BulletTime4D: Towards High Spatio-Temporal Resolution Dynamic Scene Rendering via Spike-Guided Stereo Vision

Yiqian Chang[1,2], Haoran Xu[4,2], Qinghong Ye[3,2], Jianing Li[5], Xuan Wang[1*], Wei Zhang[2], Peixi Peng[3,2*]

[1] Harbin Institute of Technology, Shenzhen, China [2] Peng Cheng Laboratory, China
[3] School of Electronic and Computer Engineering, Shenzhen Graduate School, Peking University, China
[4] Shenzhen Campus of Sun Yat-sen University, China
[5] School of Computer Science, Peking University, China
changyiqian@stu.hit.edu.cn, wangxuan@cs.hitsz.edu.cn, pxpeng@pku.edu.cn

Abstract

High spatio-temporal resolution novel-view scene rendering is crucial for applications such as sports analysis and scientific experiments. However, existing Dynamic Scene Rendering (DSR) approaches typically rely on conventional RGB cameras with limited frame rates, making it difficult to achieve high spatio-temporal resolution. In this paper, we present BulletTime4D, a high spatio-temporal resolution DSR framework, which is the first trial to integrate a spike camera with binocular RGB cameras for dynamic scene reconstruction. Specifically, we first develop a hybrid camera prototype and build a real-world dynamic scene reconstruction dataset. Then, BulletTime4D presents a multi-timescale deformation representation by combining low-frequency spatio-temporal features with high-frequency inter-frame motion features. Finally, a rendering network is designed capable of projecting 4D Gaussians into the spike domain for spike rendering, and a cross-domain supervision strategy is proposed to achieve high-frame-rate texture and color rendering. The results show that BulletTime4D outperforms state-of-the-art methods on both simulated and real-world datasets. In addition, BulletTime4D can synthesize 300 FPS novel-view renderings using stereo RGB cameras at 30 FPS and a single spike camera.

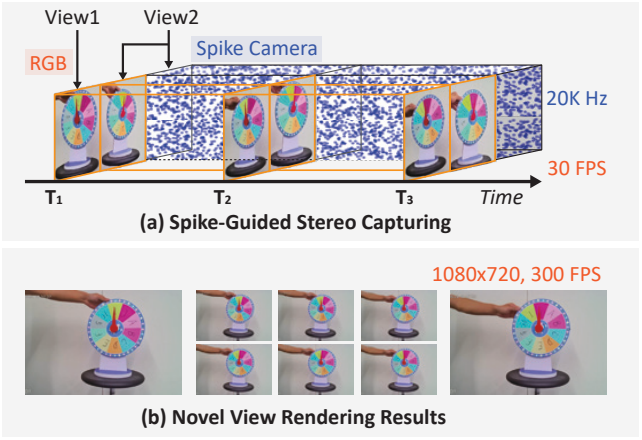

Figure 1: (a) Our hybrid camera system that combines stereo RGB cameras for capturing high-resolution frames with a spike camera for recording ultra-high-speed continuous spike streams. (b) Our BulletTime4D generates high-quality dynamic novel-view synthesis with both high spatial and temporal resolution.

Extended version —
https://github.com/changyq12/BulletTime4D.git

Introduction

Dynamic Scene Rendering (DSR) aims to synthesize images from arbitrary viewpoints and timestamps of a dynamic scene. With the advent of Neural Radiance Fields (NeRF) (Mildenhall et al. 2021) and 3D Gaussian Splatting (3DGS) (Kerbl et al. 2023), DSR methods have greatly improved in quality and diversity. This ongoing progress in DSR has enabled the capture and analysis of object motion from a 3D perspective, especially in applications like sports analysis and scientific experiments.

Existing DSR methods primarily rely on conventional RGB cameras, which may struggle to achieve high-precision reconstruction in high-speed dynamic scenes. A common approach involves using a single moving camera to capture multi-view information, but this requires prior knowledge of the camera's motion trajectory. In contrast, stereo camera setups leverage two-view spatial information, enabling more accurate depth estimation and improved dynamic scene reconstruction. While these frame-based DSR methods (Cao and Johnson 2023; Shao et al. 2023) achieve high spatial resolution in rendering, they depend on low frame rates and suffer from a sharp decline in accuracy under fast motion or blur. As a result, developing new paradigms to overcome the limitations of conventional stereo cameras for high-speed dynamic scene reconstruction remains an open issue.

Spike cameras (Zhu et al. 2019), mimicking the sampling mechanism of the retina, can capture moving scenes at ultra-high temporal frequency. These cameras generate a "spike" when the accumulated photons at a pixel exceed a threshold, enabling a frame-free imaging paradigm that preserves fine visual textures. With a temporal sampling frequency of

*Corresponding authors.

up to 20,000 Hz, spike cameras are well-suited for various high-speed vision tasks. However, they suffer from relatively low spatial resolution (e.g., Vidar camera with 400×250 pixels). While some spike-based DSR methods (Guo et al. 2024; Zhang et al. 2024; Zhu et al. 2024) attempt to synthesize novel viewpoints, most existing approaches focus on static scene reconstruction. Thus, leveraging spike cameras for high spatio-temporal dynamic scene reconstruction remains a largely unexplored challenge.

To address the aforementioned problems, we present **BulletTime4D**, a high spatio-temporal resolution DSR framework, which is the first trial to integrate a spike camera with binocular RGB cameras for dynamic scene reconstruction (see Fig. 1). In fact, the goal of this work is not to design a state-of-the-art frame-based DSR method. On the contrary, we aim at overcoming following challenges: (i) *Lack of camera setup and dataset* – How could we establish a hybrid camera prototype that could simultaneously capture multiple visual streams from a spike camera and stereo RGB cameras? (ii) *Effective model* – How to effectively model 3D dynamic scenes at high spatio-temporal resolution? (iii) *Proper supervision* - How to constrain novel-view rendering at intermediate timestamps where RGB observations are absent?

