). Extending the proposed difference-aware denoising framework to more advanced pipelines, such as NVIDIA RELAX [NVI26], improved motion estimation techniques [ZLY*21; ZRJ*15], or neural denoising approaches [BWM*23; HMS*20], would be a promising avenue for further research.

We performed all simulations at a fixed frame rate. Extending the framework to support adaptive frame rates, potentially even asynchronous per-pixel updates, would be an interesting direction for more efficient event camera simulation.

Exploring connections to differentiable rendering [LADL18; VSJ21; ZMY*20] is another promising direction for future work. Temporal differences can be viewed as finite-difference approximations of temporal derivatives, suggesting potential links between event-camera rendering and derivative-based rendering techniques.

Finally, this work primarily focuses on the rendering aspect of event camera simulation. An important next step is to evaluate how improved temporal-difference estimation translates to downstream event-camera applications, such as motion detection, tracking, reconstruction, or recognition tasks. In addition, incorporating realistic sensor hardware effects and noise models into the simulation pipeline would further improve the fidelity and practical applicability of the framework.

Acknowledgement

We thank Kehan Xu and Quinton Qu for their detailed feedback on the manuscript. This work was supported in part by the National Science Foundation under Awards No. 2403122 and 2326904.

References

[ALM*24] Alsaad, Z. M., Logan, J. V., Morath, C. P., McMahon-Crabtree, P. N., Kulesza, L., Theis, Z., Maestas, D., Graca, R., McReynolds, B., Zarkesh-Ha, P., et al. "DC characteristics of a dynamic vision sensor's photoreceptor circuit evaluated for event-based sensing in the mid-wave infrared". *Journal of Applied Physics* 136.18 (2024) 3.

[BA17] Bi, Y. and Andreopoulos, Y. "PIX2NVS: Parameterized conversion of pixel-domain video frames to neuromorphic vision streams". *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2017, 1990–1994 2, 3.

[BWM*23] Balint, M., Wolski, K., Myszkowski, K., Seidel, H.-P., and Mantiuk, R. "Neural Partitioning Pyramids for Denoising Monte Carlo Renderings". *ACM SIGGRAPH Conference Papers*. Los Angeles, CA: ACM Press, July 2023. DOI: 10/gsk4j5 10.

[CS14] Censi, A. and Scaramuzza, D. "Low-latency event-based visual odometry". *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2014, 703–710 6.

[CVD*24] Chakravarthi, B., Verma, A. A., Daniilidis, K., Fermuller, C., and Yang, Y. "Recent event camera innovations: A survey". *European Conference on Computer Vision*. Springer. 2024, 342–376 1.

[CVV*25] Chanda, K., Verma, A., Vaghela, A., Yang, Y., and Chakravarthi, B. "Event quality score (eqs): Assessing the realism of simulated event camera streams via distance in latent space". *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, 5105–5113 2.

[FLB22] Furmonas, J., Liobe, J., and Barzdenas, V. "Analytical review of event-based camera depth estimation methods and systems". *Sensors* 22.3 (2022), 1201 2.

[GDO*20] Gallego, G., Delbrück, T., Orchard, G., Bartolozzi, C., Taba, B., Censi, A., Leutenegger, S., Davison, A. J., Conradt, J., Daniilidis, K., et al. "Event-based vision: A survey". *IEEE transactions on pattern analysis and machine intelligence* 44.1 (2020), 154–180 1, 3, 6.

[GGHS20] Gehrig, D., Gehrig, M., Hidalgo-Carrió, J., and Scaramuzza, D. "Video to events: Recycling video datasets for event cameras". *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, 3586–3595 2.

[HLD21] Hu, Y., Liu, S.-C., and Delbruck, T. "v2e: From video frames to realistic DVS events". *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, 1312–1321 2, 3.

[HMS*20] Hasselgren, J., Munkberg, J., Salvi, M., Patney, A., and Lefohn, A. "Neural Temporal Adaptive Sampling and Denoising". *Computer Graphics Forum (Proceedings of Eurographics)* 39.2 (2020), 147–155. ISSN: 1467-8659. DOI: 10/gsmm6b 10.

