Geometric-Photometric Event-based 3D Gaussian Ray Tracing

Kai Kohyama [1], Yoshimitsu Aoki [1], Guillermo Gallego [2], Shintaro Shiba [1,3]
[1] Keio University, [2] Technische Universität Berlin, Science of Intelligence Excellence Cluster,
Einstein Center Digital Future, and Robotics Institute Germany, [3] The University of Tokyo.

Abstract

Event cameras offer a high temporal resolution over traditional frame-based cameras, which makes them suitable for motion and structure estimation. However, it has been unclear how event-based 3D Gaussian Splatting (3DGS) approaches could leverage fine-grained temporal information of sparse events. This work proposes GPERT, *a framework to address the trade-off between accuracy and temporal resolution in event-based 3DGS. Our key idea is to decouple the rendering into two branches: event-by-event geometry (depth) rendering and snapshot-based radiance (intensity) rendering, by using ray-tracing and the image of warped events. The extensive evaluation shows that our method achieves state-of-the-art performance on the real-world datasets and competitive performance on the synthetic dataset. Also, the proposed method works without prior information (e.g., pretrained image reconstruction models) or COLMAP-based initialization, is more flexible in the event selection number, and achieves sharp reconstruction on scene edges with fast training time. We hope that this work deepens our understanding of the sparse nature of events for 3D reconstruction. https://github.com/e3ai/gpert*

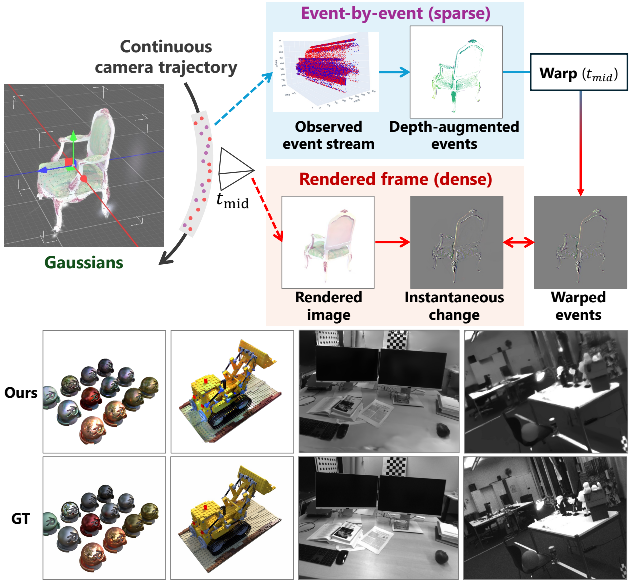


Figure 1. *Overview of the proposed method*, which takes raw events and poses as input. During the optimization of the 3D Gaussians, rendering is decoupled into two pathways: *event-by-event* (temporally dense) depth rendering and spatially dense intensity rendering. We use the image of warped events to connect these two pathways to compute both geometric and photometric losses. The color results are from a synthetic dataset, and the monochrome results are from two standard, real-world datasets.

1. Introduction

Event cameras have attracted increasing attention in computer vision and robotics due to their advantages and capabilities for various tasks [1, 2]. Unlike conventional cameras that record synchronous frames at fixed time intervals, event cameras respond asynchronously to per-pixel brightness changes with μs resolution [3–5]. This working principle makes them highly sensitive to motion, since changes are due to scene contrast and relative motion.

In parallel, Gaussian Splatting (GS) [6] has emerged as a state-of-the-art representation for photometric 3D reconstruction and novel view synthesis (NVS). Since 3D structure and motion are tightly connected in the generation of event data, it is paramount to develop 3DGS algorithms that leverage the event camera's fine-grained temporal information. Also, thanks to the event camera's minimal motion blur and high dynamic range (DR), event-based GS methods have the potential to overcome some of the limitations of frame-based GS, such as motion blur and low DR.

Previous approaches in event-based photometric 3D reconstruction (e.g., NeRF [7] and GS) typically perform two dense renderings per sample (e.g., [8]). The difference between these two rendered images is compared with the edge-like image obtained by pixel-wise aggregation of the event data, resulting in a photometric loss that drives the 3D-Gaussian optimization [8, 9]. However, as this approach requires dense (all-pixel) scene rendering twice, it not only slows down the training, but also introduces a fundamental limitation: the trade-off between accuracy and temporal window selection. A short time interval between the two renderings fails to capture subtle intensity variations that generate only a few events. Contrarily, a large interval makes the predicted edge image blurry and discards fine-grained temporal information in the observed edge. Similar observations have been made in [10, 11]: a trade-off between capturing global lighting and local details.

In this work, we fundamentally address these limitations and introduce the first framework for event-based GS that renders the dense intensity (radiance) *once* per sample (i.e., a batch of events), while keeping the rendering efficient and leveraging the high temporal resolution. Our key idea is to consider dedicated structure and appearance updates in the GS framework (Fig. 1). A "structure" pathway defines a geometric loss built on top of Contrast Maximization [12] by leveraging the known camera motion and the event-by-event (i.e., sparse) rendered depth. An "appearance" pathway defines a photometric loss between the instantaneous brightness change modeled by the rendered dense intensity and the change measured by the event data.

Our method showcases several advantages through extensive evaluations. First, it does not rely on any prior knowledge (e.g., frames or pretrained models for depth and intensity) or COLMAP [13] for initialization. Second, it achieves state-of-the-art rendering quality performance on real-world datasets, where poses and raw events are noisy, with fast training time (it is faster than the latest methods that we benchmark [8, 9, 14]). Third, it shows robustness with respect to the number of events processed per sample, without compromising accuracy.

In summary, this work presents several distinctive contributions in event-only Gaussian Splatting:

1. The proposed method decouples two different quantities in 3DGS rendering: the continuous-time spatially sparse depth, and the instantaneous dense intensity. It addresses the trade-off between accuracy and temporal resolution in existing event-based 3D reconstruction methods.
2. The comprehensive evaluation shows state-of-the-art results on real-world datasets and competitive results in simulation without relying on any prior knowledge, as opposed to existing event-based 3DGS methods.
3. The proposed framework connects event-by-event depth estimation and 3DGS, which is enabled by an efficient event-by-event ray-tracing implementation.
4. The method achieves the fastest training time among other tested state-of-the-art methods (e.g., [8, 9, 14]).

We hope that this work unblocks the potential of high temporal resolution event data in 3D reconstruction.

2. Related Work

3D Gaussian Splatting (3DGS) [6] represents the scene as a collection of anisotropic Gaussian ellipsoids and renders via differentiable splatting [6]. These methods achieve high-fidelity NVS with faster rendering and more scalable training in static and dynamic scenes [15–17] than Neural Radiance Fields (NeRFs) [7]. While most works focus on rasterization-based splatting, recently, ray-tracing approaches have emerged, which inspire our work. Notably, 3D Gaussian Ray Tracing (3DGRT) [18] casts rays against volumetric Gaussian particles, and 3DGUT [19] replaces Elliptical weighted-average raster splatting with an Unscented Transform projection of Gaussian particles.

Event cameras have inspired substantial work in reconstructing 3D geometry and scene appearance, leveraging motion information from their data modality of high temporal resolution. Several approaches propose enhanced frame-based NVS aided by the complementary information in the event data, e.g., in the presence of fast motion (images with motion blur) or deformations [11, 20–23]. Such approaches may still suffer from bottlenecks of the frame-based cameras in the system, such as low DR, and inaccuracies in sensor fusion calibration. Instead, event-only methods like our work focus on unlocking the potential of event cameras.