Toward this end, we first develop a prototype hybrid camera system and establish a real-world high spatio-temporal dynamic scene reconstruction dataset. Then, our BulletTime4D presents a multi-timescale deformation representation by combining low-frequency spatio-temporal features with high-frequency inter-frame motion features. Finally, a rendering network is designed capable of projecting 4D Gaussians into the spike domain for spike rendering, and a cross-domain supervision strategy is proposed to achieve high-frame-rate texture and color rendering. Experimental results demonstrate that our BulletTime4D outperforms state-of-the-art methods on both simulated and real-world datasets, showing its capability to reconstruct high-quality novel view images with high spatio-temporal resolution. In addition, our BulletTime4D can synthesize 300 FPS novel-view renderings using stereo RGB cameras at 30 FPS and a single spike camera. We believe that the novel problem setting will attract further research into this newly identified, yet crucial research direction.

In summary, the main contributions of this work are:

- We propose **BulletTime4D**, a *novel high spatio-temporal resolution dynamic* scene reconstruction framework (i.e.,), which first integrates a spike camera with binocular RGB cameras to synthesize arbitrary novel viewpoints.
- We present an *effective multi-timescale DSR representation* that makes complementary use of hybrid cameras to achieve high spatio-temporal resolution rendering.
- We design a *dynamic spike rendering module* that projects 4D Gaussians into the spike domain, coupled with a cross-domain supervision strategy to synthesize texture and color at high temporal resolution.
- We build a *high spatio-temporal resolution DSR dataset* using our hybrid camera system, and we believe this standardized resource will open new opportunities for research on this challenging problem.

Related Works

Dynamic Scene Rendering on Traditional Cameras

DSR aims to render images from any desired viewpoint and timestamp of a dynamic scene. With the emergence of NeRF (Mildenhall et al. 2021) and 3DGS (Kerbl et al. 2023), the quality of DSR methods has improved significantly, accompanied by a flourishing diversity of methodological developments. DSR methods based on NeRF (Mildenhall et al. 2021) are renowned for their implicit neural scene representation capability. D-NeRF (Pumarola et al. 2021) learns a deformation network that maps all scene deformations to a canonical configuration, achieving high-quality novel views for synthetic datasets. For real-world datasets captured with handheld cameras, Nerfies (Park et al. 2021a) and HyperNeRF (Park et al. 2021b) use per-frame trainable deformation codes, instead of time conditions, to deform the canonical space. Besides, DSR methods based on 3DGS (Kerbl et al. 2023) could significantly boost the rendering speed to a real-time level with the best quality. D3DGS (Yang et al. 2024) uses an MLP to learn the mapping between the center position and timestamps to the offset of dynamic 3D Gaussians in canonical space. 4DGS (Wu et al. 2024a) uses Hexplane to encode and a multi-head Gaussian deformation for decoding. Real-time 4DGS (Yang et al. 2023) decomposes the 4D Gaussian into a time-conditioned 3D Gaussian and a marginal 1D Gaussian. SwingGS (Shaw et al. 2024) introduces a novel paradigm for dynamic scene rendering with temporally local canonical spaces defined in a sliding window fashion. E-D3DGS (Bae et al. 2024) defines the deformation as a function of per-Gaussian embeddings and temporal embeddings. However, these frame-based DSR methods may struggle to produce high-quality renderings in high-speed motion scenarios due to the inherent frame rate limitations of conventional RGB cameras.

Novel View Synthesis on Neuromorphic Cameras

Neuromorphic sensors (e.g., spike cameras (Dong, Huang, and Tian 2021) and event cameras (Gallego et al. 2020)), with their high temporal resolution, have been effectively employed for novel view synthesis in high-speed motion scenarios. For the event camera, EventNeRF (Rudnev et al. 2023), Ev-NeRF (Hwang, Kim, and Kim 2023), and other works (Wu et al. 2024b; Xiong et al. 2024) have explored the reconstruction of a 3D representation from a rapidly moving event camera. More recently, Evagaussians (Yu et al. 2024b) and Eadeblur-gs (Weng et al. 2024) have fused event data and blurry frames to reconstruct high-quality images. However, event cameras could only record relative changes in light intensity. So all these event-based approaches yield limited results due to the absence of texture details in event data. In contrast, for the spike camera, each pixel could respond independently to the accumulation of photons by generating spikes. It records full visual details with ultra-high temporal resolution (i.e., 20,000 Hz). SpikeNeRF (Zhu et al. 2024) and Spike-NeRF (Guo et al. 2024) have achieved higher 3D scene reconstruction quality by combining spike streams and NeRF. In the field of 3DGS, SpikeGS (Yu et al. 2024a) reconstructed view synthesis results from a continu-

ous spike stream. In a harder setting, SpikeNVS (Dai et al. 2024) reconstructed static scenes via a moving synchronized spike-RGB camera, then SpikeGS (Guo et al. 2025) went further with Bayer-pattern spike streams from a color spike camera, Spike4DGS (Ye et al. 2025) deployed 4D Gaussian on multi-view spike cameras. However, there is no established spike-based method for addressing the challenge of rendering dynamic scenes at high spatio-temporal resolution. On this basis, this work proposes a novel hybrid DSR method to achieve high spatio-temporal resolution for DSR.

Preliminary

4D Gaussian Splatting. Gaussian Splatting (Wu et al. 2024a) has been widely used for rendering dynamic scenes. It proposes a network that learns the Gaussian deformation field to predict the deformation of each 3D Gaussian. For input 3D Gaussian $\mathcal{G}$ and time t, a spatio-temporal structure encoder $\mathcal{E}$ and a multi-head Gaussian deformation decoder $\mathcal{D}$ are used for calculating the deformations $\Delta\mathcal{G}$ as:

$$\Delta\mathcal{G} = \mathcal{D}(\mathcal{E}(\mathcal{G}, t)). \tag{1}$$

Spike Camera Sampling. Spike camera is a bio-inspired sensor which records and converts the absolute light intensity at a fairly high frame rate (up to 20,000 Hz) into accumulated voltage through photoreceptors (Zhu et al. 2019; Dong et al. 2019). If the accumulated voltage V reaches the scheduling threshold Θ, a spike will be triggered and V is reset to zero. It can be formulated as follows:

$$V(t) = \int_{t_s}^{t} \sigma \cdot L(t)dt \bmod \Theta, \tag{2}$$

where $L(t)$ represents the instant light intensity at time t, t_s is the moment when the previous spike was emitted, and σ is the constant photoelectric conversion coefficient.

