DOI: 10.1111/cgf.70321

EUROGRAPHICS 2026 / B. Masia and J. Thies
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 45 (2026), Number 2

Statistical Denoising of Transient Rendering

Oscar Pueyo-Ciutad [1], Alvaro Lopez [1] and Diego Gutierrez [1]

[1] Universidad de Zaragoza - I3A, Spain

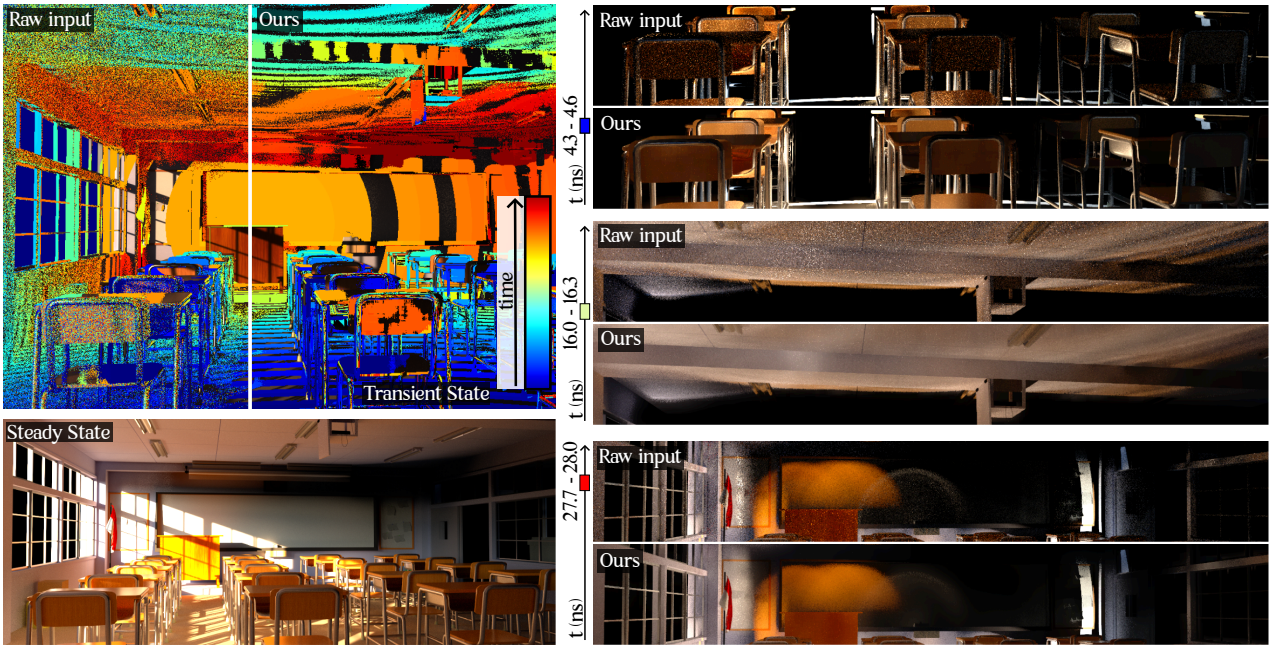

Figure 1: *Transient rendering simulates light in motion through a scene (top left). However, the stochastic nature of Monte Carlo rendering, combined with the added temporal dimension produces extremely noisy results (top left, Raw input). We denoise transient renders by relying on online sample statistics per time bin to analyze the spatio-temporal correlation of transient light transport (top left, Ours). While a naive implementation would lead to impractical memory constraints, we develop a memory-efficient algorithm by leveraging that only subsequent path vertices tend to contribute to the same time bin. Our denoiser does not require any training and achieves a remarkable reduction of noise while preserving fine spatio-temporal details (right).*

Abstract

Transient rendering simulates light in motion, measuring the time of flight from the light source to the camera. However, the stochastic nature of Monte Carlo is aggravated in transient rendering, since samples are now spread along the temporal domain. In our work, we propose to denoise transient Monte Carlo renders by exploiting the spatio-temporal correlation of transient light transport, extending a recent statistical denoising formulation. By relying on statistics, we achieve a near-optimal trade-off between reduced variance and introduced bias. We efficiently collect per-time-bin statistics in the temporal domain while avoiding impractical memory requirements, and use these collected statistics to analyze the spatio-temporal correlation and discriminate which time bins should be combined. Our statistics-based transient denoiser does not hallucinate, guarantees convergence of the result, is efficient, does not require any training and naturally handles participating media. We believe that the generality of our method might pave the way for denoising time-resolved Monte Carlo simulations in other domains, such as non-line-of-sight imaging, acoustic rendering, or absorption microscopy.

CCS Concepts

• ***Computing methodologies*** → ***Image processing;*** *Antialiasing; Image-based rendering; Computational photography;* ***Ray tracing;***

1. Introduction

Many global illumination methods approximate the render equation [Kaj86] through Monte Carlo techniques [VD98]. Although they are unbiased and achieve photorealistic results, the inherent variance of Monte Carlo introduces undesired error in form of noise in the final image. A common assumption of these techniques is that the speed of light is infinite. However, advances in ultra-fast optics and electronics have recently allowed to capture and visualize light propagating through a scene, which has in turn led to what is called *transient* rendering (see Fig. 1, top left) [JMMG17, FVW20]†.

Transient rendering methods also rely commonly on Monte Carlo to approximate the transient path integral [JMM*14]. However, naively adapting steady-state techniques to take into account the temporal domain is impractical: light with different temporal delays is stored in different time bins, which leads to extremely noisy results and makes convergence unfeasible [SSD08, JMV*12, ABW14] (see Fig. 1, Raw input).

Denoising methods trade off variance and bias by applying image filters on the rendered image. Currently, the denoising trend relies on deep learning techniques [CKS*17, Áfr25] trained on large datasets, which learn how to map a noisy input image to a noiseless output. However, this leads to hallucination from dataset bias. Furthermore, directly training these networks on transient renders require significantly more computational resources due to the new temporal dimension. Recently, Sakai et al. [SFA*24] proposed a denoiser based on pixel statistics for steady-state rendering, which works in two phases: first, the authors collect pixel statistics efficiently during rendering, then leverage such statistics to achieve a near-optimal bias-noise trade-off by comparing the statistical distribution between pixels, and discriminating which pixels should be combined.

However, adapting this approach to transient rendering is not a straightforward task. Collecting statistics requires two sets of non-linear transforms: quadratic and cubic transforms to compute the central moments (M_1, M_2, M_3) for statistics, and normalizing sample transforms such as the Box-Cox [BC18] or Yeo-Johnson [YJ00] transforms. In transient rendering, as each sample contributes to the whole temporal domain, naively ensuring that two contributions of a sample to the same time bin are not transformed separately would require a per-sample temporal dimension to guarantee the correct transformations, which in turn leads to impractical memory requirements. Furthermore, instead of only considering spatial correlation between samples as in steady-state approaches, our method needs to consider the correlations that arise in the extended spatio-temporal domain.

To make transient denoising practical, our statistics collection method exploits the fact that only subsequent vertices tend to contribute to a same time bin. We exploit this insight to drastically alleviate memory requirements, only requiring a single scalar per sample instead of a whole transient domain. Then, we leverage these statistics to exploit the spatio-temporal correlation of transient light transport in the denoising, by combining the pair-wise statistical tests with spatio-temporal filters (see Fig. 1, Ours). Last, we show how our formulation naturally generalizes to the particularly difficult problem of transient *volumetric* rendering, significantly improving results when rendering time-resolved light transport in participating media.

Our transient denoising framework does not hallucinate, guarantees convergence of the denoised result with increasing number of samples, is efficient, does not require any training, and naturally handles participating media. Moreover, since Monte Carlo simulations are a well-established technique and many domain-specific sampling strategies have been been derived for transient rendering, we hope that our work can be applied in many other fields such as optical heterodyne detection [KBW*25], non-line-of-sight imaging [RGMJ22], absorption microscopy [KMSP24], acoustic rendering [FWJ*25] or thermal imaging [TIT*18], to name a few. The source code for this project is available on our GitHub repository‡.

