High Dynamic Range Imaging with Time-Encoding Spike Camera

Zhenkun Zhu[1] **Ruiqin Xiong**[1]* **Jiyu Xie**[2] **Yuanlin Wang**[1] **Xinfeng Zhang**[3] **Tiejun Huang**[1]
[1]State Key Laboratory of Multimedia Information Processing,
School of Computer Science, Peking University
[2]Shanghai Radio Equipment Research Institute, Shanghai, China
[3]School of Computer Science and Technology, University of Chinese Academy of Sciences
`{zkzhu, wangyuanlin}@stu.pku.edu.cn`, `{rqxiong, tjhuang}@pku.edu.cn`,
`xjy646@mail.ustc.edu.cn`, `xfzhang@ucas.ac.cn`

Abstract

As a bio-inspired vision sensor, spike camera records light intensity by accumulating photons and firing a spike once a preset threshold is reached. For high-light regions, the accumulated photons may reach the threshold multiple times within a readout interval, while only one spike can be stored and read out, resulting in incorrect intensity representation and a limited dynamic range. Multi-level (ML) spike camera enhances the dynamic range by introducing a spike-firing counter (SFC) to count spikes within each readout interval for each pixel, and uses different spike symbols to represent the arrival of different amounts of photons. However, when the light intensity becomes even higher, each pixel requires an SFC with a higher bit depth, causing great cost to the manufacturing process. To address these issues, we propose time-encoding (TE) spike camera, which transforms the counting of spikes to recording of the time at which a specific number of spikes (i.e., an overflow) is reached. To encode time information with as few bits as possible, instead of directly utilising a timer, we leverage a periodic timing signal with a higher frequency than the readout signal. Then the recording of overflow moment can be transformed into recording the number of accumulated timing signal cycles until the overflow occurs. Additionally, we propose an image reconstruction scheme for TE spike camera, which leverages the multi-scale gradient features of spike data. This scheme includes a similarity-based pyramid alignment module to align spike streams across the temporal domain and a light intensity-based refinement module, which utilises the guidance of light intensity to fuse spatial features of the spike data. Experimental results demonstrate that TE spike camera effectively improves the dynamic range of spike camera. The source codes and datasets are available at https://github.com/zkzhu123/TESC.

1 Introduction

In applications such as autonomous driving and unmanned aerial vehicles, high-speed and high-dynamic-range (HDR) scenes [26; 28] frequently occur. Some specialised sensors designed for these scenarios have a bit depth exceeding 20 to accommodate dynamic ranges of 120dB or even higher. How to achieve effective imaging in high-speed and HDR scenes has become a key challenge. Conventional digital cameras usually require static scenes to achieve high-quality imaging results, as object motion during the exposure time leads to motion blur. Consequently, the applicability of conventional cameras in high-speed scenes is limited.

*Corresponding author

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

As a recently invented retinal-based camera, spike camera [19; 60; 65; 7] records light intensity by continuously accumulating photons and firing a spike when reaching a preset threshold. With a readout frequency up to 40k Hz, spike camera demonstrates significant advantages in capturing high-speed scenes. However, its support for HDR is still insufficient, as it can only indicate whether a spike is fired at a readout interval, i.e., whether a certain amount of photons have arrived within the interval. In HDR scenes [22; 12; 25; 54; 53], those high-light regions may trigger multiple spikes within a readout interval, while spike camera can only store and read out a single spike per readout interval. This prevents spike camera from accurately recording the light intensity in high-light regions.

To address these challenges, Zhu et al. [67] propose multi-level (ML) spike camera, which incorporates a spike-firing counter (SFC) to count the number of spikes. At the readout moment, ML spike selects a spike symbol as output based on the spike count. By mapping different brightness levels to distinct spike symbols, ML spike camera achieves a more precise encoding of light intensity.

However, in ultra-HDR scenes, the spike count can become excessively large, making counting and representation costly and inefficient. To this end, we propose time-encoding (TE) spike camera. Similar to ML spike camera, TE spike camera incorporates an SFC to count spikes. TE spike camera sets a maximum spike count (overflow) per readout interval. Once the overflow occurs, TE spike camera stops spike counting and instead uses the time required to reach the maximum spike count to represent the light intensity. To achieve the goal of encoding time information with as few bits as possible, instead of directly utilising a timer, we exploit two periodic clock signals that inherently exist in the spike camera: the readout signal and the higher-frequency timing signal. Since the total number of timing signal cycles within each readout interval can be predetermined based on the frequency ratio between the two signals, the recording of the overflow moment can be transformed into recording the number of accumulated timing signal cycles until the overflow occurs.

Reconstructing high-quality images from TE spike streams is one of the key tasks for TE spike camera. To fully exploit the temporal-spatial information of TE spike streams and mitigate adverse effects, such as photon shot noise and quantisation noise, we propose an encoder to extract texture and gradient features from the spike streams across different receptive fields. We propose a similarity-based pyramid alignment module to align the spike streams among temporal domains in a coarse-to-fine manner. Moreover, a light intensity-based refinement module is proposed to utilise the light intensity to guide the fusion of spatial features in the spike streams. Experimental results demonstrate that TE spike camera effectively enhances the dynamic range of spike camera, and the proposed reconstruction method outperforms other methods, achieving superior results.

2 Related Work

HDR Imaging. For conventional digital cameras, the most straightforward approach to increasing dynamic range is to enhance the full well capacity of pixels [44; 34; 41]. Multiple conversion gain techniques [39; 42; 18] extend dynamic range by applying high conversion gain for low-light regions and low conversion gain for high-light regions. Multi-exposure imaging [14; 56; 49] captures the same scene with different exposure durations and then merges these samples to construct an HDR image. The time-to-saturation technique [32; 20; 27] estimates the intensity of saturated pixels by recording the time at which pixel saturation occurs, thus extending the effective dynamic range. Other technologies such as single-photon avalanche diodes (SPADs) [37; 36; 38] and linear–logarithmic pixels [30; 23; 43] have also been employed to improve the dynamic range of image sensors.

HDR Reconstruction for Neuromorphic camera. Neuromorphic camera can be mainly divided into event camera [57; 45; 4; 1; 48; 33] and spike camera [59; 61; 51; 63; 11; 50; 10; 8; 64; 6; 9; 46; 47]. Zhang et al. [58] design an encoder-decoder network, which combines spiking neural networks (SNNs) and convolutional neural networks (CNNs), effectively generating clear visual images, even for occluded or obscured targets. Zou et al. [68] focus on the reconstruction of high-speed HDR videos based on event streams. They introduce convolutional recurrent neural networks and employ temporal consistency loss to ensure accurate and consistent reconstruction of high-speed events in HDR video sequences. Han et al. [16] introduce the NeurImg-HDR framework, which uses HDR intensity images from neuromorphic cameras to guide traditional RGB images in the luminance domain, enhancing HDR output. Expanding upon this, Han et al. [15] develop NeurImg-HDR+, which extends their method to support HDR video generation and high-resolution reconstruction.

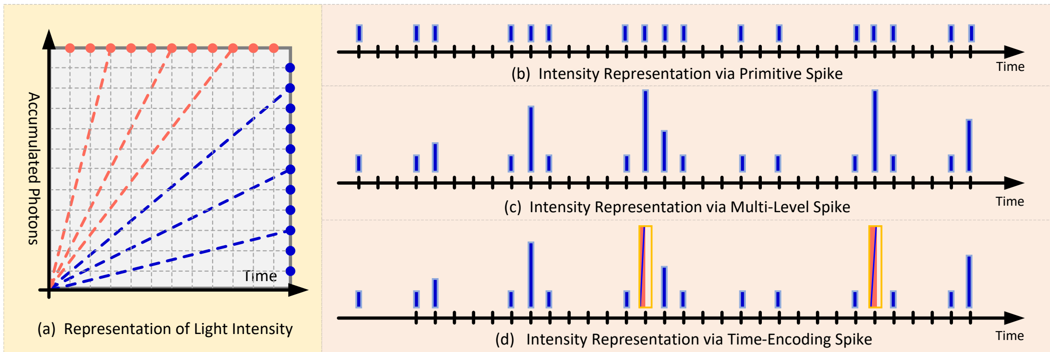

Figure 1: Mechanisms of light intensity representation with neuromorphic camera. (a) Neuromorphic camera uses the ratio of accumulated photons to time, i.e. the photon arrival rate, to represent light intensity. Two commonly adopted strategies are: (1) fixing the integration time and recording the amount of accumulated photons, and (2) fixing the amount of accumulated photons and measuring the time required to reach that amount. (b) Each spike represents a fixed amount of photons, this mechanism uses the spike interval to represent intensity. (c) This mechanism uses multi-level spike symbols to represent different amounts of photons. (d) This mechanism represents intensity by recording the time required for photon accumulation to reach a specific photon amount.

Comparison with Event camera and SPAD. Unlike spike camera, which continuously records light intensity, event camera only records the changes of light intensity, so it is difficult for event camera to perceive static objects. Cao et al. [3] focus on low- and moderate-brightness regimes (e.g., room light, outdoor sunset), where random fluctuations in the photon arrival process are the dominant source of noise. However, in high-brightness environments (e.g., outdoor daylight), leakage noise events become more prevalent, which are not modelled in their work. Therefore, the performance of such approaches in HDR scenarios remains uncertain. He et al. [17] propose an artificial micro-saccade-enhanced event camera that actively senses static scenes using a rotating wedge prism in front of the event camera. While effective in enabling the perception of static objects, this mechanism may face challenges during the initial driving phase, where both static and fast-moving objects are present. Moreover, the additional mechanical components and the resulting complex data structure may introduce higher energy consumption and increased computational burden.