Camera Prototype and Dataset

Spike-Guided Stereo Camera System. To simultaneously acquire high-resolution stereo vision information and high-frame-rate texture variation details, we carefully design an experimental setup for capturing. As shown in Fig. 2, we develop a spike-guided stereo capturing device, which consists of a bio-inspired spike camera and two optical RGB cameras. More Specifically, our hybrid camera prototype combines a spike camera (20,000 Hz and 400×250 resolution) and an optical RGB camera (30 FPS and 1080×720 resolution) using a beam splitter, which separates the incoming light and directs it to two sensors with spatial consistency. This hybrid camera could simultaneously acquire continuous spike streams and discrete RGB frames, ensuring their spatiotemporal synchronization. Then, we deploy another optical RGB camera and arrange it with the Spike-RGB synchronized camera in a stereo configuration. We use the data captured from stereo view to estimate an initial point cloud and train a dynamic 3D Gaussian. In addition, a high-speed optical RGB camera (300 FPS and 1080×720 resolution) is utilized for the evaluation of novel view synthesis. During the data collection process, temporally synchronized recordings are made from all cameras, ensuring that motion is consistently represented across different camera views.

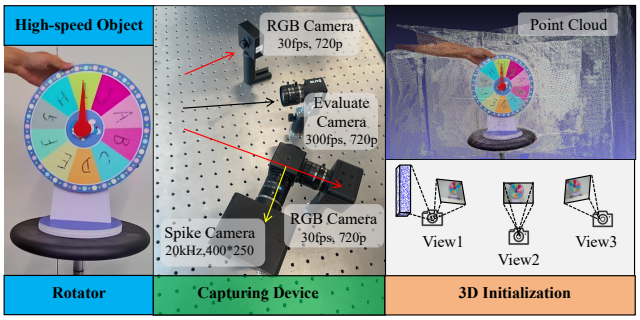

Figure 2: Our spike-guided stereo camera system. We place a high-speed object in front of our proposed capturing device to obtain RGB images and spike streams. After annotation and 3D initialization, we build benchmarks for the high spatio-temporal resolution DSR task.

Real-world and Synthetic Dataset. To establish a comprehensive evaluation benchmark, we construct two DSR datasets encompassing both real-world high-speed motion scenes and highly synthetic environments. *For real-world dataset*, we select small fast-moving objects with noticeable texture variations, such as rotating turntables and spinning cubes. To better evaluate texture reconstruction quality during motion, we add textual patterns to the object surfaces. During data collection, the capturing setup remains fixed while the object moves in front of it. Each sequence records approximately two seconds of motion to generate training and testing sets. *For synthetic dataset*, we simulate a virtual setup that mirrors the real-world device. To explore more semantically meaningful scenarios, we include animated human characters performing various fast motions such as dancing and swinging. Each synthetic sequence also spans around two seconds. The final DSR datasets consist of more than ten real-world moving objects and synthetic animated human sequences. We process the collected data using DUST3R (Wang et al. 2024) to obtain initial point clouds and camera poses for each sequence, and we provide annotations for training and testing splits.

Method

Overview

We aim to achieve dynamic scene rendering (DSR) at high spatio-temporal resolution using spike-guided stereo vision. To this end, we propose a novel DSR method called BulletTime4D, which could be formulated as:

$$\hat{C}_i^t = \mathcal{M}_B\{S, (C_1, V_1), (C_2, V_2), t, V_i\}, \tag{3}$$

where $\mathcal{M}_B$ refers to the proposed BulletTime4D model. S is the captured spike streams, (C_1, V_1) and (C_2, V_2) are the captured RGB frames and viewpoint for training, t and $V_i|i \in \{1, 2, 3\}$ are the target time and view for rendering. $\hat{C}_i^t$ is the rendered image from BulletTime4D, which is expected to achieve performance comparable to high spatio-temporal resolution benchmarks.

To achieve the goal, our BulletTime4D focuses on addressing the two critical challenges:

