Event3DGS: Event-based 3D Gaussian Splatting for Real-Time Radiance Field Rendering

TIANYI XIONG*, SHENGJIE XU*, JIAYI WU*, and CHRISTOPHER METZLER, University of Maryland, USA

Novel-view synthesis from "clean" images has achieved high fidelity through the utilization of Neural Radiance Fields (NeRF). The recent adoption of 3D Gaussian splatting (3DGS) leverages the advantage of explicit point-based representation to significantly improve the rendering speed and quality. However, swift egomotion in real-world robotic tasks induces motion blurs in input images, leading to inaccuracies and artifacts in reconstructed structure. To alleviate this issue, we propose Event3DGS, the first methodology that learns Gaussian Splatting from event-camera data. By exploiting the high temporal resolution of event cameras and explicit point-based representation, Event3DGS can reconstruct high-fidelity 3D structures solely from the event streams of fast-moving cameras. Then, our negative sampling and progressive training approaches allow for better reconstruction quality and consistency. To further optimize the appearance fidelity of the rendered scene, we explicitly model the motion blur formation process into a differentiable rasterizer and jointly use a limited number of blurred RGB images captured under high-speed egomotion alongside the corresponding event stream for appearance refinement. Experimental evaluations on multiple datasets showcase that Event3DGS achieves superior rendering quality and significantly improve the rendering speed compared with existing approaches.

CCS Concepts: • **Do Not Use This Code → Generate the Correct Terms for Your Paper**; *Generate the Correct Terms for Your Paper*; Generate the Correct Terms for Your Paper; Generate the Correct Terms for Your Paper.

Additional Key Words and Phrases: 3D Gaussian Splatting, Novel-View Synthetis, Event-based Reconstruction

ACM Reference Format:
Tianyi Xiong, Shengjie Xu, Jiayi Wu, and Christopher Metzler. 2018. Event3DGS: Event-based 3D Gaussian Splatting for Real-Time Radiance Field Rendering. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX).* ACM, New York, NY, USA, Article 111, 9 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Reconstructing a precise geometric and visually realistic 3D scene representation from a collection of 2D images taken from different viewpoints has long been a persistent challenge in the fields of computer vision and computer graphics. In recent developments, significant strides have been made in advancing this endeavor through two notable contributions: Neural Radiance Fields (NeRF) [Mildenhall et al. 2020] and 3D Gaussian Splatting (3DGS) [Kerbl et al. 2023]. Both categories of methods represent scenes as differentiable, unstructured 3D representations, enabling the rendering of new views that frequently exhibit visual fidelity comparable to evaluation images. However, a major prerequisite to achieve a high-quality radiance field based on them is that they generally require a appropriate distribution of viewpoints and high-quality sharp static input images, which are not always met in real-world robotic tasks. Within the practical implementation of robotic systems, rapid egomotion emerges as a prevalent phenomenon, notably observed in contexts characterized by high-velocity locomotion or traversal of dynamic environments. The requisite speed and agility integral to proficient robot locomotion invariably engender pronounced instances of egomotion, thereby imparting considerable challenges to diverse computational undertakings. Of particular significance, the efficacy of radiance field rendering is substantially influenced by the manifestations of swift egomotion, the dynamics of rapid egomotion can induce motion blur artifacts within rendered images, thereby compromising visual fidelity and realism, hindering the radiance field rendering methods' (such as NeRF and 3DGS) practical applicability in real-world scenes. Although recent studies [Ma et al. 2022; Oh et al. 2024; Seiskari et al. 2024; Wang et al. 2023; Wenbo and Ligang 2024; Zhao et al. 2024] have demonstrated promising advancements in reconstructing radiance fields from motion-blurred images by acquiring the capability to infer camera motion during the exposure period, they are inherently constrained by the presence of motion ambiguities and the inevitable loss of sharp geometry details, which remain unrecoverable solely from the blurry image data.

While NeRFs trained with clean RGB images achieve promising performance on 3D reconstruction and novel-view synthesis, their performance are strongly degraded with blurry input. The event camera, a novel sensing paradigm, offers several advantages over conventional frame-based cameras, particularly in scenarios characterized by fast egomotion. Diverging from conventional cameras, event cameras asynchronously record pixel-level luminance alterations, facilitating exceptional temporal resolution, minimal motion blur, high dynamic range, and negligible latency and data bandwidth. Such attributes empower event cameras to furnish a continuous flow of pixel-level intensity variations and precise and sharp scene geometry information, even in fast egomotion scenario [Hu et al. 2021]. Within radiance field rendering, the inherent capability of event cameras to precisely capture scene information at high temporal resolutions seamlessly aligns with the demands posed by radiance field rendering in fast egomotion scenarios. Employing a continuous event stream, characterized by sharp scene geometry, as a supervisory signal for radiance field rendering, holds promise in facilitating the generation of coherent and artifact-free renderings amidst rapid egomotion situations. As event cameras

*Both authors contributed equally to this research.

Authors' address: Tianyi Xiong, txiong23@umd.edu; Shengjie Xu, sjxu@umd.edu; Jiayi Wu, jiayiwu@umd.edu; Christopher Metzler, metzler@umd.edu, University of Maryland, 8125 Paint Branch Drive, College Park, USA.

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

exhibit strong capacity in capturing fast motion with low latency and data bandwidth, several works utilize event streams for training NeRFs for higher reconstruction quality. EventNeRF[Rudnev et al. 2023], EV-NeRF[Hwang et al. 2023a], and E-NeRF[Klenk et al. 2023a] extended the idea of NeRF onto the event camera by exploring static, quasi-dynamic, and moving scenes separately. These three papers propose nuanced adaptations to the NeRF architecture, tailored specifically to accommodate the sparse and asynchronous nature inherent in EC data streams. Such adaptations encompass techniques such as event encoding, temporal integration, and the formulation of bespoke loss functions. Divergences among the approaches manifest in several dimensions. For instance, EventNeRF and Ev-NeRF operationalize temporal integration to frame-like representations, while E-NeRF[Klenk et al. 2023a] opt for direct integration of event data. Moreover, methodological variances are observed in the treatment of event polarity and the architectural configurations, with distinctions particularly pronounced in the encoding networks utilized. The thematic focal points of the respective studies diverge as well, with EventNeRF[Rudnev et al. 2023]emphasizing static scene reconstruction, Ev-NeRF[Hwang et al. 2023a] prioritizing real-time rendering and dynamic scene comprehension, and E-NeRF[Klenk et al. 2023a] encompassing both static and dynamic scene reconstruction while emphasizing memory efficiency.