[HZZT23] Huang, K., Zhang, S., Zhang, J., and Tao, D. "Event-based simultaneous localization and mapping: A comprehensive survey". *arXiv preprint arXiv:2304.09793* (2023) 2.

[JMR*21] Joubert, D., Marcireau, A., Ralph, N., Jolley, A., Van Schaik, A., and Cohen, G. "Event camera simulator improvements via characterized parameters". *Frontiers in Neuroscience* 15 (2021), 702765 3.

[JZZ*20] Jiang, Z., Zhang, Y., Zou, D., Ren, J., Lv, J., and Liu, Y. "Learning event-based motion deblurring". *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, 3320–3329 2.

[KCK*22] Kallweit, S., Clarberg, P., Kolb, C., Davidovič, T., Yao, K.-H., Foley, T., He, Y., Wu, L., Chen, L., Akenine-Möller, T., Wyman, C., Crassin, C., and Benty, N. *The Falcor Rendering Framework*. Aug. 2022. URL: https://github.com/NVIDIAGameWorks/Falcor 6.

[KLD16] Kim, H., Leutenegger, S., and Davison, A. J. "Real-time 3D reconstruction and 6-DoF tracking with an event camera". *European conference on computer vision*. Springer. 2016, 349–364 2.

[KMA*15] Kettunen, M., Manzi, M., Aittala, M., Lehtinen, J., Durand, F., and Zwicker, M. "Gradient-Domain Path Tracing". *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 34.4 (July 27, 2015), 123. ISSN: 0730-0301. DOI: 10/gfzrhn 2, 6.

[KND12] Katz, M. L., Nikolic, K., and Delbruck, T. "Live demonstration: Behavioural emulation of event-based vision sensors". *2012 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE. 2012, 736–740 2.

[LADL18] Li, T.-M., Aittala, M., Durand, F., and Lehtinen, J. "Differentiable Monte Carlo Ray Tracing through Edge Sampling". *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 37.6 (Dec. 2018), 222:1–222:11. ISSN: 0730-0301. DOI: 10/gfztzx 10.

[LBL*25] Li, Z., Bai, X., Lu, J., Shen, P., Lam, E. Y., and Peng, Y. "EventTracer: Fast Path Tracing-based Event Stream Rendering". *arXiv preprint arXiv:2508.18071* (2025) 2.

[LKB*22] Lin, D., Kettunen, M., Bitterli, B., Pantaleoni, J., Yuksel, C., and Wyman, C. "Generalized Resampled Importance Sampling: Foundations of ReSTIR". *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 41.4 (July 22, 2022), 75:1–75:23. ISSN: 0730-0301. DOI: 10/gqjn7b 10.

[LKL*13] Lehtinen, J., Karras, T., Laine, S., Aittala, M., Durand, F., and Aila, T. "Gradient-Domain Metrpolis Light Transport". *ACM Transactions on Graphics* 32.4 (2013), 95:1–95:12. DOI: 10/gbdghd 2.

[LPZ*24] Liu, H., Peng, S., Zhu, L., Chang, Y., Zhou, H., and Yan, L. "Seeing motion at nighttime with an event camera". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, 25648–25658 2.

[LYL*23] Liang, J., Yang, Y., Li, B., Duan, P., Xu, Y., and Shi, B. "Coherent event guided low-light video enhancement". *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, 10615–10625 2.

[MFPA18] Mitrokhin, A., Fermüller, C., Parameshwara, C., and Aloimonos, Y. "Event-based moving object detection and tracking". *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, 1–9 2.

[MGN*22] Mahlknecht, F., Gehrig, D., Nash, J., Rockenbauer, F. M., Morrell, B., Delaune, J., and Scaramuzza, D. "Exploring event camera-based odometry for planetary robots". *IEEE Robotics and Automation Letters* 7.4 (2022), 8651–8658 2.