The key question of event-based NVS methods (NeRF and GS) lies in the measurement model: how to compare the modeled scenes, which typically display absolute intensity rendered at concrete viewpoints and times, to the acquired event data stream, which measures sparse intensity differences asynchronously at a quasi-continuous collection of times and viewpoints. They seem opposites. The typical approach in the literature consists in accumulating events into an image (of intensity increments) and computing the photometric error with respect to the difference between two rendered frames at the first and last event timestamps. While many event NeRF methods (e.g., [10, 24–26]) and GS methods (e.g., [11, 27–29]) follow such philosophy, they face the trade-off between accuracy and temporal window selection. Instead of rendering dense images, per-event loss computation has been proposed in event-NeRF [14, 30, 31] based on ray-tracing, leveraging event generation models to directly compare each observed event. However, these NeRF methods tend to suffer from considerable event noise in real-world cameras and slow rendering.

Notably, most existing works in event-based GS utilize some prior knowledge. For example, Elite-EvGS [27] utilizes pretrained event-to-video models for initialization and regularization. Event-3DGS [32], which jointly estimates sensor parameters for better photometric reconstruction, also uses video reconstruction for initialization. IncEventGS [9], which is conceived as an incremental tracking and mapping system (and therefore does not need camera pose information), uses a depth-pretrained model for bootstrapping. E-3DGS [33] uses an additional piece of data to better recover absolute intensity: exposure events obtained by controlling the camera's aperture.

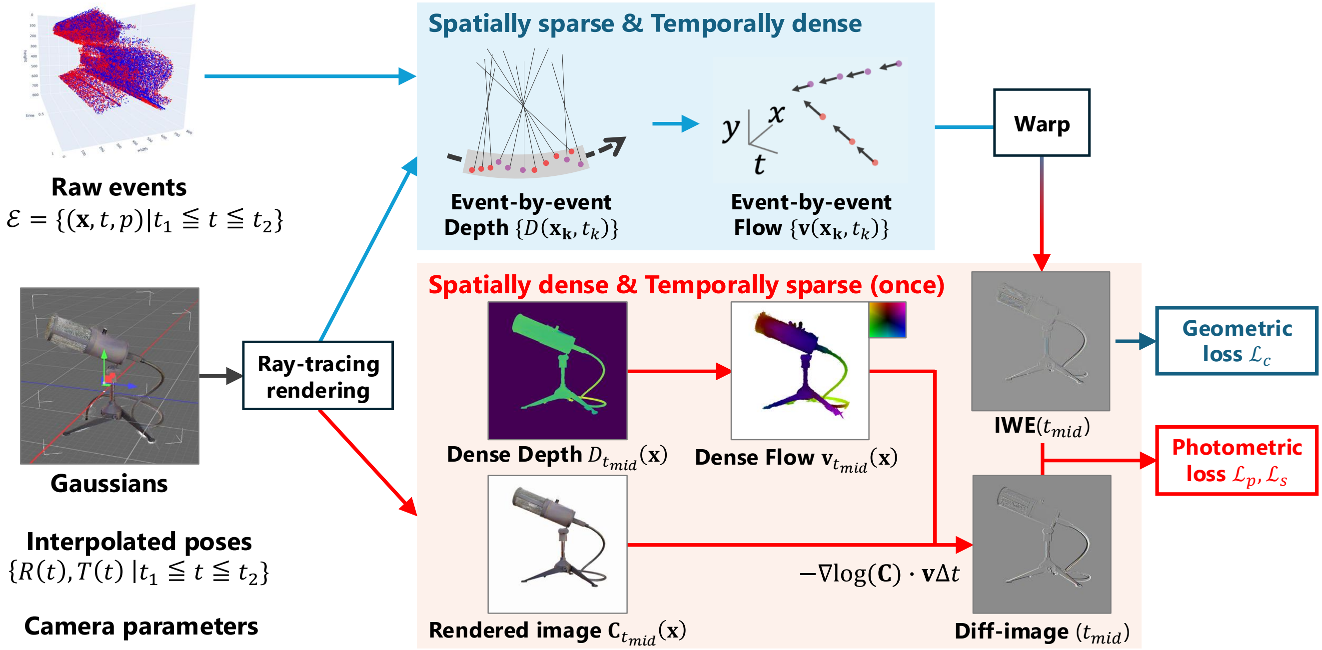


Figure 2. *Method overview*. Using ray-tracing renderer, we estimate depth for each event and compute the flow with the interpolated poses (i.e., motion field). Performing event warping produces the image of warped events at t_{mid} and computes the contrast loss. We render the dense intensity (radiance) at t_{mid} and compute the instantaneous brightness increment image, which we use for the photometric loss.

Our work falls into the category of event-only 3DGS methods, such as Event-3DGS [32] and EventSplat [8], however, with several significant differences: (i) earlier work utilize prior knowledge (e.g., the pretrained E2VID model [34]) for initial intensity recovery or initial 3D Gaussians [9], while ours does not or rely on any prior knowledge, and (ii) we explicitly incorporate geometric and photometric loss terms in the proposed render-once framework, which improve robustness with respect to the choice of the number of events processed, as opposed to using a multi-window optimization scheme in a two-rendering pipeline [8, 10]. The idea of using event warping is concurrently proposed in PAEv3D [25] (NeRF) and EF-3DGS [35] (GS using both events and frames), however, none of them tackle event-only GS or realize per-event depth rendering.

3. Methodology

The overview of our framework is shown in Figure 2. The scene is modeled via 3D Gaussians (Sec. 3.1) comprising structure and appearance parameters that interact with the event data through the optimization of a weighted loss function. The loss combines a *geometric* term that measures the goodness of fit between the event data and the modeled apparent motion, and *appearance / photometric* terms that measure the goodness of fit between the events and the brightness increment predicted by the 3D Gaussian scene model. Accordingly, we propose to decouple the processing in two branches: an event-by-event (i.e., spatially sparse but temporally dense) scene rendering of the unknown depth for geometric loss computation (Secs. 3.2 and 3.3) and a snapshot-based (i.e., spatially dense but temporally sparse) rendering for the photometric loss (Sec. 3.4). We use the image of warped events (IWE) [12] to connect both branches.

3.1. 3D Gaussian Splatting

In the typical 3D Gaussian Splatting (3DGS) setting [6], a static scene is represented as a set of N_g Gaussians $\mathcal{G} = \{(\boldsymbol{\mu}_i, \Sigma_i, \mathbf{c}_i, \alpha_i)\}_{i=1}^{N_g}$, where $\boldsymbol{\mu}_i \in \mathbb{R}^3$ denotes the 3D mean position, $\Sigma_i \in \mathbb{R}^{3\times 3}$ the covariance matrix encoding the anisotropic spatial extent, $\mathbf{c}_i \in \mathbb{R}^3$ the color and $\alpha_i \in [0, 1]$ the opacity. Each Gaussian defines a density function in space: $\mathbb{G}_i(\mathbf{X}; \boldsymbol{\mu}_i, \Sigma_i) \doteq e^{-\frac{1}{2}(\mathbf{X}-\boldsymbol{\mu}_i)^\top \Sigma_i^{-1}(\mathbf{X}-\boldsymbol{\mu}_i)}$. The rendered appearance is obtained by projecting these 3D Gaussians into the image plane and blending their contributions according to visibility and opacity. Projected Gaussians are at pixel locations $\boldsymbol{\mu}'_i = \pi(\boldsymbol{\mu}_i)$ and with covariances $\Sigma'_i \approx J_i \Sigma_i J_i^\top$, where $J_i = \frac{\partial \pi(\mathbf{X})}{\partial \mathbf{X}}\big|_{\mathbf{X}=\boldsymbol{\mu}_i}$ is the Jacobian of the projection function $\pi : \mathbb{R}^3 \to \mathbb{R}^2$ that maps world coordinates to pixel coordinates. The contribution of each Gaussian to a pixel $\mathbf{x}$ is then given by: $w_i(\mathbf{x}) = \alpha_i \mathbb{G}_i(\mathbf{x}; \boldsymbol{\mu}'_i, \Sigma'_i)$.