SPAD sensors estimate the total number of incident photons during an interval by counting the number of arrived photons within the exposure time. Sharma et al. [40] and Liu et al. [29] implicitly rely on the assumption that the number of arrived photons during the exposure time does not exceed the maximum countable capacity of the SPAD. However, when the incident photon count surpasses a certain threshold under high-illuminance conditions, conventional SPADs encounter difficulties in accurate photon counting. This limitation leads to image white-out, where bright regions are saturated. Moreover, the avalanche multiplication process in SPADs not only amplifies the signal but also inherently amplifies the noise, leading to excessive noise levels that can severely compromise the overall imaging quality, particularly under high-noise conditions.

3 Motivation

Two Approaches for Intensity Representation. Neuromorphic camera uses the ratio of accumulated photons to time, i.e. the photon arrival rate, to represent light intensity. As illustrated in Fig. 1(a), two common approaches are typically adopted. The first approach accumulates the arrived photons N_p over a fixed time interval T, yielding a photon arrival rate of $\frac{N_p}{T}$. The second approach records the time duration T_t required for the accumulated photons to reach a fixed amount N_θ, resulting in a photon arrival rate of $\frac{N_\theta}{T_t}$.

Intensity Representation via Primitive Spikes. A primitive mechanism involves accumulating photons and comparing them with a predefined threshold θ. Once the threshold is reached, a spike is fired, setting up a spike flag, and the accumulated photons are immediately released. The spike flag is periodically read out as output. In this mechanism, as shown in Fig. 1 (b), each spike corresponds to a fixed amount of photons, i.e. N_θ is fixed. This mechanism is a simple version of the second approach, where the time between spikes, that is, the spike interval, reflects the time required to accumulate N_θ photons. Consequently, the spike interval inversely correlates with light intensity and becomes shorter as the intensity increases. Denote a readout interval as T_r. In theory, the spike interval can be any integer multiple of T_r. However, in high-speed HDR scenes, the light intensity within a spike interval may vary significantly, and an excessively long spike interval can lead to degraded image quality. Assuming that the maximum acceptable spike interval is $N_s T_r$, the intensity I_1 that can be represented by this mechanism is given by:

$$I_1 = \{\frac{N_\theta}{nT_r} | n \in \mathbb{N}^+, n \le N_s\}. \tag{1}$$

Intensity Representation via Multi-Level Spikes. However, when the light intensity becomes very high, specifically when the intensity $I_2 > \frac{N_\theta}{T_r}$, multiple spikes would be fired within a readout interval. In this case, the primitive mechanism fails. To address this, a multi-level spike mechanism is introduced. This mechanism not only generates spikes but also counts them. At the readout moment, the spikes that have been fired but have not yet been read out will be encoded and output as a spike symbol with an amplitude value. As illustrated in Fig. 1 (c), the height of each spike symbol corresponds to the number of spikes fired within the readout interval. This mechanism combines elements of both the first and second approaches: it uses the spike count per readout interval to represent high-intensity regions, while relying on spike intervals to represent low-intensity regions. Assuming that the bit depth of the counter is B, the intensity I_2 that can be represented by this mechanism is given by:

$$I_2 = \{\frac{bN_\theta}{nT_r} | n, b \in \mathbb{N}^+, n \le N_s, b < 2^B\}. \tag{2}$$

Intensity Representation via Time-Encoding Spikes. However, in ultra-HDR scenes, the number of fired spikes can become excessively large, requiring a spike counter with a high bit depth. This poses a significant challenge for maintaining compact pixel sizes. To overcome this, a time-encoding mechanism is introduced. Specifically, by defining a maximum spike count P_{max} within a readout interval, once the spike count reaches P_{max}, the representation of light intensity shifts from spike count to the time T_M required to reach P_{max}. T_M is then encoded and output, providing an alternative representation of intensity through time information. The intensity I_3 that can be represented by this mechanism is given by:

$$I_3 = \begin{cases} \dfrac{bN_\theta}{nT_r}, & \text{no overflow,} \\ \dfrac{P_{\text{max}} N_\theta}{T_r} \cdot \dfrac{T_r}{T_M}, & \text{overflow,} \end{cases} \tag{3}$$

where $n, b \in \mathbb{N}^+$, $n \le N_s$, $b < P_{\text{max}}$, and $0 < T_M \le T_r$.

4 Time-Encoding Spike Camera

4.1 Working Mechanism

The working mechanism of time-encoding (TE) spike camera is shown in Fig. 2.

Spike Generation. Each pixel of TE spike camera independently captures the incoming photons and converts them into electric charges for accumulation. These charges are continuously accumulated by the integrator. Once the accumulated electric charge, denoted as $A(t)$, reaches a predefined threshold θ, the system generates a spike to represent a spike-firing and releases the accumulated electric charges under the enable operation of a periodic timing signal S_h, restarting a new "integrate-and-fire" cycle.

Spike Counting and Time Encoding. TE spike camera introduces a spike-firing counter (SFC) to count the number of fired spikes. To record the time required for the spike counting to reach P_{max} (i.e., overflow), a straightforward approach is to introduce a timer. However, encoding the timer value would consume a significant number of bits. As shown in Eq. 3, the key to time encoding lies in obtaining the ratio between T_M and T_r. T_r is controlled by the readout signal S_r. Since multiple integrate-and-fire cycles may occur within a single readout interval, the frequency of S_h is set to

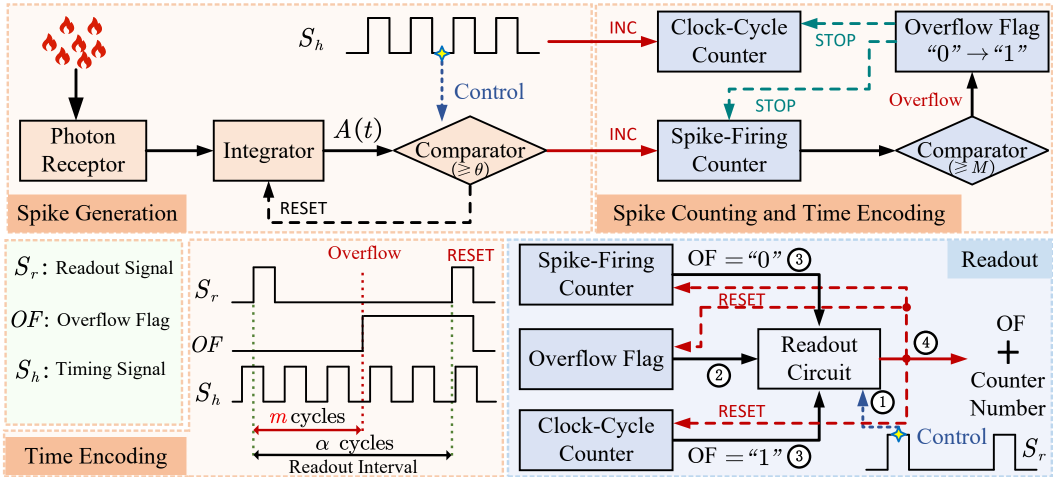

Figure 2: Working mechanism of time-encoding (TE) spike camera. Each pixel of TE spike camera continuously accumulates photons and fires a spike when a preset threshold is reached. Based on the overflow flag, TE spike camera decodes the light intensity with spike number or time information.

be significantly higher than that of S_r. Therefore, the problem of computing the ratio between T_M and T_r can be transformed into calculating the ratio between the number of S_h clock cycles until the overflow occurs, and the total number of S_h cycles within a single readout interval. To this end, we introduce a clock cycle counter (CCC) to count the number of S_h cycles. Since both S_h and S_r are periodic signals, the total number of S_h cycles within a readout interval can be determined in advance, and we denote this number as α. During a readout interval, if the spike count exceeds the maximum value that the SFC can store, an overflow occurs, setting up the overflow flag and making both the SFC and the CCC stop counting. If no overflow occurs, the SFC count spikes throughout the entire readout interval.

In the practical implementation, to facilitate manufacturing, all pixels can share a single CCC, and each pixel maintains a dedicated local register to store the number of S_h cycles provided by the CCC.

Readout. If the overflow flag is "0", the value stored in the SFC is read out along with the overflow flag, otherwise, the value in the CCC is read out with the overflow flag. Afterwards, the SFC, CCC and overflow flag are reset to prepare for the next process.

4.2 Decoding of Spike Data

Suppose the output is a $d + 1$-bit binary number. If the overflow flag is "0", the remaining d-bit (i.e., the lower d bits) binary data can be directly decoded into a decimal number, representing the total number of spikes fired during the readout interval. Then the intensity can be estimated with Eq. 2. If the overflow flag is "1", the remaining d-bit data instead represents the time information. First, the d-bit binary data is converted to a decimal number m, which represents the number of clock cycles of S_h until the overflow occurs. Since the total number of clock cycles of S_h within a readout interval is α, the $\frac{T_r}{T_m}$ in Eq. 3 can be approximated as $\frac{\alpha}{m}$.

4.3 Dynamic Range

According to Eq. 3, the minimum representable light intensity $I_3^{\min} = \frac{N_\theta}{N_s T_r}$. The maximum representable light intensity is jointly determined by the bit depths of the SFC and the CCC. Suppose the bit depths of the SFC and CCC are B_1 and B_2, respectively and we define an overflow as occurring when the overflow flag of the SFC becomes "1", i.e. $P_{\max} = 2^{B_1}$. Then the maximum light intensity can be expressed as $I_3^{\max} = \frac{2^{B_1} N_\theta}{T_r}(2^{B_2} - 1)$. In this case, the dynamic range of TE spike camera is $D_{TE} = 20 \log_{10}(I_3^{\max}/I_3^{\min}) = 20 \log_{10}(2^{B_1} \cdot (2^{B_2} - 1) N_s)$. From Eq. 2, we can obtain the dynamic range of ML spike camera $D_{ML} = 20 \log_{10}((2^B - 1) N_s)$. When we set $B_1 + B_2 > B$, we can obtain $D_{TE} > D_{ML}$.