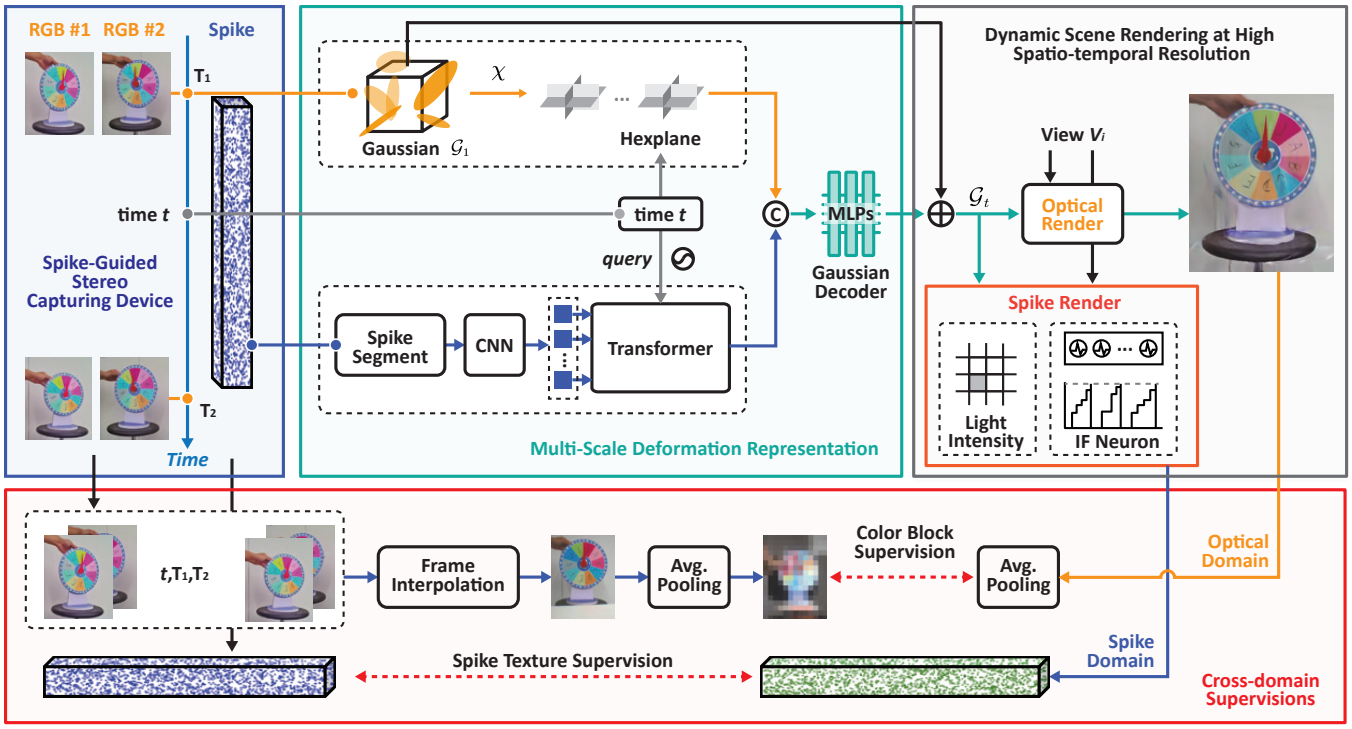

Figure 3: The main framework of our proposed BulletTime4D. BulletTime4D first presents a **Multi-timescale Deformation Representation** by combining low-frequency spatio-temporal features with high-frequency inter-frame motion features. Then, a **Dynamic Spike Rendering Module** is designed to be capable of projecting 4D Gaussians into the spike domain for spike rendering, followed by a **Cross-domain Supervision** strategy to achieve high-frame-rate texture and color rendering.

- **Q1**: How to effectively model the 3D dynamic scenes at high spatial-temporal resolution?
- **Q2**: How to constrain novel-view rendering at intermediate timestamps where low-frequency RGB frames observations are absent?

Multi-Scale Deformation Representation

To solve the challenge **Q1**, we characterize scene motion across two temporal scales: coarse and fine.

- **Coarse-scale Deformation** is achieved by training a dynamic Gaussian field based on RGB keyframes, which primarily captures the relatively stable or slow-changing components of objects or scenes.
- **Fine-scale Deformation** focuses on high-speed motion details and sharp variations. It refines the positions and attributes of Gaussians between two consecutive keyframe times using spike data, thereby generating supplemental deformations.

Through joint training, our proposed multi-scale deformation representation could capture the overall motion trajectory at the coarse temporal scale and compensate for high-speed motion blur and distortion at the fine temporal scale, enabling high spatio-temporal resolution rendering.

Coarse-scale Deformation Representation. Specifically, for a given target time t and canonical Gaussians $\mathcal{G}$, a dedicated spatial-temporal feature could be extracted via a multi-scale deformation network:

$$f_d^m = \mathcal{E}^c(\mathcal{X}, t) + \mathcal{E}^f(\overline{\mathcal{S}}, t), \tag{4}$$

where $\mathcal{E}^c$ and $\mathcal{E}^f$ are the deformation encoders at coarse-scale and fine-scale, $\mathcal{X}$ is the position of Gaussians $\mathcal{G}$, $\overline{\mathcal{S}}$ is a part of the spike stream $\mathcal{S}$ surrounding time t, and f_d^m is the proposed multi-Scale deformation representation.

For $\mathcal{E}^c$, we employ the same spatial-temporal structure encoder as Eq. 1, which contains a small MLP ϕ_d and a multi-resolution HexPlane $\mathcal{H}$:

$$f_d^c = \mathcal{E}^c(\mathcal{X}, t) = \phi_d(\mathcal{H}(\mathcal{X}, t)), \tag{5}$$

where f_d^c is the coarse-scale deformation representation.

Fine-scale Deformation Representation. For $\mathcal{E}^f$, we extract the texture features of the spike stream around time t and utilize a Transformer (Vaswani et al. 2017) to infer the temporal relationship between texture features, therefore generating supplemental motion details.

Firstly, we obtain spike streams surrounding time t:

$$\overline{\mathcal{S}} = \mathcal{S}[T_1 : T_2], \tag{6}$$

where T_1 and T_2 are the two nearest RGB frame timestamps surrounding the target time t, and $\mathcal{S}[T_1 : T_2]$ means the captured spike stream from time T_1 to T_2.

Next, a trained convolution network Φ (with frozen parameters) is utilized for extracting texture features from each spike in the stream $\overline{\mathcal{S}}$:

$$f_{tex}(i) = \Phi(\mathcal{S}[t_i]), \tag{7}$$

where $t_i \in [T_1, T_2]$ and the texture features $f_{tex}(i)$ extracted from each spike form a feature sequence $\overline{F}_{tex}$.

To extract detailed temporal variation features at the target time t, we first encode the texture feature sequence and target time as tokens:

$$\overline{F}^e_{tex} = \overline{F}_{tex} + \mathrm{PE}(\overline{F}_{tex}),\ Q^e_t = \mathrm{PE}(t), \tag{8}$$

where $\mathrm{PE}(\cdot)$ is the sinusoidal positional encoding. Then, a Multi-Head Attention (MHA) is applied to capture temporal correlations in the spike stream features F^e_{tex}, followed by a Cross Attention (CA) for predicting the motion dynamics at query time t:

$$f^f_d = \mathrm{CA}(\mathrm{MHA}(\overline{F}^e_{tex}), Q^e_t), \tag{9}$$

where f^f_d is the fine-scale deformation representation.