The rendered color for each pixel $\mathbf{C}(\mathbf{x})$ is approximated by alpha compositing along the camera ray with correct ordering and blending based on depth:

$$\mathbf{C}(\mathbf{x}) = \textstyle\sum_{i=1}^{N} \mathbf{c}_i\, w_i(\mathbf{x}) \prod_{j=1}^{i-1}(1-w_j(\mathbf{x})). \tag{1}$$

Finally, to obtain a differentiable depth rendering, we associate each Gaussian with a mean depth value $Z_i = \mathbf{e}_3^\top \boldsymbol{\mu}_i$ in camera coordinates. The rendered depth $D(\mathbf{x})$ is then given by the opacity-weighted expectation:

$$D(\mathbf{x}) = \frac{\sum_{i=1}^{N} Z_i\, w_i(\mathbf{x}) \prod_{j<i}(1-w_j(\mathbf{x}))}{\sum_{i=1}^{N} w_i(\mathbf{x}) \prod_{j<i}(1-w_j(\mathbf{x})) + \epsilon}. \tag{2}$$

3.2. Event-by-event Ray Tracing

An event camera asynchronously captures visual changes as soon as the log-intensity L at a pixel $\mathbf{x}$ exceeds a threshold C_{th}: $\Delta L(\mathbf{x}_k, t_k) \doteq L(\mathbf{x}_k, t_k) - L(\mathbf{x}_k, t_k - \Delta t_k) = p_k\, C_{\text{th}}$. Each event $e_k \doteq (\mathbf{x}_k, t_k, p_k)$ specifies the space-time coordinates $(\mathbf{x}_k, t_k)$ and polarity $p_k \in \{+1, -1\}$ of the change.

Events are sparse in pixel space and quasi-continuous (dense) in time. To fully leverage sparsity, the rendering of the 3DGS should be sparse rather than image rasterization. Hence, we propose the framework of *event-by-event rendering* in the 3DGS pipeline, inspired by recent advances in ray-tracing GS [18, 19]. The idea that each event should also carry information about depth, i.e., *depth-augmented events*, originates in [36] for the context of SLAM.

For each event e_k, we render the corresponding depth $D(\mathbf{x}_k, t_k)$, which is now a function of both space and time. To this end, at each timestamp t_k, we compute the interpolated camera pose $(R(t_k), T(t_k))$ and the ray through the camera's optical center and pixel $\mathbf{x}_k$. Finally, GPU-accelerated ray tracing enables us to efficiently render event-by-event depth $\mathcal{D} \doteq \{D(\mathbf{x}_k, t_k)\}_{k=1}^{N_e}$, as illustrated in Fig. 3, column (b).

Assuming a stationary scene viewed by a moving camera with linear and angular velocities $\mathbf{V}$ and $\boldsymbol{\omega}$, respectively, the per-event depth D can be used to compute the per-event apparent motion via the motion field equation (Sec. 8.2) [37]:

$$\mathbf{v}(\mathbf{x}, t) = \frac{1}{D(\mathbf{x}, t)} A(\mathbf{x})\mathbf{V} + B(\mathbf{x})\boldsymbol{\omega}. \tag{3}$$

See the example in Fig. 3, column (d).

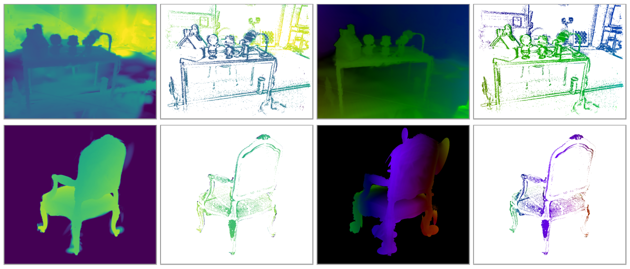

(a) Dense depth (b) Sparse depth (c) Dense flow (d) Sparse flow

Figure 3. *Visualization of dense/sparse depth and optical flow.* Sparse depth and optical flow are not simply obtained by masking the dense counterparts, but by actual event-by-event ray tracing (Sec. 3.2). Top: using real events (EDS). Bottom: using synthetic events. The flow color notation is specified in Fig. 2.

3.3. Geometric Loss

To guide the estimation of the 3DGS parameters, we consider a geometric loss that is computed in an unsupervised manner following the Contrast Maximization (CMax) framework [12] that is widely used for various motion estimation tasks [38–47]. Under the brightness constancy assumption, events $\mathcal{E} \doteq \{e_k\}_{k=1}^{N_e}$ are caused by moving edges and can be motion-compensated by a warping operation if their motion is known: $\mathcal{E}'_{t_{\text{ref}}} \doteq \{e'_k\}_{k=1}^{N_e}$, where $e'_k \doteq (\mathbf{x}'_k, t_{\text{ref}}, p_k)$, at a reference time t_{ref}. We formulate the warp using the spatio-temporal optical flow $\mathbf{v}(\mathbf{x}, t)$ [48], which in the 3DGS setting can be obtained using (3),

$$\mathbf{x}'_k = \mathbf{x}_k + (t_k - t_{\text{ref}})\, \mathbf{v}(\mathbf{x}_k, t_k). \tag{4}$$

Then, the warped events are aggregated to produce an image or histogram of warped events (IWE, top branch of Fig. 2)

$$\text{IWE}(\mathbf{x}; t_{\text{ref}}, D) \doteq \textstyle\sum_{k=1}^{N_e} b_k C_{\text{th}} \delta(\mathbf{x} - \mathbf{x}'_k), \tag{5}$$

where $b_k = p_k$ if polarity is used and $b_k = 1$ if polarity is not used. The Dirac delta is approximated by a Gaussian, $\delta(\mathbf{x} - \boldsymbol{\mu}) \approx \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \sigma^2 = 1\text{px})$.

The IWE measures the alignment between the event data and the candidate motion $\mathbf{v}$. The true motion $\mathbf{v}^*$ leads to a sharp IWE, with motion-compensated edges. Hence, as geometric loss we use the IWE sharpness (without polarity, $b_k = 1$), normalized by the value at zero flow [48]:

$$\begin{gathered} \mathcal{L}_{\text{c}} \doteq G(\mathbf{0}; -) \,/\, G(\mathbf{v}(D); t_{\text{ref}}), \\ G(\mathbf{v}(D); t_{\text{ref}}) = \tfrac{1}{|\Omega|} \textstyle\int_{\Omega} \|\nabla \text{IWE}(\mathbf{x}; t_{\text{ref}}, D)\|_1 d\mathbf{x}. \end{gathered} \tag{6}$$