We're excited to implement this similar idea to the domain of a new 3D scene reconstruction method 3D Gaussian Splatting. The emerging 3D Gaussian Splatting[Kerbl et al. 2023] (3D-GS) significantly improves the rendering speed to a real-time level by explicitly modeling the scene as 3D Gaussians. During inference, 3D Gaussians are rendered into camera views via splatting-based rasterization.

In this work, we propose Event3DGS, the first methodology that leverages the advantages of gaussian splatting to facilitate high-fidelity 3D reconstruction and real time rendering based solely on event data. Event3DGS is trained from events in a self-supervised manner by comparing the approximate difference between views computed by accumulate the observed events polarities against the difference between the rendering views. Our findings demonstrate the feasibility of rendering the accurate geometry of the scene solely from an event stream input by 3DGS. Notably, in order to further optimize the appearance fidelity of the rendered scene, we explicitly model the motion blur formation process into a differentiable rasterizer, and jointly use a limited number of blurred RGB images captured under high-speed egomotion and the event stream within the corresponding exposure time as supervision signals for appearance refinement. Our results show that our optimization strategy makes targeted use of information in event and blurred images, which makes our approach facilitates the explicit 3D representation recovery of scenes during rapid egomotion. Through the fusion of events and sparsely collected blur images, Event3DGS enables comparable and often more visually appealing rendering quality than prevailing pipeline approaches, while achieving faster rendering speed at lower data bandwidth and memory footprint. Our contributions can be summarized as follows:

(1) To the best of our knowledge, this is the first work to produce explicit 3D Gaussian splatting scene representations solely from event data.
(2) With a methodological approach incorporating negative sampling and differential supervision of event data, specifically tailored to accommodate 3D Gaussian Splatting (3DGS), we achieves accurate 3D geometry reconstruction and real-time rendering functionalities.
(3) To enhance appearance fidelity in the rendered scene, we integrate the motion blur formation process into a differentiable rasterizer. Through use a limited number of blurred RGB images and the corresponding event stream as joint supervision, Event3DGS enables optional appearance refinement to further get more visually appealing rendering quality.

2 RELATED WORK

2.1 Novel View Synthesis and 3D Gaussian Splatting

In computer graphics and computer vision, dense photorealistic rendering of a scene in a 3D-consistent manner and novel view synthesis is a critical task involving generating images of a scene from viewpoints that were not observed during data acquisition. This process finds applications in various domains, including virtual reality, robotics, autonomous driving, and augmented reality. While effective, traditional methods such as ray casting and ray marching often suffer from computational inefficiency, particularly when confronted with large-scale datasets. Recent advancements in novel view synthesis have witnessed the emergence of Neural Radiance Fields (NeRF) [Mildenhall et al. 2020], a groundbreaking technique introduced by Mildenhall et al. NeRF [Mildenhall et al. 2020] represents a neural network-based approach to scene representation and rendering, wherein a volumetric scene is modeled as a continuous function that maps 3D spatial coordinates to radiance values. By leveraging neural networks to approximate this function, NeRF [Mildenhall et al. 2020] achieves photorealistic rendering results with high fidelity and fine details, surpassing traditional rendering techniques in realism and accuracy.

However, while NeRF [Mildenhall et al. 2020] has demonstrated remarkable capabilities in synthesizing novel views of scenes, it poses challenges in terms of computational complexity and scalability, particularly for large-scale scenes with intricate geometry and texture details. Moreover, NeRF [Mildenhall et al. 2020] requires substantial amounts of training data and computational resources, limiting its applicability in real-time or interactive scenarios. In light of these challenges, recent research has explored alternative approaches to scene representation and rendering that offer a balance between efficiency and visual fidelity. Kerbl et al. introduced 3D Gaussian splatting (3DGS) [Kerbl et al. 2023], a technique with roots in computer graphics dating back to the seminal work of Westover [Westover 1991]. 3DGS involves projecting volumetric data onto a 2D image plane using Gaussian distributions to emulate smooth transitions and preserve essential details. While traditionally used in volume rendering, 3DGS has garnered renewed interest in the context of scene representation and rendering, particularly as a complement to NeRF-based methods. Unlike NeRF [Mildenhall et al. 2020], which operates on implicit 3D representations, Gaussian splatting offers an explicit representation of the scene, enabling efficient rendering and synthesis of novel views without the need for complex neural network architectures or extensive training data.

3DGS emerges as a promising alternative, offering a principled approach to project volumetric data onto a 2D image plane. Central to its methodology is the projection of the learnable 3D Gaussian onto a 2D image plane, facilitated by the application of Gaussian distributions to emulate smooth transitions and preserve salient details. This process entails the transformation of voxel points onto the image plane, the generation of Gaussian functions centered at these points, and the cumulative aggregation of their contributions to yield the final rendered image.

2.2 Event-based 3D Reconstruction and Radiance Field Rendering

Event-based and event-aided 3D reconstruction [Baudron et al. 2020; Chamorro et al. 2022; Muglikar et al. 2021; Wang et al. 2022; Xiao et al. 2022; Zhu et al. 2019] and radiance field rendering [Bhattacharya et al. 2024; Cannici and Scaramuzza 2024; Hwang et al. 2023b; Ma et al. 2023; Qi et al. 2023; Rudnev et al. 2023] represent a paradigm shift in computer vision and graphics, revolutionizing the perception and rendering of dynamic scenes with high temporal resolution and accuracy. Traditional frame-based imaging systems capture scenes at fixed intervals, resulting in motion blur and latency issues, particularly in fast-moving scenarios. Event-based sensors, inspired by biological vision systems, detect changes in luminance (events) asynchronously and with microsecond precision at the pixel level. This asynchronous operation enables event-based systems to capture fast-moving objects with minimal motion blur and latency, making them ideal for applications requiring real-time perception and interaction, such as robotics, augmented reality, and autonomous driving.