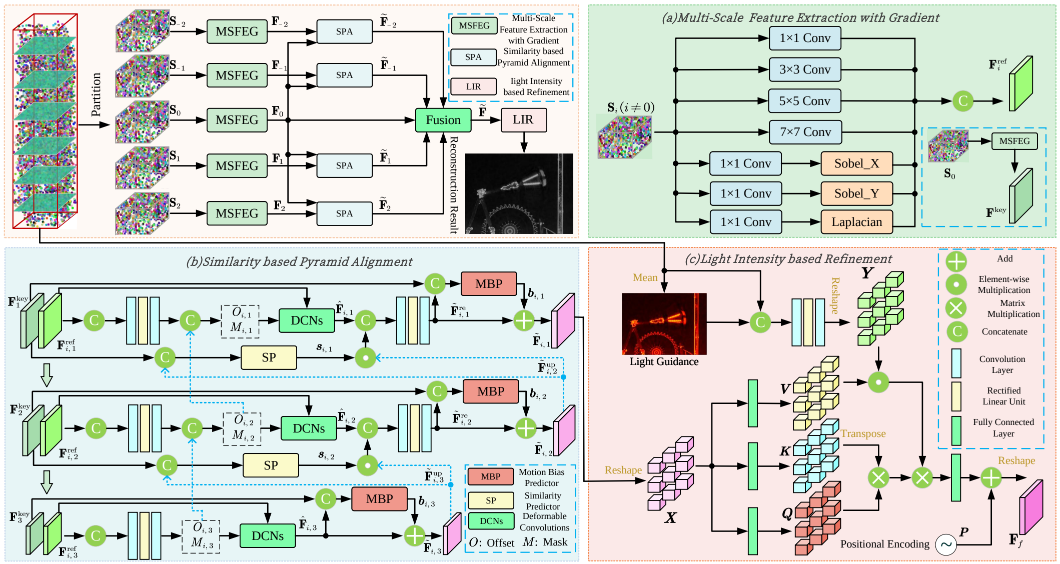

Figure 3: Architecture of the proposed reconstruction model. The MSFEG module extracts multi-scale features as well as gradient information to facilitate the following alignment process. The SPA module aligns the spike streams among temporal domains in a coarse-to-fine manner. The LIR module utilises the guidance from light intensity to fuse the spatial features of the spike streams.

5 Image Reconstruction for Time-Encoding Spike Camera

5.1 Overall Architecture

The overall architecture of the proposed reconstruction method is shown in Fig. 3. For the given TE spike stream $\mathbf{S} \in \mathbb{R}^{H \times W \times T}$, we segment $\mathbf{S}$ into five spike sub-streams $\{\mathbf{S}_i\}_{i=-2}^{2}$ to reconstruct the scene at time t_0, where i represents the time index. Each sub-stream $\mathbf{S}_i$ is centered at moment t_i with a window radius ω_h, defined as:

$$\mathbf{S}_i(p) = \{\mathbf{S}(p,t)\}_{t=t_i-\omega_h}^{t_i+\omega_h}. \tag{4}$$

Here, t_0 is designated as the key moment, while $\{t_i\}_{i\in\{-2,-1,1,2\}}$ represent reference moments. We utilise an encoder to extract multi-scale features as well as gradient information $\{\mathbf{F}_i\}_{i=-2}^{2}$ from these five spike sub-streams. To facilitate effective fusion, we design a similarity-based pyramid alignment module to align the spike streams among temporal domains in a coarse-to-fine manner. Moreover, a light intensity-based refinement module is proposed to utilise the guidance from light intensity to fuse the spatial features of the spike streams.

5.2 Multi-Scale Feature Extraction with Gradient

As shown in Fig. 3 (a), we employ convolution kernels of sizes 1, 3, 5 and 7 for feature extraction, facilitating the capture of motion at varying scales through different receptive fields. The spike stream generated by TE spike camera comprises multiple spike symbols, where higher values correspond to stronger light intensity, thereby embedding rich gradient information. This gradient information is crucial for precise motion alignment. To leverage it effectively, we apply horizontal and vertical Sobel operators, along with a Laplacian operator, to extract essential features from the spike stream. More information about gradient extraction can be found in our supplementary.

5.3 Similarity-based Pyramid Alignment

To facilitate the effective fusion of reference features and the key feature, we employ a pyramid structure to achieve coarse-to-fine feature alignment. Specifically, we propose a similarity-based pyramid alignment (SPA) module. As illustrated in Fig. 3 (b), we use convolution layers to downsample the extracted features and construct a three-level pyramid. The key feature at t_0 moment at the l-th pyramid layer is denoted as $\mathbf{F}_l^{\text{key}}$ and the reference feature at t_i moment at the l-th pyramid layer is represented as $\mathbf{F}_{i,l}^{\text{ref}}$. First, $\mathbf{F}_3^{\text{key}}$ and $\mathbf{F}_{i,3}^{\text{ref}}$ are concatenated and then passed through two convolution layers to generate the offset $O_{i,3}$ and mask $M_{i,3}$. Subsequently, $\mathbf{F}_{i,3}^{\text{ref}}$ is aligned to $\mathbf{F}_3^{\text{key}}$ by deformable convolutions (DCNs) [5; 66]:

$$\hat{\mathbf{F}}_{i,3}(p) = \sum_{i=0}^{n} \omega_i \cdot \mathbf{F}_{i,3}^{\text{ref}}(p + p_i + O_{i,3}(p, p_i)) \cdot M_{i,3}(p, p_i), \tag{5}$$

where $\hat{\mathbf{F}}_{i,3}$ denotes the aligned feature at the third pyramid layer. The coordinate $p = (x, y)$ denotes the center location, n refers to the number of sampling locations, ω_i represents the i-th weight, and p_i denotes the i-th fixed offset.

To mitigate motion blur and potential artefacts in occluded regions, we propose a motion bias predictor (MBP) module to evaluate the bias between $\hat{\mathbf{F}}_{i,3}$ and $\mathbf{F}_3^{\text{key}}$. The bias is then added to $\hat{\mathbf{F}}_{i,3}$ to obtain the refined feature $\tilde{\mathbf{F}}_{i,3}$.

The offset $O_{i,3}$ and mask $M_{i,3}$ are propagated and fused across different pyramid levels through upsampling, as illustrated in Fig. 3 (b). Similarly, the aligned feature $\tilde{\mathbf{F}}_{i,3}$ is also propagated and fused between pyramid levels via upsampling. However, the feature information of $\tilde{\mathbf{F}}_{i,3}$ may not be reliable. Additionally, downsampling during pyramid construction can lead to information loss, making the upsampled $\tilde{\mathbf{F}}_{i,3}^{\text{up}}$ potentially uncorrelated with $\mathbf{F}_2^{\text{key}}$. To mitigate this issue, we design a similarity predictor (SP) module, which takes the upsampled $\tilde{\mathbf{F}}_{i,3}^{\text{up}}$ and $\mathbf{F}_2^{\text{key}}$ as inputs and outputs a similarity feature $\mathbf{s}$. This similarity feature is then used to guide the fusion of the DCNs-aligned $\hat{\mathbf{F}}_{i,2}$ and the upsampled feature $\tilde{\mathbf{F}}_{i,3}^{\text{up}}$. A bias is also estimated to facilitate the fusion. The aligned reference features are concatenated with the key feature and passed through a fusion module for preliminary integration and obtain $\tilde{\mathbf{F}}$. More details about the SPA module can be found in our supplementary.

5.4 Light Intensity-based Refinement

After performing motion alignment at different time instances, we further refine the preliminary integration result $\tilde{\mathbf{F}}$ using spatial domain information. Inspired by [2], considering that the texture details in regions with stronger illumination are generally more reliable, we propose a light intensity-based refinement (LIR) module, as shown in Fig. 3 (c).

In fact, the values at different moments and locations in the spike stream are linear with the light intensity. By averaging the spike stream along the temporal direction, we obtain a rough estimation of the light intensity at the key moment. This estimation is then concatenated with the spike stream to generate an intensity-based feature $\mathbf{I}$ for refinement through a two-layer convolution network.

To capture long-range spatial correlations while maintaining computational efficiency, we apply a multi-head self-attention network to extract the spatial domain information of $\tilde{\mathbf{F}}$. $\tilde{\mathbf{F}}$ is reshaped into tokens $\mathbf{X} \in \mathbb{R}^{HW \times C}$ and $\mathbf{X}$ is further split into k heads:$\{\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_k\}$. For each head, three fully connected layers are applied to project $\mathbf{X}_i$ into query $\mathbf{Q}_i \in \mathbb{R}^{HW \times \frac{C}{k}}$, key $\mathbf{K}_i \in \mathbb{R}^{HW \times \frac{C}{k}}$ and value $\mathbf{V}_i \in \mathbb{R}^{HW \times \frac{C}{k}}$. Then, we use the intensity-based feature $\mathbf{I}$ to guide the computation of self-attention. In detail, $\mathbf{I}$ is reshaped into $\mathbf{Y} \in \mathbb{R}^{HW \times C}$and split into k heads: $\{\mathbf{Y}_1, \mathbf{Y}_2, ..., \mathbf{Y}_k\}$. Then the self-attention of each head can be calculated as:

$$\text{att} = (\mathbf{Y}_i \circ \mathbf{V}_i)\text{softmax}(\frac{\mathbf{K}_i^T \mathbf{Q}_i}{\gamma_i}), \tag{6}$$

where $\circ$ means element-wise multiplication and $\gamma_i \in \mathbb{R}^1$ is a learnable scaling factor. Subsequently, k heads are concatenated to pass a fully connected layer and then plus a positional encoding $\mathbf{P} \in \mathbb{R}^{HW \times C}$ to produce the output tokens $\mathbf{X}_o \in \mathbb{R}^{HW \times C}$. $\mathbf{X}_o$ is then reshaped and the output feature $\mathbf{F}_o \in \mathbb{R}^{H \times W \times C}$ is obtained. Finally, we use one convolution layer to adjust the channel number of $\mathbf{F}_o$ and derive the final output $\mathbf{F}_f \in \mathbb{R}^{H \times W \times 1}$.