[MHH*20] Mohamed, S. A., Haghbayan, M.-H., Heikkonen, J., Tenhunen, H., and Plosila, J. "Towards real-time edge detection for event cameras based on lifetime and dynamic slicing". *The International Conference on Artificial Intelligence and Computer Vision*. Springer. 2020, 584–593 2.

[MKA*15] Manzi, M., Kettunen, M., Aittala, M., Lehtinen, J., Durand, F., and Zwicker, M. "Gradient-Domain Bidirectional Path Tracing". *Proceedings of EGSR (Experimental Ideas & Implementations)*. 2015. DOI: 10/gf2hcw 2.

[MKD*16] Manzi, M., Kettunen, M., Durand, F., Zwicker, M., and Lehtinen, J. "Temporal Gradient-Domain Path Tracing". *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 35.6 (Dec. 5, 2016). ISSN: 0730-0301. DOI: 10/f9cpsw 2, 6.

[MYMK24] Manabe, Y., Yatagawa, T., Morishima, S., and Kubo, H. "Monte Carlo Path Tracing and Statistical Event Detection for Event Camera Simulation". *2024 IEEE International Conference on Computational Photography (ICCP)*. IEEE. 2024, 1–10 2, 3.

[NVI26] NVIDIA. *NVIDIA Real-Time Denoisers (NRD)*. https://github.com/NVIDIA-RTX/NRD. Accessed: 2026-06-01. 2026 10.

[RGS18] Rebecq, H., Gehrig, D., and Scaramuzza, D. "ESIM: an open event camera simulator". *Conference on robot learning*. PMLR. 2018, 969–982 2, 3.

[RZL*18] Ramesh, B., Zhang, S., Lee, Z. W., Gao, Z., Orchard, G., and Xiang, C. "Long-term object tracking with a moving event camera." *Bmvc*. 2018, 241 2.

[SBM18] Scheerlinck, C., Barnes, N., and Mahony, R. "Continuous-time intensity estimation using event cameras". *Asian Conference on Computer Vision*. Springer. 2018, 308–324 3, 6.

[SDK*24] Shariff, W., Dilmaghani, M. S., Kielty, P., Moustafa, M., Lemley, J., and Corcoran, P. "Event cameras in automotive sensing: A review". *IEEE Access* 12 (2024), 51275–51306 2.

[SGT*26] Sun, P., Guan, B., Tao, J., Yu, Z., Bai, X., Shang, Y., and Yu, Q. "High dynamic range imaging based on an asymmetric event-SVE camera system". *Optics Express* 34.4 (2026), 7174–7189 3.

[SKW*17] Schied, C., Kaplanyan, A., Wyman, C., Patney, A., Chaitanya, C. R. A., Burgess, J., Liu, S., Dachsbacher, C., Lefohn, A., and Salvi, M. "Spatiotemporal Variance-Guided Filtering: Real-time Reconstruction for Path-Traced Global Illumination". *Proceedings of High Performance Graphics*. New York, NY, USA: ACM, 2017, 2:1–2:12. ISBN: 978-1-4503-5101-0. DOI: 10/ggd8dg 1, 2.

[SPD18] Schied, C., Peters, C., and Dachsbacher, C. "Gradient Estimation for Real-Time Adaptive Temporal Filtering". *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 1.2 (Aug. 2018), 24:1–24:16. ISSN: 2577-6193. DOI: 10/ggd8dh 2–4.

[TYKM23] Tsuji, Y., Yatagawa, T., Kubo, H., and Morishima, S. "Event-Based Camera Simulation Using Monte Carlo Path Tracing with Adaptive Denoising". *2023 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2023, 301–305 2, 3, 8.

[VGB16] Vasco, V., Glover, A., and Bartolozzi, C. "Fast event-based Harris corner detection exploiting the advantages of event-driven cameras". *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE. 2016, 4144–4149 2.