Event-based 3D reconstruction leverages the spatiotemporal information provided by event data to reconstruct the 3D geometry and appearance of dynamic scenes with unprecedented speed and accuracy. By exploiting the temporal resolution of event data, these techniques enable the reconstruction of scenes with fine temporal detail, surpassing the capabilities of traditional frame-based methods. Event-based sensors offer advantages in low-light conditions and high dynamic range environments, further enhancing their applicability across various domains. Moreover, Event-aided approaches combine event data with traditional frame-based imaging to enhance the robustness and accuracy of 3D reconstruction and radiance field rendering, particularly in challenging scenarios with dynamic lighting conditions or occlusions. Weikersdorfer et al. [Weikersdorfer et al. 2013] demonstrated event-based stereo reconstruction, showcasing the feasibility of reconstructing 3D scenes from event data captured by stereo event cameras. Advancements in event-based radiance field rendering have expanded the capabilities of event-based systems beyond reconstruction to include denser and visually captivating rendering of real world scenes. [Hwang et al. 2023b] proposed an event-based neural radiance field (Ev-NeRF) framework that capitalizes on the multi-view consistency inherent in NeRF, offering a robust self-supervision signal for mitigating erroneous measurements and extracting coherent underlying structures from noisy raw event data. This methodology yields a cohesive 3D structure capable of delivering high-fidelity observations. Rudnev et al. [Rudnev et al. 2023] introduced an approach (EventNeRF) for 3D-consistent, dense and photorealistic novel view synthesis using just a single colour event stream as input. It was trained exclusively in a self-supervised fashion using event data while maintaining the original resolution of the color event channels. Their evaluations shows that their approach not only provides denser and visually captivating renderings compared to existing methods but also demonstrates robustness in demanding scenarios characterized by rapid motion and low-light conditions.

Furthermore, some existing works [Cannici and Scaramuzza 2024; Qi et al. 2023] enhance pipeline performance in fast motion and low-light scenes by integrating event streams into RGB-based radiative field rendering pipeline frameworks (NeRF or 3DGS) for deblurring. For example, Qi et al. [Qi et al. 2023] introduced a novel approach termed Event-Enhanced NeRF (E2NeRF), integrating event streams into the learning framework of neural volumetric representation. Their methodology incorporates a blur rendering loss and an event rendering loss, aimed at guiding the network by modeling real blur processes and event generation processes, respectively. Their evaluation substantiates that their methodology effectively leverages the intrinsic association between events and images, yielding superior quality compared to prior image-based or event-based NeRF approaches in complex and low-light scenes. Cannici et al. [Cannici and Scaramuzza 2024] achieve surpassing existing deblurring NeRFs by explicitly modeling the blur formation process and utilizing event double integrals as additional model-based priors.

While the mentioned approaches successfully incorporates the NeRF formulation with raw event data, resulting in 3D rendering of exceptional visual fidelity, NeRF's algorithmic framework demands substantial computational resources, rendering it less viable for real-time applications. This limitation poses a challenge for lightweight and real-time efficient event camera systems. Furthermore, the implicit representation of the model makes it difficult to edit and integrate with traditional 3D graphics processing pipelines.

However, our proposed Event3DGS pipeline integrates 3D Gaussian Splatting (3DGS) [Kerbl et al. 2023] with real-time rendering capabilities and asynchronous event stream, providing an efficient, deterministic and interpretable depiction of scene geometry, and easy-to-edit high-fidelity 3D rendering capability. It allows seamless integration with established graphics pipelines and enabling streamlined optimization processes. In addition, Event3DGS is extremely robust to fast motion, low light, and high dynamic range scenarios where it is difficult for RGB cameras to obtain high-quality images. By fusing the hardware advantages of the event camera with the efficient rendering capabilities of 3DGS, our pipeline provides real-time 3D rendering of a wider range of real-world scenes with low latency, low data bandwidth, ultra-low power consumption and allows acquisition devices to perform 3D mapping work at a higher operating speed.

3 METHODS

3.1 Preliminary

3D Gaussian Splatting (3DGS) [Kerbl et al. 2023] explicitly represents a scene with a group of anisotropic 3D Gaussians (ellipsoids). Each

Gausssian is defined by a full 3D covariance matrix Σ with its center point (mean) μ:

$$G(x) = e^{-\frac{1}{2}x^T\Sigma^{-1}x} \tag{1}$$

To preserve the valid positive semi-definite property during optimization, the covariance matrix is decomposed into $\Sigma = RSS^TR^T$, where $S \in \mathbb{R}_+^{\not\in}$ represents scaling factors and $R \in SE(3)$ is the rotation matrix. Each Gaussian is also described with an opacity factor$\sigma \in \mathbb{R}$, and spherical harmonics $C \in \mathbb{R}^k$ for modeling view-dependent effects.

During optimization, 3D Gaussian splatting adaptively control the density of Gaussians via densification in areas with large view-space positional gradients and pruning points with low opacity. For rendering, the 3D Gaussians $G(x)$ are first projected onto the 2D imaging plane $G'(x)$, then a tile-based rasterizer is proposed to enable fast sorting and α-blending. The color of pixel u is calculated via blending N ordered overlapping points:

$$C(u) = \sum_{i=1}^{N} c_i\alpha_i \prod_{j=1}^{i-1}(1-\alpha_j) \tag{2}$$

where $c_i = f(C_i)$ is the color modeled via spherical harmonics, and $\alpha_i = \sigma_i G_i'(u)$ is the multiplication of opacity and the transformed 2D Gaussian.

Event cameras record a continuous stream of event tuples $\{e_k = (t_k, u_k, p_k)\}_{k=1}^N$, where t is the timestamp, $u_k = (x_k, y_k)$ is the pixel coordinates and $p_k \in \{+1, -1\}$ is the polarity of each events.

3.2 Neutralization Minimization using Adaptive Integral Bins

In event-based radiance field rendering pipelines, the slicing strategy of the event stream affects rendering quality of the scene. This effect is more obvious in our pipeline. Since we use polarity during the accumulation process, it is inevitable to lose information due to neutralization during the accumulation process. In order to mitigate the information loss during the accumulation process, we design a neutralization-aware event slicing strategy. Notably, some existing works have stated that using constant short windows leads to poor propagation of high-level illumination, and using constant long windows often leads to poor local detail [Rudnev et al. 2023]. Our slicing strategy considers the number of events and the neutralization moment to adaptively sample the length of the event integration window. (1) perform slicing when the number of events reaches the threshold; (2) perform slicing where neutralization occurs on many pixels (set threshold manually). Our slicing strategy not only ensures the diversity of window lengths but also minimizes the loss of detailed information caused by neutralization.

3.3 Parameters Separable Alternating Optimization

Event data with high temporal resolution is able to provide differential supervision signals with sharp scene structure, which allows 3D gaussian splatting (3DGS) to perform fine reconstruction of scene structure under fast egomotion. The multi-view consistency of 3DGS provides a powerful self-supervision signal, which enables the learnable gaussians to continuously approximate the ground truth of geometric structure of the scene during the optimization process. However, time-differenced event data lacks some appearance components the scene, such as the absolute value of pixel intensity, true tone, local texture, etc. Therefore, 3D scene representations optimized only through event data often suffer from tone shifts and lack texture details.