Finally, we could get the multi-scale deformation representation:

$$f^m_d = f^c_d + f^f_d. \tag{10}$$

This deformation representation would be sent to the multi-head Gaussian deformation decoder in Eq. 1 to get a high spatio-temporal-resolution deformable Gaussian field $\mathcal{G}_t$.

Dynamic Spike Rendering Module

To solve the challenge **Q2**, we need to leverage the real continuous spike stream between two low-frame-rate RGB frames as supervision, compensating for the absence of ground-truth RGB in intermediate frame rendering. However, to the best of our knowledge, there are currently few end-to-end method that directly renders dynamic Gaussians into spike streams. Therefore, we propose **a dynamic spike rendering module** that projects the deformable Gaussian into the spike domain and renders spike information directly.

Specifically, due to the use of a beam splitter, the light beams reaching the spike camera and the RGB camera at view V_1 have identical intensity, which could be obtained through Gaussian rasterization: $\mathcal{G}_t$:

$$I(t) = \sum_i c^i_t \alpha^i_t \prod_{j=1}^{i-1} (1 - \alpha^j_t), \tag{11}$$

where c^i_t and α^i_t are the color and density of the i-th Gaussian in the deformable Gaussian field $\mathcal{G}_t$.

Then, due to the intrinsic parameters, the actual light intensity received by the spike camera should be modified by changing the resolution and adding sensor noise:

$$I^s(t) = \mathrm{Downsample(I(t))}, \tag{12}$$

$$\hat{I}^s(t) == \frac{1}{\frac{Q_r}{I^s(t)+N_p+N_d} + N_{rnu} + N_q} + N_c, \tag{13}$$

where $\hat{I}^s(t)$ is the actual light intensity received by the spike camera, $\mathrm{Downsample}(\cdot)$ means downsampling the light intensity from RGB resolution to spike resolution. Q_r is the relative quantity matrix of electric charge, N_p, N_d,N_{rnu},N_q, and N_c represent shot noise, dark current noise, response nonuniformity noise, quantization noise, and truncation noise, respectively (Zhu et al. 2023).

According to the sampling mechanism of spike cameras, the spike data at t is accumulated and fired from the continuous light intensities in a time interval $(t - \tau, t)$. We have proved in the appendix that this accumulation and firing process could be simulated by using a spike neuron named Integrate-and-Fire (IF) (Gerstner et al. 2014; Fang et al. 2021):

$$\hat{S}(t) = \mathrm{IF}(\overline{I}^s(t - \tau, t)), \tag{14}$$

where $\overline{I}^s(t-\tau, t) = \{\hat{I}^s(t-\tau), ..., \hat{I}^s(t)\}$ is the light intensity sequence. Since the time interval τ is extremely short, the dynamic scene could be approximately seen as static during this period:

$$\overline{I}^s(t - \tau, t) \simeq \{\hat{I}^s(t)..., \hat{I}^s(t)\}, \tag{15}$$

where $\{\hat{I}^s(t)..., \hat{I}^s(t)\}$ is a sequence with m unchanged light intensities, m is a preset number. Thus, we could calculate the final rendered spike data:

$$\hat{S}(t) \simeq \mathrm{IF}(\{\hat{I}^s(t)..., \hat{I}^s(t)\}). \tag{16}$$

Cross-domain Supervisions

Spike Supervision on the Specific View. In the view V_1, our Spike-RGB Stereo Camera could capture real spike data at a high frame rate. Therefore, we project the dynamic Gaussians onto this view to render the corresponding spike information and compare the rendered 2D spike output with the actual captured 2D spike stream, serving as a spike texture supervision for high-frame-rate motion:

$$\mathcal{L}^{\mathrm{spike}}_t = ||S(t) - \hat{S}(t)||_1, \tag{17}$$

where $S(t)$ is the captured spike at time t in view V_1.

Optical Supervision on Each View. Since our goal is to render high spatio-temporal resolution color images, color constraints are also needed in addition to motion texture constraints. However, the RGB images at high frame rates are absent, so we use frame interpolation to provide approximate color supervision.

Specifically, we first extract the two nearest RGB frame timestamps surrounding the target time t as T_1 and T_2. For view V_i, its RGB frames at time T_1 and T_2 could be captured from the optical camera and defined as $C^{T_1}_i$ and $C^{T_2}_i$. We use the video interpolation method RIFE (Huang et al. 2022) to generate the interpolated image $\tilde{C}^t_i$ at time t:

$$\tilde{C}^t_i = \mathrm{RIFE}(C^{T_1}_i, C^{T_2}_i, \epsilon), \tag{18}$$

$$\epsilon = (t - T_1)/(T_2 - T_1), t \in (T_1, T_2), \tag{19}$$

where $\mathrm{RIFE}(\cdot)$ is the trained interpolation network and ϵ is the coefficient of linear interpolation.

It is important to note that the interpolated image is generated via linear interpolation from low-frame-rate RGB images, which cannot faithfully capture the fine-grained motion details and texture variations present in true high-frame-rate imagery. Therefore, we apply average pooling to the interpolated image, divide it into multiple patches, and use only the average color of each patch as a block supervision:

$$\mathcal{L}^{\mathrm{optical}}_t = ||\mathrm{Avepool}(\tilde{C}^t_i) - \mathrm{Avepool}(\hat{C}^t_i)||_1, \tag{20}$$