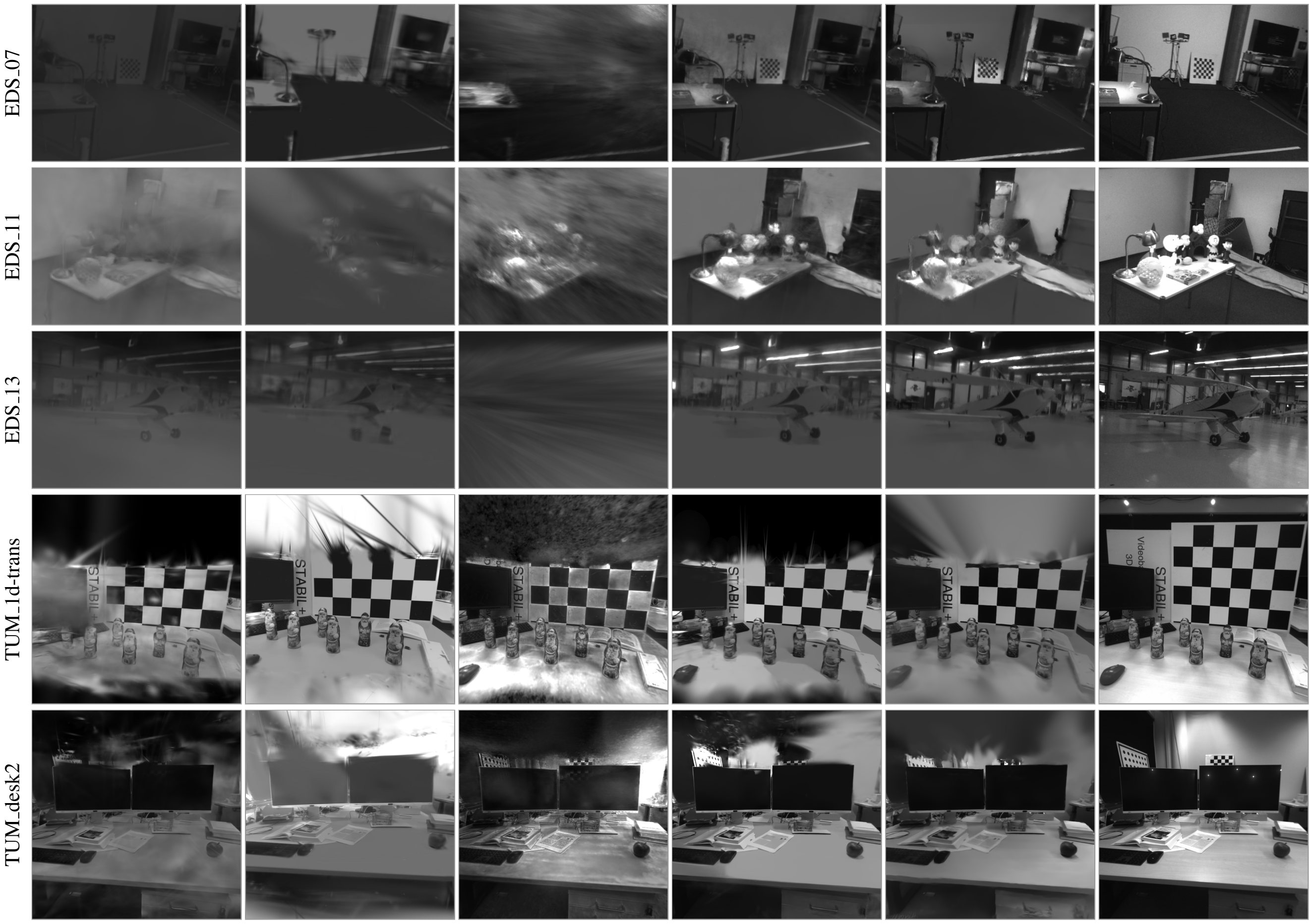
Notice that we use the reciprocal of the contrast objective due to the minimization formulation, and the L^1-norm because it performs well for depth estimation [49].

3.4. Photometric Loss

The IWE (5) represents not only motion-corrected edges, but also their strength (e.g., intensity gradient) with respect to the flow direction [12, 52]. Hence, we may use the IWE to design not only geometric loss terms but also photometric ones (bottom branch of Fig. 2, and examples in Fig. 3 columns (a), (c)). This is inspired by methods in the literature that define losses on brightness increment images obtained from grayscale information [50, 53–55].

Specifically, following the event generation model [1], the prediction of the scene's edge strength at time t_{ref} is:

$$\hat{H}(\mathbf{x}; t_{\text{ref}}) \doteq \frac{\partial \log \mathbf{C}}{\partial t} \Delta t \approx -\nabla \log(\mathbf{C}) \cdot \mathbf{v} \Delta t, \tag{7}$$



(a) E2VID + 3DGS (b) IncEventGS [9] (c) Robust E-NeRF [14] (d) EventSplat [8] (e) Ours (f) GT

Figure 4. *Results on the real-world datasets EDS [50] and TUM-VIE [51].* The event camera's field of view in the TUM dataset is narrower than the GT (i.e., frame camera) in the vertical direction.

where $\mathbf{C} \equiv \mathbf{C}(\mathbf{x})$ is the rendered frame (1) from the viewpoint of camera pose $(R(t_{\text{ref}}), T(t_{\text{ref}}))$ and $\mathbf{v} \equiv \mathbf{v}(\mathbf{x}, t_{\text{ref}})$ is the motion field (3) obtained using the rendered depth at time t_{ref} (bottom branch of Fig. 2). This corresponds to the instantaneous rate of brightness change in the optical flow direction [52]. Note that $\mathbf{C}$ may represent color for the simulated/color event cameras, or gray (intensity) for the standard event cameras. The dense-pixel (i.e., radiance) rendering happens once in each optimization step (see Sec. 5.1).

Finally, photometric errors between the IWE (with $b_k = p_k$) and its prediction (7) are defined by the L^2-norm and the Structural Similarity Index Measure (SSIM) [56]:

$$\begin{aligned} \mathcal{L}_{\text{p}} &\doteq \frac{1}{|\Omega|} \|\text{IWE}(\mathbf{x}; t_{\text{ref}}) - \hat{H}(\mathbf{x}; t_{\text{ref}})\|^2, \\ \mathcal{L}_{\text{s}} &\doteq \text{SSIM}\big(\text{IWE}(\mathbf{x}; t_{\text{ref}}),\ \hat{H}(\mathbf{x}; t_{\text{ref}})\big). \end{aligned} \tag{8}$$

We find that warping is more useful to leverage the high temporal resolution than the simple pixel-wise accumulation of polarities used in most event-based GS and NeRF literature ([8–10]), because the latter: (i) may result in blurry edge images that discard the fine temporal resolution, (ii) incurs neutralization (cancellation of event polarities), (iii) requires two dense intensity renderings to compute the photometric loss, (iv) omits a dependency on the unknown depth/flow that can be useful during optimization [45].

