SpikeGS: Reconstruct 3D Scene Captured by a Fast-Moving Bio-Inspired Camera

Yijia Guo[1]*, Liwen Hu[1]*, Yuanxi Bai[2], Jiawei Yao[3], Lei Ma[1,2]†, Tiejun Huang[1]

[1]National Key Laboratory for Multimedia Information Processing, School of Computer Science, Peking University
[2]National Biomedical Imaging Center, Peking University
[3]University of Washington
{2301112015, 230121059}@stu.pku.edu.cn, {liwenhu, leima, tjhuang}@pku.edu.cn, jwyao@uw.edu

Abstract

3D Gaussian Splatting (3DGS) has been proven to exhibit exceptional performance in reconstructing 3D scenes. However, the effectiveness of 3DGS heavily relies on sharp images, and fulfilling this requirement presents challenges in real-world scenarios particularly when utilizing fast-moving cameras. This limitation severely constrains the practical application of 3DGS and may compromise the feasibility of real-time reconstruction. To mitigate these challenges, we proposed Spike Gaussian Splatting (SpikeGS), the first framework that integrates the Bayer-pattern spike streams into the 3DGS pipeline to reconstruct 3D scenes captured by a fast-moving high temporal resolution color spike camera **in one second**. With accumulation rasterization, interval supervision, and a special designed pipeline, SpikeGS realizes continuous spatiotemporal perception while extracts detailed structure and texture from Bayer-pattern spike stream which is unstable and lacks details. Extensive experiments on both synthetic and real-world datasets demonstrate the superiority of SpikeGS compared with existing spike-based and deblurring 3D scene reconstruction methods.

Code — https://spikegs.github.io.

Introduction

Speed remains a critical challenge in the area of 3D reconstruction. The achievement of Neural Radiance Field (Mildenhall et al. 2021) has been notably successful. However, the training and rendering processes associated with the neural implicit representation entail a substantial time commitment, resulting in a significant bottleneck. 3DGS (Kerbl et al. 2023) has effectively addressed this challenge, remarkably improving reconstruction speeds to minutes and achieving real-time rendering, while InstantSplat (Fan et al. 2024) has further reduced reconstruction time to seconds. However, ultra-fast reconstruction speeds do not inherently guarantee a real-time reconstruction pipeline. Both 3DGS and the majority of methods relying on 3DGS, such as InstantSplat, require sharp image inputs, resulting in the necessity for slow camera movements to prevent motion blur. Capturing even the most basic scenes often needs minutes or more. In the near future, the speed of camera capture, rather than training speed, may become the limiting factor in 3D reconstruction. Presently, the demand for sharp images also limits the application of 3DGS in scenarios when cameras are mounted on high-speed vehicles, e.g., trains and UAVs. Although many deblurring methods (Chen and Liu 2024; Zhao, Wang, and Liu 2024) attempt to address these challenges and free the constraint on camera motion, the limitations of traditional RGB camera sensors greatly hinder their ability to mitigate motion blur effectively. Bio-inspired cameras, characterized by their high temporal resolution, are considered promising solutions to address these issues at their core. Bio-inspired cameras such as event and spike (Huang et al. 2023) cameras have been widely applied in 3D reconstruction (Rudnev et al. 2023; Xiong et al. 2024; Zhu et al. 2024) of high-speed scenes with significant success. However, event-based camera approaches may lack adequate visual texture and frequently rely on supplementary traditional cameras for reconstruction (Qi et al. 2023), thereby constraining their applicability. Spike cameras, which capture rich visual textures, are more suitable for independent 3D reconstruction. However, prior solutions (Guo et al. 2024b; Zhu et al. 2024) have encountered difficulties in training and rendering efficiency, which posing an insurmountable challenge for real-time rendering and

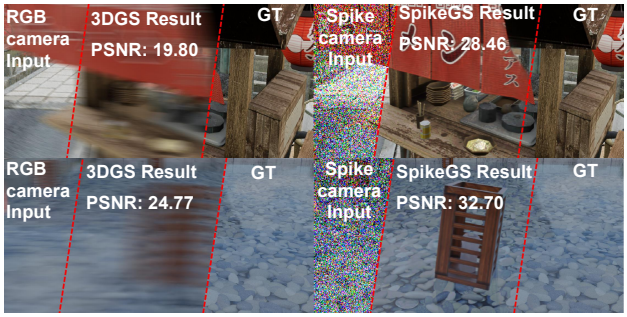

Figure 1: Brief introduction of SpikeGS. Given high temporal resolution but texture lacking spike stream captured by a fast-moving spike camera **in one second**, our method can reconstruct a sharp radiance field in minutes and achieves clear real-time rendering results.

*These authors contributed equally.
†Corresponding author.

fast reconstruction. These methods also have not undergone extensive testing in real-world high-speed scenarios and have encountered challenges in reconstructing Bayer-pattern spike streams as the first-generation spike camera can only reproduce grayscale signals. Besides, as depicted in Figure 1, the Bayer-pattern spike stream lacks both geometric consistency and time-domain stability. The rapid reconstruction of precise structures and accurate irradiance from unstable spike streams poses a significant challenge.