Although RGB images with severe motion blur are difficult to be directly used for radiance field training due to the degradation of the structure of the scene, the true hue and texture information contained in them become complementary to the event data. In order to further improve the fidelity of the appearance and maintain the sharp details of the scene structure, we sparsely insert a very small number of blur RGB images as supervision signals of directional appearance and opacity, and design an parameters separable alternating optimization strategy on the combined data of the event stream and the blur RGB images. In addition, we explicitly incorporate the formation process of motion blur distortion into the rasterization process (see Sec. 3.4) by integrating information over a short camera trajectory (during the exposure interval). This mitigates the undesirable effects of appearance degradation in blurry images on supervision.

Separable setting of parameters in alternating optimization is key to our method maintaining sharp structural details of the scene while improving the fidelity of appearance components from joint supervision of blur images and event data. We divide the learnable parameters into two groups. The structural parameters include the position (mean) μ and covariance matrix Σ of 3D Gaussians, and the appearance parameters include opacity α and spherical harmonics (SH) coefficients. When high temporal resolution differential signals from event data serve as supervision, we perform gradient descent on the structural parameters of 3D Gaussians to approximate the sharp structure of the scene, while the coarse appearance parameters (supervised only by event data) are frozen or given an extremely low learning rate. In contrast, when the integrated signal from the blur RGB image is added to the supervised iteration, we only use this signal to perform gradient descent on the refined appearance parameters of the 3D Gaussians (which are additionally cached and not shared with the coarse appearance parameters) to refine the appearance components of the scene. Synchronously, the structural parameters are optimized under the supervision of the aligned event differential signal.

In order to ensure efficient optimization and rapid convergence of the model, we carefully determined the timing of intervention of alternating optimization in the overall training process. We find that because the structural parameters of 3D Gaussians were optimized at a higher learning rate under the supervision of event data in the early iterations of training, and a small amount of blur RGB supervision signals were intermittently called during the optimization process. Therefore, the refined appearance parameters do not converge quickly in the early iterations of training but continue to oscillate. Based on this observation, we allow supervision of event data in the early iterations of training to perform gradient descent on all parameters to increase convergence speed and avoid redundancy. After the structural parameters initially converge (the corresponding learning rate stabilizes at a lower value for a certain number of iterations), the blur RGB supervision signal will be

intermittently called in subsequent iterations to perform gradient descent on the refined appearance parameters.

3.4 Differentiable Blur-aware Rasterization

In the case of high-speed egomotion, images captured by RGB cameras often contain severe motion blur, so it is difficult to provide multi-view consistent appearance supervision for training under the original rasterization method of 3DGS (fixed camera pose). We aim to optimize appearance parameters of the learnable 3D Gaussians using a given small amount of motion blurred inputs, which are able to improve visual fidelity while maintaining sharp scene structure. In the realm of physics, camera motion blur stems from the amalgamation of irradiance induced by the movement of the camera. According to the physical image formation, camera motion blur is produced by the integration of irradiance during camera movement, which can be mathematically represented as following equation:

$$\mathbf{I}_{blur} = \int_{\tau_s}^{\tau_e} \mathbf{I}(\mathbf{P}_\tau)\, d\tau \approx \frac{1}{N} \sum_{i=1}^{N} \mathbf{I}(\mathbf{P}_{\tau_i}) \tag{3}$$

where $\mathbf{I}_{blur}$ represents blurry image, $\mathbf{I}(\mathbf{P}_\tau)$ is latent sharp image captured from the camera pose $\mathbf{P}_\tau \in SE(3)$. To simplify the integral calculation, we approximate it as a finite sum of N irradiance $\mathbf{I}(\mathbf{P}_{\tau_i})$, where τ_i are the midpoint timestamps of a finite number of event integration windows (EIW) within the exposure interval (from τ_s to τ_e).

In order to incorporate motion effects due to camera movement during frame capture modeling into the differentiable rasterization process of the 3DGS pipeline, we incorporate the above physical formation process of motion blur into the rendering equation:

$$\widetilde{C}_{blur}(x, y, \mathbf{P}_{\frac{\tau_s+\tau_e}{2}}, \mathcal{G}) = g(\frac{1}{N_{EIW}} \sum_{i=1}^{N_{EIW}} \widetilde{C}(x, y, \mathbf{P}_{\tau_i}, \mathcal{G})) \tag{4}$$

where $\widetilde{C}_{blur}$ denotes the blurry color of the pixel(x,y) of output image given by blur-aware volumetric rendering, $\mathcal{G}$ is the 3D Gaussian model parameters, $g(\cdot)$ is a gamma correction function, $g(R, G, B) = (R^{\frac{1}{\gamma}}, G^{\frac{1}{\gamma}}, B^{\frac{1}{\gamma}})$ with $\gamma = 2.2$, N_{EIW} represents the number of event integration windows within the exposure interval.

3.5 Loss Function

Following [Kerbl et al. 2023], We first calculate the log illuminance different between two timestamps t and t_0, and compute the L1 loss with the accumulated event maps.

$$\mathcal{L}_u(t_0, t) = \text{L1\_loss}\ (\log I_u(t) - \log I_u(t_0), E_u(t_0, t)) \tag{5}$$

Similar to [Klenk et al. 2023b; Rudnev et al. 2023], to improve overall reconstruction consistency and robustness, we also compute loss on negative pixels where no events are triggered. To further improve robustness, apply random dropout with probability p on the computed pixels. The total loss can be written as

$$\mathcal{L}(t_0, t) = \text{Dropout}(\sum_{E_u(t_0,t)\neq 0} L_u(t0, t) + \lambda \sum_{E_u(t_0,t)=0} L_u(t0, t), p) \tag{6}$$

4 EXPERIMENTS

4.1 Experimental Settings

Our implementation is based on 3DGS[Kerbl et al. 2023]. We train our model on a single NVIDIA RTXA4000 GPU for 30k iterations, which takes arounds x minutes. We set the scale of pointcloud initialization from 2.6 to 0.2 for fit the scale of training scenes. Other hyperparmeters and optimizers are set as default.

Synthetic Dataset. [Rudnev et al. 2023] generate seven sequences via rendering a 360° rotation of camera around each 3D object with 1000 views and simulating the event streams accordingly.