Table 1: Comparison of quantitative results on the synthesized HDM-HDR-2014 dataset (♠ means only one parsing branch is utilized).

ML Reconstruction	PSNR-μ ↑	SSIM-μ ↑	HDR-VDP↑	HDR-VQM↓	TE Reconstruction	PSNR-μ ↑	SSIM-μ ↑	HDR-VDP↑	HDR-VQM↓
TFI_ML	15.50	0.161	3.056	0.943	TFI_TE	17.37	0.235	3.367	0.933
TFP_ML	16.13	0.250	3.778	0.970	TFP_TE	17.74	0.317	3.412	0.970
Spk2ImgNet_ML	26.69	0.779	7.209	0.459	Spk2ImgNet_TE	29.64	0.830	7.892	0.357
BSF_ML	26.94	0.782	7.338	0.457	BSF_TE	29.64	0.826	7.825	0.370
MambaSpike ♠	26.42	0.767	7.189	0.507	MambaSpike_TE	28.94	0.828	7.802	0.400
MambaSpike	27.30	0.788	7.405	0.443	Ours	**30.86**	**0.853**	**8.055**	**0.325**

Table 2: Comparison of quantitative results on the synthesized Kalantari13 dataset (♠ means only one parsing branch is utilized).

ML Reconstruction	PSNR-μ ↑	SSIM-μ ↑	HDR-VDP↑	HDR-VQM↓	TE Reconstruction	PSNR-μ ↑	SSIM-μ ↑	HDR-VDP↑	HDR-VQM↓
TFI_ML	24.55	0.539	3.777	1.222	TFI_TE	27.52	0.740	3.882	1.228
TFP_ML	22.38	0.699	3.232	1.288	TFI_TE	23.88	0.737	3.321	1.288
Spk2ImgNet_ML	26.76	0.863	8.512	0.361	Spk2ImgNet_TE	30.29	0.937	9.587	0.138
BSF_ML	28.28	0.872	8.619	0.345	BSF_TE	31.74	0.937	9.586	**0.132**
MambaSpike ♠	25.99	0.859	8.584	0.334	MambaSpike_TE	29.06	0.929	9.516	0.151
MambaSpike	27.85	0.874	8.689	0.325	Ours	**33.65**	**0.943**	**9.606**	0.139

6 Experiments

6.1 Spike Simulator

Based on ML spike camera and TE spike camera we propose, we design two corresponding spike simulators to generate spike streams from video sequences. We convert them to grayscale to represent the light intensity in the external environment. The photoelectric conversion rate η is set to 0.7. Additionally, Poisson noise is added to simulate the shot noise during the photon arrival process. For ML spike camera, we utilise an SFC with a bit depth of 8. For TE spike camera, we utilise an SFC with a bit depth of 4 and a CCC with a bit depth of 4. Suppose a pixel value at the moment t is $I(t)$, the threshold of ML spike camera is set to $\max(I)/2^7$, and the threshold of TE spike camera is set to $\max(I)/(2^4 \times (2^4 - 1))$. More explanations of our settings can be found in our supplementary.

6.2 Dataset

We use part of HDM-HDR-2014 dataset [13] for training and the other part of HDM-HDR-2014 dataset, along with Kalantari13 dataset [22] for testing. Since Kalantari13 dataset does not have ground truth, we use the result of HDRFlow [52] (one of the SOTA HDR reconstruction methods) as the ground truth. More dataset settings can be found in our supplementary.

6.3 Implementation Details

During the training process, we randomly crop the spike streams to a spatial size of 96×96 and apply random horizontal and vertical flips for data augmentation. The network is trained for 60 epochs with a batch size of 4. We use the Adam [24] optimizer with parameter $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The initial learning rate is set to $1e - 4$ and is halved every 10 epochs. To guide the training, we employ the $\mathcal{L}_1$ loss function to compute the difference between the estimated $\hat{I}(t_0)$ and the ground truth $I_{gt}(t_0)$:

$$\mathcal{L} = ||\hat{I}(t_0)/\eta - I_{gt}(t_0)||_1. \tag{7}$$

6.4 Comparison with ML Spike Camera

In our experiments, we use PSNR-μ, SSIM-μ, HDR-VDP-3 [31] and HDR-VQM [35] as evaluation metrics, where μ means the HDR images are tone-mapped with μ law [21], and we apply $\mu = 5000$ in our experiments.

To demonstrate the advantages of TE spike camera for image reconstruction, we adapt several reconstruction methods for ML spike camera in [67] to TE spike camera. These methods include two

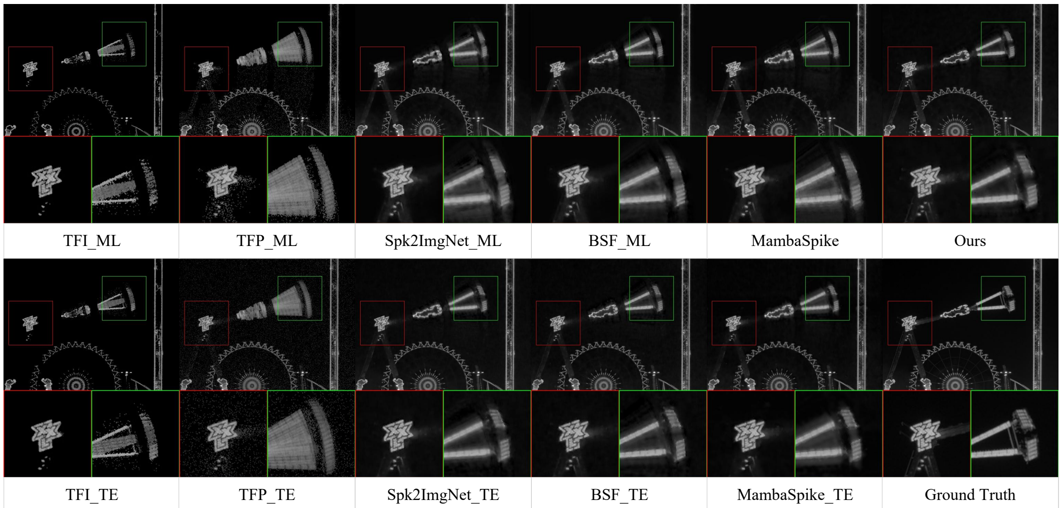

Figure 4: Visual comparisons of different reconstruction methods on the synthesized HDM-HDR-2014 dataset.

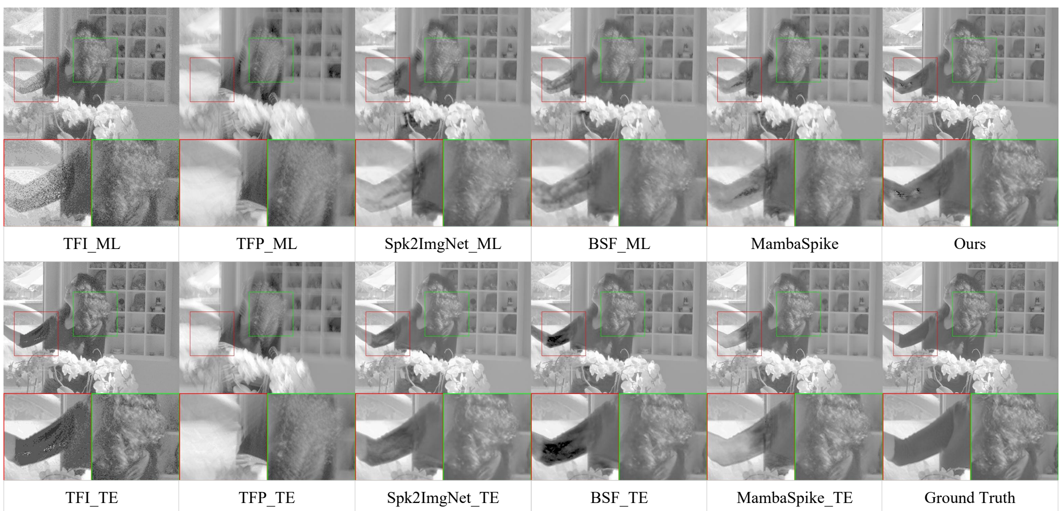

Figure 5: Visual comparisons of different reconstruction methods on the synthesized Kalantari13 dataset.

training-free approaches: TFI_ML and TFP_ML, and three deep learning methods: Spk2ImgNet_ML, BSF_ML and MambaSpike.

The spike streams are segmented into slices as input. Following the configuration in MambaSpike [67], each slice for the training-free methods consists of 9 continuous frames, while each slice for the deep learning methods contains 61 continuous frames. The average value for each sequence is computed, and the overall average across all testing sequences is used as the final experimental result.

Adapted from TFI_ML and TFP_ML, TFI_TE and TFP_TE utilise a spike interval and spike-firing rate in a temporary window to predict light intensity, respectively. For three deep-learning methods, Spk2ImgNet_ML, BSF_ML and MambaSpike, we modify the input from ML spike streams to TE spike streams, resulting in the modified versions: Spk2ImgNet_TE, BSF_TE, and MambaSpike_TE. For fair comparisons, We retrain all three deep-learning methods on our training set. Unlike ML spike camera, which reads only the major part of the SFC at the readout moment, TE spike camera reads all the data from the SFC. As a result, there is no need to predict the remaining part of the SFC, thereby eliminating the need for multiple parsings of the spike stream. Consequently, we utilise one spike parsing in MambaSpike_TE.

Table 3: Ablation studies of the proposed method.