2. Related work

2.1. Transient rendering

Time-of-flight devices leverage ultra-fast sensors to capture the propagation of light through time [VWJ*13, HHGH13, SXV*16], which triggered a new set of rendering techniques for transient light transport. Existing works developed a general framework for transient light transport, for both forward (e.g., [JMM*14, PAJ19, PVG19]) and differentiable [YKC*21] applications. However, sampling methods from conventional, steady-state rendering can not be directly applied into transient rendering without a detrimental performance in convergence.

Jarabo et al. [JMM*14] introduced the transient path integral, proposing temporal density estimation and importance-sample techniques, suited for participating media. Later, the photon beams approach [JNT*11, KGH*14] was extended into the transient domain [MGJ*19], again achieving great results for homogeneous media, but introducing bias in the form of beam discretization error. Recently, distance and direction sampling with ellipsoidal connections were derived from transient diffusion theory [HDJJ24], but again the method is suited for homogeneous participating media.

Pan et al. [PAJ19] generalized instant radiosity for time-resolved global illumination between diffuse surfaces. Pediredla et al. [PVG19] proposed ellipsoidal path connections to impose restrictions on path length, which provides great results for narrow time-gated rendering, but it is not as efficient for full transient rendering. Recently, Royo et al. [RCGP23] published an efficient time-resolved simulator over the Dr. JiT [JSRV22] and Mitsuba3 [JSR*22] rendering frameworks.

2.2. Denoising

Classical denoising approaches span a wide variety of formulations, using image-filtering kernels such as Gaussian, bilateral filter

† Funded by the European Union. Views and opinions expressed are those of the author(s) and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the granting authority can be held responsible for them.

‡ https://github.com/alv02/StatisticalDenoisingTransientRendering

or non-local means [RKZ12, VAN*19], risk estimators [LWC12], Bayes [BB17], wavelet [ODR09, DSHL10, SKW*17], or weighted recursive least squares for volumetric denoising [IGMM22]. Many of these methods rely on G-Buffers (a set of per-pixel, low-level features such as albedo and normals) and variance. Meta-denoisers combine multiple denoising techniques [ZZXY21], and post-denoisers use a raw input and an input denoiser [BHHM20, FFJ22, BHHM22, GIGM22, WQH*24]. Other approaches rely on histogram distributions [BB17, SD12], but suffer from resolution issues. This would become even more constraining when applying them to transient rendering, due to the extra temporal dimension.

Learning approaches such as NVIDIA OptiX AI-Accelerated Denoiser [CKS*17] or Intel Open Image Denoise (OIDN) [Áfr25] take as input the noisy image and G-Buffer information to predict the denoised output. To train these models, large datasets of noisy inputs and noiseless references need to be generated, requiring huge computational resources (a problem aggravated in the extended spatio-temporal domain). In real-time approaches, neural methods learn the coherence between adjacent frames [BWM*23, CHH*24, XXZ*25]. Furthermore, these approaches are more focused on producing visually-pleasing results, but usually introduce bias and lead to hallucinations in the denoised output.

Denoising transient renders is a relatively unexplored field. Existing research has focused on denoising Poisson noise from captures [LCH*20, RHL*22]; in turn, we denoise Monte Carlo noise from transient renderings. Furthermore, existing methods rely on lossy signal compression (using neural encoder-decoders or mixtures of exponentially modified Gaussians), which inevitably impacts the quality of details.

Our work leverages the recent statistical filtering approach of Sakai et al. [SFA*24], which collects online, per-pixel statistics (central moments) during rendering which are then compared and analyzed to minimize noise. However, naively adapting this approach to transient rendering becomes impractical due to its excessive memory requirements. We show how to extend the statistics collection into the time domain and leverage spatio-temporal correlations, while keeping our method memory efficient. Our transient denoiser is independent of the sampling strategy used, it handles both surface-based and volumetric light transport, and does not rely on computationally-expensive learning techniques.

3. Method

We first recall the transient path integral [JMM*14], and then extend a recent statistical formulation [SFA*24] to denoise transient renders. Next, we present an efficient approach to collect transient sample statistics. Finally, we combine both into our statistics-based spatio-temporal denoising.

3.1. Transient path integral

In the path integral [VD98], the image pixel intensity I is computed as an integral over the space of light transport paths Ω. In the *transient* path integral [JMM*14], we must also integrate over the space of scattering delays ΔT (e.g. phosphorescence) of all light transport

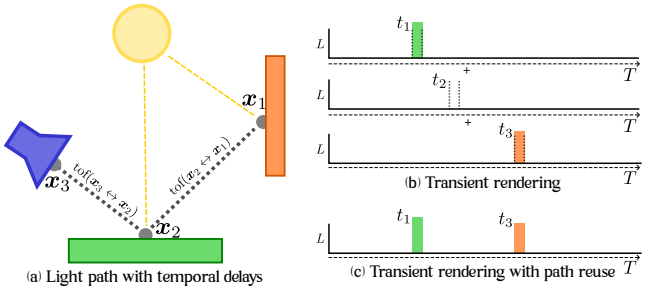

Figure 2: ***Transient rendering and path reuse.*** *(a) Transient rendering takes into account the temporal delays of light between path vertices $\boldsymbol{x}_i$. (b) Regular transient rendering repeats the rendering T times (number of time bins) gating the sensor for each time bin t_i, discarding light paths whose temporal delay $\bar{t}$ is not contained within the gated bin, implying re-rendering T times. (c) Transient rendering with path reuse stores the contribution L of each light path to the time bin t_i corresponding to its path delay $\bar{t}$, rendering once for the whole transient domain.*

paths Ω (see Fig. 2a), as

$$I = \int_{\Omega}\int_{\Delta T} f(\bar{\boldsymbol{x}}, \overline{\Delta t})\, d\mu(\overline{\Delta t})\, d\mu(\bar{\boldsymbol{x}}), \tag{1}$$

where $\bar{\boldsymbol{x}} = \boldsymbol{x}_0 \dots \boldsymbol{x}_k$ are the spatial coordinates of the $k+1$ vertices of a path ($\boldsymbol{x}_0$ is on the light source, $\boldsymbol{x}_k$ is on the camera sensor, and $\boldsymbol{x}_1 \dots \boldsymbol{x}_{k-1}$ are intermediate scattering vertices). The differential measure $d\mu(\bar{\boldsymbol{x}})$ denotes integration for surface area and volume media vertices, $\overline{\Delta t} = \Delta t_0 \dots \Delta t_k$ defines the sequence of scattering delays, and $d\mu(\overline{\Delta t})$ is the temporal integration at each path vertex.

The transient path contribution $f(\bar{\boldsymbol{x}}, \overline{\Delta t})$ is similar to the steady-state path contribution, but considers time-varying emission L_e, path throughput $\mathfrak{T}$, and sensor importance W_e as:

$$f(\bar{\boldsymbol{x}}, \overline{\Delta t}) = L_e(\boldsymbol{x}_0 \to \boldsymbol{x}_1, \Delta t_0)\, \mathfrak{T}(\bar{\boldsymbol{x}}, \overline{\Delta t})\, W_e(\boldsymbol{x}_{k-1} \to \boldsymbol{x}_k, \Delta t_k), \tag{2}$$

where W_e defines both spatio-angular sensitivity and the region of time being evaluated (which is usually a finite interval of interest in the temporal domain).

The time of arrival of light to the sensor can be computed from the propagation delays and scattering delays Δt as

$$\bar{t} = \sum_{i=0}^{k-1} (\mathrm{tof}(\boldsymbol{x}_i \leftrightarrow \boldsymbol{x}_{i+1}) + \Delta t_i), \tag{3}$$

where $\mathrm{tof}(\boldsymbol{x}_i \leftrightarrow \boldsymbol{x}_{i+1}) = \|\boldsymbol{x}_i - \boldsymbol{x}_{i+1}\|\, \eta / c$, with η being the index of refraction of the medium and c the speed of light in vacuum.