4.2 Quantitative Evaluation

Synthetic Sequences. It can be seen from the Table 1 that, our Event3DGS almost provide better performance than the competitors, E2VID + NeRF and EventNeRF[Rudnev et al. 2023].

4.3 Qualitative Evaluation

Synthetic Sequences. Our results shows that (See Fig. 4.3 and Fig. 4.3). Event3DGS enables comparable and often more visually appealing rendering quality than prevailing pipeline approaches, while achieving faster rendering speed at lower data bandwidth and memory footprint.

Real Sequences. Our results shows that (See Fig. 4.3). Event3DGS enables consistent rendering quality in the real world data.

5 NERFSTUDIO IMPLEMENTATION

We leverage the Nerfstudio framework, depicted in Fig. 4, which categorizes NeRF methods into a sequence of fundamental building blocks. We adopt this base template and the corresponding Splatfacto method, a Gaussian Splatting implementation in Nerfstudio, to design our Event3DGS method (https://github.com/jayhsu0627/Event3DGS).

The pipeline comprises two key components: the DataManager and the Model. The DataManager is tasked with (1) parsing image formats via a DataParser and (2) generating RayBundles and RayGT objects, necessary for training and inference. RayBundle objects encapsulate ray origins and viewing directions, while RayGT objects, required solely during training, contain ground truth (GT) information for loss computation. For instance, GT pixel values can supervise rendered rays using an L2 loss. These rays are subsequently fed into the Model's forward pass, which queries Fields and renders quantities as RayOutputs. Ultimately, the entire Pipeline is supervised end-to-end with a loss function.

To avoid conflation with 3DGS [Kerbl et al. 2023] and Nerfstudio's "Splatfacto", [Tancik et al. 2023], we refer to our implementation as "Esplatfacto." Akin to Nerfacto's amalgamation of various techniques, the authors of Nerfstudio posit that Splatfacto will be a blend of different Gaussian Splatting methodologies. Consequently, we designed our method based on this template to capitalize on its potential impact.

5.1 Dataloader

As Nerfstudio stores its images as files and camera poses as `JSON` files, we must convert the event data to the ground truth in Nerfstudio

Table 1. Quantitative comparison (PSNR↑) on synthetic event-sequences

Methods	chair	drums	ficus	hotdog	lego	materials	mic	Average
E2VID + NeRF	24.12	19.71	24.97	24.38	20.17	22.01	23.08	22.64
EventNeRF[Rudnev et al. 2023]	30.62	27.43	31.94	30.26	25.84	24.10	31.78	28.85
Event3DGS (w/o blur images)	30.962	27.645	31.249	30.799	27.7	29.232	31.908	29.928

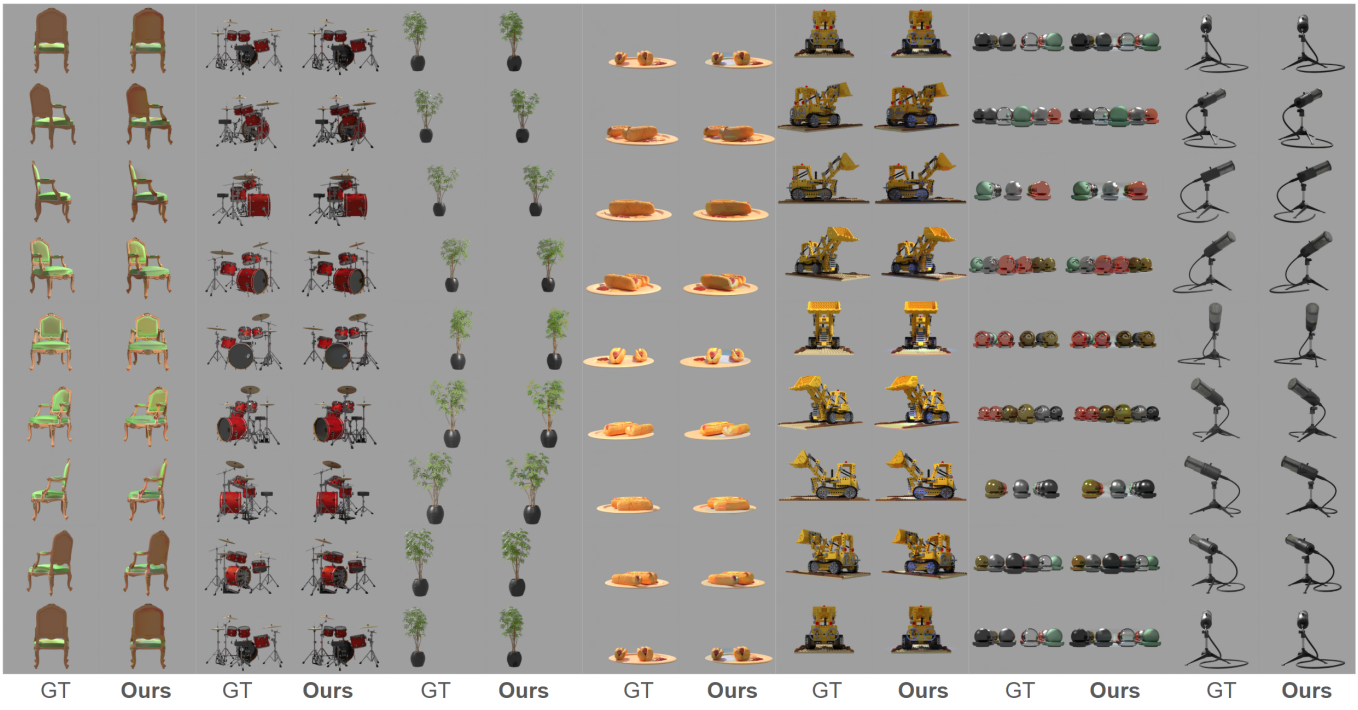

Fig. 1. Qualitative comparison on synthetic sequences with motion blur

format. Instead of using the `ns-process-data` command designed in Nerfstudio, we programmed our data processing script as follows.

Initially, we load the line-by-line event data as four NumPy arrays: timestamps, x coordinates, y coordinates, and polarity values. Subsequently, we employ the accumulation method to obtain raw event-based images between t_0 and t. Due to the random backtracking of t_0 for a given t, we store the corresponding pair in the generated camera pose `JSON` so that when called during training, the loss function can view the difference between these paired camera poses.

$$t_0 \sim U[t - L_{max}, t) \tag{7}$$

Next, we apply a Bayer filter to obtain a colorized image. Akin to [Rudnev et al. 2023], our Bayer filter $F \in \mathbb{R}^{W \times H \times 3}$, where $W \times H$ is the image resolution, and F consists of the following tiled 2×2 pattern: [[[1, 0, 0], [0, 1, 0]], [[0, 1, 0], [0, 0, 1]]] (RGGB filter).