3.5. Combined Loss Function

For each slice of events $\mathcal{E}$ we use the middle timestamp as a reference, $t_{\text{ref}} \doteq t_{\text{mid}}$. The total loss is a weighted sum of the event-alignment loss (CMax) and the photometric losses:

$$\mathcal{L} \doteq \lambda_{\text{c}}\mathcal{L}_{\text{c}} + \lambda_{\text{p}}\mathcal{L}_{\text{p}} + \lambda_{\text{s}}\mathcal{L}_{\text{s}}. \tag{9}$$

3.6. Initialization

The initialization of the 3D Gaussians is important. For example, it is common practice for frame-based GS methods to use COLMAP [13] to favor initial Gaussians on scene texture and edges. Indeed, prior work EventSplat [8] uses

		EDS [50]						**TUM-VIE** [51]		
Metric	**Method**	**Avg.**	03	07	08	11	13	**Avg.**	1d-trans	desk2
PSNR ↑	E2VID + 3DGS	15.510	15.670	15.050	14.030	13.830	18.960	9.524	9.382	9.664
	Robust E-NeRF (ICCV'23) [14]	16.250	19.190	14.780	14.750	14.430	18.100	11.790	9.612	13.970
	IncEventGS (CVPR'25) [9]	15.210	14.130	15.760	15.890	13.830	16.460	10.090	10.130	10.050
	EventSplat (CVPR'25) [8]	18.860	**20.780**	19.140	17.530	**17.790**	19.050	–	–	–
	Ours	**19.470**	19.040	**20.240**	**21.030**	16.730	**20.300**	**13.090**	**11.970**	**14.200**
SSIM ↑	E2VID + 3DGS	0.692	0.716	0.689	0.642	0.691	0.723	0.516	0.525	0.507
	Robust E-NeRF (ICCV'23) [14]	0.739	**0.846**	0.815	0.735	0.569	0.729	0.573	0.504	0.642
	IncEventGS (CVPR'25) [9]	0.691	0.756	0.684	0.692	0.648	0.676	0.533	0.536	0.529
	EventSplat (CVPR'25) [8]	0.792	0.835	0.816	0.745	0.789	0.774	–	–	–
	Ours	**0.816**	0.819	**0.855**	**0.814**	**0.790**	**0.804**	**0.716**	**0.665**	**0.766**
LPIPS ↓	E2VID + 3DGS	0.375	0.266	0.378	0.402	0.415	0.415	0.759	0.790	0.728
	Robust E-NeRF (ICCV'23) [14]	0.543	0.324	0.476	0.567	0.700	0.650	0.588	0.721	0.454
	IncEventGS (CVPR'25) [9]	0.561	0.356	0.557	0.631	0.588	0.674	0.685	0.707	0.663
	EventSplat (CVPR'25) [8]	0.362	**0.239**	0.351	0.424	**0.391**	**0.407**	–	–	–
	Ours	**0.357**	0.272	**0.335**	**0.369**	0.396	0.414	**0.411**	**0.497**	**0.324**

Table 1. *Results on standard, real-world datasets EDS and TUM-VIE.* Best in bold.

intensity reconstruction and runs COLMAP for initialization. However, it relies on the pretrained E2VID model [34] as prior. We propose using the $\mathrm{IWE}(\mathbf{x}; t_{\mathrm{mid}})$ without polarity and the rendered image $\mathbf{C}(\mathbf{x})$ for initialization, keeping the rest of the pipeline untouched. This favors initial 3D Gaussians around scene structures because the IWE responds to edges. We find that IWEs produce better initialization than images of pixel-wise accumulation of events because of their sharpness, which narrows down the initial possible locations of the Gaussian centers (see Sec. 8.5).

4. Experiments

4.1. Datasets, Metrics, and Baselines

Datasets. We use standard datasets for event-based NeRF and GS works, both on simulated and real data. *EDS* [50] is a real-world dataset of indoor scenarios, recorded with a VGA event camera (640 × 480 px), an RGB camera, an IMU, and ground-truth poses from motion capture. The sequences include challenging scenes, such as flickering light sources. *TUM-VIE* [51] is another real-world dataset, acquired with an HD event camera (1280 × 720 px, i.e., 1 megapixel) and with ground-truth poses. It consists of indoor and outdoor sequences recorded with the sensor rig mounted on a helmet. We use indoor sequences following prior work. *Robust E-NeRF* [14] contributes a synthetic color event dataset with a 800×800 px resolution and color pixels following the Bayer pattern.

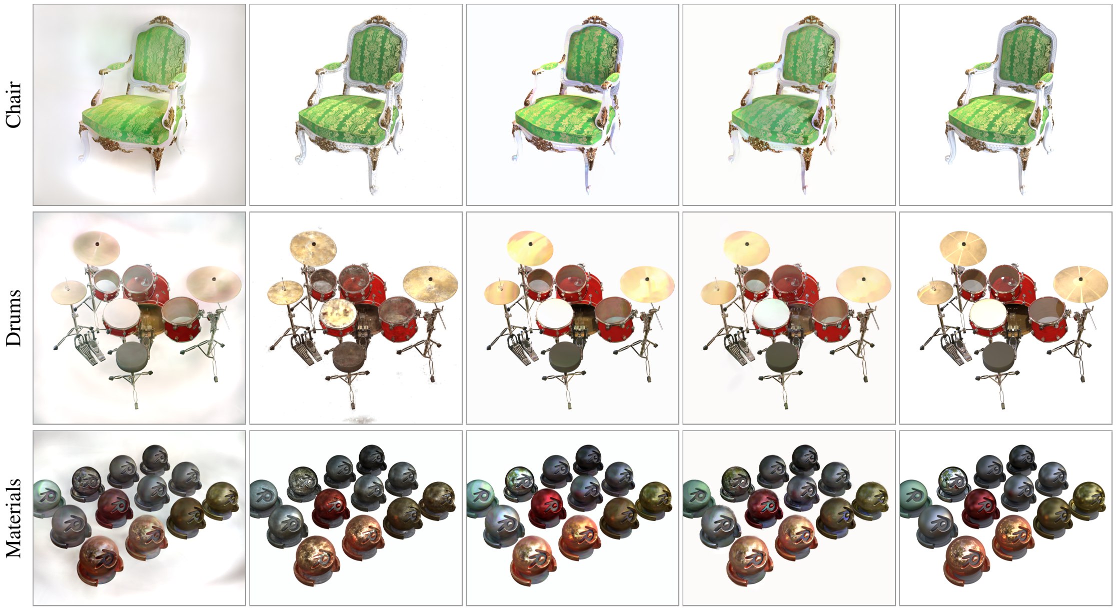
Evaluation Metrics. Following prior work [8, 14], reconstruction performance is measured with standard metrics on view synthesis quality: Peak Signal-to-Noise Ratio (PSNR), SSIM, and Learned Perceptual Image Patch Similarity (LPIPS). Real-world datasets use poses from colocated frame-based cameras for the evaluation of rendering. In addition, we follow prior work and apply gamma correction before computing the evaluation metrics.

Baselines. Our baselines are among the best event-only NeRF and GS methods in the literature. First, we use the two-stage approach of E2VID image reconstruction [34] and frame-based GS, termed "E2VID + 3DGS". For event-based GS methods (the two-rendering approaches), we retrain *IncEventGS* [9], and copy the results from *EventSplat* [8] because it has no available code. We also compare with the state-of-the-art NeRF method *Robust E-NeRF* [14] that uses event-by-event loss computation.

Hyper-parameters. For all sequences the contrast threshold is set to $C_{\mathrm{th}} = 0.25$. The loss weights in (9) are set to $\lambda_{\mathrm{c}} = 0.125, \lambda_{\mathrm{p}} = 500, \lambda_{\mathrm{s}} = 1$. The number of events is $N_e = 125\mathrm{k}$ for EDS and synthetic data [14], and $N_e = 500\mathrm{k}$ for TUM-VIE. We further test the robustness of the method to the choice of N_e. The initialization steps are 10k, and the entire training steps are 40k for all sequences.

