Joint Demosaicing and Denoising for Spike Camera

Yanchen Dong[1], Ruiqin Xiong[1*], Jing Zhao[2, 1], Jian Zhang[3], Xiaopeng Fan[4], Shuyuan Zhu[5], Tiejun Huang[1]

[1] National Key Laboratory for Multimedia Information Processing, School of Computer Science, Peking University
[2] National Computer Network Emergency Response Technical Team
[3] School of Electronic and Computer Engineering, Peking University
[4] School of Computer Science and Technology, Harbin Institute of Technology
[5] School of Information and Communication Engineering, University of Electronic Science and Technology of China
yanchendong@stu.pku.edu.cn, {rqxiong, jzhaopku, zhangjian.sz, tjhuang}@pku.edu.cn, fxp@hit.edu.cn, eezsy@uestc.edu.cn

Abstract

As a neuromorphic camera with high temporal resolution, spike camera can capture dynamic scenes with high-speed motion. Recently, spike camera with a color filter array (CFA) has been developed for color imaging. There are some methods for spike camera demosaicing to reconstruct color images from Bayer-pattern spike streams. However, the demosaicing results are bothered by severe noise in spike streams, to which previous works pay less attention. In this paper, we propose an iterative joint demosaicing and denoising network (SJDD-Net) for spike cameras based on the observation model. Firstly, we design a color spike representation (CSR) to learn latent representation from Bayer-pattern spike streams. In CSR, we propose an offset-sharing deformable convolution module to align temporal features of color channels. Then we develop a spike noise estimator (SNE) to obtain features of the noise distribution. Finally, a color correlation prior (CCP) module is proposed to utilize the color correlation for better details. For training and evaluation, we designed a spike camera simulator to generate Bayer-pattern spike streams with synthesized noise. Besides, we captured some Bayer-pattern spike streams, building the first real-world captured dataset to our knowledge. Experimental results show that our method can restore clean images from Bayer-pattern spike streams. The source codes and dataset are available at https://github.com/csycdong/SJDD-Net.

Introduction

Neuromorphic camera is a type of image sensor designed to mimic the structure and functionality of the human visual system, which sends asynchronous signals with a high temporal resolution. Besides the well-known event camera (Litzenberger et al. 2006; Lichtsteiner, Posch, and Delbruck 2008; Posch, Matolin, and Wohlgenannt 2010) providing light intensity changes to record motion, there is another recently invented neuromorphic camera called spike camera (Dong, Huang, and Tian 2017; Dong et al. 2019; Huang et al. 2023). Different from event camera which records the ***relative*** change of light intensity, spike camera records the ***absolute*** light intensity. Spike camera is designed for emerging vision applications, such as autonomous driving and unmanned aerial vehicle. By continuously accumulating photons and firing spike signals, spike camera is able to record dynamic scenes with high-speed motion.

To recover the scenes from captured spike streams, there have been some works (L. Zhu, S. Dong, T. Huang, and Y. Tian 2019; Zhao, Xiong, and Huang 2020; Zhao et al. 2021b; Zheng et al. 2021; Chen et al. 2022) about spike camera reconstruction, which focus on the first-generation spike camera that can only reproduce gray-scale image signals. Recently, spike camera with color filter array (CFA) has been developed to record dynamic scenes with color information and meanwhile enjoy the benefits of spike imaging. In addition to noise in conventional digital cameras such as thermal noise and readout noise, there is also quantization noise in spike streams, as a result of the mechanism of spike cameras. At present, there are some spike camera demosaicing methods (Dong et al. 2022) to reconstruct color images from Bayer-pattern spike streams. However, the previous demosaicing methods pay less attention to the noise in spike streams and the importance of exploring an efficient representation of Bayer-pattern spike streams. In summary, color imaging of spike camera meets the following challenges:

(1) **Severe noise in spike streams.** Due to the imaging mechanism, the noise of spike cameras is severe. The existing spike camera demosaicing methods focus more on inferring the missing pixels of each color channel, not paying enough attention to the noise in spike streams.

(2) **Representation of Bayer-pattern spike streams.** Previous spike camera demosaicing methods tend to employ an intuitive way to represent spike camera data, without considering the motion and missing pixels of color channels.

(3) **Lack of Bayer-pattern spike datasets.** For better speed and performance, the task can be handled by deep learning-based methods. However, there are no released datasets for the training or evaluation of deep models.