Case	MSFE	Gradient	DCNs	SP	MBP	LIR	PSNR-$\mu \uparrow$	SSIM-$\mu \uparrow$
1	✓						25.68	0.680
2	✓	✓					26.35	0.697
3	✓	✓	✓				29.54	0.818
4	✓	✓	✓	✓			30.04	0.828
5	✓	✓	✓	✓	✓		30.32	0.824
6	✓	✓	✓	✓	✓	✓	**30.86**	**0.853**

As shown in Tab. 1, reconstruction methods based on TE spike camera significantly outperform those methods based on ML spike camera. Furthermore, the proposed method achieves the best overall performance in all the metrics. The visualised reconstruction results in Fig. 4 clearly demonstrate the advantages of TE spike camera-based methods, particularly in terms of texture detail preservation and motion alignment.

Testing results on the synthesized Kalantari13 dataset are presented in Tab. 2 and Fig. 5. Methods based on the TE spike camera exhibit a significant performance improvement over their ML counterparts, with the proposed method achieving the best results. This demonstrates that the proposed method exhibits strong generalization ability.

More visualised reconstruction results, analysis of noise influence and complexity analysis can be found in our supplementary.

6.5 Ablation Studies

We investigate the effects of the proposed multi-scale feature extraction with gradient (MSFEG) module, similarity-based pyramid alignment (SPA) module, and light intensity-based refinement (LIR) module. We use the synthesized HDM-HDR-2014 dataset for the ablation studies. The results are presented in Tab. 3. For the MSFEG module, Cases (1-2) highlight the effectiveness of the gradient features. For the SPA module, Cases (3-5) demonstrate the effectiveness of the deformable convolutions (DCNs), similarity predictor (SP) module, and Motion Bias Predictor (MBP) module, respectively. Finally, Case (6) illustrates the effectiveness of the LIR module.

7 Limitations

Current validation is limited to simulations, and additional challenges are anticipated in real-world applications.

8 Conclusion

We propose time-encoding (TE) spike camera, a novel advancement over multi-level (ML) spike camera. TE spike camera incorporates an additional clock cycle counter (CCC) to record the time of spike counting. When the spike count reaches a certain value, TE spike camera stops counting and retrieves the time required for the counting from the CCC. By computing the ratio of this time to the readout interval, the overflow moment can be represented. Furthermore, we propose an image reconstruction scheme for TE spike camera, we focus on the gradient features of spike data, and propose a similarity-based pyramid alignment module to align the spike streams among temporal domains. Moreover, a light intensity-based refinement module is proposed to utilise the guidance from light intensity to fuse the spatial features. Experimental results demonstrate that TE spike camera effectively enhances the dynamic range of spike camera.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China under Grant 62072009 and 22127807, and 62521007.

References

[1] Cadena, P.R.G., Qian, Y., Wang, C., Yang, M.: Spade-e2vid: Spatially-adaptive denormalization for event-based video reconstruction. IEEE TIP **30**, 2488–2500 (2021)

[2] Cai, Y., Bian, H., Lin, J., Wang, H., Timofte, R., Zhang, Y.: Retinexformer: One-stage retinex-based transformer for low-light image enhancement. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 12504–12513 (2023)

[3] Cao, R., Galor, D., Kohli, A., Yates, J.L., Waller, L.: Noise2image: noise-enabled static scene recovery for event cameras. Optica **12**(1), 46–55 (2025)

[4] Choi, J., Yoon, K.J., et al.: Learning to super resolve intensity images from events. In: CVPR. pp. 2768–2776 (2020)

[5] Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: ICCV. pp. 764–773 (2017)

[6] Dong, Y., Xiong, R., Fan, X., Yu, Z., Tian, Y., Huang, T.: Self-supervised learning for color spike camera reconstruction. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 6231–6240 (2025)

[7] Dong, Y., Xiong, R., Fan, X., Zhu, S., Wang, J., Huang, T.: Dynamic scene reconstruction for color spike camera via zero-shot learning. IEEE Transactions on Computational Imaging (2025)

[8] Dong, Y., Xiong, R., Zhang, J., Yu, Z., Fan, X., Zhu, S., Huang, T.: Super-resolution reconstruction from bayer-pattern spike streams. In: CVPR. pp. 24871–24880 (2024)

[9] Dong, Y., Xiong, R., Zhao, J., Fan, X., Zhang, X., Huang, T.: Color spike camera reconstruction via long short-term temporal aggregation of spike signals. IEEE Transactions on Image Processing (2025)

[10] Dong, Y., Xiong, R., Zhao, J., Zhang, J., Fan, X., Zhu, S., Huang, T.: Joint demosaicing and denoising for spike camera. In: AAAI. vol. 38, pp. 1582–1590 (2024)

[11] Dong, Y., Xiong, R., Zhao, J., Zhang, J., Fan, X., Zhu, S., Huang, T.: Learning a deep demosaicing network for spike camera with color filter array. IEEE TIP **33**, 3634–3647 (2024)

[12] Eilertsen, G., Kronander, J., Denes, G., Mantiuk, R.K., Unger, J.: Hdr image reconstruction from a single exposure using deep cnns. ACM TOG **36**(6), 178:1–178:15 (2017)

[13] Froehlich, J., Grandinetti, S., Eberhardt, B., Walter, S., Schilling, A., Brendel, H.: Creating cinematic wide gamut hdr-video for the evaluation of tone mapping operators and hdr-displays. In: Digital photography X. vol. 9023, pp. 279–288 (2014)

[14] Gulve, R., Sarhangnejad, N., Dutta, G., Sakr, M., Nguyen, D., Rangel, R., Chen, W., Xia, Z., Wei, M., Gusev, N., et al.: A 39,000 subexposures/s cmos image sensor with dual-tap coded-exposure data-memory pixel for adaptive single-shot computational imaging. In: 2022 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits). pp. 78–79. IEEE (2022)

[15] Han, J., Yang, Y., Duan, P., Zhou, C., Ma, L., Xu, C., Huang, T., Sato, I., Shi, B.: Hybrid high dynamic range imaging fusing neuromorphic and conventional images. IEEE TPAMI **45**(7), 8553–8565 (2023)

[16] Han, J., Zhou, C., Duan, P., Tang, Y., Xu, C., Xu, C., Huang, T., Shi, B.: Neuromorphic camera guided high dynamic range imaging. In: CVPR. pp. 1730–1739 (2020)

[17] He, B., Wang, Z., Zhou, Y., Chen, J., Singh, C., Li, H., Gao, Y., Shen, S., Wang, K., Cao, Y., et al.: Microsaccade-inspired event camera for robotics (2024)

[18] Hsu, P.H., Lee, Y.R., Chen, C.H., Hung, C.C.: A low-noise area-efficient column-parallel adc with an input triplet for a 120-db high dynamic range cmos image sensor. IEEE Transactions on Very Large Scale Integration (VLSI) Systems **31**(12), 1939–1949 (2023)

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

[20] Ikeno, R., Mori, K., Uno, M., Miyauchi, K., Isozaki, T., Takayanagi, I., Nakamura, J., Wuu, S.G., Bainbridge, L., Berkovich, A., et al.: A 4.6-μm, 127-db dynamic range, ultra-low power stacked digital pixel sensor with overlapped triple quantization. IEEE Transactions on Electron Devices **69**(6), 2943–2950 (2022)

[21] Kalantari, N.K., Ramamoorthi, R., et al.: Deep high dynamic range imaging of dynamic scenes. ACM TOG **36**(4), 144:1–144:12 (2017)

[22] Kalantari, N.K., Shechtman, E., Barnes, C., Darabi, S., Goldman, D.B., Sen, P.: Patch-based high dynamic range video. ACM TOG **32**(6), 202:1–202:8 (2013)

[23] Kavadias, S., Dierickx, B., Scheffer, D., Alaerts, A., Uwaerts, D., Bogaerts, J.: A logarithmic response cmos image sensor with on-chip calibration. IEEE Journal of Solid-state circuits **35**(8), 1146–1152 (2000)

[24] Kingma, D.P.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

[25] Lee, S., An, G.H., Kang, S.J.: Deep recursive hdri: Inverse tone mapping using generative adversarial networks. In: ECCV. pp. 596–611 (2018)

[26] Li, Z., Liao, J., Tang, C., Zhang, H., Li, Y., Bian, Y., Sheng, X., Feng, X., Li, Y., Gao, C., et al.: Ustc-td: A test dataset and benchmark for image and video coding in 2020s. IEEE Transactions on Multimedia (2025)

[27] Liu, C., Bainbridge, L., Berkovich, A., Chen, S., Gao, W., Tsai, T.H., Mori, K., Ikeno, R., Uno, M., Isozaki, T., et al.: A 4.6 μm, 512$\times$ 512, ultra-low power stacked digital pixel sensor with triple quantization and 127db dynamic range. In: 2020 IEEE International Electron Devices Meeting (IEDM). pp. 16–1. IEEE (2020)

[28] Liu, S., Zhang, X., Sun, L., Liang, Z., Zeng, H., Zhang, L.: Joint hdr denoising and fusion: A real-world mobile hdr image dataset. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13966–13975 (2023)

[29] Liu, Y., Gutierrez-Barragan, F., Ingle, A., Gupta, M., Velten, A.: Single-photon camera guided extreme dynamic range imaging. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1575–1585 (2022)

[30] Loose, M., Meier, K., Schemmel, J.: A self-calibrating single-chip cmos camera with logarithmic response. IEEE Journal of Solid-state circuits **36**(4), 586–596 (2001)

[31] Mantiuk, R.K., Hammou, D., Hanji, P.: Hdr-vdp-3: A multi-metric for predicting image differences, quality and contrast distortions in high dynamic range and regular content. arXiv preprint arXiv:2304.13625 (2023)