4.2. Results on Real-World Datasets

Figure 4 shows the results on EDS and TUM-VIE datasets. Throughout the scenes, the proposed method consistently achieves successful reconstructions (we encourage readers to watch the video). Notably, our reconstructions recover fine details: (i) gradual (mild) intensity changes, e.g., shadows and reflections on the desk in *TUM-desk2*, (ii) fewer artifacts due to noisy events, e.g., walls on *EDS-07,11*. (iii) sharp edges in details, e.g., airplane and background in *EDS-13*. Also, EDS sequences contain lots of events due to flickering lights. Surprisingly, our method converges



(a) E2VID + 3DGS (b) Robust E-NeRF [14] (c) EventSplat [8] (d) Ours (e) GT

Figure 5. *Qualitative results on the color synthetic dataset [14].*

Metric	Method	Avg.	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic
PSNR ↑	E2VID + 3DGS	19.290	21.390	19.860	19.900	15.550	18.170	20.080	20.100
	Robust E-NeRF (ICCV'23) [14]	**28.190**	**30.240**	23.150	**30.710**	18.070	27.340	24.980	32.870
	EventSplat (CVPR'25) [8]	28.140	28.690	**25.810**	29.900	**22.910**	**29.220**	**27.160**	**33.270**
	Ours	23.110	26.420	23.340	25.360	17.760	18.080	23.500	27.300
SSIM ↑	E2VID + 3DGS	0.917	0.934	0.915	0.922	0.897	0.895	0.901	0.957
	Robust E-NeRF (ICCV'23) [14]	0.945	**0.958**	0.897	**0.971**	**0.953**	0.934	0.923	0.981
	EventSplat (CVPR'25) [8]	**0.953**	0.953	**0.947**	0.966	0.940	**0.945**	**0.936**	**0.986**
	Ours	0.927	0.941	0.921	0.938	0.911	0.901	0.910	0.968
LPIPS ↓	E2VID + 3DGS	0.118	0.076	0.094	0.108	0.208	0.145	0.125	0.069
	Robust E-NeRF (ICCV'23) [14]	0.057	**0.040**	0.091	**0.022**	**0.095**	0.074	**0.052**	0.029
	EventSplat (CVPR'25) [8]	**0.051**	0.047	**0.052**	0.028	0.098	**0.055**	0.060	**0.015**
	Ours	0.074	0.054	0.066	0.046	0.160	0.097	0.061	0.032

Table 2. *Quantitative results on the color synthetic dataset [14].* The Bayer pattern is challenging for the proposed warp-based method.

and successfully reconstructs the scene while relying on the contrast loss, which may be sensitive to flickering events.

The quantitative comparison is reported in Tab. 1. Our method consistently achieves state-of-the-art results: *the best results on average across all three metrics*, despite not relying on pretrained depth estimation models [9], or video-guided initialization and cubic splines for pose interpolation [8]. Notice that there are some limitations of the quantitative evaluation on the real-world sequences, such as the high dynamic range (HDR) of event cameras, and the disparity between the event camera and the frame camera.

4.3. Results on Synthetic Data

Due to the influence of synthetic RGB-based novel-view-synthesis datasets [7], the method is also tested on such data, converted into events via an event camera simulator [57]. Note that such sequences are unrealistic because they lack the noise and most dynamic effects characteristic of event cameras. Results are shown in Fig. 5. The RGB Bayer pattern is challenging for warp-based methods, such as the proposed one, since (i) warped pixels may not fall into the same location among different colors [52], which complicates the demosaicing operation, and (ii) the color distribution is imbalanced (green pixels are twice as many as red/blue). Nonetheless, our method achieves successful color reconstruction. Following the same color correction

Metric	Method	Synthetic [14]	EDS	TUM-VIE
PSNR ↑	Ours	**23.110**	**19.470**	13.090
	w/o contrast loss	9.600	15.520	**13.450**
	w/o initialization	20.820	17.340	11.360
SSIM ↑	Ours	**0.927**	**0.816**	**0.716**
	w/o contrast loss	0.810	0.744	0.715
	w/o initialization	0.908	0.759	0.633
LPIPS ↓	Ours	**0.073**	**0.357**	0.411
	w/o contrast loss	0.405	0.581	**0.405**
	w/o initialization	0.121	0.442	0.561

Table 3. *Ablation on the contrast loss and the initialization.*

steps as [8, 14], our results show fewer object artifacts and fewer floaters on the background. Quantitative results are given in Tab. 2, where we achieve competitive values.

Figure 3 shows rendered depth (sparse or dense) obtained on this data and real-world data. We find that Gaussian-based depth estimation achieves high-quality results, especially around occlusions. We report the quantitative comparisons with EMVS [58] in the supplementary.

4.4. Runtime

The training takes 30–45 minutes for EDS and synthetic sequences [14], and 80–130 minutes for TUM-VIE. The rendering takes roughly 3 ms for $N_g = 0.1M$, and 30 ms for $N_g = 1M$, using a PyTorch implementation on an NVIDIA RTX6000 (Ada). Our method is significantly faster than other methods: both Robust E-NeRF [14] and IncEventGs [9] take 3 h to train on EDS under the same settings. EventSplat [8] does not have publicly available code but reports 1–3 h for the same number of iterations on EDS.

5. Ablations

5.1. Effect of Temporal Window Selection

We further investigate the efficacy of our two-branch pipeline, which renders intensity just once (Fig. 2). Most event-based GS methods render dense intensity twice and subtract one from another to obtain an edge-like image (i.e., $\Delta\mathbf{C} = \mathbf{C}(t_2) - \mathbf{C}(t_1)$) that is compared to the brightness increment obtained by pixel-wise accumulation of the event data. A clear advantage of the proposed pipeline over the above "render-twice" pipeline is the robustness with respect to the choice of N_e. Figure 6 reports reconstruction performance for different N_e, using two sequences from the TUM-VIE dataset: *1d-trans* and *desk2*. For larger N_e the edges become more blurry in the render-twice pipeline, and therefore, the reconstruction quality degrades. However, the proposed render-once pipeline shows consistent results regardless of N_e, which is desirable because it is a sensible parameter that depends on many factors, such as camera resolution, scene texture, and camera motion.

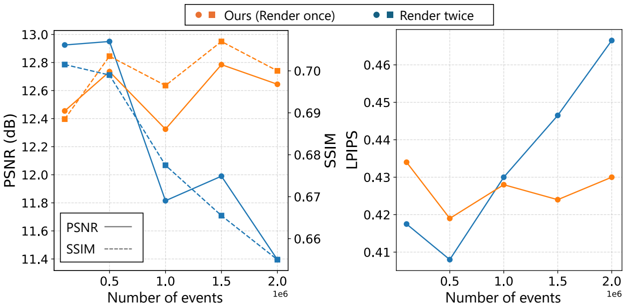


Figure 6. *Robustness with respect to the time window selection.* We compare the proposed pipeline and its render-twice variant for different numbers of events N_e. Due to the warp that reduces blur in the scene, the proposed method shows robustness against the choice of N_e, achieving consistently good values.

5.2. Contrast Loss and Initialization

We conduct ablation studies on the contrast loss and initialization, as shown in Tab. 3. Here, "w/o initialization" starts from random 10^5 Gaussians and skips the proposed initialization step (Sec. 3.6). Our method achieves the best or second-best results among all metrics and datasets. Notably, the SSIM improves with the contrast loss, showcasing the efficacy of the proposed ray-tracing rendering and loss.

6. Limitations

Our method follows an unsupervised approach based on the contrast loss, which assumes brightness constancy and therefore suffers in the presence of flickering events. Although the pipeline converges on the EDS dataset, we find that the presence of large amounts of flickering events make appearance recovery and depth estimation results unstable.

The proposed framework assumes static scenes, and is not expected to work well on dynamic scenes. However, following recent advances in frame-based 4D GS [16, 59], event-based 4D GS would be a relevant future direction.