The transient path integral can be numerically approximated using a Monte Carlo estimator

$$\langle I \rangle = \frac{1}{n} \sum_{j=1}^{n} \frac{f(\overline{\boldsymbol{x}_j}, \overline{\Delta t_j})}{p(\overline{\boldsymbol{x}_j}, \overline{\Delta t_j})}, \tag{4}$$

which averages n random paths $(\bar{\boldsymbol{x}}, \overline{\Delta t})$ from a spatio-temporal probability distribution p. As a result of the time-gating, transient rendering Monte Carlo estimators produce extremely noisy results when compared to steady-state.

3.2. Transient statistical formulation

We propose to denoise transient renders using general spatio-temporal filters by extending the recent statistical denoising framework of Sakai et al. [SFA*24], to explicitly and efficiently take into account the temporal domain. To achieve a near-optimal bias-noise trade-off, we seek to obtain a denoised estimator $\tilde{\theta}_j^t$ (where j represents the pixel and t the time bin) as a convex combination of spatio-temporally nearby noisy estimators $\hat{\theta}_i^\tau$ (where i is the pixel and τ the time bin) as

$$\tilde{\theta}_j^t = \sum_\tau \sum_i w_{ij}^{\tau t} \hat{\theta}_i^\tau, \tag{5}$$

where $w_{ij}^{\tau t}$ are the weights assigned to the estimator $\hat{\theta}_i^\tau$. The goal of transient denoising is to achieve an optimal trade-off between variance and bias. To accomplish this, we seek to obtain the best weight $w_{ij}^{\tau t}$ that minimizes the mean squared error (MSE) as

$$\begin{aligned} \{w_{ij}^{\tau t,*}\} &= \arg\min_{\{w_{ij}^{\tau t}\}} \sum_t \sum_j \mathrm{MSE}(\tilde{\theta}_j^t, \theta_j^t) \\ \mathrm{MSE}(\tilde{\theta}_j^t, \theta_j^t) &= \mathbb{E}[(\tilde{\theta}_j^t - \theta_j^t)^2] = \mathrm{Var}(\tilde{\theta}_j^t) + \mathrm{Bias}(\tilde{\theta}_j^t, \theta_j^t)^2. \end{aligned} \tag{6}$$

However, as ground-truths θ_j^t are unknown, we can only use the noisy estimators $\hat{\theta}_i^\tau$ to achieve the denoised estimators $\tilde{\theta}_j^t$.

Our statistical denoiser must also have a number of key properties, including symmetry, convergence with number of samples, and identity with itself [SFA*24]. Moreover, whether a pair of estimators $(\hat{\theta}_i^\tau, \hat{\theta}_j^t)$ should be combined or not should depend only on the statistics of such pair of estimators. Last, statistics should be computed online to prevent prohibitive memory or performance requirements.

Classic filters such as a 3D Gaussian filter or a Joint Bilateral Filter do not analyze the input estimators and use pre-defined weights independent of the samples. Instead, we adapt steady-state statistical estimators to the temporal domain to denoise time-gated images. Our transient estimators are obtained from the mean, central moments M_l^t (where l is the moment order), and skewness $(M_3/M_2^{3/2})$. The central moments [KK51] can be computed as

$$M_l^t = \frac{1}{n} \sum_j^n \left(X_j^t - \overline{X^t}\right)^l, \tag{7}$$

where X_j^t represents each sample j of a time bin t, and $\overline{X^t}$ is the sample mean of the time bin t.

Unfortunately, collecting these estimators $\hat{\theta}_j^t$ naively is impractical due to high memory constraints. In the following section, we show how to efficiently collect them for transient rendering.

3.3. Efficient collection of transient statistics

Jarabo et al. [JMM*14] realized that, in order to efficiently render light transport in the temporal domain, it is necessary to *reuse* paths. Instead of rendering the time gate of each time bin independently, which leads to repeating the rendering process T times (one per time bin, see Fig. 2b), path reuse *simultaneously* estimates all the temporal dimension at once by storing the contribution of a light path into the corresponding time bin of the path temporal delay. As a result, each light path $\bar{\boldsymbol{x}}$ contributes to the full temporal domain T (Fig. 2c).

Algorithm 1 Statistics collection. In the **sample** function of a Monte Carlo integrator, we use a single scalar value L per sample, and accumulate on it the path contribution until the temporal delay exceeds the current time bin width h. Then, we dump into the final transient values through the **update_stats** function, which synchronizes parallel dumps through the **sync_accum** function.

```
1:  def sample (...):
2:      L = 0, 𝔗 = 1
3:      t_last_bin = −1 , t̄ = 0
4:      while path is active :
5:          surface_it = ray_intersect(...)
            # Propagation delay between path vertices
6:          t̄ += surface_it.tof
            ...
            # Next Event Estimation
7:          dir_sample = emitter.sample(surface_it)
8:          t_bin = ⌊(t̄ + dir_sample.tof − t_start)/h⌋
9:          if t_bin ≠ t_last_bin :
10:             update_stats(L, t_last_bin, ...)
11:             L = 0,  t_last_bin = t_bin
12:         L += 𝔗 · dir_sample.L
            ...
13:     update_stats(L, t_last_bin, ...)

14: def update_stats (L, t_bin, xy):
15:     v = transform(L)
16:     non_zero += sync_accum(1, t_bin, xy)
17:     (X1 X2 X3) += sync_accum((v v² v³), t_bin, xy)
```

Algorithm 2 Estimator $\hat{\theta}$ computation. We compute the final estimator $\hat{\theta}$ and its variance $Var(\hat{\theta})$ from the central moments (μ, σ^2, M_3) of the statistics X_1, X_2, X_3, the samples-per-pixel spp and count of non-zero samples.

```
1:  def compute_estimators (X1, X2, X3, spp, non_zero):
2:      z = transform(0)
        # Filling with transformed zeros
3:      (X1, X2, X3) += (spp − non_zero) (z, z², z³)
4:      μ = X1/n
5:      σ² = n/(n−1) (X2/n − μ²)
6:      M3 = X3/n − 3μσ² − μ³
        # Correct means to account for skewness [Cur23]
7:      θ̂ = μ + M3/(6σ² n)
8:      Var(θ̂) = σ²/n
```

However, it is not possible to just update the statistics for each light contribution at each path vertex $\boldsymbol{x}_k$, since there are two sets of non-linear transforms involved: quadratic and cubic transforms for central moments, and normalizing sample transforms such as the Box-Cox [BC18] or Yeo-Johnson [YJ00] transforms. Furthermore, close path vertices $(\boldsymbol{x}_{k-1}, \boldsymbol{x}_k, \boldsymbol{x}_{k+1})$ usually contribute to the

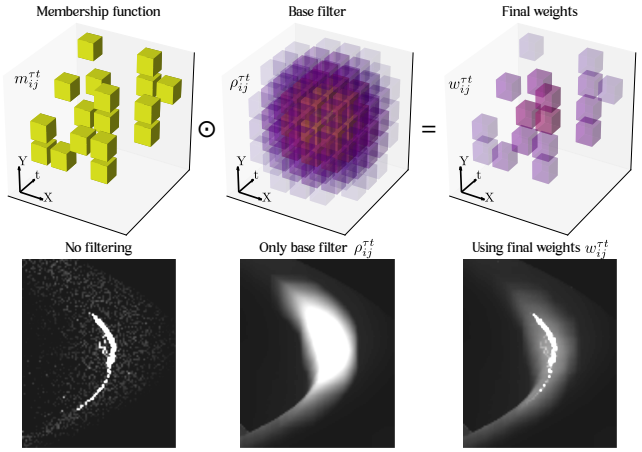

Figure 3: ***Spatio-temporal denoising.*** *We leverage the spatio-temporal correlation of transient light transport for denoising. Our denoiser is composed of a membership function $m_{ij}^{\tau t}$ deciding which transient pixels should be combined together, and a base filter $\rho_{ij}^{\tau t}$, which combined leads to the final weights $w_{ij}^{\tau t}$. While the raw render is noisy and the base filter adds a significant amount of bias, our weights achieve a near-optimal trade-off between bias and noise.*

same time bin. As a result, naively collecting samples while ensuring correctness would require tracking an independent temporal domain for each sample, which would translate into an impractical memory usage: for a scene with height (H), width (W), transient (T), and samples per pixel (spp) of 1024, and using 32-bit floats, it would add up to 4 TB just for light contributions.