[VSJ21] Vicini, D., Speierer, S., and Jakob, W. "Path Replay Backpropagation: Differentiating Light Paths Using Constant Memory and Linear Time". *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 40.4 (July 19, 2021), 108:1–108:14. ISSN: 0730-0301. DOI: 10/gnngwp 10.

[WNSM21] Wang, Z., Ng, Y., Scheerlinck, C., and Mahony, R. "An asynchronous kalman filter for hybrid event cameras". *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, 448–457 6.

[XYW*21] Xu, F., Yu, L., Wang, B., Yang, W., Xia, G.-S., Jia, X., Qiao, Z., and Liu, J. "Motion deblurring with real events". *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, 2583–2592 2.

[YLS20] Yang, L., Liu, S., and Salvi, M. "A survey of temporal antialiasing techniques". *Computer graphics forum*. Vol. 39. 2. Wiley Online Library. 2020, 607–621 3.

[ZJL*15] Zwicker, M., Jarosz, W., Lehtinen, J., Moon, B., Ramamoorthi, R., Rousselle, F., Sen, P., Soler, C., and Yoon, S.-E. "Recent Advances in Adaptive Sampling and Reconstruction for Monte Carlo Rendering". *Computer Graphics Forum (Proceedings of Eurographics State of the Art Reports)* 34.2 (May 2015), 667–681. ISSN: 1467-8659. DOI: 10/f7k6kj 2.

[ZLY*21] Zeng, Z., Liu, S., Yang, J., Wang, L., and Yan, L.-Q. "Temporally Reliable Motion Vectors for Real-time Ray Tracing". *Computer Graphics Forum (Proceedings of Eurographics)* 40.2 (2021), 79–90. ISSN: 1467-8659. DOI: 10/gsmm6n 10.

[ZMY*20] Zhang, C., Miller, B., Yan, K., Gkioulekas, I., and Zhao, S. "Path-Space Differentiable Rendering". *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 39.4 (July 8, 2020). ISSN: 0730-0301, 1557-7368. DOI: 10/gg8xc2 10.

[ZRJ*15] Zimmer, H., Rousselle, F., Jakob, W., Wang, O., Adler, D., Jarosz, W., Sorkine-Hornung, O., and Sorkine-Hornung, A. "Path-Space Motion Estimation and Decomposition for Robust Animation Filtering". *Computer Graphics Forum (Proceedings of the Eurographics Symposium on Rendering)* 34.4 (July 1, 2015), 131–142. ISSN: 0167-7055. DOI: 10/f7mb34 10.

Appendix A: Details of SVGF Weights

The depth weight suppresses filtering across geometric edges:

$$w_z(p,q) = \exp\left(-\frac{|z(p)-z(q)|}{\sigma_z|\nabla z(p)\cdot(p-q)|+\epsilon}\right), \tag{23}$$

which accounts for both absolute depth differences and local depth gradients. The normal weight enforces consistency in surface normals $\mathbf{n}$:

$$w_n(p,q) = \max(0,\mathbf{n}(p)\cdot\mathbf{n}(q))^{\sigma_n}, \tag{24}$$

penalizing samples with misaligned normals. Finally, the luminance weight adapts filtering based on signal variation:

$$w_l^{(i)}(p,q) = \exp\left(-\frac{|l^{(i)}(p)-l^{(i)}(q)|}{\sigma_l\sqrt{g_{3\times3}(\mathrm{Var}(l^{(i)}(p)))}+\epsilon}\right), \tag{25}$$

where $g_{3\times3}$ denotes a 3×3 Gaussian filter applied to the local variance estimate to modulate sensitivity to noise. Each term is controlled by user-defined parameters σ_z, σ_n, and σ_l. By iteratively applying this filter at increasing kernel scales, SVGF effectively reduces noise while preserving important scene features.

Appendix B: Proof of False Event Probability

We analyze the number of false events caused by temporal noise increments. Let the event threshold be $C>0$, and let Δn_t denote the noise increment at time step t. We assume that the signal phase within one threshold cell is uniformly distributed (either due to initial bias or long accumulation), i.e.,

$$U_t \sim \mathrm{Unif}(0,C), \tag{26}$$

where U_t represents the distance from the current signal value to the next lower threshold. Conditioned on a noise increment $\Delta n_t=\delta$, the false-event count equals the number of threshold boundaries crossed by shifting the phase U_t by δ.