To address the issues, we propose a joint demosaicing and denoising network (SJDD-Net) for spike cameras, based on the observation model of spike camera JDD. Inspired by optimization derivation, SJDD-Net is an iterative network, which considers both the temporal and color correlation. To learn the latent and motion-aligned representation, we first design a color spike representation (CSR). In particular, we utilize the motion offset of each color channel to jointly align the temporal features, resulting in an offset-sharing de-

*Corresponding author.

formable convolution module in CSR. Considering the severe noise in spike streams, a spike noise estimator (SNE) is proposed to learn the features of spike noise in color channels. Then we proposed a color correlation prior (CCP) to reconstruct the color image, utilizing the correlation of color channels. Besides, we developed a simulator according to the structure of the spike camera with CFA, generating a large number of Bayer-pattern spike streams for training and evaluation. To evaluate our performance on real-world data, we also capture some spike streams and build the first real-world captured spike camera demosaicing dataset.

Experiments show that our method can construct color images from both synthesized and real-world Bayer-pattern spike streams, with promising visual quality. The main contributions of this paper can be summarized as follows:

- Based on the observation model of spike camera JDD, we propose an iterative joint demosaicing and denoising network for spike cameras, SJDD-Net. Both the temporal and color correlations are considered in our method to reconstruct color images with fine details.
- We design a color spike representation to learn the latent and motion-aligned representation from Bayer-pattern spike streams, in which an offset-sharing deformable convolution module is proposed to jointly align the temporal features of each color channel.
- We build a Bayer-pattern spike stream (BSS) dataset, which is the first real-world captured spike camera demosaicing dataset to the best of our knowledge.

Related Work

Reconstruction from Event Camera Data. Event camera (Litzenberger et al. 2006; Lichtsteiner, Posch, and Delbruck 2008; Posch, Matolin, and Wohlgenannt 2010), which is a well-known neuromorphic camera, sends asynchronous events with a high temporal resolution to record light intensity changes of scenes. How to reconstruct the scenes from signals recorded by event cameras is an active topic in the field. As event cameras provide only relative intensity changes, reconstructing scenes from event streams is challenging. As early research, an Extended Kalman Filter (Kim et al. 2008) is proposed to reconstruct images from event signals. Bardow *et al.* (Bardow, Davison, and Leutenegger 2016) utilize the primal-dual algorithm to jointly infer optical flow and light intensity. Besides, Munda *et al.* (Munda, Reinbacher, and Pock 2018) and Scheerlinck *et al.* (Scheerlinck, Barnes, and Mahony 2019) try to reconstruct intensity images with direct event integration. Inspired by the performance of deep learning, some works (Rebecq et al. 2019a,b; Scheerlinck et al. 2020) focus on the design of deep neural networks, achieving performance improvement.

Reconstruction from Spike Camera Data. Spike camera (Dong, Huang, and Tian 2017; Dong et al. 2019; Huang et al. 2023), as another neuromorphic camera, accumulates incoming photons continuously and generates spike streams to record the light intensity in dynamic scenes. In recent years, some reconstruction methods for spike cameras have been proposed to obtain images from spike streams. Zhu *et al.* (L. Zhu, S. Dong, T. Huang, and Y. Tian 2019) propose TFI that infers light intensity by inter-spike intervals, and TFP that utilizes the count of spikes within a temporal window. However, TFI often produces noisy results, and TFP tends to generate blurry results. To overcome these limitations, researchers have tried to reconstruct dynamic scenes from spike streams by mimicking human vision, as done in TVS (Zhu et al. 2020). Inspired by the success of deep learning, Spk2ImgNet (Zhao et al. 2021b) utilizes a deep convolutional neural network to reconstruct dynamic scenes from spike streams, achieving excellent performance. To reconstruct high-resolution images, some research (Zhao et al. 2021a, 2023) aims at spike camera super-resolution reconstruction. Besides, there are also works (Xia et al. 2023b; Chang et al. 2023) on reconstruction from both spike stream and image frames. All the methods above focus on grayscale spike streams. Aiming for recently invented spike cameras with color filter array (CFA), 3DRI (Dong et al. 2022) is proposed to first handle the color reconstruction task.