To overcome this, we instead accumulate the light contribution of a light path $\bar{\boldsymbol{x}}$ into a single value L as long as it contributes to the same time bin (see Algorithm 1). Once the temporal delay $\bar{t}$ of the path at vertex $\boldsymbol{x}_k$ is greater than the width of the current time bin, we dump the path contribution into the final transient value and update its statistics. Therefore, we only track a single value per sampled light path $\bar{\mathbf{x}}$, reducing by a factor of T the memory consumption while leveraging the benefits of path reuse. In the previous example, the memory usage is reduced by a factor of 1024, yielding a significantly more practical memory usage of 4 GB.

Furthermore, online updates of central moments [Men15] can not be computed from a variant of the batched Welford algorithm [Wel62]. This is due to the concurrent evaluation of many different samples in a tensorized framework [JSRV22], since the number of paths concurrently updating the statistics of a certain time bin is unknown. This requires a temporal dimension to obtain the batched statistics before dumping to the final transient values, leading to the aforementioned impractical memory requirements. Therefore, we rely on the expanded definition of central moments (i.e. $M_2 = \sigma^2 = E[X_2] - E[X_1]^2$) and only carry X_1, X_2, and X_3, from which we compute the noisy estimator $\hat{\theta}_j^t$ as shown in Algorithm 2.

3.4. Transient denoising

Given our statistical formulation for transient rendering (Section 3.2) and our efficient algorithm for collecting those statistics (Section 3.3), we now combine them for transient denoising.

In transient rendering, since correlations arise in the spatio-temporal domain, we aim to obtain the weights $w_{ij}^{\tau t}$ that achieve the bias-variance trade-off in such higher-dimensional space. Taking into account the temporal dimension, we obtain the optimal weights w^* that minimize the MSE from Eq. (6) by assuming unbiased estimators, setting the derivative with respect to the weight to zero [SFA*24], and solving for w^* as

$$w^* = \frac{2(\theta_i^\tau - \theta_j^t)^2 + \mathrm{Var}(\hat{\theta}_i^\tau) + \mathrm{Var}(\hat{\theta}_j^t)}{2\left((\theta_i^\tau - \theta_j^t)^2 + \mathrm{Var}(\hat{\theta}_i^\tau) + \mathrm{Var}(\hat{\theta}_j^t)\right)}. \tag{8}$$

However, noisy estimators yield noisy weights, and using them directly introduces additional bias. This bias can be limited by enforcing binary membership functions

$$m_{ij}^{\tau t} = 1 \text{ if } (1 - w^*) > \gamma,\ 0 \text{ otherwise}, \tag{9}$$

where the threshold γ is the discriminativity of $m_{ij}^{\tau t}$, similar to a Welch's t-test, $t < \gamma_w$ and $\gamma_w = \sqrt{1/(2\gamma) - 1}$. Membership functions $m_{ij}^{\tau t}$ adaptively discriminate which estimators $\hat{\theta}_j^t$ should be combined together. Two estimators $(\hat{\theta}_j^t, \hat{\theta}_i^\tau)$ are combined only when the difference between their means (introduced bias) is small relative to the sum of their variances (present noise).

To further enhance image quality, we can combine the membership functions $m_{ij}^{\tau t}$ with existing spatio-temporal filters $\rho_{ij}^{\tau t}$, such as a Joint Bilateral Filter (JBF) –which uses low-level features from G-Buffers such as geometric normal and texture albedo to filter neighbors– with an added temporal radius. We compute the weights $w_{ij}^{\tau t}$ in Eq. (5) by combining the binary membership functions $m_{ij}^{\tau t}$ with such spatio-temporal filter $\rho_{ij}^{\tau t}$ as

$$w_{ij}^{\tau t} = \frac{\rho_{ij}^{\tau t} m_{ij}^{\tau t}}{\sum_\tau \sum_i \rho_{ij}^{\tau t} m_{ij}^{\tau t}}, \tag{10}$$

as illustrated in Fig. 3.

Participating media. Our statistics-based transient denoiser naturally supports denoising of transient volumetric renders by replacing the base filter $\rho_{ij}^{\tau t}$ from a modified JBF to a 3D Gaussian filter. This is motivated by the fact that the JBF would wrongly discriminate neighbors based on the low-level features from G-Buffers, which are tailored to surface features. Therefore, when denoising participating media we rely solely on the membership functions $m_{ij}^{\tau t}$ to discriminate which estimands should be combined together.

4. Results

In this section we show the performance of our statistical transient denoiser on a variety of scenes. Our algorithm for rendering and statistics collection with path reuse is implemented in mitransient [RCGP23], a library for transient rendering developed over the Dr. JiT [JSRV22] and Mitsuba3 [JSR*22] frameworks. Our statistical transient denoiser is implemented in PyTorch [PGM*19].