In this paper, we combine spike streams with 3DGS to devise a tailored 3D reconstruction pipeline, achieving the reconstruction of scenes captured by spike cameras in one second for the first time. To reconstruct detailed structures and textures from high-temporal resolution but with noisy visual texture spike streams, we carefully designed accumulation rasterization and interval supervision to enhance continuous spatiotemporal perception. Specifically, accumulation rasterization effectively utilizes long-term temporal information to mitigate temporal aliasing and motion blur, thereby facilitating precise per-pixel detail restoration and achieving sharp reconstruction results. In parallel, interval supervision contributes to the recovery of intricate geometric structures and enables rapid initialization of point clouds and camera poses. To validate our approach, we have created the first dataset specifically for 3D reconstruction from high-speed motion spike streams, which also serves as the first dataset for 3D reconstruction from Bayer-pattern spike streams. Our experiments demonstrate that our method not only achieves state-of-the-art 3D reconstruction quality for high-speed scenes from spike streams but also exhibits exceptional robustness to speed variations. SpikeGS is also compatible with the first-generation spike camera, as delineated in the appendix. Our main contributions are as follows:

- We propose SpikeGS, **the first framework** integrating Bayer-pattern spike streams into the 3DGS pipeline for reconstructing 3D scenes using a fast-moving spike camera. Our method is tailored for spike-based supervision, ensuring high robustness across various speeds.
- We contribute SpikeGS dataset, **the first dataset** for 3D reconstruction from high-speed real-world spike scenes, **the first dataset** for 3D reconstruction from color spike camera, and **the first dataset** featuring explicit speed annotations and varying speeds within the same scene. We also provide a standardized paradigm for capturing indoor high-speed spike 3D reconstruction scenes.
- Our experiments demonstrated that our method not only achieves state-of-the-art 3D reconstruction quality for high-speed scenes compared to all current methods but also exhibits exceptional robustness to speed variations. We also validated and proved the feasibility of spike camera reconstruction of open outdoor scenes using synthetic datasets for the first time.

Related Work

Novel View Synthesis on Traditional Cameras

Novel View Synthesis (NVS) is the process of generating new images from viewpoints different from those of the original captures. NeRF-based approaches (Mildenhall et al. 2021; Barron et al. 2021, 2022; Wang et al. 2021; Zhang et al. 2020) have become a standard technique in this field and demonstrate that implicit radiance fields can effectively learn scene representations and synthesize high-quality novel views. Further advancements (Müller et al. 2022; Chen et al. 2022; Fridovich-Keil et al. 2022) have been made to improve the training and rendering of NeRF with advanced implicit scene representations. 3D Gaussian splatting (Kerbl et al. 2023) further adopts a discrete 3D Gaussian representation of scenes, significantly accelerating the training and rendering of radiance fields with explicit scenes. It has attracted considerable research interest in the field of generation (Yi et al. 2023; Chen et al. 2023), relighting (Liang et al. 2023; Guo et al. 2024a) and dynamic 3D scene reconstruction (Wu et al. 2023; Yang et al. 2023).

Novel View Synthesis on Bio-inspired Cameras

Bio-inspired sensors have shown their advantages in most computer version problems including novel views synthesizing. EventNeRF (Rudnev et al. 2023) and Ev-nerf (Hwang, Kim, and Kim 2023) synthesize the novel view in scenarios such as high-speed movements that would not be conceivable with a traditional camera by event supervision. Nonetheless, these works assume that event streams are temporally dense and low-noise which is inaccessible in practice. Robust e-NeRF (Low and Lee 2023) incorporates a more realistic event generation model to directly and robustly reconstruct NeRFs under various real-world conditions. DE-NeRF (Ma et al. 2023) and E2NeRF (Qi et al. 2023) extend event nerf to dynamic scenes and severely blurred images as NeRF researchers did. Recently, many researchers (Xiong et al. 2024) introduced 3D Gaussian Splatting framework for reconstructing appearance and geometry solely from event data, significantly improved the speed and quality of reconstruction. However, due to the absence of texture details in event data, all these approaches yield limited results. Spike cameras (Dong, Huang, and Tian 2021) can capture more texture information. NeRF based on spike cameras (Guo et al. 2024b; Zhu et al. 2024) achieves higher 3D scene reconstruction quality than event-based methods. However, the above methods still cannot quickly and efficiently reconstruct real-world 3D scenes while the camera moves at high speeds. They also have problems in dealing with Bayer-pattern spike streams from color spike camera.

Datasets

Synthetic SpikeGS Dataset

We generate spike streams using six synthetic scenes from NeRF (Mildenhall et al. 2021) and three synthetic scenes from Deblur-nerf (Ma et al. 2022). Specifically, we first adjust the camera frame rate in Blender to match the frame rate of the spike camera and render images. Further, we use the state-of-the-art spike camera simulator SCSim (Hu et al. 2024) to convert the image sequences into Bayer-pattern spike streams. Besides, to investigate the 3D reconstruction performance under different speeds, we vary the motion speed of objects in each scene and generate another set

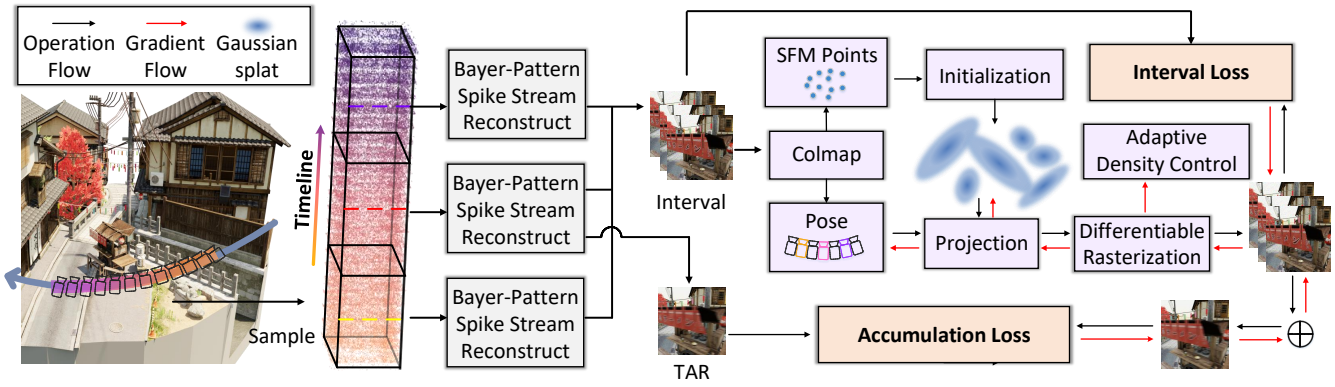

Figure 2: Overview of our SpikeGS. We first reconstruct Bayer-pattern spike streams into spike intervals and spike accumulation (details are shown in Fig. 5). Unlike 3DGS, we adopt spike intervals to initialize SFM points, camera poses, and Gaussian splats. We then embed the time accumulation process into the rasterizer to calibrate the colorization while maintaining multi-view consistency. By progressively optimizing the 3DGS parameters using an accumulation loss and an interval loss, our method facilitates high-quality 3DGS reconstruction.