Joint Demosaicing and Denoising. Demosaicing (Malvar, He, and Cutler 2004; Kiku et al. 2013; Monno et al. 2017) is to reconstruct color images from sub-sampled data using CFAs. However, the input data is usually noisy in practical applications, involving thermal noise, dark current noise, readout noise, and so on. Therefore, many works focus on demosaicing and denoising simultaneously. JDD methods can be divided into traditional methods (Hirakawa and Parks 2006; Condat and Mosaddegh 2012; Khashabi et al. 2014) and deep learning methods (Gharbi et al. 2016; Liu et al. 2020; Guo, Liang, and Zhang 2021). Traditional methods often handle the task based on the image prior or mathematical models. For example, Condat *et al.* (Condat and Mosaddegh 2012) handle the task by minimizing total variation. Deep learning methods usually depend on training an end-to-end model. As the green channel enjoys twice the sampling rate and better quality, a self-guidance network (Liu et al. 2020) is proposed to guide the inference of all missing values using initially estimated green channels. Similarly, the green channel prior is also used in GCPNet (Guo, Liang, and Zhang 2021). Besides, Qian *et al.* study the effects of pipelines on joint demosaicing, denoising, and super-resolution, proposing TENet (Qian et al. 2022) with state-of-the-art performance for the mixture problem.

Problem Statement

Spike Camera Model

Spike camera consists of an array of $H \times W$ pixels where each pixel accumulates the incoming photons continuously and fires spike signals asynchronously. For each pixel, the received incoming photons will be transferred to voltage. Once the voltage reaches a certain threshold θ, the pixel fires a spike signal and resets its accumulator, restarting a new *integration and firing* (IF) cycle (Dong, Huang, and Tian 2017; Zhao et al. 2021c). In particular, the accumulator voltage of a pixel (x, y) for an arbitrary time t can be formulated as:

$$\mathbf{A}(x, y, t) = \int_0^t \eta \cdot \mathbf{I}(x, y, \delta) d\delta \mod \theta, \tag{1}$$

where $\mathbf{I}(x, y, \delta)$ denotes the instantaneous arriving rate of incoming photons of pixel (x, y) at time δ, and η denotes

the photoelectric conversion rate. The spike signal should be checked and read out immediately in the ideal spike camera model. However, the *checking and resetting* (CR) process is controlled by a clock signal in the real hardware implementation. Thus the reading time of the spike signals is quantified with a constant CR period T, resulting in some quantization errors. To be specific, the spike camera adopts high-speed CR to check spike signals of each pixel (e.g., at 40,000Hz), resulting in a binary ("1" for spike, and "0" for no spike) spike stream with a shape of $N \times H \times W$.

Color Imaging with Spike Camera

To capture color information of dynamic scenes, the Bayer pattern CFA is recently employed on the spike sensor, so that the spike camera can produce spike signals of different color channels at different pixels according to the Bayer pattern color arrangement. As a result, the output spike frames from the spike camera can be split into three color channels (i.e., red, green, and blue) with missing pixels.

During an extremely short time, the rapidly changing scene can be represented by a sequence of latent light intensity frames $\{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_n\}$. Then the i-th latent light intensity frame $\mathbf{y}_i$ can be written as:

$$\mathbf{y}_i = \mathbf{M} \cdot \Omega_{k \to i} \cdot \mathbf{x}, \quad i \in \{1, 2, ..., n\}, \tag{2}$$

where $\mathbf{x}$ denotes the target light intensity to be reconstructed from the spike frames captured within the period, $\mathbf{M}$ denotes the Bayer-pattern color mask, k denotes the middle time index of the period, and $\Omega_{k \to i}$ denotes the motion transform matrix from k to i. However, there is noise from the sensor in the latent intensity frames, and we have:

$$\widetilde{\mathbf{y}}_i = \mathbf{y}_i + \mathbf{n}_i, \tag{3}$$

where $\widetilde{\mathbf{y}}_i$ denotes the i-th latent light intensity frame recorded on the sensor, and $\mathbf{n}_i$ denotes the noise. From the latent light intensity frames, the spike stream with missing pixels in each color channel can be generated as:

$$\{\mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_{n'}\} = \mathcal{I}(\{\widetilde{\mathbf{y}}_1, \widetilde{\mathbf{y}}_2, ..., \widetilde{\mathbf{y}}_n\}, \theta), \tag{4}$$

where $\mathcal{I}(\cdot)$ denotes the IF process, and θ denotes the firing threshold mentioned above. So far, we have obtained the Bayer-pattern spike camera data. JDD for Spike camera is to predict the missing pixels and reconstruct a clean color image with three complete channels from $\{\mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_{n'}\}$, thus achieving color imaging of spike camera.