[32] Mori, K., Yasuda, N., Miyauchi, K., Isozaki, T., Takayanagi, I., Nakamura, J., Chien, H.C., Fu, K., Wuu, S.G., Berkovich, A., et al.: A 4.0 μm stacked digital pixel sensor operating in a dual quantization mode for high dynamic range. IEEE Transactions on Electron Devices **69**(6), 2957–2964 (2022)

[33] Mostafavi, M., Wang, L., Yoon, K.J.: Learning to reconstruct hdr images from events, with applications to depth and flow prediction. IJCV **129**, 900–920 (2021)

[34] Murata, M., Kuroda, R., Fujihara, Y., Otsuka, Y., Shibata, H., Shibaguchi, T., Kamata, Y., Miura, N., Kuriyama, N., Sugawa, S.: A high near-infrared sensitivity over 70-db snr cmos image sensor with lateral overflow integration trench capacitor. IEEE transactions on electron devices **67**(4), 1653–1659 (2020)

[35] Narwaria, M., Da Silva, M.P., Le Callet, P.: Hdr-vqm: An objective quality measure for high dynamic range video. Signal Processing: Image Communication **35**, 46–60 (2015)

[36] Ogi, J., Takatsuka, T., Hizu, K., Inaoka, Y., Zhu, H., Tochigi, Y., Tashiro, Y., Sano, F., Murakawa, Y., Nakamura, M., et al.: A 124-db dynamic-range spad photon-counting image sensor using subframe sampling and extrapolating photon count. IEEE Journal of Solid-State Circuits **56**(11), 3220–3227 (2021)

[37] Ota, Y., Morimoto, K., Sasago, T., Shinohara, M., Kuroda, Y., Endo, W., Maehashi, Y., Maekawa, S., Tsuchiya, H., Abdelahafar, A., et al.: A 0.37 w 143db-dynamic-range 1mpixel backside-illuminated charge-focusing spad image sensor with pixel-wise exposure control and adaptive clocked recharging. In: 2022 IEEE International Solid-State Circuits Conference (ISSCC). vol. 65, pp. 94–96. IEEE (2022)

[38] Park, B., Choi, H.S., Jeong, J., Kim, T., Lee, M.J., Chae, Y.: A 113.3-db dynamic range 600 frames/s spad x-ray detector with seamless global shutter and time-encoded extrapolation counter. IEEE Journal of Solid-State Circuits **58**(11), 2965–2975 (2023)

[39] Park, D., Lee, S.W., Han, J., Jang, D., Kwon, H., Cha, S., Kim, M., Lee, H., Suh, S., Joo, W., et al.: A 0.8 μm smart dual conversion gain pixel for 64 megapixels cmos image sensor with 12k e-full-well capacitance and low dark noise. In: 2019 IEEE International Electron Devices Meeting (IEDM). pp. 16–2. IEEE (2019)

[40] Sharma, S., Rongali, G., Mitra, K.: Transforming single photon camera images to color high dynamic range images. In: International Conference on Computer Vision and Image Processing. pp. 135–149. Springer (2024)

[41] Shike, H., Kuroda, R., Kobayashi, R., Murata, M., Fujihara, Y., Suzuki, M., Harada, S., Shibaguchi, T., Kuriyama, N., Hatsui, T., et al.: A global shutter wide dynamic range soft x-ray cmos image sensor with backside-illuminated pinned photodiode, two-stage lateral overflow integration capacitor, and voltage domain memory bank. IEEE transactions on electron devices **68**(4), 2056–2063 (2021)

[42] Stefanov, K.D., Prest, M.J.: Pinned photodiode imaging pixel with floating gate readout and dual gain. IEEE Transactions on Electron Devices **70**(6), 3136–3139 (2023)

[43] Storm, G., Henderson, R., Hurwitz, J., Renshaw, D., Findlater, K., Purcell, M.: Extended dynamic range from a combined linear-logarithmic cmos image sensor. IEEE Journal of Solid-State Circuits **41**(9), 2095–2106 (2006)

[44] Sugawa, S., Akahane, N., Adachi, S., Mori, K., Ishiuchi, T., Mizobuchi, K.: A 100 db dynamic range cmos image sensor using a lateral overflow integration capacitor. In: ISSCC. 2005 IEEE International Digest of Technical Papers. Solid-State Circuits Conference, 2005. pp. 352–603. IEEE (2005)

[45] Wang, L., Kim, T.K., Yoon, K.J.: Eventsr: From asynchronous events to image reconstruction, restoration, and super-resolution via end-to-end adversarial learning. In: CVPR. pp. 8315–8325 (2020)

[46] Wang, Y., Zhang, Y., Xiong, R., Zhang, J., Zhang, X., Huang, T.: Super-resolving dynamic scenes with spike camera via multi-frame sequential alignment with motion propagation. IEEE Transactions on Image Processing (2025)

[47] Wang, Y., Zhang, Y., Xiong, R., Zhao, J., Zhang, J., Fan, X., Huang, T.: Spk2srimgnet: Super-resolve dynamic scene from spike stream via motion aligned collaborative filtering. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 11416–11426 (2025)

[48] Weng, W., Zhang, Y., Xiong, Z.: Event-based video reconstruction using transformer. In: ICCV. pp. 2563–2572 (2021)

[49] Wocial, T., Stefanov, K.D., Martin, W.E., Barnes, J.R., Jones, H.R.: A method to achieve high dynamic range in a cmos image sensor using interleaved row readout. IEEE Sensors Journal **22**(22), 21619–21627 (2022)

[50] Xia, L., Ding, Z., Zhao, R., Zhang, J., Ma, L., Yu, Z., Huang, T., Xiong, R.: Unsupervised optical flow estimation with dynamic timing representation for spike camera. In: NeurIPS. vol. 36, pp. 48070–48082 (2024)

[51] Xia, L., Zhao, J., Xiong, R., Huang, T.: Svfi: spiking-based video frame interpolation for high-speed motion. In: AAAI. vol. 37, pp. 2910–2918 (2023)

[52] Xu, G., Wang, Y., Gu, J., Xue, T., Yang, X.: Hdrflow: Real-time hdr video reconstruction with large motions. In: CVPR. pp. 24851–24860 (2024)

[53] Yan, Q., Chen, W., Zhang, S., Zhu, Y., Sun, J., Zhang, Y.: A unified hdr imaging method with pixel and patch level. In: CVPR. pp. 22211–22220 (2023)

[54] Yan, Q., Zhang, L., Liu, Y., Zhu, Y., Sun, J., Shi, Q., Zhang, Y.: Deep hdr imaging via a non-local network. IEEE TIP **29**, 4308–4322 (2020)

[55] Yang, Y., Han, J., Liang, J., Sato, I., Shi, B.: Learning event guided high dynamic range video reconstruction. In: CVPR. pp. 13924–13934 (2023)

[56] Yin, P.H., Lu, C.W., Wang, J.S., Chang, K.L., Lin, F.K., Chen, P.: A 368× 184 optical under-display fingerprint sensor comprising hybrid arrays of global and rolling shutter pixels with shared pixel-level adcs. IEEE Journal of Solid-State Circuits **56**(3), 763–777 (2021)

[57] Yu, L., Yang, W., et al.: Event-based high frame-rate video reconstruction with a novel cycle-event network. In: ICIP. pp. 86–90 (2020)

[58] Zhang, X., Liao, W., Yu, L., Yang, W., Xia, G.S.: Event-based synthetic aperture imaging with a hybrid network. In: CVPR. pp. 14235–14244 (2021)

[59] Zhao, J., Xie, J., Xiong, R., Zhang, J., Yu, Z., Huang, T.: Super resolve dynamic scene from continuous spike streams. In: ICCV. pp. 2533–2542 (2021)

[60] Zhao, J., Xiong, R., Xie, J., Shi, B., Yu, Z., Gao, W., Huang, T.: Reconstructing clear image for high-speed motion scene with a retina-inspired spike camera. IEEE TCI **8**, 12–27 (2021)

[61] Zhao, J., Xiong, R., Zhang, J., Zhao, R., Liu, H., Huang, T.: Learning to super-resolve dynamic scenes for neuromorphic spike camera. In: AAAI. vol. 37, pp. 3579–3587 (2023)

[62] Zhao, J., Zhang, S., Ma, L., Yu, Z., Huang, T.: Spikingsim: A bio-inspired spiking simulator. In: 2022 IEEE International Symposium on Circuits and Systems (ISCAS). pp. 3003–3007. IEEE (2022)

[63] Zhao, R., Xiong, R., Zhang, J., Yu, Z., Zhu, S., Ma, L., Huang, T.: Spike camera image reconstruction using deep spiking neural networks. IEEE Transactions on Circuits and Systems for Video Technology **34**(6), 5207–5212 (2023)

[64] Zhao, R., Xiong, R., Zhang, J., Zhang, X., Yu, Z., Huang, T.: Optical flow for spike camera with hierarchical spatial-temporal spike fusion. In: AAAI. vol. 38, pp. 7496–7504 (2024)

[65] Zheng, Y., Zheng, L., Yu, Z., Huang, T., Wang, S.: Capture the moment: High-speed imaging with spiking cameras through short-term plasticity. IEEE TPAMI **45**(7), 8127–8142 (2023)

[66] Zhu, X., Hu, H., Lin, S., Dai, J.: Deformable convnets v2: More deformable, better results. In: CVPR. pp. 9308–9316 (2019)

[67] Zhu, Z., Xiong, R., Zhao, J., Zhao, R., Fan, X., Zhu, S., Huang, T.: High dynamic range imaging for dynamic scenes based on multi-level spike camera. IEEE TCSVT **35**(6), 5394 – 5406 (2025)

[68] Zou, Y., Zheng, Y., Takatani, T., Fu, Y.: Learning to reconstruct high speed and high dynamic range videos from events. In: CVPR. pp. 2024–2033 (2021)

A Supplementary

A.1 Module Design

A.1.1 Multi-Scale Feature Extraction with Gradient

The horizontal Sobel operator G_x, vertical Sobel operator G_y, and the Laplacian operator L applied in our methods are:

$$\begin{aligned}
G_x &= \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix}, \\
G_y &= \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, \\
L &= \begin{bmatrix} 0 & +1 & 0 \\ +1 & -4 & +1 \\ 0 & +1 & 0 \end{bmatrix}.
\end{aligned} \tag{8}$$

We assign the weights of these operators to 3×3 convolution kernels, and these weights remain fixed throughout the training process.