We first consider the case $\delta\geq0$. Starting from phase $U\in[0,C)$, after the increment the new position becomes $U+\delta$, and the number of crossed thresholds is

$$K(U,\delta) = \left\lfloor\frac{U+\delta}{C}\right\rfloor. \tag{27}$$

Averaging over the uniform phase gives

$$\mathbb{E}[K\mid\Delta n_t=\delta] = \frac{1}{C}\int_0^C\left\lfloor\frac{u+\delta}{C}\right\rfloor du. \tag{28}$$

Now decompose δ as $\delta=mC+r$ with residue r; then

$$\left\lfloor\frac{u+\delta}{C}\right\rfloor = m+\left\lfloor\frac{u+r}{C}\right\rfloor. \tag{29}$$

Since $u\in[0,C)$, the second term is equal to 1 only when $u\geq C-r$, which occurs over an interval of length r. Therefore,

$$\mathbb{E}[K\mid\Delta n_t=\delta] = m+\frac{r}{C} = \frac{\delta}{C}. \tag{30}$$

For $\delta<0$, the same argument applies symmetrically, and the expected *absolute* number of crossings is

$$\mathbb{E}[K\mid\Delta n_t=\delta] = \frac{|\delta|}{C}. \tag{31}$$

Taking the expectation over the distribution of Δn_t yields

$$\mathbb{E}[K] = \mathbb{E}[\mathbb{E}[K\mid\Delta n_t]] = \frac{1}{C}\mathbb{E}[|\Delta n_t|]. \tag{32}$$

If the noise profile is Gaussian, $\Delta n_t\sim\mathcal{N}(0,\sigma_\Delta^2)$, the above equation becomes

$$\mathbb{E}[K] = \sqrt{\frac{2}{\pi}}\frac{\sigma_\Delta}{C}. \tag{33}$$

In practice, larger L_2 deviations are generally associated with larger L_1 deviations,

$$\mathbb{E}[K]\propto\frac{\sigma_\Delta}{C}. \tag{34}$$

Appendix C: Temporal Difference with Motion Alignment

Let $w_t(s)=s+\Delta_t s$ denote the motion-compensated pixel correspondence and define the motion-aligned temporal difference as

$$\Delta_M I_t[s] = I_t[s]-I_{t-1}[w_t(s)]. \tag{35}$$

Applying temporal accumulation gives

$$I'_t[s] = \alpha I_t[s]+(1-\alpha)I'_{t-1}[w_t(s)], \tag{36}$$

$$I'_{t-1}[w_t(s)] = \alpha I_{t-1}[w_t(s)]+(1-\alpha)I'_{t-2}[w_{t-1}(w_t(s))]. \tag{37}$$

Subtracting the two equations yields

$$\Delta_M I'_t[s] = \alpha\Delta_M I_t[s]+(1-\alpha)\Delta_M I'_{t-1}[w_t(s)]. \tag{38}$$

Unlike the non-motion-aligned formulation, the recursive update contains no additional correction term. However, the quantity required for event generation is the non-motion-aligned temporal difference:

$$I_t[s]-I_{t-1}[s] = \Delta_M I_t[s]+\underbrace{\left(I_{t-1}[w_t(s)]-I_{t-1}[s]\right)}_{\text{correction}}, \tag{39}$$

where the correction term now appears as a spatial residual in the previous frame.

From an implementation perspective, bilinear fetching of $I_{t-1}[w(s)]$ does not guarantee a matching sample pair (other pixels may use different random seeds) so we need to explicitly splat the primary hit point (GBuffer) and random seed from previous frame into the current frame following the motion vector, which is considerably more complicated. Thus, we have not implemented the motion-aligned formulation, but a comparison between the two approaches is an interesting direction for future work.