Noise Model of Spike Camera

Similar to conventional cameras, there are also diverse noises like photon shot noise, readout noise, and thermal noise in spike cameras. In many works (Foi et al. 2008; Wei et al. 2020; Byun, Cha, and Moon 2021), the real-world image sensor noise can be modeled with the Poisson-Gaussian noise (GPN). Under this model, the camera noise $\mathbf{n}_i^c$ can be divided into source-independent Gaussian noise like readout noise, and source-dependent Poisson noise like photon shot noise. The Poisson-Gaussian noise $\mathbf{n}_i^c$ of the i-th latent light intensity frame $\mathbf{y}_i$ can be represented by following a heteroscedastic Gaussian distribution (Foi et al. 2008) as:

$$\mathbf{n}_i^c \sim \mathcal{N}(0, \sigma_g^2 + \sigma_p^2 \cdot \mathbf{y}_i), \tag{5}$$

where σ_g and σ_p denote the noise level parameter of Gaussian noise and Poisson noise. In practice, the parameters are decided by the internal design of the sensor. For spike cameras, noise is severe and can not be ignored. To be specific, we have $\mathbf{n}_i = \mathbf{n}_i^c + \mathbf{n}_i^q$, where $\mathbf{n}_i^q$ denotes the quantization noise that comes from the quantization errors. Due to the constant CR period in hardware implementation, quantization noise is prone to occur. For example, the voltage of a pixel almost reaches θ before CR, while the signal for the time point is 0. In addition, the voltage of a pixel reaches $n \cdot \theta, n \in \mathbb{Z}^+$, while the signal is the same as the pixels with voltage just reaching the firing threshold θ.

Method

JDD for Spike Camera

As mentioned above, the Bayer-pattern spike stream is binary data with spatial missing pixels and noise in each color channel, which involves high-speed motion. Therefore, the data is challenging to be directly processed. Inspired by previous works (Hu et al. 2022; Zhao et al. 2022; Xia et al. 2023a) using a representation to interpret gray-scale spikes, we tend to design a latent representation for Bayer-pattern spikes. With Bayer-pattern spike frames and the corresponding color mask $\mathbf{M}$ as input, the representation is designed to produce the sum of latent intensity frames without missing pixels and motion offset, which can be modeled as:

$$\mathcal{R}(\{\mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_{n'}\}, \mathbf{M}) = \sum_{i=1}^{n} \Omega_{k \to i}^{-1} \cdot \mathbf{M}^{-1} \widetilde{\mathbf{y}}_i, \tag{6}$$

where $\Omega_{k \to i}^{-1}$ denotes the inverse motion transform matrix from time point i to k, $\mathbf{M}^{-1}$ denotes the inverse transform of sub-sampling by the color mask, and $\widetilde{\mathbf{y}}_i$ denotes the i-th latent light intensity frame recorded on the sensor. Based on the observation model in Eqn. (2), Eqn. (3), and the representation in Eqn. (6), the optimization function of the spike camera JDD can be written by:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \sum_{i=1}^{n} \frac{1}{2} \|\mathbf{M} \cdot \Omega_{k \to i} \cdot \mathbf{x} - \widetilde{\mathbf{y}}_i\|_2^2 + \lambda \Phi(\mathbf{x}), \tag{7}$$

where $\Phi(\mathbf{x})$ denotes a regularization term, which contains the image prior of $\mathbf{x}$, and λ is a weight parameter. It's a common strategy to convert an optimization problem to more tractable sub-problems. To separately solve the optimization function, we can employ half-quadratic splitting (HQS) to split it into two sub-objective functions. To be specific, by introducing equivalent inverse transformation and an auxiliary variable $\mathbf{z}$, Eqn. (7) can be formulated as follows:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \sum_{i=1}^{n} \frac{1}{2} \|\mathbf{x} - \Omega_{k \to i}^{-1} \cdot \mathbf{M}^{-1} \widetilde{\mathbf{y}}_i\|_2^2 + \lambda \Phi(\mathbf{z}), \quad \text{s.t. } \mathbf{z} = \mathbf{x}. \tag{8}$$

Then the objective function can be optimized by iteratively optimizing two sub-objective functions corresponding to the fidelity term and the prior term:

$$\begin{cases} \mathbf{z}_{k+1} = \arg\min_{\mathbf{z}} \frac{\mu}{2} \|\mathbf{z} - \mathbf{x}_k\|_2^2 + \lambda \Phi(\mathbf{z}), & \text{(9a)} \\ \mathbf{x}_{k+1} = \arg\min_{\mathbf{x}} \sum_{i=1}^{n} \frac{1}{2} \|\mathbf{x} - \Omega_{k \to i}^{-1} \cdot \mathbf{M}^{-1} \widetilde{\mathbf{y}}_i\|_2^2 + \mu \|\mathbf{x} - \mathbf{z}_{k+1}\|_2^2. & \text{(9b)} \end{cases}$$