We compare our results against two other approaches: the Joint

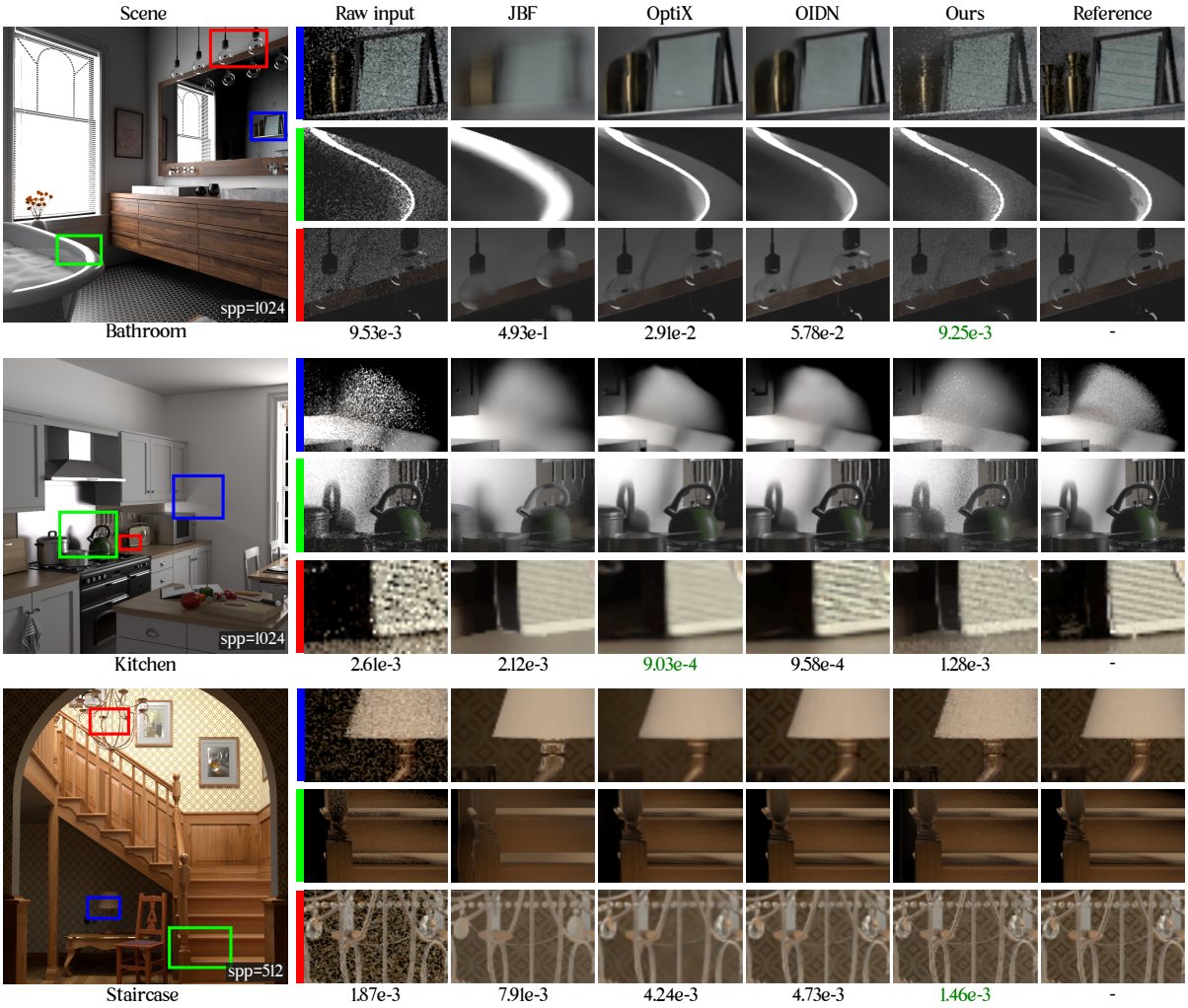

Figure 4: ***Results comparison.*** *We compare our statistics-based spatio-temporal transient denoiser against a Joint Bilateral Filter (JBF) (the base filter used in our method), OptiX (NVIDIA OptiX AI-Accelerated Denoiser) and OIDN (Intel's Open Image Denoise). We use the RMSE to compare each method with the reference, and each inset shows a different time bin. JBF blurs out most of the image details, to the point of eliminating many scene features. Learning-based methods aim to obtain more visually pleasant results at the cost of introducing bias (see e.g. the mirror reflection in* BATHROOM *or the shadow in* KITCHEN*), and hallucinating details (such as the caustic discontinuity). The bath inset is rendered with 128 spp. Please refer to the supplemental material for videos of transient light transport of these scenes.*

Table 1: *Perceptual metrics (FLIP, SSIM) and reference (RMSE) of the compared methods. Green shows best, and yellow second best. First two decimals of SSIM (always 99) are removed for clarity.*

	BATHROOM			KITCHEN			STAIRCASE		
Method	FLIP ↓	SSIM ↑	RMSE ↓	FLIP ↓	SSIM ↑	RMSE ↓	FLIP ↓	SSIM ↑	RMSE ↓
Raw input	0.4290	0.9956	0.0095	0.4673	0.6127	0.2613	0.2867	0.9971	0.1867
JBF	0.2631	0.3710	0.4935	0.3341	0.7068	0.2120	0.2289	0.9486	0.7907
OptiX	0.2497	0.9960	0.0291	0.3126	0.9474	0.0903	0.1542	0.9957	0.4242
OIDN	0.2212	0.9967	0.0578	0.2620	0.9390	0.0958	0.1256	0.9964	0.4737
Ours	0.2474	0.9969	0.0093	0.3162	0.9096	0.1282	0.1832	0.9984	0.1462

Bilateral Filter (JBF) used as base filter $\rho_{ij}^{t\tau}$ in our method (to showcase the importance of memberships), OptiX (NVIDIA OptiX AI-Accelerated Denoiser) [CKS*17] and OIDN (Intel's Open Image Denoise) [Áfr25] as state-of-the-art deep learning denoisers. We do not use temporal denoising in OptiX by treating transient data as a video, since we observed it worsened its performance.

Fig. 4 shows the results against a reference computed with 30k spp. JBF over-smooths most of the finer image details, since it combines transient pixels according to their normals and albedos, ignoring the actual lighting. In contrast, our method discriminates which transient pixels should be combined together based on transient statistics, which incorporate lighting information and thus preserves finer details. Learning-based methods (OptiX and OIDN) may fail to reconstruct some of the finer details while hallucinating others, such as the reflections in the BATHROOM scene (top

row) or the shadow in KITCHEN. Table 1 shows quantitative results of RMSE and perceptual metrics (SSIM and FLIP). Our method yields good results using only the transient statistics collected during rendering, without requiring training on a dataset.

4.1. Parameter analysis

We analyze next the components and properties of our statistical transient denoiser. In particular, we analyze the influence of sample transforms and statistical parameters (Fig. 5), the influence of the spatio-temporal radius (Fig. 6), and the convergence with respect to the samples per pixel (Fig. 7).

Sample transforms. We analyze in Fig. 5 the performance of the Identity, Box-Cox [BC18], and Yeo-Johnson [YJ00] sample transforms, varying the transform parameter $\lambda \in [0.25, 1.50]$. We also test the influence of the threshold $\gamma = \{0.01, 0.05, 0.1\}$. We fix the spatial radius to 5 pixels and the temporal radius to 1 time bin.

The distribution of samples depends heavily on the characteristics of the scene. We found that, for transient rendering, the Identity and the Yeo-Johnson transform with γ between 0.05 and 0.1 consistently offers the best performance across scenes. This is different from steady-state statistical denoising [SFA*24], where the Box-Cox transform seemed to be the best option. We believe this difference arises due to most of the light contributions reaching the sensor in transient rendering are near zero, and the Box-Cox transform applies directly a power transform over x as

$$\text{Box-Cox}\ (x,\lambda) = \frac{x^\lambda - 1}{\lambda}, \quad \lambda \neq 0,\ x > 0, \tag{11}$$

which yields a disproportionate transformation near zero values. In contrast, the Identity transform does not transform the sample, while the Yeo-Johnson transform prevents said distortion issue by shifting x before applying the power transform:

$$\text{Yeo-Johnson}\ (x,\lambda) = \frac{(x+1)^\lambda - 1}{\lambda}, \quad \lambda \neq 0,\ x \geq 0. \tag{12}$$

Spatio-temporal denoising. We analyze in Fig. 6 the influence of spatio-temporal denoising on the RMSE for different time bin width h. We observe that exploiting the spatio-temporal correlation of transient light transport is crucial for narrow h. However, as h increases, temporal correlation progressively loses relevance with respect to spatial correlation, since wider h progressively approximate a steady-state render. For this test, we used 512 spp and the Identity transform with $\gamma = 0.05$.

Convergence. We analyze in Fig. 7 the convergence of our method in the KITCHEN, BATHROOM and STAIRCASE scenes, and compare it against learning-based denoisers (OptiX and OIDN). Our method consistently improves the RMSE, while the performance of learning-based methods depend heavily on the scene, even performing significantly *worse* than the raw, noisy input (see BATHROOM and STAIRCASE). This is probably due to the amount of information learning-based methods need to hallucinate in areas of very fine detail, as we showed in Fig. 4.

4.2. Volumetric results

Our statistical formulation for transient render denoising naturally generalizes to transient volumetric rendering with participating media. We show in Fig. 8 the performance of our method in the BANNER and DISNEY CLOUD scenes, and compare our results against a base 3D Gaussian filter. It can be seen how, while the 3D Gaussian filter blurs the images excessively, our method better preserves fine structural details.

5. Discussion

Transient light transport is prone to noisy results, due to the low number of valid samples (photons) in each time bin. Our work extends the range of statistics-based techniques for Monte Carlo denoising to the transient domain, offering an efficient solution that minimizes noise while preserving fine structural details, does not hallucinate and does not require training, while guaranteeing convergence with increasing number of samples.