For each camera's pose, we load the camera-to-world poses generated by EventNeRF [Rudnev et al. 2023] in the COLMAP/OpenCV convention. We then process s the matrix by flipping, swapping, and inverting vertically to align with the OpenGL/Blender coordinate system used in Nerfstudio. The final operation scales the scene to a "NeRF-sized" scale. To process dataset, we iterate through the event stream from beginning to end and save the colorized image between timestamps t_0 and t to calculate its accumulated difference accordingly.

5.2 DataManager and DataParser

The DataManager (`esplatfacto_datamanager`) is responsible for turning posed images into RayBundles, which are slices of 3D space that start at a camera origin. Within the DataManager, the DataParser (`esplatfacto_dataparser`) first loads the input images and camera data. Once the images are properly loaded and formatted, the DataManager iterates through the ground truth data, generating RayBundles and ground truth supervision.

Compared to vanilla 3DGS or NeRF, the challenge of event-based 3DGS lies in the fact that each supervision relies on both a random previous frame and the deterministic current frame. Consequently, we need to pass through both concurrent groups of camera poses, denoted as `cameras`, and its corresponding `pre_cameras`. In each

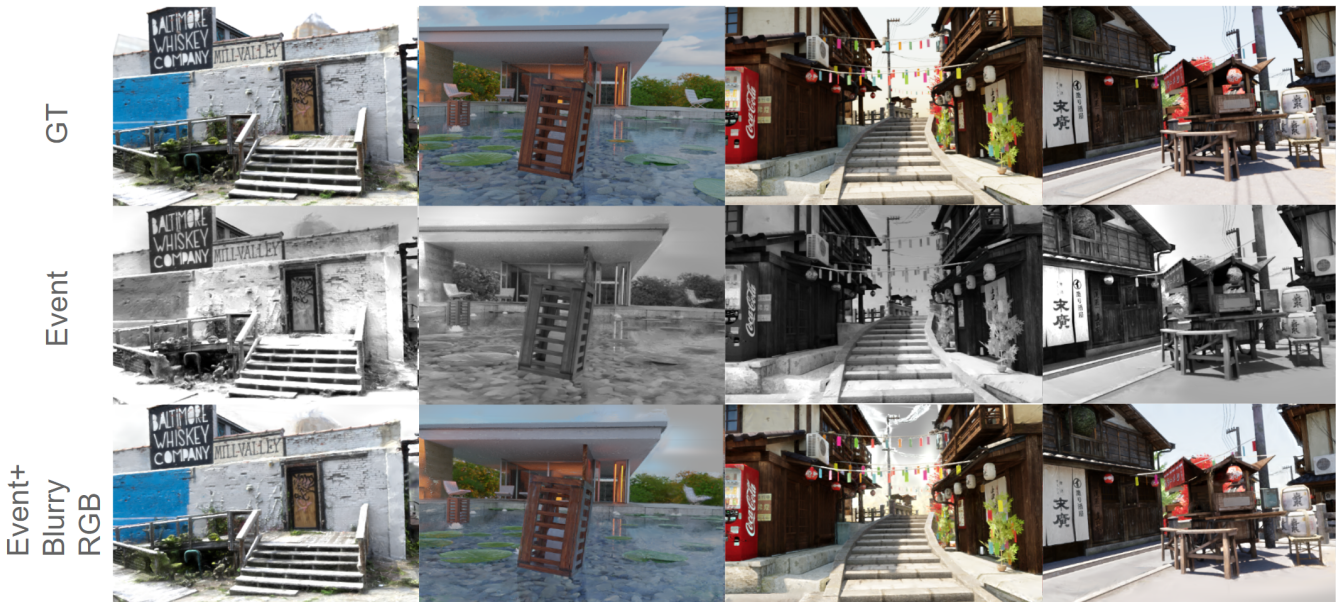

Fig. 2. Qualitative comparison on synthetic sequences with motion blur

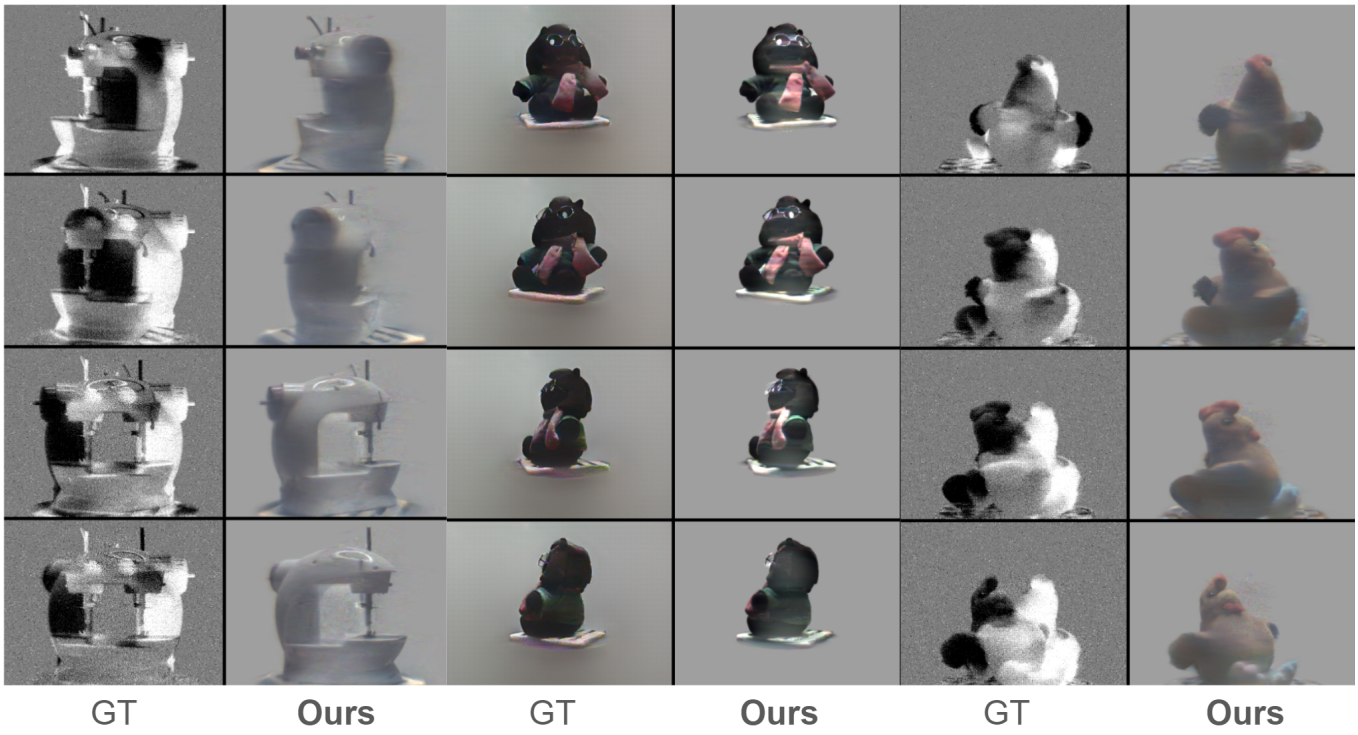

Fig. 3. Qualitative comparison on real sequences with motion blur

training step, we request a t_0 camera pose named `camera_pre` and a `camera_pre` at timestep t by calling `esplatfacto_datamanager`.

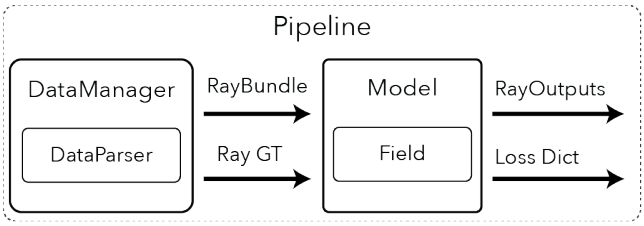

Fig. 4. Pipeline components. Each method in Nerfstudio is implemented as a custom Pipeline. DataManager processes input images into bundles of rays (RayBundles) that get rendered by the Model to produce a set of NeRF outputs (RayOutput). A dictionary of losses supervises the pipeline end-to-end.

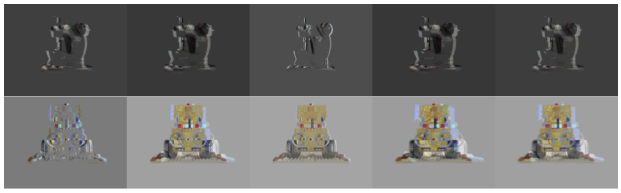

Fig. 5. Processed 'Sewing' (real-world) and 'Lego'(synthesis).

5.3 RayBundle, RaySample, and Frustum