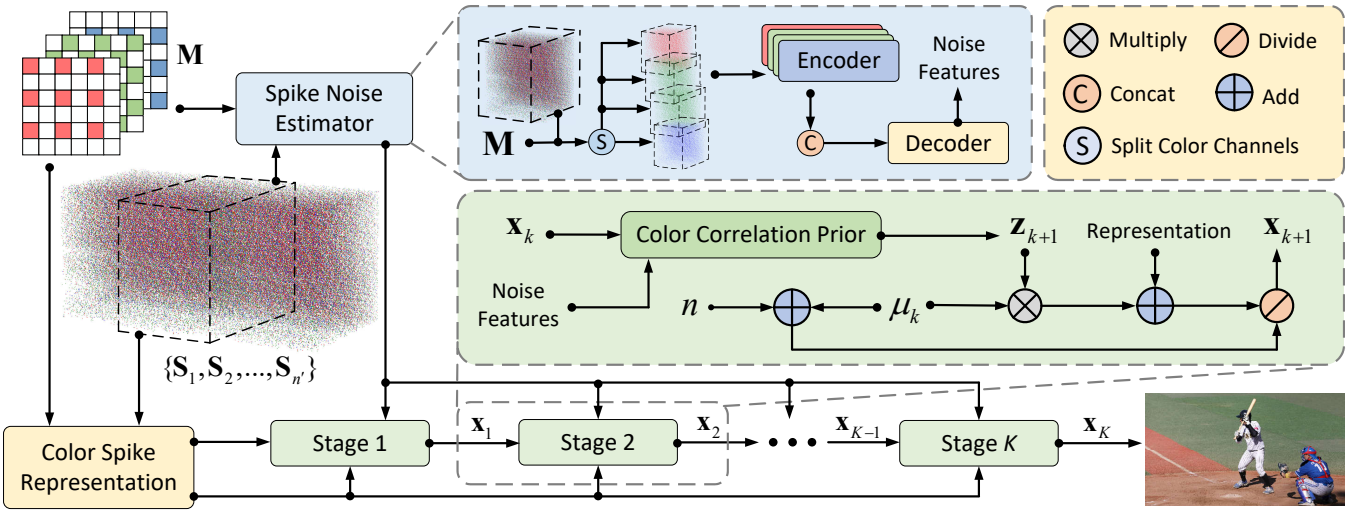

Figure 1: The overall architecture of our proposed SJDD-Net, which consists of K iteration stages. The spike noise estimator (SNE) is to learn the spike noise feature from input spike frames. The color spike representation (CSR) is to learn temporal aligned features without missing pixels. The color correlation prior (CCP) is to model the image prior by a neural network.

Eqn. (9a) denotes the sub-objective function related to the color image prior, where μ/λ is a parameter about the noise level. As the noise level of the spike stream is unknown, we can employ a spike noise estimator $\mathcal{E}(\cdot)$ to estimate the noise information, with Bayer-pattern spike stream $\{\mathbf{S}_2, ..., \mathbf{S}_{n'}\}$ and the color mask $\mathbf{M}$ as input. Then Eqn. (9a) can be modeled by a color image prior function $\mathcal{P}(\cdot)$ as follows:

$$\mathbf{z}_{k+1} = \mathcal{P}\left(\mathbf{x}_k, \mathcal{E}(\{\mathbf{S}_2, ..., \mathbf{S}_{n'}\}, \mathbf{M})\right). \tag{10}$$

In particular, the prior term function can be handled by a deep neural network to utilize powerful non-linear modeling capabilities, which will be discussed later. Finally, based on Eqn. (6), the quadratic regularized least-squares problem Eqn. (9b) related to the fidelity term can be solved by:

$$\mathbf{x}_{k+1} = \frac{1}{n+\mu} \cdot (\mu \mathbf{z}_{k+1} + \mathcal{R}(\{\mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_{n'}\}, \mathbf{M})). \tag{11}$$