We foresee different directions to improve the performance of our method. Relying on statistical tests from noisy estimators offers an efficient solution that does not depend on trained data, although noise may affect the performance of the statistical tests. A possible solution could be using a hybrid approach; although end-to-end learning pipelines are computationally prohibitive due to the excessive amount of samples needed for dataset generation, some parts of the denoising pipeline could leverage learned data. For example, instead of relying on classical spatio-temporal filters such as the Joint Bilateral Filter, a neural network may be able to extract some initial structure, or leverage the statistical information to obtain optimal kernel weights. Furthermore, future work could analyze the potential interplay between method parameters and perceptual error metrics, or combine our formulation with concurrent and future works on statistical error reduction such as Sakai et al. [SFWH25], by leveraging their multi-transform denoising framework and variance-aware adaptive sampling strategies.

Focusing only on denoising while ignoring the sampling component of rendering is not optimal, as a naive sampling leads to some time bins with extremely few samples. We believe that our formulation could strongly benefit from sampling techniques that distribute samples evenly in the transient domain. Our denoiser would then not be hindered by these small set of extremely noisy estimators.

Collecting statistics in transient rendering leads to an execution time overhead of 0.03 ms per pixel with 1024 spp and 100 time bins in transient volumetric rendering, tested on a AMD Ryzen 7 9800*X3D* 4.7/5.2GHz CPU and a MSI GeForce RTX 5070 Ti GPU. Based on the similarities with the steady-state denoiser, our statistics transient denoiser could be implemented with an execution time of just 1.125 s for a resolution of $H \times W \times T = 360 \times 960 \times 100$, while the noisy transient render of the DISNEY CLOUD scene with 60k spp takes around 15 minutes. This showcases the importance of a denosing component in the transient render pipeline.

Our work may help pave the way to denoise time-gated simulations in different domains, where accuracy is preferred over visually-pleasant but possibly biased results, such as acoustic

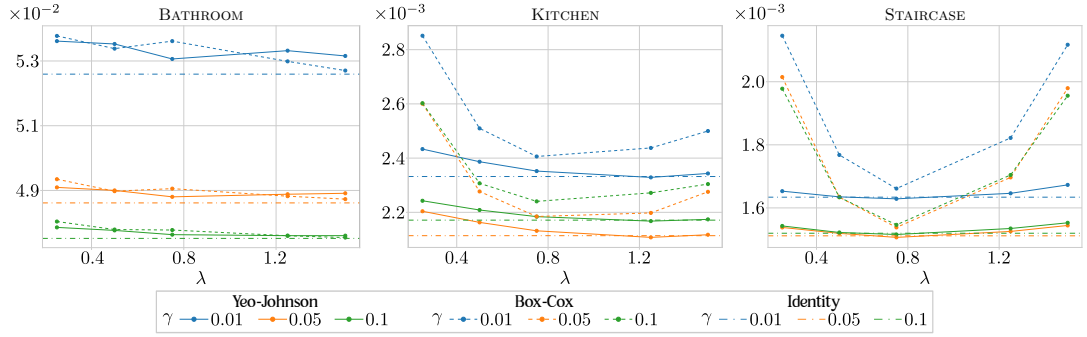

Figure 5: ***Error varying statistics parameters.*** *We analyze for the* Bathroom, Kitchen, *and* Staircase *scenes the influence of varying the parameters and sample transforms used in the Root-Mean-Square-Error (RMSE): We vary the sample transform function between the Identity, Box-Cox [BC18], and Yeo-Johnson [YJ00] transforms with different influences* λ *(x-axis), and membership threshold* γ *(showing the significance level). In transient rendering, since the distribution of samples is different than in steady-state rendering, the best sample transform function is no longer the Box-Cox but either the Identity or the Yeo-Johnson transform with* γ *between* 0.05 *and* 0.1.

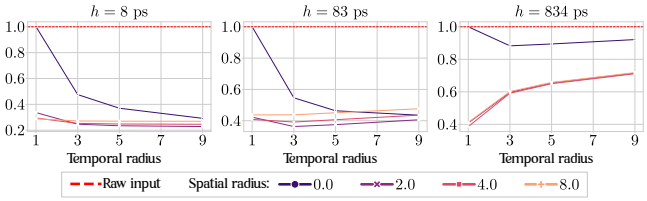

Figure 6: ***Analysis of temporal radius and bin width*** *h. We show the evolution of the RMSE varying the temporal radius of the kernel for different spatial radius. Leveraging the spatio-temporal correlation is key to reduce error for narrow time bins (*$h = 8$ ps*). As the bin width increases (*$h = 83$ ps*), spatio-temporal correlation loses relevance in favor of spatial-only correlation. With wide time bins (*$h = 834$ ps*), spatial correlation is only needed since transient renders present a similar spatial correlation to steady-state renders, and very weak temporal correlation. Tested in the* Kitchen *scene.*

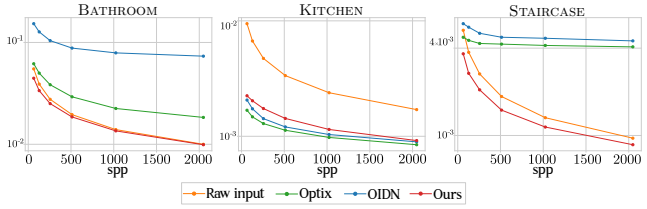

Figure 7: ***Convergence with samples-per-pixel.*** *We test the convergence of our method against learning-based denoisers, varying the number of samples per pixel in the three scenes. Our method converges with increased number of samples. While learning-based methods are not reliable to improve the result, our method consistently improves the RMSE. This is due to the bias introduced by learning-based methods, while ours only uses the statistics present from rendering to achieve a bias-noise trade-off.*

simulations [FWJ*25], thermal imaging [TIT*18] or transient spectroscopy [KMSP24], or domains that share similarities with time-gating, such as optical heterodyne detection simulations [KBW*25].

Acknowledgments

This work has received funding from the European Union's EUROPEAN DEFENSE FUND under grant agreement No 101103242. The views and opinions expressed herein are those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission; neither the European Union nor the granting authority can be held responsible for them. Additionally, Oscar Pueyo-Ciutad was supported by the FPU22/02432 predoctoral grant. Thanks to Guillermo Enguita, Maria Pena, Alfonso Lopez, Diego Royo and Jorge Garcia-Pueyo from the Graphics and Imaging Lab for their help in preparing scenes and figures, and proofreading the manuscript.

References

[ABW14] Ament M., Bergmann C., Weiskopf D.: Refractive radiative transfer equation. *ACM Trans. Graph. 33*, 2 (Apr. 2014).

[Áfr25] Áfra A. T.: Intel® Open Image Denoise, 2025.

[BB17] Boughida M., Boubekeur T.: Bayesian collaborative denoising for monte carlo rendering. *Computer Graphics Forum 36*, 4 (2017), 137–153.

[BC18] Box G. E. P., Cox D. R.: An analysis of transformations. *Journal of the Royal Statistical Society: Series B (Methodological) 26*, 2 (12 2018), 211–243.

[BHHM20] Back J., Hua B.-S., Hachisuka T., Moon B.: Deep combiner for independent and correlated pixel estimates. *ACM Trans. Graph. 39*, 6 (Nov. 2020). URL: https://doi.org/10.1145/3414685.3417847, doi:10.1145/3414685.3417847.

[BHHM22] Back J., Hua B.-S., Hachisuka T., Moon B.: Self-supervised post-correction for monte carlo denoising. In *ACM SIGGRAPH 2022 Conference Proceedings* (New York, NY, USA, 2022), SIGGRAPH '22, Association for Computing Machinery. URL: https://doi.org/10.1145/3528233.3530730, doi:10.1145/3528233.3530730.