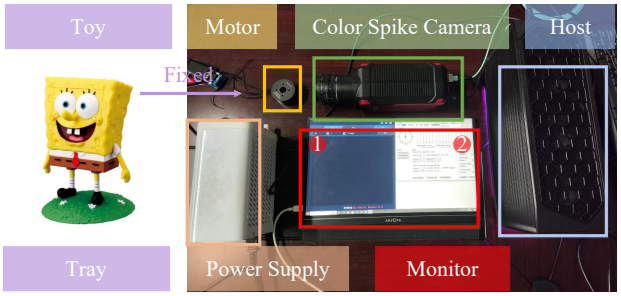

Figure 3: The setup for capturing the real-world dataset, SpikeGS-dataset. First, we fix the toy onto a tray, then the tray is fixed to a motor. Finally, we use a spike camera to capture the high-speed rotating toy. ①(②) controls the sampling of the spike camera (the rotation speed of the motor).

of spike streams with higher speeds. As a result, the high-speed spike streams have a size of $1000\times1000\times1500$, which indicates that the motion speed is about 2500 rpm. The low-speed spike streams have a size of $1000 \times 1000 \times 2500$, which indicates that the motion speed is about 1500 rpm.

Real-world SpikeGS Dataset

In fact, there is still a significant gap between synthetic and real data for bio-inspired cameras. Therefore, constructing a real 3D high-speed scene dataset is crucial for the evaluation of methods. As shown in Fig. 3, we carefully design an experimental setup to capture spike streams. Based on the setup, we first propose a 3D high-speed scene dataset for color spike cameras. SpikeGS dataset contains a total of 3×7 scenes, where 3 represents 3 different motion speeds (700 rpm, 1000 rpm, 1400 rpm) and 7 represents 7 different objects. For each dynamic scene, we capture a one-second spike stream and its size is $20000 \times 1000 \times 1000$.

Method

Preliminary:3D Gaussian Splatting

3D Gaussian Splatting constitutes an explicit 3D scene representation utilizing point clouds, with Gaussian splats employed to depict the scene's structure. In this representation, every Gaussian splat G is parameterized by an anisotropic covariance $\Sigma \in \mathbb{R}^{3\times3}$ and the center $x \in \mathbb{R}^3$.

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

The covariance matrix Σ of a 3D Gaussian scatter can be analogized to the characterization of an ellipsoidal form. Consequently, it can be further factorized into a rotation matrix denoted as $R \in SO(3)$ and a scale matrix denoted as $S \in \mathbb{R}^3$, permitting independent optimization:

$$\Sigma = RSS^TR^T \tag{2}$$

To render our 3D Gaussian splats in 2D, we employ the technique of splatting to position the Gaussian splats on the camera planes:

$$\Sigma' = JW\Sigma W^TJ^T \tag{3}$$

Where J is the Jacobian of the affine approximation of the projective transformation and W is the viewing transformation. Following this, the pixel color is obtained by alpha-blending N sequentially layered 2D Gaussian splats from front to back:

$$C = \sum_{i\in N} c_i\alpha_i \prod_{j=1}^{i-1}(1-\alpha_j) \tag{4}$$

Where c_i is the color of each point and α_i is given by evaluating a 2D Gaussian with covariance Σ multiplied with a learned per-point opacity.

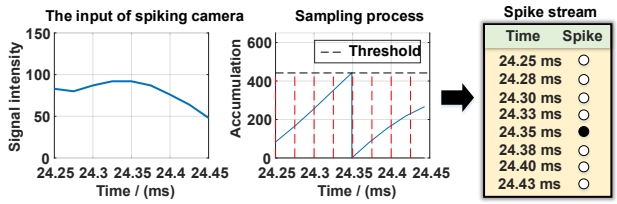

Figure 4: Working principle of each pixel in spike camera. Black (white) circle denotes a (no) spike.

Color Spike Camera Model

Each pixel on the color spike camera model converts the light signal into a current signal and accumulates the input current. As shown in Fig. 4, for pixel $\mathbf{x} = (x, y)$, if the accumulation of input current reaches a fixed threshold ϕ, a spike is fired and then the accumulation can be reset as,

$$A(\mathbf{x}, t) = A_{\mathbf{x}}(t) \bmod \phi = \int_0^t L_C(\mathbf{x}, \tau) d\tau \bmod \phi, \quad (5)$$

where $A(\mathbf{x}, t)$ is the accumulation at time t, $A_{\mathbf{x}}(t)$ is the accumulation without reset before time t, $L_C(\mathbf{x}, \tau)$ is the input current of a certain color at time τ (proportional to light intensity) and $C \in \{R, G, B\}$ denote the color determined by the Bayer-pattern color layout. Note that in a real camera, noise can interfere with the input current L_C and threshold ϕ. As shown in Fig. 4, each spike is read out at discrete time $nT, n \in \mathbb{N}$ (T is a micro-second level). The output of the color spike camera is a spatial-temporal binary stream S with $H \times W \times N$ size.**Please refer to appendix for more details**.