The RayBundle is a primitive that represents a slice through 3D space. By specifying the interval bin spacing, the RayBundle generates RaySample, which represents sampled chunks of 3D space along each ray. These chunks, represented as Frustums, can be encoded either as point samples or as Gaussians with mean and covariance. A visualization of this abstraction can be found in Fig. 6.

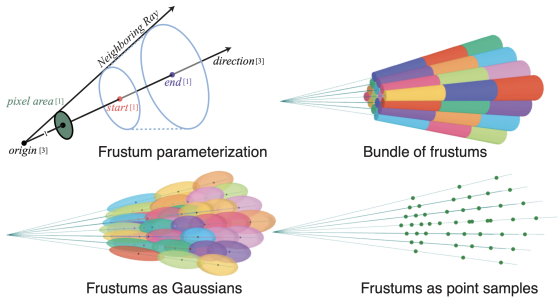

Fig. 6. Different versions of raybundle, in our case, we used the frustum as Gaussians.

5.4 Model and Field

The RayBundles are sent to Models (`ESplatfactoModelConfig`) as input, which samples them into RaySamples. The RaySamples are consumed by Fields to turn Frustums into color or density. We remain most parts of the `Splatfacto` except we feed the loss function in Eq. 10 to the training Pipeline.

$$\hat{L} = L(t) - L(t_0), L = F \odot E(t_0, t) \tag{8}$$

$$\mathcal{L}_1 = \text{MSE}(\hat{L}, L), \mathcal{L}_{\text{D-SSIM}} = \frac{1 - \text{SSIM}(L, \hat{L})}{2} \tag{9}$$

$$\mathcal{L}_{\text{3DGS}} = (1 - \lambda)\mathcal{L}_1 + \lambda\mathcal{L}_{\text{D-SSIM}} \tag{10}$$

5.5 Esplatfacto training results

From the above description of our method, we should note that the Esplatfacto approach is a naive implementation for training a 3D Gaussian splatting model. As such, our vanilla Esplatfacto method suffers from several drawbacks. We can observe background color inconsistency in the ground truth image generation, as shown in Fig. 5, which is caused by normalization during the debayering filter recovery stage. It is also worth noting that the debayering method differs between the training and ground truth generation stages. Furthermore, we did not apply clipping or masking in our training, resulting in numerous background floaters in our training viewer compared to EventNeRF, which clips the training region of interest to a cylindrical shape. Additionally, we did not incorporate negative sampling, as discussed in Sec. 3.4.

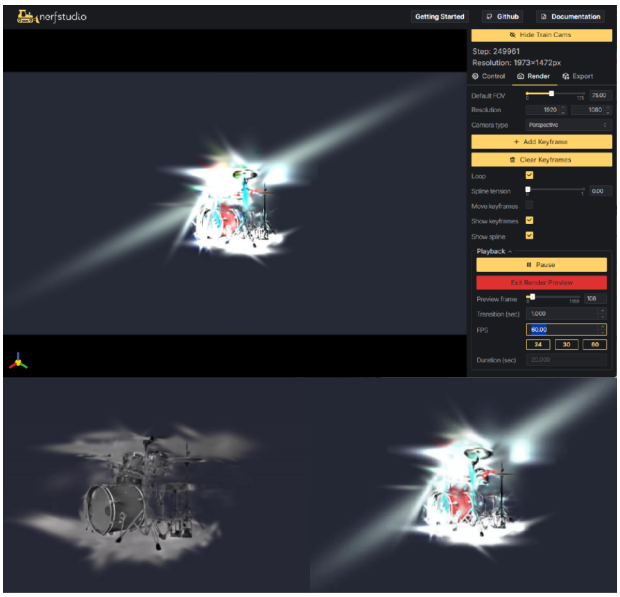

Fig. 7. The overall Esplatfacto training interface in NerfStudio (top). Grayscale training result of 'drums' (bottom left), RGB training result of 'drums' (bottom right).