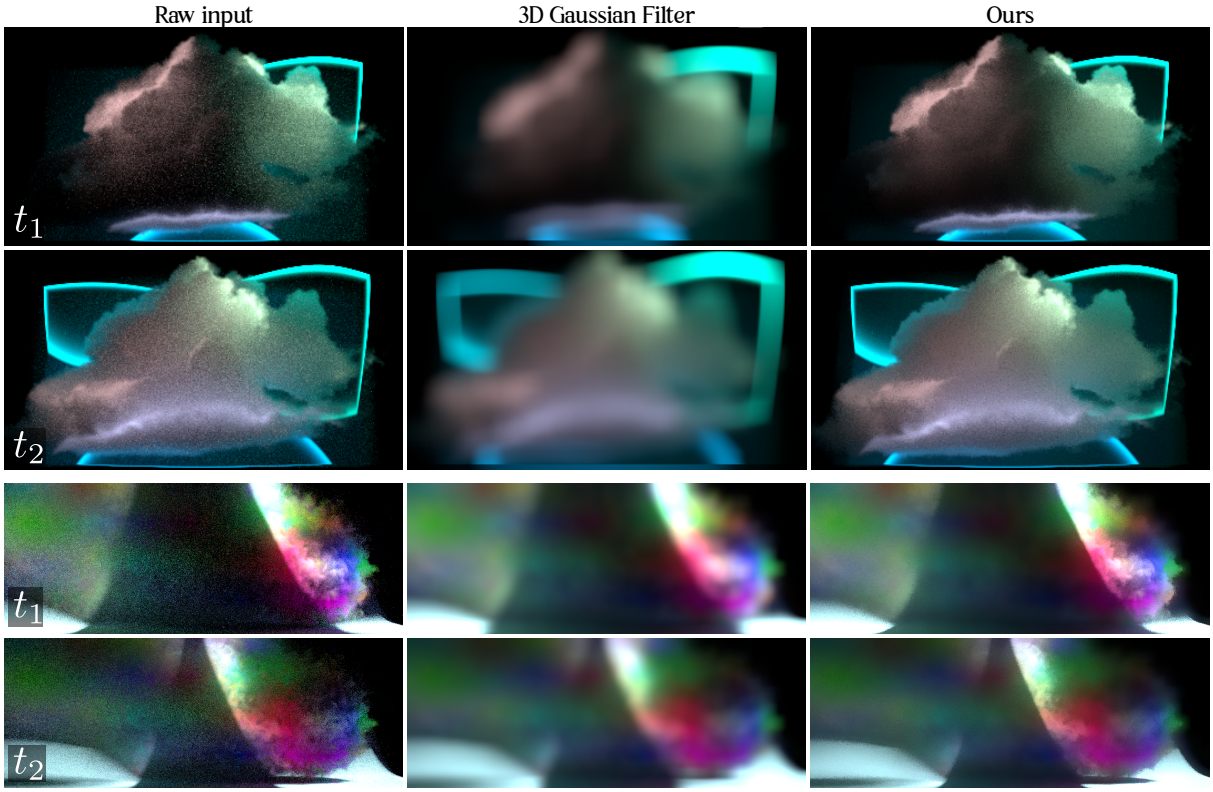

Figure 8: ***Denoising participating media.*** *We compare our method against the base 3D Gaussian Filter in the* DISNEY CLOUD *volume with three lights (pink, green, blue) and the* BANNER *scene with only lateral lighting, rendered with* 60*k and* 1024 *spp, respectively. While the Gaussian filter just blurs the image, our method enhances image quality by discriminating which time bins should be combined to preserve fine spatio-temporal detail while denoising high-noise regions. We refer the reader to the supplemental material for videos of transient light transport of these scenes.*

[BWM*23] BALINT M., WOLSKI K., MYSZKOWSKI K., SEIDEL H.-P., MANTIUK R.: Neural partitioning pyramids for denoising monte carlo renderings. In *ACM SIGGRAPH 2023 Conference Proceedings* (New York, NY, USA, 2023), SIGGRAPH '23, Association for Computing Machinery. URL: https://doi.org/10.1145/3588432.3591562, doi:10.1145/3588432.3591562.

[CHH*24] CHOI H., HONG S., HA I., KANG N., MOON B.: Online neural denoising with cross-regression for interactive rendering.

[CKS*17] CHAITANYA C. R. A., KAPLANYAN A. S., SCHIED C., SALVI M., LEFOHN A., NOWROUZEZAHRAI D., AILA T.: Interactive reconstruction of monte carlo image sequences using a recurrent denoising autoencoder. *ACM Trans. Graph. 36*, 4 (July 2017).

[Cur23] CURTO J. D.: Confidence intervals for means and variances of nonnormal distributions. *Communications in Statistics - Simulation and Computation 52*, 9 (2023), 4414–4430.

[DSHL10] DAMMERTZ H., SEWTZ D., HANIKA J., LENSCH H. P. A.: Edge-Avoiding À-TrousWavelet Transform for fast Global Illumination Filtering. In *High Performance Graphics* (2010), Doggett M., Laine S., Hunt W., (Eds.), The Eurographics Association.

[FFJ22] FIRMINO A., FRISVAD J. R., JENSEN H. W.: Progressive denoising of monte carlo rendered images. *Computer Graphics Forum 41*, 2 (2022), 1–11.

[FVW20] FACCIO D., VELTEN A., WETZSTEIN G.: Non-line-of-sight imaging. *Nature Reviews Physics 2*, 6 (Jun 2020), 318–327.

[FWJ*25] FINNENDAHL U., WORCHEL M., JÜTERBOCK T., WUJECKI D., BRINKMANN F., WEINZIERL S., ALEXA M.: Differentiable geometric acoustic path tracing using time-resolved path replay backpropagation. *ACM Trans. Graph. 44*, 4 (July 2025).

[GIGM22] GU J., IGLESIAS-GUITIAN J. A., MOON B.: Neural jamesstein combiner for unbiased and biased renderings. *ACM Trans. Graph. 41*, 6 (Nov. 2022). URL: https://doi.org/10.1145/3550454.3555496, doi:10.1145/3550454.3555496.

[HDJJ24] HE Q., DU D., JIANG H., JIN X.: Darts: Diffusion approximated residual time sampling for time-of-flight rendering in homogeneous scattering media. *ACM Trans. Graph. 43*, 6 (Nov. 2024).

[HHGH13] HEIDE F., HULLIN M. B., GREGSON J., HEIDRICH W.: Low-budget transient imaging using photonic mixer devices. *ACM Trans. Graph. 32*, 4 (July 2013).

[IGMM22] IGLESIAS-GUITIAN J. A., MANE P., MOON B.: Real-time denoising of volumetric path tracing for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics 28*, 7 (2022), 2734–2747. doi:10.1109/TVCG.2020.3037680.

[JMM*14] JARABO A., MARCO J., MUNOZ A., BUISAN R., JAROSZ W., GUTIERREZ D.: A framework for transient rendering. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia) 33*, 6 (Nov. 2014).

[JMMG17] JARABO A., MASIA B., MARCO J., GUTIERREZ D.: Recent advances in transient imaging: A computer graphics and vision perspective. *Visual Informatics 1*, 1 (2017), 65–79.

[JMV*12] JARABO A., MASIA B., VELTEN A., RASKAR R., GUTIÉR-

REZ D.: Femto-photography: Capturing light in motion. *Jornada de Jóvenes Investigadores del I3A* (May 2012), 15.

[JNT*11] JAROSZ W., NOWROUZEZAHRAI D., THOMAS R., SLOAN P.-P., ZWICKER M.: Progressive photon beams. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia) 30*, 6 (Dec. 2011).

[JSR*22] JAKOB W., SPEIERER S., ROUSSEL N., NIMIER-DAVID M., VICINI D., ZELTNER T., NICOLET B., CRESPO M., LEROY V., ZHANG Z.: Mitsuba 3 renderer, 2022. https://mitsuba-renderer.org.

[JSRV22] JAKOB W., SPEIERER S., ROUSSEL N., VICINI D.: Dr.jit: a just-in-time compiler for differentiable rendering. *ACM Trans. Graph. 41*, 4 (July 2022).

[Kaj86] KAJIYA J. T.: The rendering equation. *SIGGRAPH Comput. Graph. 20*, 4 (Aug. 1986), 143–150.