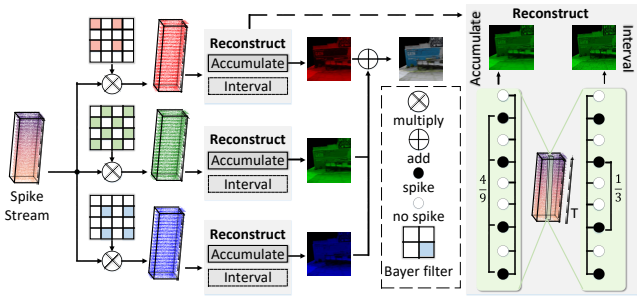

Figure 5: **Left:** Overview of Bayer-pattern spike stream reconstruction. We employ a Bayer filter to extract the spike streams of a certain color and calculate accumulating/interval results separately. **Right:** Reconstruction details. We estimate light intensity by accumulating the number of spikes over a period of time (equation 6) or calculating the interval between adjacent spikes (equation 13).

Time Accumulation Rasterization

Due to the instability of the visual texture in a single-frame spike, utilizing the spike stream directly for supervising the training of 3DGS may result in serious multi-view inconsistency and incorrect structure. Therefore, we accumulate long-term temporal information on spike streams to get stable texture. As shown in Fig. 5, we mathematically represented this process as follows:

$$I_{ac}(t_1, t_N) = \phi / N \sum_{t_i} S(P_{t_i}) \quad (6)$$

Where $I_{ac}(t_1, t_N)$ is the result of spike stream accumulation from t_1 to t_N, P_{t_i} is the camera pose at time $t_i \in \{t_1, t_2, \ldots, t_N\}$, ϕ is the threshold and $S(P_{t_i})$ is the spike at camera pose P_{t_i}. Additionally, a 2×2 Bayer filter, as shown in Fig. 5, is employed to extract the spike streams of a certain color and accumulate them separately. The discreet nature of the spike stream in time necessitates the use of summation rather than integration to denote the accumulation process. If the photoelectric conversion process is disregarded, from equation 5 and equation 6 we have:

$$I_{ac}(t_1, t_N) = \phi / N \sum_{t_i} \int_{t_{i-1}}^{t_i} c(P_\tau) d\tau \bmod \phi \quad (7)$$

Where $c(P_\tau)$ is the radiance at camera pose P_τ. Further we have:

$$I_{ac}(t_1, t_N) \approx 1/N \int_{t_1}^{t_N} c(P_\tau) d\tau \quad (8)$$

We have proved in the appendix that the error of the above approximation does not exceed $1/N$. To align the accumulation process of the spike stream, we also improved the rasterization of 3DGS to a time accumulation rasterization by simulating the physical accumulation process of photons:

$$C_{ac}(t_1, t_N) = N^{-1} \int_{t_1}^{t_N} \sum_{i \in N} c_i(P_\tau) \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j) d\tau \quad (9)$$

Considering computability, equation 9 can be approximated by a discrete summation:

$$C_{ac}(t_1, t_N) = N^{-1} \sum_{\tau=t_1}^{t_N} \sum_{i \in N} c_i(P_\tau) \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j) \quad (10)$$

Given the high temporal resolution of the spike stream, the changes in camera pose between consecutive frames are minimal. Consequently, the rendering results between adjacent frames can be assumed to exhibit approximately linear variations. Therefore, the average rendering results of consecutive frames can be approximated by the result from the middle frame. In our paper, we use n keyframes (where n $<$ 10) to represent the entire cumulative process. The equation 10 can be further expressed as:

$$C_{ac}(t_1, t_N) = n^{-1} \sum_{i=1}^{n} \sum_{i \in N} c_i(P_i) \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j) \quad (11)$$

Where P_i is the camera pose at the keyframe i. The above process is fully differentiable. The derivation of differentiable rendering is also provided in the appendix. Finally, we employ an accumulation loss to minimize their photometric error, expressed as:

$$\mathcal{L}_{ac} = (1 - \lambda_1) \|C_{ac} - I_{ac}\|_2 + \lambda_1 SSIM(C_{ac}, I_{ac}) \quad (12)$$