6 CONCLUSION

In this project, we explore the possibility of applying event-streams onto the state-of-the-art 3D Gaussian Splatting for scene reconstruction and novel-view synthesis. With the formulation of event-based optimization, 3DGS is able to reconstruct clear 3D structures solely from event-streams. Our simple yet effective negative-sampling and robust training further boosts model performance, outperforming previous EventNeRF by +1 PSNR while reducing the training time from hours to around 8 minutes. And we also use limited number

of blurry RGB images to refine the appearance. In addition, our pipeline also works well in forward looking examples. We also integrate our method into the open-source project NeRF-Studio, making it more usable and scalable for future research.

REFERENCES

Alexis Baudron, Zihao W. Wang, Oliver Cossairt, and Aggelos K. Katsaggelos. 2020. E3D: Event-Based 3D Shape Reconstruction. arXiv:2012.05214 [cs.CV]

Anish Bhattacharya, Ratnesh Madaan, Fernando Cladera, Sai Vemprala, Rogerio Bonatti, Kostas Daniilidis, Ashish Kapoor, Vijay Kumar, Nikolai Matni, and Jayesh K. Gupta. 2024. EvDNeRF: Reconstructing Event Data With Dynamic Neural Radiance Fields. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 5846–5855.

Marco Cannici and Davide Scaramuzza. 2024. Mitigating Motion Blur in Neural Radiance Fields with Events and Frames. arXiv:2403.19780 [cs.CV]

William Chamorro, Joan Sola, and Juan Andrade-Cetto. 2022. Event-based line SLAM in real-time. *IEEE Robotics and Automation Letters* 7, 3 (2022), 8146–8153.

Yuhuang Hu, Shih-Chii Liu, and Tobi Delbruck. 2021. v2e: From Video Frames to Realistic DVS Events. arXiv:2006.07722 [cs.CV]

Inwoo Hwang, Junho Kim, and Young Min Kim. 2023a. Ev-nerf: Event based neural radiance field. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 837–847.

Inwoo Hwang, Junho Kim, and Young Min Kim. 2023b. Ev-NeRF: Event Based Neural Radiance Field. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 837–847.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics* 42, 4 (July 2023). https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/

Simon Klenk, Lukas Koestler, Davide Scaramuzza, and Daniel Cremers. 2023a. E-nerf: Neural radiance fields from a moving event camera. *IEEE Robotics and Automation Letters* 8, 3 (2023), 1587–1594.

Simon Klenk, Lukas Koestler, Davide Scaramuzza, and Daniel Cremers. 2023b. E-NeRF: Neural Radiance Fields from a Moving Event Camera. *IEEE Robotics and Automation Letters* (2023).

Li Ma, Xiaoyu Li, Jing Liao, Qi Zhang, Xuan Wang, Jue Wang, and Pedro V. Sander. 2022. Deblur-NeRF: Neural Radiance Fields from Blurry Images. arXiv:2111.14292 [cs.CV]

Qi Ma, Danda Pani Paudel, Ajad Chhatkuli, and Luc Van Gool. 2023. Deformable Neural Radiance Fields using RGB and Event Cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 3590–3600.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*.

Manasi Muglikar, Guillermo Gallego, and Davide Scaramuzza. 2021. ESL: Event-based structured light. In *2021 International Conference on 3D Vision (3DV)*. IEEE, 1165–1174.

Jeongtaek Oh, Jaeyoung Chung, Dongwoo Lee, and Kyoung Mu Lee. 2024. DeblurGS: Gaussian Splatting for Camera Motion Blur. arXiv:2404.11358 [cs.CV]

Y. Qi, L. Zhu, Y. Zhang, and J. Li. 2023. E2NeRF: Event Enhanced Neural Radiance Fields from Blurry Images. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE Computer Society, Los Alamitos, CA, USA, 13208–13218. https://doi.org/10.1109/ICCV51070.2023.01219

Viktor Rudnev, Mohamed Elgharib, Christian Theobalt, and Vladislav Golyanik. 2023. EventNeRF: Neural Radiance Fields from a Single Colour Event Camera. In *Computer Vision and Pattern Recognition (CVPR)*.

Otto Seiskari, Jerry Ylilammi, Valtteri Kaatrasalo, Pekka Rantalankila, Matias Turkulainen, Juho Kannala, Esa Rahtu, and Arno Solin. 2024. Gaussian Splatting on the Move: Blur and Rolling Shutter Compensation for Natural Camera Motion. arXiv:2403.13327 [cs.CV]

Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David Mcallister, Justin Kerr, and Angjoo Kanazawa. 2023. Nerfstudio: A Modular Framework for Neural Radiance Field Development. In *ACM SIGGRAPH 2023 Conference Proceedings* (<conf-loc>, <city>Los Angeles</city>, <state>CA</state>, <country>USA</country>, </conf-loc>) *(SIGGRAPH '23)*. Association for Computing Machinery, New York, NY, USA, Article 72, 12 pages. https://doi.org/10.1145/3588432.3591516

Peng Wang, Lingzhe Zhao, Ruijie Ma, and Peidong Liu. 2023. BAD-NeRF: Bundle Adjusted Deblur Neural Radiance Fields. arXiv:2211.12853 [cs.CV]

Ziyun Wang, Kenneth Chaney, and Kostas Daniilidis. 2022. Evac3d: From event-based apparent contours to 3d models via continuous visual hulls. In *European conference on computer vision*. Springer, 284–299.

David Weikersdorfer, Raoul Hoffmann, and Jörg Conradt. 2013. Simultaneous localization and mapping for event-based vision systems. In *Proceedings of the 9th international conference on Computer Vision Systems*. 133–142.

Chen Wenbo and Liu Ligang. 2024. Deblur-GS: 3D Gaussian Splatting from Camera Motion Blurred Images. *Proc. ACM Comput. Graph. Interact. Tech. (Proceedings of I3D 2024)* 7, 1 (2024), 13 pages. https://doi.org/10.1145/3651301

Lee A Westover. 1991. *SPLATTING: A Parallel, Feed-Forward Volume Rendering Algorithm*. Technical Report. USA.

Kun Xiao, Guohui Wang, Yi Chen, Jinghong Nan, and Yongfeng Xie. 2022. Event-based dense reconstruction pipeline. In *2022 6th International Conference on Robotics and Automation Sciences (ICRAS)*. IEEE, 172–177.

Lingzhe Zhao, Peng Wang, and Peidong Liu. 2024. BAD-Gaussians: Bundle Adjusted Deblur Gaussian Splatting. arXiv:2403.11831 [cs.CV]

Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. 2019. Unsupervised event-based learning of optical flow, depth, and egomotion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 989–997.

Received 20 February 2024