Proposed JDD Network for Spike Camera

Overall Architecture. Inspired by the optimization process, we design an iterative network (SJDD-Net) for spike camera JDD as shown in Fig. 1, with K iteration stages. The input of SJDD-Net is a short clip of the Bayer-pattern spike stream $\{\mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_{n'}\}$, and the output is a color image. Considering the motion and missing pixels, the spike stream and color mask are first passed to the color spike representation (CSR) module $\mathcal{R}(\cdot)$ for aligned spike representation, which utilizes the temporal correlation and motion consistency of color channels. At the same time, the spike stream is also passed to our spike noise estimator (SNE) module $\mathcal{E}(\cdot)$ to learn the spike noise features. Specifically, to estimate the pixel-level noise features from each color channel, the spike stream is split into four channels according to the color mask. Then the features from each color channel are extracted via the corresponding residual block-based encoder. After that, the features are concatenated as the input of the decoder with several convolution layers to generate the global noise features. Initialized from the representation, $\mathbf{x}_0$ is passed to stage one. According to Eqn. (10), we propose a color correlation prior (CCP) module $\mathcal{P}(\cdot)$, utilizing the correlation of color channels and spike noise features for reconstruction. With the learnable parameter μ_k, we calculate $\mathbf{x}_1$ from the CCP output $\mathbf{z}_1$ and the representation based on Eqn. (11), producing the output of this stage, and so on in the following stages. In the end, we obtain the final color image $\mathbf{x}_K$ from the Bayer-pattern spike stream.

Color Spike Representation. According to Eqn. (6), we design a color spike representation (CSR) module to learn temporal aligned features without missing pixels as shown in Fig. 2. To split out the spikes of each color channel for independent encoding, we first multiply the spike frames with the mask for each color, resulting in the spike frames with missing pixels. Utilizing the temporal continuity of spike frames, we propose a multi-scale 3D encoder to handle the missing pixels, where three 3D convolution layers with different kernel sizes are applied to the spike frames for multi-scale temporal feature extraction. With the features from the three paths, we learn weights to multiply the features for rescaling and temporally fuse the features via convolution. After that, we pass the features to our proposed offset-sharing deformable convolution (OSDCN) module to align the features. To be specific, we divide the features of each color channel into temporal overlapping feature blocks. For each block, we extract deeper features and concatenate them with the features of the temporally middle block for offset learning. As the color channels are captured at the same time, they can be regarded as sharing the same motion offset. Thus we learn the sharing offset for each feature block from the offsets of the three channels. Then the block is aligned using the sharing offset. Besides, we add the color mask to the input of offset learning for each color, so that the missing pixels can be located. As a result, all the feature blocks are

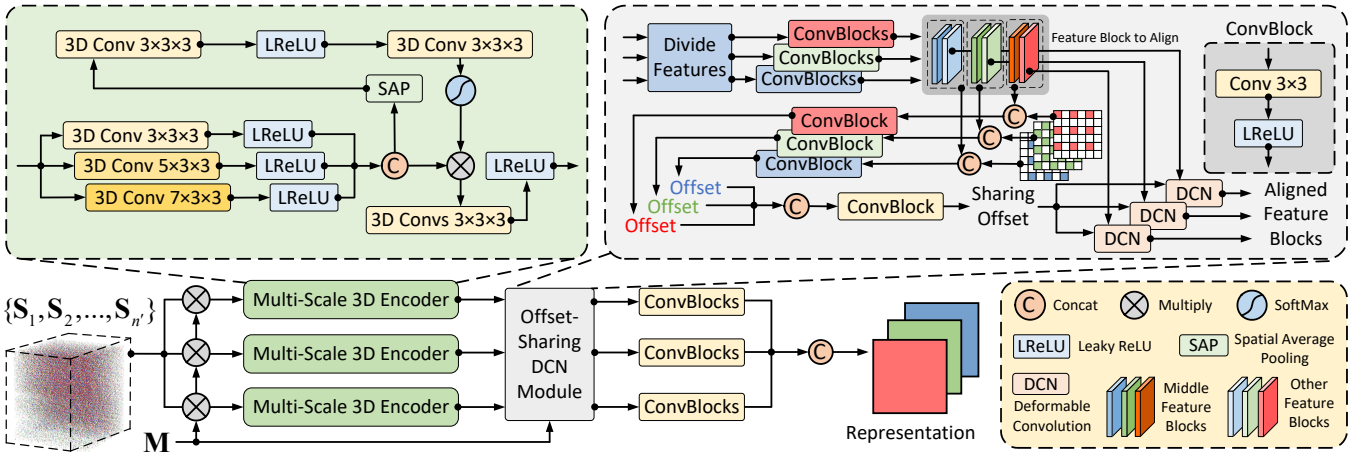

Figure 2: Illustration of the proposed color spike representation (CSR) module for Bayer-pattern spike camera data.

aligned. Finally, we fuse the blocks of each channel, resulting in the representation of the Bayer-pattern spike stream.