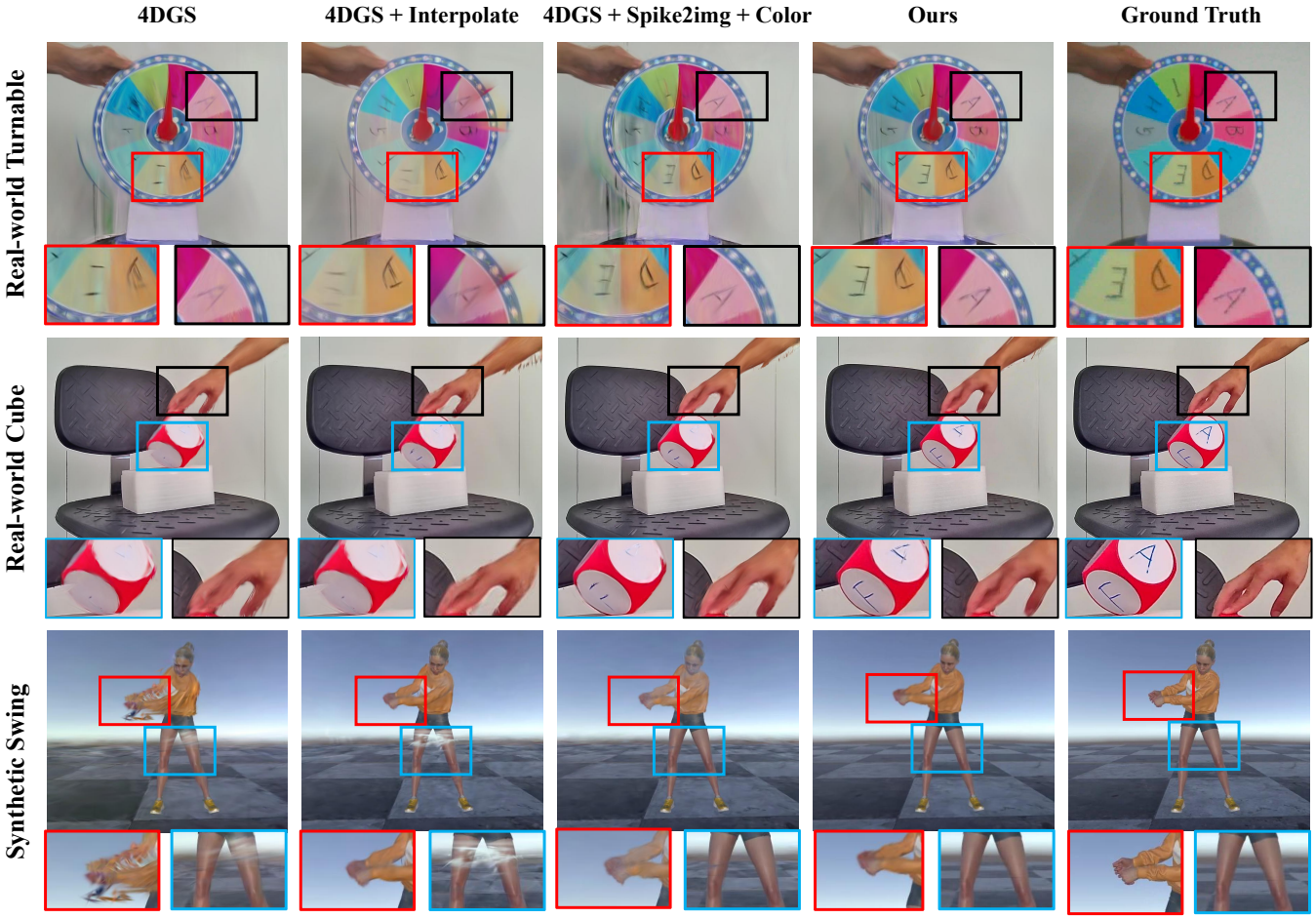

Figure 4: Quantitative comparison with other methods on the dataset of our real-world and synthetic datasets. We mainly compare our method with some SOTA approaches. In contrast, our method delivers both superior outlines and clear details.

Therefore, the final loss of our BulletTime4D is:

$$\mathcal{L}_{\text{Bullet4D}} = \mathcal{L}_t^{\text{spike}} + \mathcal{L}_t^{\text{optical}} + \mathcal{L}_t^{\text{frame}}, \tag{21}$$

$$\mathcal{L}_t^{\text{frame}} = \beta * ||C_i^t - \hat{C}_i^t||_1, \tag{22}$$

$$\beta = \begin{cases} 1, \text{if t} \in \{\text{T}_1, \text{T}_2, ...\}, \\ 0, \text{if t} \notin \{\text{T}_1, \text{T}_2, ...\}, \end{cases} \tag{23}$$

where $\{T_1, T_2, ...\}$ means the time sequence for low-frequency RGB frames and C_i^t is the captured real RGB frames in view V_i at time t.

Experiments

Experimental Setups

Competitors. Due to the relative lack of specific methods for dynamic novel view synthesis at high spatio-temporal resolution, we deploy the comparison using some two-stage rendering approaches. First, we choose some direct inter-frame generation approaches: "Repeat" (extending a video from 30fps to 300fps by repeating each frame 10 times), "Interpolate" (using the frame interpolation method such as SimpleFlow (Tao et al. 2012) and RIFE (Huang et al. 2022)), "Spike2img + Color" (using spike information to synthesize grayscale images such as TFI, TFP (Zhu et al. 2019) and deploy ColorTransfer (Reinhard et al. 2002)), and None (keeping the inter-frame as empty information), and then use them to generate inter-frame RGB frames at the target high frequency for training. Second, we conduct SOTA dynamic scene rendering approaches on these generated training frames. We choose 4DGS (Wu et al. 2024a) and E-D3DGS (Bae et al. 2024). All experiments are deployed on our proposed real-world and synthetic datasets.