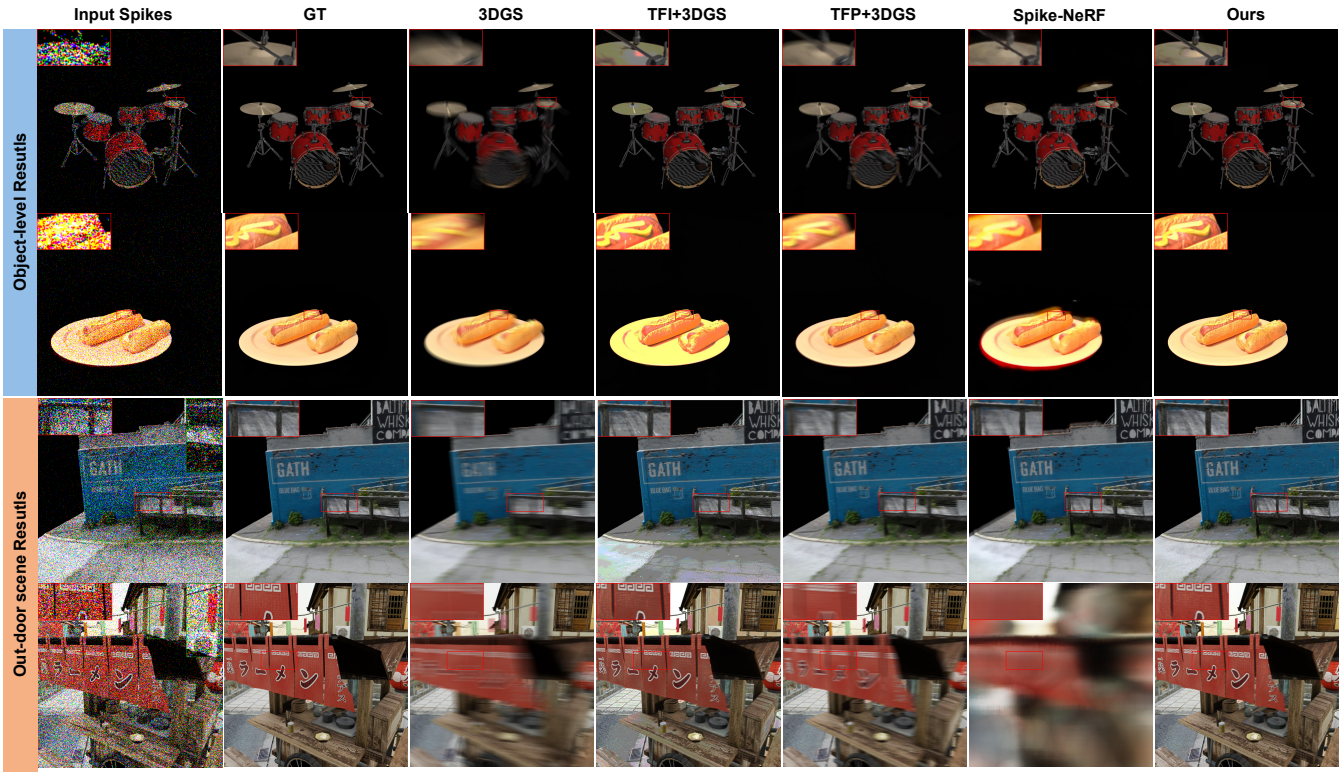

Figure 6: Qualitative comparisons on synthetic datasets. Our SpikeGS achieved sharper, more consistent structures without motion blur like 3DGS or TFP+3DGS artifacts like Spike-NeRF, and color distortion like TFI+3DGS.

Mutually Constrained Training and Loss Function

While using time accumulation rasterization can help recover texture details and geometric information effectively, it tends to hinder the rapid convergence of Gaussian splats during the early stages of training, significantly impacting training efficiency. Meanwhile, the accuracy of point cloud initialization and the geometric precision during the early stages of Gaussian splats training have a significant impact on the final reconstruction quality of 3DGS. We found that whether using a single frame spike or accumulated spike streams, the results of point cloud initialization and the initial training were unsatisfactory. In our paper, we use spike intervals, which have higher sensitivity and flexibility to the time domain, to initialize point clouds and use them for initial training, as shown in Fig. 2. In order to obtain more accurate color and texture information from the spike interval (Zhu et al. 2019), we added additional processing to the spike interval, just like Fig. 5:

$$I_{in} = \frac{\phi}{t_1 - t_2} \tag{13}$$

Where ϕ represents the threshold. t_1 and t_2 represent the times corresponding to two adjacent trigger spikes ($S(t_1) = S(t_2) = 1$). We used a 2 × 2 Bayer filter the same as accumulation. Using the spike interval allows the recovery of geometric details using minimal numbers of spikes (typically t1-t2 <10). However, spike intervals have significant time-domain aliasing, which manifests as color deviations and is highly susceptible to noise interference. Therefore, we gradually reduce its impact during later stages of training, and ultimately only use it to adjust the geometric parameters of the Gaussian, such as position and opacity. Rendering results corresponding to the spike intervals can be represented as:

$$C_{in} = \sum_{i \in N} c_i(P_{(t_1+t_2)/2})\alpha_i \prod_{j=1}^{i-1}(1-\alpha_j) \tag{14}$$

We also employ an interval loss which can be expressed as :

$$\mathcal{L}_{in} = (1-\lambda_1)\|C_{in} - I_{in}\|_2 + \lambda_1 SSIM(C_{in}, I_{in}) \tag{15}$$

Our final loss function can be represented as:

$$\mathcal{L}_{final} = \lambda_{accu}\mathcal{L}_{in} + \lambda_{in}\mathcal{L}_{in} \tag{16}$$

Where λ_{accu} and λ_{in} are hyperparameters. λ_{accu} starts at 0 in the early stages of training and gradually increases with the number of epochs.

Experiment

Implementation Details

Our code is based on 3DGS (Kerbl et al. 2023) and we train the models for 30000 iterations on one NVIDIA A800 GPU with the same optimizer and hyper-parameters as 3DGS. The color spike camera shown in Fig. 3 is **Spike M1K40-H4-Gen3** with 1000×1000 resolution and the motor shown in Fig. 3 is **DMR4315**.

Datasets	object			outdoor scene			average		
Methods|Metrics	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
3DGS	26.948	0.880	0.120	23.379	0.689	0.447	25.163	0.785	0.284
TFI+3DGS	<u>32.877</u>	0.850	<u>0.035</u>	26.955	<u>0.823</u>	<u>0.253</u>	<u>29.916</u>	0.837	<u>0.144</u>
TFP+3DGS	30.775	0.830	0.078	<u>27.506</u>	0.760	0.355	29.140	0.795	0.217
S2I+3DGS	30.421	<u>0.946</u>	0.060	27.313	0.762	0.284	28.867	<u>0.854</u>	0.171
SpikeNeRF	12.478	0.828	0.176	11.421	0.505	0.460	11.949	0.666	0.318
Spike-NeRF	28.243	0.924	0.081	18.142	0.621	0.413	23.192	0.772	0.247
Delur-3DGS	28.467	0.918	0.090	23.549	0.699	0.412	26.008	0.809	0.251
ours	**35.860**	**0.960**	**0.030**	**30.915**	**0.856**	**0.210**	**33.387**	**0.908**	**0.120**

Table 1: Quantitative comparisons of novel view synthesis on synthetic datasets. The best and second-best performances are respectively highlighted in bold and underscored with an underline. Our SpikeGS demonstrates the best NVS quality.