Color Correlation Prior. To model the image prior, we propose a color correlation prior module according to Eqn. (10), with $\mathbf{x}_k$ and the noise features from SNE as the input. The CCP module contains three encoders and one decoder as shown in Fig. 3. Considering the global spike noise, we first concatenate each color channel with the noise features from SNE as the input of their encoder to learn deep features. Utilizing the correlation of colors, we sum the features of each downsampling phase for all the encoders and pass them to the corresponding upsampling phase of the decoder, so that the features of each color can be considered together for output with better texture and details.

Experiments

Dataset

Synthesized Data. For training, a large number of spike stream-color image pairs are required. However, it's hard to collect so many high-quality data pairs, especially in scenes with high-speed motion. Inspired by the success of the simulator in (Zhao et al. 2021b), we propose a simulator to generate Bayer-pattern spike streams with synthesized noise for spike camera JDD. As shown in Fig. 4, we regard input video as the scene to record. Assuming the motion between two adjacent frames is consistent, we approximate the continuous dynamic light intensity variation process via estimated motion information. Then the light intensity frames are sub-sampled according to the color mask to simulate the CFA on the sensor. After that, we generate noise and add it to the sub-sampled light intensity of each color channel according to the noise model. Finally, we simulate accumulating light intensity and check the accumulated value periodically for firing spikes, resulting in simulated spike frames.

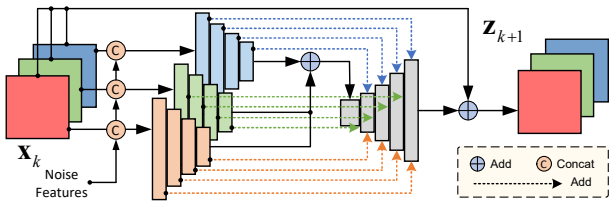

Figure 3: The color correlation prior (CCP) module.

With the simulator, we use videos from REDS (Nah et al. 2019) and DAVIS (Pont-Tuset et al. 2017) to generate data. The scene changes of the former mainly come from camera motion, while those of the latter come from object motion. As a result, we generated 1,950 spike sequences for training. We also generated a DAVIS-based dataset (DSPK) with 120 sequences and a REDS-based dataset (RSPK) with 60 sequences for evaluation. In the experiments, we first employ the Gaussian noise (GN) model to evaluate non-blind JDD performance, where the noise level parameter σ is sampled from $U(0, 15)$. To evaluate the performance of our method on real-world captured data, we then employ the Gaussian-Poisson noise (GPN) model in the simulator, which is closer to real spike camera noise. σ_p, σ_g are both sampled from $U(0, 5)$ in the simulator. Therefore, we generated two versions of the training and evaluation datasets using the GN model and the GPN model, respectively.

Real-World Captured Data. To further evaluate the blind JDD performance of our method, we build a Bayer-pattern spike stream (BSS) dataset captured by a spike camera with CFA, which is the first real-world dataset to our knowledge. In the dataset, there are 28 sequences with a shape of 1000×1000, captured from both indoor and outdoor scenes with a 20,000 Hz sampling rate. Notably, BSS includes instances of both camera and object motion.

Implementation Details

In our implementation, the stage number K of iterations is set to 3, which will be further discussed in our ablation study. The CCP module in each iteration stage shares the same parameters. The number of input spike frames N is set to 39. We randomly crop the spike frames into 64×64 patches and set the batch size to 8. For data augmentation, we randomly perform horizontal and vertical flips on the input frames. During training, our models are optimized using

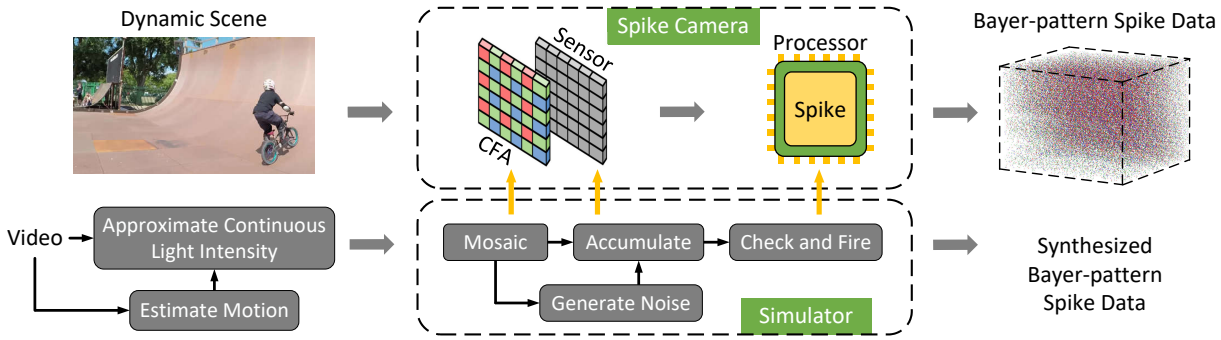

Figure 4: The pipeline of our designed spike camera simulator to generate Bayer-pattern spike camera data.