Implementation Details. Firstly, we use the multi-scale deformation representation to get a dedicated deformable Gaussian. The ϕ_d is a one-layer MLP with an output dimension of 256. We choose the ResNet50 (He et al. 2016) as the trained convolution network. For the Transformer, the dimension of the PE is 256. The MHA encoder uses 4 heads with 512-dimensional forward features. The dimension of the temporal query and output of the CA decoder is 256. Secondly, a dynamic spike rendering module is utilized to project the deformable Gaussian field into the spike domain and render spike information directly. We use the bilinear interpolation as the method for downsampling. The parameters of spike camera noises are obtained by capturing a uni-

Methods	PSNR↑	SSIM↑	LPIPS↓
4DGS	17.83	84.0	0.313
+ Repeat	18.04	84.2	0.311
+ Interpolate	18.45	84.5	0.309
+ Spike2img + Color	18.95	85.1	0.306
E-D3DGS	18.08	84.3	0.311
+ Repeat	18.15	84.3	0.310
+ Interpolate	18.82	84.9	0.307
+ Spike2img + Color	19.25	85.3	0.303
Ours	**20.12**	**86.3**	**0.297**

Table 1: Average quantitative evaluation on the real-world dataset.Unit: PSNR-dB ↑ , SSIM ↑ , LPIPS ↓.

Methods	PSNR↑	SSIM↑	LPIPS↓
4DGS	27.88	89.2	0.213
+ Repeat	28.55	89.5	0.206
+ Interpolate	29.25	89.9	0.198
+ Spike2img + Color	29.38	89.8	0.200
E-D3DGS	28.12	89.4	0.210
+ Repeat	28.66	89.7	0.205
+ Interpolate	29.45	90.0	0.196
+ Spike2img + Color	29.60	90.3	0.199
Ours	**30.48**	**90.7**	**0.187**

Table 2: Average quantitative evaluation on the synthetic dataset.Unit: PSNR-dB ↑ , SSIM ↑ , LPIPS ↓.

form light scene and recording the intensity. m is set to 32. The kernel size and stride of the average pooling $Avepool$ are set to 4. The total experiments are conducted on a single NVIDIA GTX 4090 with PyTorch, and the optimization for a single scene typically takes about 30 minutes to converge and 40 FPS when rendering. For the metrics of both Real-world and synthetic datasets, we employ three widely-used image quality assessment metrics, PSNR (Wu et al. 2024a), SSIM (Wang et al. 2004), and LPIPS (Zhang et al. 2018).

Performance Comparisons

Comparisons on Real-world Dataset. We present a detailed comparison of our method against those two-stage rendering approaches on several example high-speed scenes. As shown in Table 1, our method achieves superior performance compared to the SOTA rendering approaches Specifically, for the comparison with 4DGS, (i) in the 1st row, we use only 30fps RGB frames to supervise the 4DGS; (ii) in the 2nd row, we expand the train data from 30fps to 300fps by repeating each frame 10 times; (iii) in the 3rd row, we use RIFE to interpolate intermediate frames for 4DGS supervision; (iv) in the 4th row, we use TFI to translate the spike into color images and compensate the loss of intermediate frame supervision in 4DGS. In qualitative experiments, we focus on the generated texture details. Fig. 4 shows that we could generate high-quality texture details. For example, our rendering could **generate clear alphabets "A, B, C…"** on the Turntable scene while others could not.

Methods	PSNR↑
Coarse DSR + $\mathcal{L}_t^{\text{frame}}$	17.83
Coarse DSR + Fine DSR + $\mathcal{L}_t^{\text{frame}}$	18.50
Multi DSR + Spike Render + $\mathcal{L}_t^{\text{spike}}$ + $\mathcal{L}_t^{\text{frame}}$	19.65
Multi DSR + Spike Render + $\mathcal{L}_{Bullet4D}$	**20.12**

Table 3: The contribution of each component. Evaluated on the real-world dataset.

Comparisons on Synthetic Dataset. As for the synthetic dataset, we deploy the same comparison on different animated human motions. As demonstrated in Table 2, our method outperforms the SOTA two-stage rendering methods. We also present the qualitative results in Fig. 4, where: (i) 4DGS purely based on 30fps RGB frames almost failed to reconstruct the outlines and details of each scene. (ii) "4DGS + Interpolate" and "4DGS + Spike2img + Color" could only reconstruct the outlines with texture details missing. (iii) In contrast, our method demonstrates superior performance, generating clearer and more detailed novel views.

Ablation Study

Contribution of Each Component. We conduct an ablation study to assess the contribution of each component in our BulletTime4D. As shown in Table 3, the results are evaluated by adding each module gradually: (i) The baseline results are shown in the first row, which uses only RGB frames for supervision to train a coarse deformation. (ii) In the second row, we incorporate the fine-scale deformation representation to build a complete multi-scale deformation. (iii) The third row indicates that we render the deformable Gaussians into the spike domain and deploy the spike texture loss. Compared with the baseline, the result in this line has the largest improvement, indicating that the "spike rendering with spike supervision" is the most effective module. (iv) The last row is our full model, which incorporates another color block loss. Performance validates the effectiveness of our design for the spike-guided DSR task.

Conclusion

This paper presents BulletTime4D, a high spatio-temporal resolution DSR framework, which is the first to integrate a spike camera with binocular RGB cameras for dynamic scene reconstruction. We develop a hybrid camera prototype and propose a multi-timescale deformation representation that combines low-frequency spatio-temporal features with high-frequency inter-frame motion features, a spike-rendering module that projects 4D Gaussians into the spike domain. Experimental results show that BulletTime4D outperforms state-of-the-art methods and can synthesize 300 FPS novel-view renderings using stereo RGB cameras operating at 30 FPS together with a single spike camera. We believe this hybrid prototype will provide insight into next-generation high-speed cameras.

Acknowledgments

The study was funded by the Shenzhen Science and Technology Program (KQTD20240729102051063), the National Natural Science Foundation of China under contracts No. 62422602, No. 62372010, No. 62425101, No. 62332002, No. 62372010, No. 62088102, the Major Key Project of Pengcheng Laboratory (PCL2025A02), and the Key Laboratory of Guangdong Province (2022B1212010005). Computing support was provided by Pengcheng Cloudbrain.

References

Bae, J.; Kim, S.; Yun, Y.; Lee, H.; Bang, G.; and Uh, Y. 2024. Per-Gaussian Embedding-Based Deformation for Deformable 3D Gaussian Splatting. *arXiv preprint arXiv:2404.03613*.