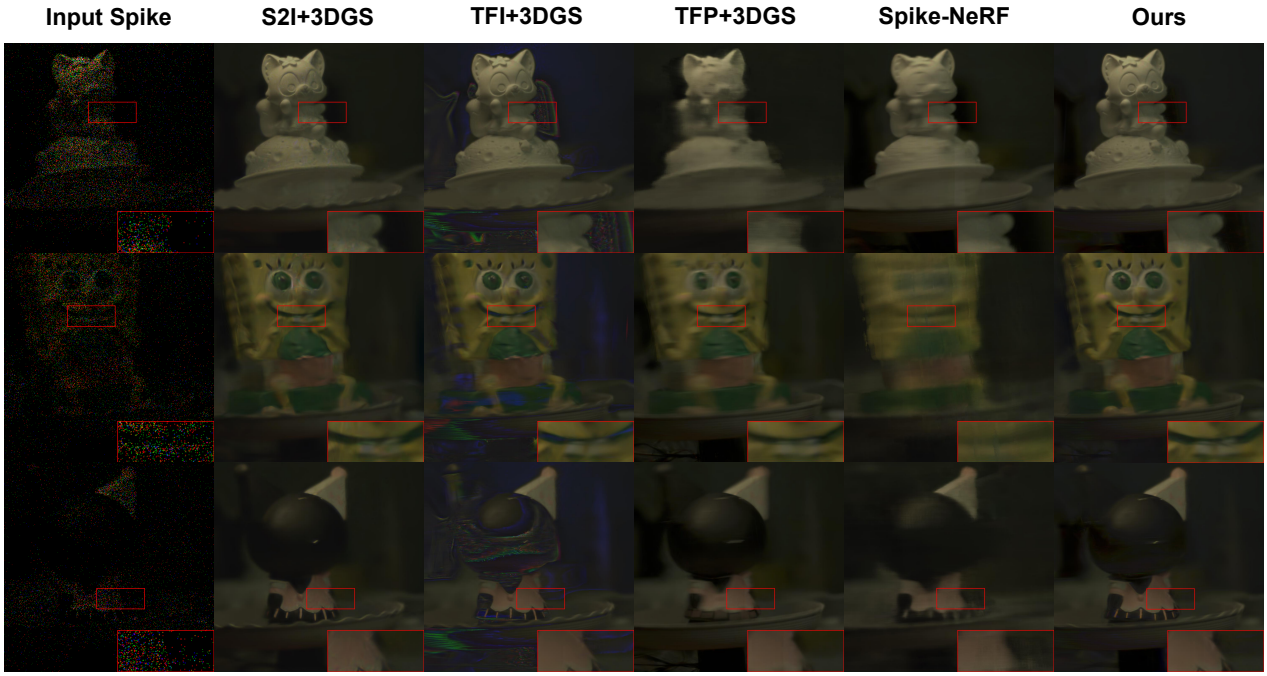

Figure 7: Qualitative comparisons on real-world datasets. Our SpikeGS also significantly outperforms the baseline on real-world datasets. We shows the results from high-speed scene.

Experimental Settings

We first compare our method with 3D-GS (Kerbl et al. 2023) and deblur-GS (Chen and Liu 2024) directly trained on the blurry images. Then, we compare our method against several 3D scene reconstruction methods for spike cameras, such as SpikeNeRF (Zhu et al. 2024) and Spike-NeRF (Guo et al. 2024b). Finally, given the absence of existing spike-based 3DGS methods, we compared our method with three spike-based image reconstruction methods with 3DGS: TFI (Zhu et al. 2019)+3DGS, TFP (Zhu et al. 2019)+3DGS and Spk2imgNet (Zhao et al. 2021) +3DGS. We conduct experiments using proposed SpikeGS datasets to evaluate our method's performance on both synthetic and real-world scenes. For synthetic datasets, we evaluate the synthesized novel view in terms of PSNR, SSIM, and LPIPS. For real-world datasets, we evaluate the synthesized novel view using a set of No-Reference Image Quality Assessment (NR-IQA) metrics since we don't have ground truth. The adopted metrics include NIQE (Mittal, Soundararajan, and Bovik 2012) and IL-NIQE (Zhang, Zhang, and Bovik 2015).

Comparisons on Synthetic Datasets

We compare our method with the aforementioned baselines on synthetic datasets. The results in Tab. 1 indicate that our SpikeGS achieved state-of-the-art novel view synthesis quality compared to other 3D reconstruction methods based on spike cameras and the deblur method based on traditional RGB camera. On average, our method achieves a +3.47dB higher PSNR, a 6.32% higher SSIM and a 14% lower LPIPS compared to the best performance among all baselines. We also perform qualitative experiments on synthetic data. Fig. 6 demonstrated that our SpikeGS achieved sharper, more consistent structures. Meanwhile, we validated the accuracy of the geometry by comparing the pre-

Scene Speed	High Speed		Mid Speed		Low Speed		Average	
Methods|Metrics	NIQE ↓	IL-NIQE ↓	NIQE ↓	IL-NIQE ↓	NIQE ↓	IL-NIQE ↓	NIQE ↓	IL-NIQE ↓
TFP+3DGS	10.9712	85.9904	11.0427	82.7746	10.5385	78.5193	10.8508	86.1067
S2I+3DGS	10.9634	<u>78.6108</u>	10.9318	<u>73.2949</u>	<u>10.4655</u>	<u>69.8803</u>	10.7869	<u>73.9287</u>
Spike-NeRF	**9.6009**	85.5263	<u>9.4205</u>	83.6021	10.6877	78.9751	<u>9.9030</u>	82.7012
Ours	<u>9.7297</u>	**74.8154**	**9.2678**	**69.8146**	**8.2760**	**63.7877**	**9.0912**	**69.4726**

Table 2: Quantitative comparisons of novel view synthesis on real-world datasets. The best and second-best performances are respectively highlighted in bold and underscored with an underline. Our SpikeGS demonstrates the best NVS quality.