A.1.2 Similarity-based Pyramid Alignment

When $l = 1, 2$, the operation in the l_th layer of the similarity-based pyramid alignment module can be expressed as:

$$\begin{aligned}
\hat{\mathbf{F}}_{i,l}(p) &= \sum_{i=0}^{n} \omega_i \cdot \mathbf{F}^{\text{ref}}_{i,l}(p + p' + O_{i,l}(p, p')) \cdot M_{i,l}(p, p'), \\
\mathbf{s}_{i,l} &= \text{SP}(\text{Cat}(\tilde{\mathbf{F}}^{\text{up}}_{i,l+1}, \mathbf{F}^{\text{key}}_{l})), \\
\tilde{\mathbf{F}}^{\text{re}}_{i,l} &= \text{Conv}(\text{Cat}(\hat{\mathbf{F}}_{i,l}, \mathbf{s}_{i,l} \circ \tilde{\mathbf{F}}^{\text{up}}_{i,l+1})), \\
\mathbf{b}_{i,l} &= \text{MBP}(\text{Cat}(\tilde{\mathbf{F}}^{\text{re}}_{i,l}, \mathbf{F}^{\text{key}}_{l})) \\
\tilde{\mathbf{F}}_{i,l} &= \tilde{\mathbf{F}}^{\text{re}}_{i,l} + \mathbf{b}_{i,l}
\end{aligned} \tag{9}$$

where Cat means concatenation and $\circ$ means element-wise multiplication.

The fusion module consists of a two-layer convolution.

A.1.3 Motion Bias Predictor and Similarity Predictor

For the motion bias predictor (MBP) and similarity predictor (SP), to ensure the inference speed of the model, we adopt a two-layer convolution and a ReLU activation for realization. Even with this lightweight design, a significant performance improvement can be observed. For different scenarios and requirements, more complex models can be employed as replacements.

A.2 Experiments

A.2.1 Spike Simulator

According to Eq. 2, when the bit depth of the ML spike camera is 8, the threshold should be set to $2^8 - 1$. However, to reduce output bandwidth, Zhu et al. [67] introduced a dual-buffer mechanism in the design of the ML spike camera, in which only the bit index of the most significant bit (MSB) of the SFC is output at each readout moment. As a result, Eq. 2 is modified to:

$$I_2 = \{\frac{bN_\theta}{nT_r}, n, b \in \mathbb{N}^+, n \le N_s, b \le 2^{(B-1)}\}. \tag{10}$$

Therefore, the threshold is set to 2^7. Since the output mode of the ML spike camera is not the focus of this paper, we adopt Eq. 2 for clarity and ease of understanding. In the experiments, to ensure a fair comparison, we use Eq. 10 consistent with [67].

A.2.2 Dataset Settings

For HDM-HDR-2014 dataset and Kalantari13 dataset, we first extract the .hdr or .exr files from each sequence and convert the images to grayscale to simulate external light intensity. To facilitate subsequent data processing and neural network input, we identify the maximum pixel value within each sequence and normalise all the images in each sequence accordingly.

Following the settings of [67], we partition HDM-HDR-2014 dataset into sub-sequences, each containing a continuous scene, with a maximum length of 400 frames and no repeated frames between sub-sequences. This results in a total of 57 sub-sequences. We utilize 30 sub-sequences for training and reserve the remaining 27 sub-sequences for testing. To maintain the independence of the training and testing sets, sub-sequences derived from the same original sequence are assigned exclusively to either the training or testing set.

For the training set, we resize the original images from 1920×1080 to 240×135 to facilitate fast training. For the testing set, we crop the original images to a resolution of 512×384.

A.2.3 μ Law

Following the settings of [21], μ law is defined as:

$$\mathcal{T}(H_Y) = \frac{log(1+\mu H_Y)}{log(1+\mu)}, \tag{11}$$

where H_Y is the generated HDR image after normalized to $[0, 1]$, and $\mathcal{T}$ is the tone-mapping operator and μ is the amount of compression. In our experiments, we set μ to 5000 to maintain consistency with standard settings.

A.2.4 Comparison with Event-RGB Hybrid Method

We compare the proposed method with the one of the SOTA Event-RGB hybrid method HDRev [55]. We retrained it on our own training dataset. Since HDRev generates color images, we converted them to grayscale for a fair comparison. The results on the synthesized HDM-HDR-2014 dataset are shown Tab. 4. The proposed method achieves the best result.

Table 4: Comparison with Event-RGB hybrid method on the synthesized HDM-HDR-2014 dataset.

Metric	HDRev(event only)	HDRev(RGB only)	HDRev	Ours
PSNR-μ ↑	13.40	14.35	22.47	**30.86**
SSIM-μ ↑	0.545	0.546	0.777	**0.853**

The results on the synthesized Kalantari13 dataset are shown in Tab. 5. The proposed method achieves the best result for PSNR-μ metric and HDRev achieves the best result for SSIM-μ metric.

Table 5: Comparison with Event-RGB hybrid method on the synthesized Kalantari13 dataset.

Metric	HDRev(event only)	HDRev(RGB only)	HDRev	Ours
PSNR-μ ↑	15.08	12.63	28.05	**33.65**
SSIM-μ ↑	0.773	0.684	**0.972**	0.943

A.2.5 Analysis of Noise Influence

In our previous experimental design, we modelled the Poisson shot noise as follows: We multiplied each pixel's normalised intensity value (ranging from 0 to 1) by 60,000 to obtain the average photon count per pixel during each readout interval. Then, we used the 'numpy.random.poisson' function to simulate the randomness of photon arrivals according to Poisson statistics.

We also modelled the quantization noise: the residual photons at the readout moment were preserved, and the time to accumulate a certain number of spikes was rounded down to the nearest integer.

Furthermore, we simulated the dark current noise. We assumed that the number of electrons induced by dark current follows a Gaussian distribution with a mean of 400 and a standard deviation of 50. If the normalised intensity at time t is I, the total number of generated electrons is computed as:

$$0.7 \times \mathcal{P}(60000 \times I) + \mathcal{N}(\mu = 400, \sigma = 50) \tag{12}$$

where 0.7 represents the photoelectric conversion rate. The results on HDM-HDR-2014 dataset are shown in Tab. 6.

Table 6: Noise influence on the synthesized HDM-HDR-2014 dataset.

Metric	Spk2ImgNet_TE	BSF_TE	Ours
PSNR-μ ↑	28.43	28.57	**28.63**
SSIM-μ ↑	**0.810**	0.798	**0.810**

The results on Kalantari13 dataset are shown in Tab. 7.

Table 7: Noise influence on the synthesized Kalantari13 dataset.

Metric	Spk2ImgNet_TE	BSF_TE	Ours
PSNR-μ ↑	29.78	31.95	**33.36**
SSIM-μ ↑	0.929	0.931	**0.936**

Additionally, we referred to [62], which recorded the actual dark current behaviour of spike camera. The results showed that the time intervals between dark current-induced spikes roughly follow a Gaussian distribution with a mean of 140 and a standard deviation of 50. Based on this observation, the total number of generated electrons is computed as:

$$0.7 \times \mathcal{P}(60000 \times I) + \frac{60000}{\mathcal{N}(\mu = 140, \sigma = 50)} \tag{13}$$

To ensure a reasonable electron count caused by dark current, we clip the $\frac{60000}{\mathcal{N}(\mu=140,\sigma=50)}$ to the range $(0, 15000)$.The results on HDM-HDR-2014 dataset are shown in Tab. 8.

Table 8: Noise influence on the synthesized HDM-HDR-2014 dataset.

Metric	Spk2ImgNet_TE	BSF_TE	Ours
PSNR-μ ↑	25.73	25.81	**25.82**
SSIM-μ ↑	0.725	0.751	**0.752**

The results on Kalantari13 dataset are shown in Tab. 9.

The introduction of dark current noise inevitably degrades the reconstruction quality to some extent. Nevertheless, our reconstruction method still achieves the best performance, particularly on Kalantari13 dataset. We sincerely appreciate your comment, which has helped make our system design more comprehensive and reasonable.

A.2.6 Complexity Analysis

We calculate the parameter size, FLOPs, and testing time for five deep-learning-based methods: Spk2ImgNet_TE, BSF_TE, MambaSpike_TE and Ours. All tests are conducted on an Ubuntu 20.04 system with an Intel Core i7 CPU and an RTX 3090 GPU. The testing time is measured by the time required to infer a single image using each method. The results are shown in Tab. 10. Although our method has a relatively large parameter size and FLOPs, it achieves high processing speed due to its efficient high-speed parallel design.

Table 9: Noise influence on the synthesized Kalantari13 dataset.

Metric	Spk2ImgNet_TE	BSF_TE	Ours
PSNR-μ ↑	27.81	29.95	**31.16**
SSIM-μ ↑	0.897	0.900	**0.904**

Table 10: Comparisons of parameter size, FLOPs and testing time.

Method	Parameter	FLOPs	Testing Time
Spk2ImgNet_TE	3.760M	1.955T	0.418s
BSF_TE	2.071M	0.887T	0.231s
MambaSpike_TE	4.418M	1.919T	0.004s
Ours	6.479M	3.711T	0.005s

A.2.7 Experimental Results

Further visualizations of the experimental results are shown in Fig. 6, Fig. 7, Fig. 8, Fig. 9, and Fig. 10.