7. Conclusion

We propose the first framework for event-based Gaussian Splatting that fully leverages the spatio-temporal properties of event data. Our method targets the majority of modern event cameras, which are monochrome (i.e., grayscale) and with VGA or higher (1 megapixel) resolution. The rendering pipeline consists of two explicit pathways: spatially sparse and temporally dense (i.e., event-by-event) pathway for geometry (depth) recovery, and spatially dense and temporally sparse (i.e., a snapshot) pathway for appearance (radiance) estimation. A thorough evaluation reveals that the proposed method (i) achieves state-of-the-art performance on real-world data without using extra priors, and (ii) effectively tackles the trade-off revolved around the choice of the number of events to process.

Acknowledgments

We thank Mr. Yura Toshiya and Dr. Hidenobu Matsuki for useful discussions. This work was partially supported by JST-Research and Development Program for Next-generation Edge AI Semiconductors Grant Number JPMJES2513. This work has been partially supported by the German Federal Ministry of Research, Technology and Space (BMFTR) under the Robotics Institute Germany (RIG). Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2002/1 "Science of Intelligence" – project number 390523135.

References

[1] Guillermo Gallego, Tobi Delbruck, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew Davison, Jörg Conradt, Kostas Daniilidis, and Davide Scaramuzza, "Event-based vision: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 154–180, 2022.

[2] Hadi AliAkbarpour, Ahmad Moori, Javad Khorramdel, Erik Blasch, and Omar Tahri, "Emerging trends and applications of neuromorphic dynamic vision sensors: A survey," *IEEE Sensors Reviews*, vol. 1, p. 14–63, 2024.

[3] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck, "A 128×128 120 dB 15 μs latency asynchronous temporal contrast vision sensor," *IEEE J. Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, 2008.

[4] Christoph Posch, Teresa Serrano-Gotarredona, Bernabe Linares-Barranco, and Tobi Delbruck, "Retinomorphic event-based vision sensors: Bioinspired cameras with spiking output," *Proc. IEEE*, vol. 102, pp. 1470–1484, Oct. 2014.

[5] Thomas Finateu, Atsumi Niwa, Daniel Matolin, Koya Tsuchimoto, Andrea Mascheroni, Etienne Reynaud, Pooria Mostafalu, Frederick Brady, Ludovic Chotard, Florian LeGoff, Hirotsugu Takahashi, Hayato Wakabayashi, Yusuke Oike, and Christoph Posch, "A 1280x720 back-illuminated stacked temporal contrast event-based vision sensor with 4.86μm pixels, 1.066Geps readout, programmable event-rate controller and compressive data-formatting pipeline," in *IEEE Int. Solid-State Circuits Conf. (ISSCC)*, pp. 112–114, 2020.

[6] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis, "3D Gaussian splatting for real-time radiance field rendering," *ACM Trans. Graph.*, vol. 42, July 2023.

[7] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng, "NeRF: representing scenes as neural radiance fields for view synthesis," *Commun. ACM*, vol. 65, p. 99–106, Dec. 2021.

[8] Toshiya Yura, Ashkan Mirzaei, and Igor Gilitschenski, "EventSplat: 3D Gaussian splatting from moving event cameras for real-time rendering," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 26876–26886, 2025.

[9] Jian Huang, Chengrui Dong, Xuanhua Chen, and Peidong Liu, "IncEventGS: Pose-free Gaussian splatting from a single event camera," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 26933–26942, 2025.

[10] Viktor Rudnev, Mohamed Elgharib, Christian Theobalt, and Vladislav Golyanik, "EventNeRF: Neural radiance fields from a single colour event camera," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 4992–5002, 2023.

[11] Tianyi Xiong, Jiayi Wu, Botao He, Cornelia Fermuller, Yiannis Aloimonos, Heng Huang, and Christopher A Metzler, "Event3DGS: Event-based 3D Gaussian splatting for high-speed robot egomotion," *Conf. on Robot Learning (CoRL)*, 2024.

[12] Guillermo Gallego, Henri Rebecq, and Davide Scaramuzza, "A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 3867–3876, 2018.

[13] Johannes Lutz Schönberger and Jan-Michael Frahm, "Structure-from-motion revisited," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2016.

[14] Weng Fei Low and Gim Hee Lee, "Robust e-NeRF: NeRF from sparse & noisy events under non-uniform motion," in *Int. Conf. Comput. Vis. (ICCV)*, pp. 18289–18300, 2023.

[15] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao, "2D Gaussian splatting for geometrically accurate radiance fields," in *SIGGRAPH*, pp. 1–11, 2024.

[16] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang, "4D Gaussian splatting for real-time dynamic scene rendering," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 20310–20320, 2024.

[17] Jonas Kulhanek, Songyou Peng, Zuzana Kukelova, Marc Pollefeys, and Torsten Sattler, "WildGaussians: 3D Gaussian splatting in the wild," *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2024.

[18] Nicolas Moenne-Loccoz, Ashkan Mirzaei, Or Perel, Riccardo de Lutio, Janick Martinez Esturo, Gavriel State, Sanja Fidler, Nicholas Sharp, and Zan Gojcic, "3D Gaussian ray tracing: Fast tracing of particle scenes," *ACM Trans. Graph.*, vol. 43, no. 6, pp. 1–19, 2024.

[19] Qi Wu, Janick Martinez Esturo, Ashkan Mirzaei, Nicolas Moenne-Loccoz, and Zan Gojcic, "3DGUT: Enabling distorted cameras and secondary rays in Gaussian splatting," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 26036–26046, 2025.

[20] Hiroyuki Deguchi, Mana Masuda, Takuya Nakabayashi, and Hideo Saito, "E2GS: Event enhanced Gaussian splatting," in *IEEE Int. Conf. Image Process. (ICIP)*, pp. 1676–1682, 2024.

[21] Marco Cannici and Davide Scaramuzza, "Mitigating motion blur in neural radiance fields with events and frames," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2024.

[22] Junhao He, Jiaxu Wang, Jia Li, Mingyuan Sun, Qiang Zhang, Jiahang Cao, Ziyi Zhang, Yi Gu, Jingkai Sun, and Renjing Xu, "DEGS: Deformable event-based 3D Gaussian splatting from RGB and event stream," *IEEE Trans. Vis. Comput. Graphics*, vol. 32, no. 2, pp. 1698–1712, 2026.

[23] Yufei Deng, Yuanjian Wang, Rong Xiao, Chenwei Tang, Jizhe Zhou, Jiahao Fan, Deng Xiong, Jiancheng Lv,

and Huajin Tang, "EBAD-Gaussian: Event-driven bundle adjusted deblur Gaussian splatting," *arXiv preprint arXiv:2504.10012*, 2025.

[24] Inwoo Hwang, Junho Kim, and Young Min Kim, "Ev-NeRF: Event based neural radiance field," in *IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, pp. 837–847, 2023.

[25] Jiaxu Wang, Junhao He, Ziyi Zhang, and Renjing Xu, "Physical priors augmented event-based 3D reconstruction," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, pp. 16810–16817, 2024.

[26] Yuanjian Wang, Yufei Deng, Rong Xiao, Jiahao Fan, Chenwei Tang, Deng Xiong, and Jiancheng Lv, "SaENeRF: Suppressing artifacts in event-based neural radiance fields," in *Int. Joint Conf. Neural Netw. (IJCNN)*, 2025.