Scene	High Speed			Low Speed			Average		
Method|Metrics	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
3DGS	25.225	0.850	0.139	26.948	0.880	0.120	26.08	0.865	0.130
Deblur-3DGS	26.422	<u>0.897</u>	0.110	28.467	0.918	0.090	27.44	0.907	0.100
Ours w/o $\mathcal{L}_{in}$	32.50	0.868	0.063	<u>35.75</u>	<u>0.948</u>	<u>0.031</u>	34.12	0.908	0.047
Ours w/o $\mathcal{L}_{accu}$	29.95	0.859	0.088	32.42	0.943	0.063	31.18	0.901	0.075
Ours-3 views	31.90	0.862	0.061	34.19	0.940	0.043	33.04	0.901	0.052
Ours-5 views	<u>32.94</u>	0.870	<u>0.052</u>	35.34	0.947	0.035	<u>34.14</u>	<u>0.908</u>	<u>0.043</u>
Ours-9 views	**33.45**	**0.921**	**0.0467**	**35.86**	**0.960**	**0.030**	**34.65**	**0.941**	**0.038**

Table 3: Quantitative ablation on $\mathcal{L}_{in}$, $\mathcal{L}_{accu}$, and numbers of camera poses on scenes with different speeds. The best and second-best performances are respectively highlighted in bold and underscored with an underline. All of them have a great influence on NVS quality.

Methods|Metrics	PSNR ↑	SSIM ↑	LPIPS ↓
3DGS	18.757	0.885	0.137
TFI+3DGS	<u>22.72</u>	0.327	0.096
TFP+3DGS	22.69	0.729	0.104
S2I+3DGS	21.41	<u>0.912</u>	<u>0.094</u>
SpikeNeRF	10.42	0.103	0.260
Spike-NeRF	11.72	0.280	0.220
Delur-3DGS	20.98	0.843	0.115
ours	**27.360**	**0.913**	**0.059**

Table 4: Quantitative comparisons of Depth Estimation on synthetic datasets. The best and second-best performances are respectively highlighted in bold and underscored with an underline. Our SpikeGS demonstrates the best quality.

cision of depth estimation. Tab. 4 shows that the geometric results reconstructed by our SpikeGS method significantly outperforms baselines. Visual results are shown in Fig 8.

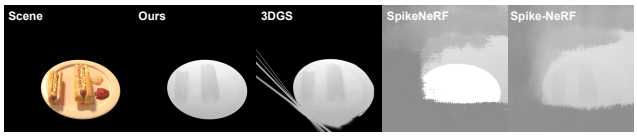

Figure 8: Qualitative comparisons on depth estimation. Our SpikeGS achieves significantly better structural details.

Comparisons on Real-world Datasets

Due to the lack of data from other modalities, we did not compare with non-spike methods on real-world data. According to Table 2, our approach demonstrated superior performance compared to other spike-based methods across three different scene speeds: high, medium, and low. This outcome validates the applicability of our method across a broad range of speeds. Fig. 7 demonstrates that our SpikeGS achieves more visually appealing results without introducing blurring, high-frequency noise, and erroneous backgrounds.

Ablation

Numbers of Camera Poses. We conduct ablations to investigate the effect of the number of camera poses (key frames) optimized in the accumulation. We select two speeds from our synthetic dataset for evaluation. As depicted in Tab. 3, the number of camera poses had a significant impact on the quality of reconstruction results, and the impact is more pronounced at high speeds. Here, we also provide a comparison with Deblur-3DGS (10 views) (Chen and Liu 2024) and 3DGS (Kerbl et al. 2023), which demonstrates that our approach demonstrates spectacular superiority over methods based on traditional RGB camera.

Accumulation Loss and Interval Loss. We also conduct novel view synthesis experiments to investigate the effect of interval and accumulation on the same dataset. Tab. 3 shows that both interval and accumulation loss significantly enhance novel view synthesis quality. Moreover, the impact of accumulation loss remains consistent across both low-speed and high-speed scenarios while the influence of interval loss is notably more pronounced in high-speed scenarios.

Conclusion

This paper introduces Spike Gaussian Splatting (SpikeGS), the first framework that integrates the Bayer-pattern spike streams into the 3DGS pipeline to reconstruct a 3D scene via a fast-moving spike camera. With accumulation rasterization, interval supervision, and a specially designed pipeline, SpikeGS reconstructs detailed geometry and texture from spike streams which are unstable and lack details, addressing the challenges associated with real-world Bayer-pattern spike streams.

Acknowledgments

This work was supported by National Scienceand Technology Major Project (2022ZD0116305)

References

Barron, J. T.; Mildenhall, B.; Tancik, M.; Hedman, P.; Martin-Brualla, R.; and Srinivasan, P. P. 2021. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5855–5864.

Barron, J. T.; Mildenhall, B.; Verbin, D.; Srinivasan, P. P.; and Hedman, P. 2022. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5470–5479.

Chen, A.; Xu, Z.; Geiger, A.; Yu, J.; and Su, H. 2022. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision*, 333–350. Springer.

Chen, W.; and Liu, L. 2024. Deblur-GS: 3D Gaussian Splatting from Camera Motion Blurred Images. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 7(1): 1–15.

Chen, Y.; Chen, Z.; Zhang, C.; Wang, F.; Yang, X.; Wang, Y.; Cai, Z.; Yang, L.; Liu, H.; and Lin, G. 2023. Gaussianeditor: Swift and controllable 3d editing with gaussian splatting. *arXiv preprint arXiv:2311.14521*.