[KBW*25] KIM J., BENKO C., WRENNINGE M., VILLEMIN R., BARBER Z., JAROSZ W., PEDIREDLA A.: A Monte Carlo rendering framework for simulating optical heterodyne detection. *ACM Transactions on Graphics (Proceedings of SIGGRAPH) 44*, 4 (Aug. 2025).

[KGH*14] KŘIVÁNEK J., GEORGIEV I., HACHISUKA T., VÉVODA P., ŠIK M., NOWROUZEZAHRAI D., JAROSZ W.: Unifying points, beams, and paths in volumetric light transport simulation. *ACM Trans. Graph. 33*, 4 (July 2014). URL: https://doi.org/10.1145/2601097.2601219, doi:10.1145/2601097.2601219.

[KK51] KENNEY J., KEEPING E.: *Mathematics of Statistics*. No. parte 2. Van Nostrand, 1951.

[KMSP24] KIM J., MULTHAUP J., SNEHA M., PEDIREDLA A.: Efficient time sampling strategy for transient absorption spectroscopy. In *2024 IEEE International Conference on Computational Photography (ICCP)* (2024), pp. 1–12.

[LCH*20] LIANG Y., CHEN M., HUANG Z., GUTIERREZ D., NOZ A. M., MARCO J.: Compression and denoising of time-resolved light transport. *Opt. Lett. 45*, 7 (Apr 2020), 1986–1989.

[LWC12] LI T.-M., WU Y.-T., CHUANG Y.-Y.: Sure-based optimization for adaptive sampling and reconstruction. *ACM Trans. Graph. 31*, 6 (Nov. 2012).

[Men15] MENG X.: Simpler online updates for arbitrary-order central moments, 2015.

[MGJ*19] MARCO J., GUILLÉN I., JAROSZ W., GUTIERREZ D., JARABO A.: Progressive transient photon beams. *Computer Graphics Forum 38*, 6 (2019), 19–30.

[ODR09] OVERBECK R. S., DONNER C., RAMAMOORTHI R.: Adaptive wavelet rendering. *ACM Trans. Graph. 28*, 5 (Dec. 2009), 1–12.

[PAJ19] PAN X., ARELLANO V., JARABO A.: Transient instant radiosity for efficient time-resolved global illumination. *Computers & Graphics 83* (2019), 107–113.

[PGM*19] PASZKE A., GROSS S., MASSA F., LERER A., BRADBURY J., CHANAN G., KILLEEN T., LIN Z., GIMELSHEIN N., ANTIGA L., DESMAISON A., KOPF A., YANG E., DEVITO Z., RAISON M., TEJANI A., CHILAMKURTHY S., STEINER B., FANG L., BAI J., CHINTALA S.: Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035.

[PVG19] PEDIREDLA A., VEERARAGHAVAN A., GKIOULEKAS I.: Ellipsoidal path connections for time-gated rendering. *ACM Trans. Graph. 38*, 4 (July 2019).

[RCGP23] ROYO D., CRESPO M., GARCIA-PUEYO J.: mitransient. https://github.com/diegoroyo/mitransient, 2023.

[RGMJ22] ROYO D., GARCÍA J., MUÑOZ A., JARABO A.: Non-line-of-sight transient rendering. *Computers & Graphics* (2022).

[RHL*22] ROYO D., HUANG Z., LIANG Y., SONG B., NOZ A. M., GUTIERREZ D., MARCO J.: Structure-aware parametric representations for time-resolved light transport. *Opt. Lett. 47*, 19 (Oct 2022), 5212–5215. URL: https://opg.optica.org/ol/abstract.cfm?URI=ol-47-19-5212, doi:10.1364/OL.465316.

[RKZ12] ROUSSELLE F., KNAUS C., ZWICKER M.: Adaptive rendering with non-local means filtering. *ACM Trans. Graph. 31*, 6 (Nov. 2012).

[SD12] SEN P., DARABI S.: On filtering the noise from the random parameters in monte carlo rendering. *ACM Trans. Graph. 31*, 3 (May 2012).

[SFA*24] SAKAI H., FREUDE C., AUZINGER T., HAHN D., WIMMER M.: A statistical approach to monte carlo denoising. In *SIGGRAPH Asia 2024 Conference Papers* (New York, NY, USA, 2024), SA '24, Association for Computing Machinery.

[SFWH25] SAKAI H., FREUDE C., WIMMER M., HAHN D.: Statistical error reduction for monte carlo rendering. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers* (New York, NY, USA, 2025), SA Conference Papers '25, Association for Computing Machinery. URL: https://doi.org/10.1145/3757377.3763995, doi:10.1145/3757377.3763995.

[SKW*17] SCHIED C., KAPLANYAN A., WYMAN C., PATNEY A., CHAITANYA C. R. A., BURGESS J., LIU S., DACHSBACHER C., LEFOHN A., SALVI M.: Spatiotemporal variance-guided filtering: real-time reconstruction for path-traced global illumination. In *Proceedings of High Performance Graphics* (New York, NY, USA, 2017), HPG '17, Association for Computing Machinery.

[SSD08] SMITH A., SKORUPSKI J., DAVIS J.: *Transient Rendering*. Tech. Rep. UCSC-SOE-08-26, School of Engineering, University of California, Santa Cruz, February 2008.

[SXV*16] SHIN D., XU F., VENKATRAMAN D., LUSSANA R., VILLA F., ZAPPA F., GOYAL V. K., WONG F. N. C., SHAPIRO J. H.: Photon-efficient imaging with a single-photon camera. *Nature Communications 7*, 1 (Jun 2016), 12046.

[TIT*18] TANAKA K., IKEYA N., TAKATANI T., FUNATOMI T., KUBO H., MUKAIGAWA Y.: Time-resolved light transport decomposition for thermal photometric stereo. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2018), p. 4804–4813.

[VAN*19] VICINI D., ADLER D., NOVÁK J., ROUSSELLE F., BURLEY B.: Denoising deep monte carlo renderings. *Computer Graphics Forum 38*, 1 (2019), 316–327.

[VD98] VEACH E., DEPARTMENT S. U. C. S.: *Robust Monte Carlo Methods for Light Transport Simulation*. No. n.º 1610 in Report (Stanford University. Computer Science Department). Stanford University, Department of Computer Science, 1998.

[VWJ*13] VELTEN A., WU D., JARABO A., MASIA B., BARSI C., JOSHI C., LAWSON E., BAWENDI M., GUTIERREZ D., RASKAR R.: Femto-photography: capturing and visualizing the propagation of light. *ACM Trans. Graph. 32*, 4 (July 2013).

[Wel62] WELFORD B. P.: Note on a method for calculating corrected sums of squares and products. *Technometrics 4*, 3 (1962), 419–420.

[WQH*24] WANG Q., QIAO P., HUO Y., ZHAI S., XIE Z., HUA W., BAO H., LIU T.: Neural kernel regression for consistent monte carlo denoising. *ACM Trans. Graph. 43*, 6 (Nov. 2024). URL: https://doi.org/10.1145/3687949, doi:10.1145/3687949.

[XXZ*25] XU C., XU X., ZHANG J., LIU Y., CAO Y., ZHAO L.: Real-time neural denoising for volume rendering using dual-input feature fusion network. *Computer Graphics Forum 44*, 6 (2025), e70276. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.70276, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.70276, doi:https://doi.org/10.1111/cgf.70276.

[YJ00] YEO I.-K., JOHNSON R. A.: A new family of power transformations to improve normality or symmetry. *Biometrika 87*, 4 (2000), 954–959.

[YKC*21] YI S., KIM D., CHOI K., JARABO A., GUTIERREZ D., KIM M. H.: Differentiable transient rendering. *ACM Trans. Graph. 40*, 6 (Dec. 2021).

[ZZXY21] ZHENG S., ZHENG F., XU K., YAN L.-Q.: Ensemble denoising for monte carlo renderings. *ACM Trans. Graph. 40*, 6 (Dec. 2021). URL: https://doi.org/10.1145/3478513.3480510, doi:10.1145/3478513.3480510.