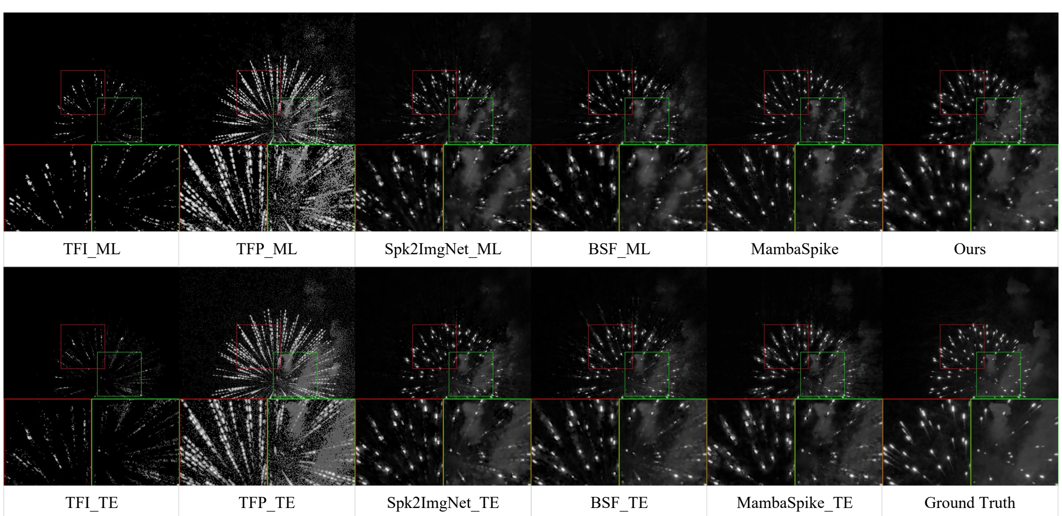

Figure 6: Visual comparisons of different reconstruction methods (part 1).

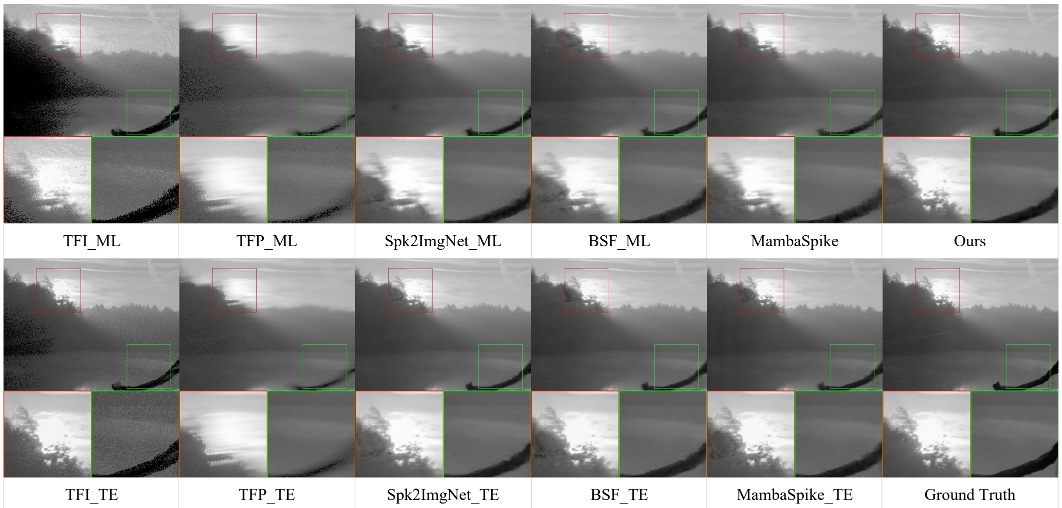

Figure 7: Visual comparisons of different reconstruction methods (part 2).

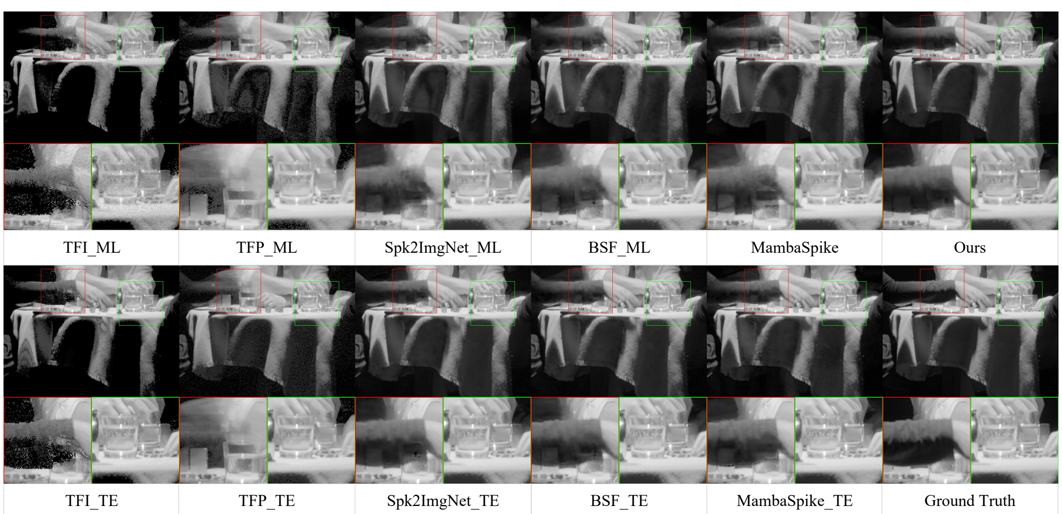

Figure 8: Visual comparisons of different reconstruction methods (part 3).

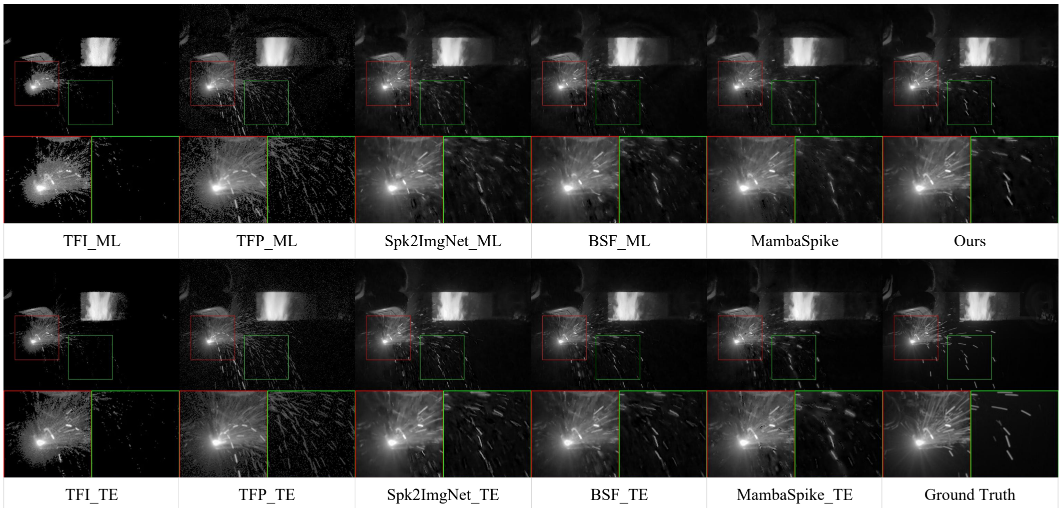

Figure 9: Visual comparisons of different reconstruction methods (part 4).

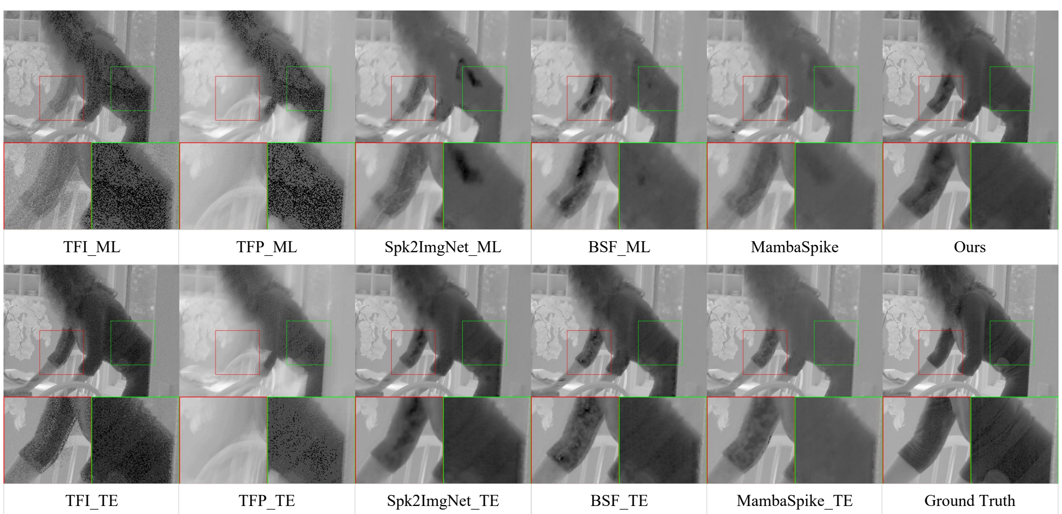

Figure 10: Visual comparisons of different reconstruction methods (part 5).

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: In Abstract, we point out that we target HDR imaging for spike camera. In the Introduction part, we claim the paper's contributions.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We make some descriptions in the Limitation part in Supplementary.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provide detailed formulas.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide detailed descriptions about how to reproduce our results in the section of Experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We will release our code after publication.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide these details in our experimental part.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We apply a random seed for our method.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide this information in the Experiments part.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We conduct the research conforming with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to

generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: No risk.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have assured that.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [No]

Justification: We will release our code and dataset after publication.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects..

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

Justification: LLM is only used for text polishing.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.