Dong, S.; Huang, T.; and Tian, Y. 2021. Spike camera and its coding methods. *arXiv preprint arXiv:2104.04669*.

Fan, Z.; Cong, W.; Wen, K.; Wang, K.; Zhang, J.; Ding, X.; Xu, D.; Ivanovic, B.; Pavone, M.; Pavlakos, G.; et al. 2024. Instantsplat: Unbounded sparse-view pose-free gaussian splatting in 40 seconds. *arXiv preprint arXiv:2403.20309*.

Fridovich-Keil, S.; Yu, A.; Tancik, M.; Chen, Q.; Recht, B.; and Kanazawa, A. 2022. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5501–5510.

Guo, Y.; Bai, Y.; Hu, L.; Guo, Z.; Liu, M.; Cai, Y.; Huang, T.; and Ma, L. 2024a. PRTGS: Precomputed Radiance Transfer of Gaussian Splats for Real-Time High-Quality Relighting. *arXiv preprint arXiv:2408.03538*.

Guo, Y.; Bai, Y.; Hu, L.; Liu, M.; Guo, Z.; Ma, L.; and Huang, T. 2024b. Spike-NeRF: Neural Radiance Field Based On Spike Camera. *arXiv preprint arXiv:2403.16410*.

Hu, L.; Ma, L.; Guo, Y.; and Huang, T. 2024. SCSim: A Realistic Spike Cameras Simulator. arXiv:2405.16790.

Huang, T.; Zheng, Y.; Yu, Z.; Chen, R.; Li, Y.; Xiong, R.; Ma, L.; Zhao, J.; Dong, S.; Zhu, L.; et al. 2023. 1000× faster camera and machine vision with ordinary devices. *Engineering*, 25: 110–119.

Hwang, I.; Kim, J.; and Kim, Y. M. 2023. Ev-NeRF: Event based neural radiance field. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 837–847.

Kerbl, B.; Kopanas, G.; Leimkühler, T.; and Drettakis, G. 2023. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4): 1–14.

Liang, Z.; Zhang, Q.; Feng, Y.; Shan, Y.; and Jia, K. 2023. Gs-ir: 3d gaussian splatting for inverse rendering. *arXiv preprint arXiv:2311.16473*.

Low, W. F.; and Lee, G. H. 2023. Robust e-NeRF: NeRF from Sparse & Noisy Events under Non-Uniform Motion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 18335–18346.

Ma, L.; Li, X.; Liao, J.; Zhang, Q.; Wang, X.; Wang, J.; and Sander, P. V. 2022. Deblur-nerf: Neural radiance fields from blurry images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12861–12870.

Ma, Q.; Paudel, D. P.; Chhatkuli, A.; and Van Gool, L. 2023. Deformable Neural Radiance Fields using RGB and Event Cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3590–3600.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1): 99–106.

Mittal, A.; Soundararajan, R.; and Bovik, A. C. 2012. Making a "completely blind" image quality analyzer. *IEEE Signal processing letters*, 20(3): 209–212.

Müller, T.; Evans, A.; Schied, C.; and Keller, A. 2022. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4): 1–15.

Qi, Y.; Zhu, L.; Zhang, Y.; and Li, J. 2023. E2nerf: Event enhanced neural radiance fields from blurry images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 13254–13264.

Rudnev, V.; Elgharib, M.; Theobalt, C.; and Golyanik, V. 2023. EventNeRF: Neural radiance fields from a single colour event camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4992–5002.

Wang, Z.; Wu, S.; Xie, W.; Chen, M.; and Prisacariu, V. A. 2021. NeRF–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*.

Wu, G.; Yi, T.; Fang, J.; Xie, L.; Zhang, X.; Wei, W.; Liu, W.; Tian, Q.; and Wang, X. 2023. 4d gaussian splatting for real-time dynamic scene rendering. *arXiv preprint arXiv:2310.08528*.

Xiong, T.; Wu, J.; He, B.; Fermuller, C.; Aloimonos, Y.; Huang, H.; and Metzler, C. A. 2024. Event3DGS: Event-based 3D Gaussian Splatting for Fast Egomotion. *arXiv preprint arXiv:2406.02972*.

Yang, Z.; Yang, H.; Pan, Z.; Zhu, X.; and Zhang, L. 2023. Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. *arXiv preprint arXiv:2310.10642*.

Yi, T.; Fang, J.; Wu, G.; Xie, L.; Zhang, X.; Liu, W.; Tian, Q.; and Wang, X. 2023. Gaussiandreamer: Fast generation from text to 3d gaussian splatting with point cloud priors. *arXiv preprint arXiv:2310.08529*.

Zhang, K.; Riegler, G.; Snavely, N.; and Koltun, V. 2020. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*.

Zhang, L.; Zhang, L.; and Bovik, A. C. 2015. A feature-enriched completely blind image quality evaluator. *IEEE Transactions on Image Processing*, 24(8): 2579–2591.

Zhao, J.; Xiong, R.; Liu, H.; Zhang, J.; and Huang, T. 2021. Spk2imgnet: Learning to reconstruct dynamic scene from continuous spike stream. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11996–12005.

Zhao, L.; Wang, P.; and Liu, P. 2024. Bad-gaussians: Bundle adjusted deblur gaussian splatting. *arXiv preprint arXiv:2403.11831*.

Zhu, L.; Dong, S.; Huang, T.; and Tian, Y. 2019. A retina-inspired sampling method for visual texture reconstruction. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, 1432–1437. IEEE.

Zhu, L.; Jia, K.; Zhao, Y.; Qi, Y.; Wang, L.; and Huang, H. 2024. SpikeNeRF: Learning Neural Radiance Fields from Continuous Spike Stream. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6285–6295.