[27] Zixin Zhang, Kanghao Chen, and Lin Wang, "Elite-EvGS: Learning event-based 3D Gaussian splatting by distilling event-to-video priors," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, pp. 13972–13978, 2025.

[28] Sohaib Zahid, Viktor Rudnev, Eddy Ilg, and Vladislav Golyanik, "E-3DGS: Event-based novel view rendering of large-scale scenes using 3D Gaussian splatting," *Int. Conf. 3D Vision (3DV)*, 2025.

[29] Tao Liu, Runze Yuan, Yi'ang Ju, Xun Xu, Jiaqi Yang, Xiangting Meng, Xavier Lagorce, and Laurent Kneip, "GS-EVT: Cross-modal event camera tracking based on Gaussian splatting," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, pp. 4587–4593, 2025.

[30] Simon Klenk, Lukas Koestler, Davide Scaramuzza, and Daniel Cremers, "E-NeRF: Neural radiance fields from a moving event camera," *IEEE Robot. Autom. Lett.*, vol. 8, no. 3, pp. 1587–1594, 2023.

[31] Chaoran Feng, Wangbo Yu, Xinhua Cheng, Zhenyu Tang, Junwu Zhang, Li Yuan, and Yonghong Tian, "AE-NeRF: Augmenting event-based neural radiance fields for non-ideal conditions and larger scenes," *AAAI Conf. Artificial Intell.*, vol. 39, pp. 2924–2932, Apr. 2025.

[32] Hanqian Han, Jianing Li, Henglu Wei, and Xiangyang Ji, "Event-3DGS: Event-based 3D reconstruction using 3D Gaussian splatting," *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 37, pp. 128139–128159, 2024.

[33] Xiaoting Yin, Hao Shi, Yuhan Bao, Zhenshan Bing, Yiyi Liao, Kailun Yang, and Kaiwei Wang, "E-3DGS: 3D Gaussian splatting with exposure and motion events," *Applied Optics*, vol. 64, no. 14, pp. 3897–3908, 2025.

[34] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza, "High speed and high dynamic range video with an event camera," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 6, pp. 1964–1980, 2021.

[35] Bohao Liao, Wei Zhai, Zengyu Wan, Zhixin Cheng, Wenfei Yang, Tianzhu Zhang, Yang Cao, and Zheng-Jun Zha, "EF-3DGS: Event-aided free-trajectory 3D Gaussian splatting," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2025.

[36] David Weikersdorfer, David B. Adrian, Daniel Cremers, and Jörg Conradt, "Event-based 3D SLAM with a depth-augmented dynamic vision sensor," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, pp. 359–364, 2014.

[37] Emanuele Trucco and Alessandro Verri, *Introductory Techniques for 3-D Computer Vision*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1998.

[38] Guillermo Gallego and Davide Scaramuzza, "Accurate angular velocity estimation with an event camera," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 632–639, 2017.

[39] Haram Kim and H. Jin Kim, "Real-time rotational motion estimation with contrast maximization over globally aligned events," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 6016–6023, 2021.

[40] Shintaro Shiba, Yoshimitsu Aoki, and Guillermo Gallego, "A fast geometric regularizer to mitigate event collapse in the contrast maximization framework," *Adv. Intell. Syst.*, p. 2200251, 2022.

[41] Xin Peng, Ling Gao, Yifu Wang, and Laurent Kneip, "Globally-optimal contrast maximisation for event cameras," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3479–3495, 2022.

[42] Cheng Gu, Erik Learned-Miller, Daniel Sheldon, Guillermo Gallego, and Pia Bideau, "The spatio-temporal Poisson point process: A simple model for the alignment of event camera data," in *Int. Conf. Comput. Vis. (ICCV)*, pp. 13495–13504, 2021.

[43] Shuang Guo and Guillermo Gallego, "CMax-SLAM: Event-based rotational-motion bundle adjustment and SLAM system using contrast maximization," *IEEE Trans. Robot.*, vol. 40, pp. 2442–2461, 2024.

[44] Shuang Guo and Guillermo Gallego, "Event-based photometric bundle adjustment," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 47, no. 10, pp. 9280–9297, 2025.

[45] Shuang Guo, Friedhelm Hamann, and Guillermo Gallego, "Unsupervised joint learning of optical flow and intensity with event cameras," in *Int. Conf. Comput. Vis. (ICCV)*, pp. 7980–7989, 2025.

[46] Ryo Yamaki, Shintaro Shiba, Gallego Guillermo, and Yoshimitsu Aoki, "Iterative event-based motion segmentation by variational contrast maximization," in *IEEE Conf. Comput. Vis. Pattern Recog. Workshops (CVPRW)*, pp. 4918–4927, 2025.

[47] Shintaro Shiba, Yoshimitsu Aoki, and Guillermo Gallego, "Simultaneous motion and noise estimation with event cameras," in *Int. Conf. Comput. Vis. (ICCV)*, pp. 6959–6969, 2025.

[48] Shintaro Shiba, Yoshimitsu Aoki, and Guillermo Gallego, "Secrets of event-based optical flow," in *Eur. Conf. Comput. Vis. (ECCV)*, pp. 628–645, 2022.

[49] Shintaro Shiba, Yannick Klose, Yoshimitsu Aoki, and Guillermo Gallego, "Secrets of event-based optical flow, depth, and ego-motion by contrast maximization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 12, pp. 7742–7759, 2024.

[50] Javier Hidalgo-Carrió, Guillermo Gallego, and Davide Scaramuzza, "Event-aided direct sparse odometry," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 5781–5790, June 2022.

[51] Simon Klenk, Jason Chui, Nikolaus Demmel, and Daniel Cremers, "TUM-VIE: The TUM stereo visual-inertial event dataset," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, pp. 8601–8608, 2021.

[52] Zelin Zhang, Anthony Yezzi, and Guillermo Gallego, "Formulating event-based image reconstruction as a linear inverse problem with deep regularization using optical flow,"

IEEE Trans. Pattern Anal. Mach. Intell., vol. 45, no. 7, pp. 8372–8389, 2023.

[53] Samuel Bryner, Guillermo Gallego, Henri Rebecq, and Davide Scaramuzza, "Event-based, direct camera tracking from a photometric 3D map using nonlinear optimization," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, pp. 325–331, 2019.

[54] Federico Paredes-Vallés and Guido C. H. E. de Croon, "Back to event basics: Self-supervised learning of image reconstruction for event cameras via photometric constancy," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 3445–3454, 2021.

[55] Shintaro Shiba, Friedhelm Hamann, Yoshimitsu Aoki, and Guillermo Gallego, "Event-based background oriented schlieren," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 4, pp. 2011–2026, 2024.

[56] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, pp. 600–612, Apr. 2004.

[57] Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza, "ESIM: an open event camera simulator," in *Conf. on Robot Learning (CoRL)*, vol. 87 of *Proc. Machine Learning Research*, pp. 969–982, PMLR, 2018.

[58] Henri Rebecq, Guillermo Gallego, Elias Mueggler, and Davide Scaramuzza, "EMVS: Event-based multi-view stereo—3D reconstruction with an event camera in real-time," *Int. J. Comput. Vis.*, vol. 126, pp. 1394–1414, Dec. 2018.

[59] Minghao Yin, Yukang Cao, Songyou Peng, and Kai Han, "Splat4D: Diffusion-enhanced 4D Gaussian splatting for temporally and spatially consistent content creation," in *SIGGRAPH*, pp. 1–10, 2025.

[60] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny, "VGGT: Visual Geometry Grounded Transformer," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 5294–5306, 2025.