Cao, A.; and Johnson, J. 2023. Hexplane: A fast representation for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 130–141.

Dai, G.; Wang, Z.; Xu, Q.; Lu, M.; Chen, W.; Shi, B.; Zhang, S.; and Huang, T. 2024. Spikenvs: Enhancing novel view synthesis from blurry images via spike camera. *arXiv preprint arXiv:2404.06710*.

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

Dong, S.; Zhu, L.; Xu, D.; Tian, Y.; and Huang, T. 2019. An efficient coding method for spike camera using inter-spike intervals. *arXiv preprint arXiv:1912.09669*.

Fang, W.; Yu, Z.; Chen, Y.; Huang, T.; Masquelier, T.; and Tian, Y. 2021. Deep Residual Learning in Spiking Neural Networks. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 21056–21069. Curran Associates, Inc.

Gallego, G.; Delbrück, T.; Orchard, G.; Bartolozzi, C.; Taba, B.; Censi, A.; Leutenegger, S.; Davison, A. J.; Conradt, J.; Daniilidis, K.; et al. 2020. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1): 154–180.

Gerstner, W.; Kistler, W. M.; Naud, R.; and Paninski, L. 2014. *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press.

Guo, Y.; Bai, Y.; Hu, L.; Liu, M.; Guo, Z.; Ma, L.; and Huang, T. 2024. Spike-nerf: Neural radiance field based on spike camera. In *IEEE International Conference on Multimedia and Expo*, 1–6. IEEE.

Guo, Y.; Hu, L.; Bai, Y.; Yao, J.; Ma, L.; and Huang, T. 2025. Spikegs: Reconstruct 3d scene captured by a fast-moving bio-inspired camera. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 3293–3301.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Huang, Z.; Zhang, T.; Heng, W.; Shi, B.; and Zhou, S. 2022. Real-time intermediate flow estimation for video frame interpolation. In *European Conference on Computer Vision*, 624–642. Springer.

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.

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.

Park, K.; Sinha, U.; Barron, J. T.; Bouaziz, S.; Goldman, D. B.; Seitz, S. M.; and Martin-Brualla, R. 2021a. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5865–5874.

Park, K.; Sinha, U.; Hedman, P.; Barron, J. T.; Bouaziz, S.; Goldman, D. B.; Martin-Brualla, R.; and Seitz, S. M. 2021b. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228*.

Pumarola, A.; Corona, E.; Pons-Moll, G.; and Moreno-Noguer, F. 2021. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10318–10327.

Reinhard, E.; Adhikhmin, M.; Gooch, B.; and Shirley, P. 2002. Color transfer between images. *IEEE Computer Graphics and Applications*, 21(5): 34–41.

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.

Shao, R.; Zheng, Z.; Tu, H.; Liu, B.; Zhang, H.; and Liu, Y. 2023. Tensor4d: Efficient neural 4d decomposition for high-fidelity dynamic reconstruction and rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16632–16642.

Shaw, R.; Nazarczuk, M.; Song, J.; Moreau, A.; Catley-Chandar, S.; Dhamo, H.; and Pérez-Pellitero, E. 2024. Swings: sliding windows for dynamic 3D gaussian splatting. In *European Conference on Computer Vision*, 37–54. Springer.

Tao, M.; Bai, J.; Kohli, P.; and Paris, S. 2012. SimpleFlow: A Non-iterative, Sublinear Optical Flow Algorithm. In *Computer Graphics Forum*, volume 31, 345–353. Wiley Online Library.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Wang, S.; Leroy, V.; Cabon, Y.; Chidlovskii, B.; and Revaud, J. 2024. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20697–20709.

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612.

Weng, Y.; Shen, Z.; Chen, R.; Wang, Q.; and Wang, J. 2024. Eadeblur-gs: Event assisted 3d deblur reconstruction with gaussian splatting. *arXiv preprint arXiv:2407.13520*.

Wu, G.; Yi, T.; Fang, J.; Xie, L.; Zhang, X.; Wei, W.; Liu, W.; Tian, Q.; and Wang, X. 2024a. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20310–20320.

Wu, J.; Zhu, S.; Wang, C.; and Lam, E. Y. 2024b. Ev-gs: Event-based gaussian splatting for efficient and accurate radiance field rendering. In *IEEE 34th International Workshop on Machine Learning for Signal Processing*, 1–6. IEEE.

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 e-prints*, arXiv–2406.

Yang, Z.; Gao, X.; Zhou, W.; Jiao, S.; Zhang, Y.; and Jin, X. 2024. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20331–20341.

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

Ye, Q.; Chang, Y.; Li, J.; Xu, H.; Wang, X.; Zhang, W.; Tian, Y.; and Peng, P. 2025. Spike4DGS: Towards High-Speed Dynamic Scene Rendering with 4D Gaussian Splatting via a Spike Camera Array. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Yu, J.; Peng, X.; Lu, Z.; Kneip, L.; and Wang, Y. 2024a. Spikegs: Learning 3d gaussian fields from continuous spike stream. In *Proceedings of the Asian Conference on Computer Vision*, 4280–4298.

Yu, W.; Feng, C.; Tang, J.; Yang, J.; Tang, Z.; Jia, X.; Yang, Y.; Yuan, L.; and Tian, Y. 2024b. Evagaussians: Event stream assisted gaussian splatting from blurry images. *arXiv preprint arXiv:2405.20224*.

Zhang, J.; Chen, K.; Chen, S.; Zheng, Y.; Huang, T.; and Yu, Z. 2024. Spikegs: 3d gaussian splatting from spike streams with high-speed camera motion. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 9194–9203.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern recognition*, 586–595.

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

Zhu, L.; Zheng, Y.; Geng, M.; Wang, L.; and Huang, H. 2023. Recurrent spike-based image restoration under general illumination. In *Proceedings of the 31st ACM International Conference on Multimedia*, 8251–8260.