Noise	Dataset	Metric	MPRNet	FastDVDNet	TENet		GCPNet			Ours
			3DRI	3DRI	TFI	TFP	None	TFI	TFP	None
GN $\sigma = 0$	DSPK	PSNR	29.85	25.01	23.88	22.34	33.54	34.46	33.76	**34.72**
		SSIM	0.9291	0.8751	0.7995	0.7573	0.9452	**0.9535**	0.9475	0.9532
	RSPK	PSNR	28.92	25.71	25.36	22.91	31.73	32.77	32.11	**32.78**
		SSIM	0.8940	0.8787	0.8074	0.6829	0.9186	**0.9315**	0.9242	0.9256
GN $\sigma = 5$	DSPK	PSNR	29.73	24.96	23.85	22.33	33.06	33.89	33.21	**34.23**
		SSIM	0.9270	0.8715	0.7922	0.7549	0.9351	0.9414	0.9391	**0.9468**
	RSPK	PSNR	28.80	25.66	25.21	22.91	31.19	32.07	31.44	**32.28**
		SSIM	0.8898	0.8747	0.8012	0.6821	0.9072	0.9157	0.9116	**0.9169**
GN $\sigma = 10$	DSPK	PSNR	29.60	24.89	23.51	22.22	30.71	31.03	31.18	**33.23**
		SSIM	0.9209	0.8596	0.7570	0.7476	0.8463	0.8430	0.8918	**0.9319**
	RSPK	PSNR	28.47	25.46	24.73	22.85	29.11	29.47	29.42	**31.20**
		SSIM	0.8794	0.8619	0.7707	0.6780	0.8324	0.8283	0.8560	**0.8959**
GN $\sigma = 15$	DSPK	PSNR	29.27	24.73	22.75	22.01	27.12	27.36	28.51	**32.01**
		SSIM	0.9115	0.8355	0.6774	0.7307	0.6796	0.6920	0.8036	**0.9077**
	RSPK	PSNR	28.05	25.19	23.74	22.73	26.25	26.35	26.97	**29.99**
		SSIM	0.8644	0.8393	0.7049	0.6667	0.7134	0.7111	0.7718	**0.8702**
Random GPN	DSPK	PSNR	29.92	24.96	23.74	22.38	31.91	34.06	33.41	**34.45**
		SSIM	0.9227	0.8633	0.7811	0.7495	0.8922	0.9473	0.9418	**0.9499**
	RSPK	PSNR	33.99	25.30	25.16	22.97	30.15	32.46	31.94	**32.63**
		SSIM	0.8906	0.8608	0.7857	0.6796	0.8690	0.9260	0.9190	**0.9254**

Table 1: PSNR(dB) and SSIM comparison on the synthesized DSPK and RSPK datasets with various noise models.

Adam optimizer (Kingma and Ba 2014) with a learning rate initially set as 10^{-4}. The learning rate is scaled by 0.8 every 50 epochs. Besides, we use L_2 loss for training. All the models are trained using one NVIDIA GTX 1080Ti GPU.

Comparative Results

Quantitative Comparison. Since there are no JDD algorithms for spike cameras, we designed some methods to compare. First, we connect the SOTA spike camera demosaicing method 3DRI (Dong et al. 2022) with a single-frame denoising method MPRNet (Zamir et al. 2021) and a multi-frame denoising method FastDVDNet (Scheerlinck et al. 2020). With the pixel-independent spike camera reconstruction method TFI and TFP (L. Zhu, S. Dong, T. Huang, and Y. Tian 2019) for preprocessing, we perform JDD with a single-frame JDDSR method TENet (Qian et al. 2022) and a multi-frame JDD method GCPNet(Guo, Liang, and Zhang 2021). Considering the input of GCPNet is multi-frames, we try to directly input spike frames. As a result, we have 7 comparison methods as shown in Table 1. Then we evaluate models trained with random GN on DSPK and RSPK with $\sigma \in \{0, 5, 10, 15\}$ and models trained with random GPN on datasets with random GPN. In our settings, the former is non-blind JDD, while the latter is blind. For the non-blind experiments, noise levels are provided for the methods except ours and MPRNet-3DRI involving no noise level input.

According to Table 1, our method has significant performance superiority over other methods in both non-blind and blind experiments. As the binary spike streams are quite different from raw images, GCPNet with spike input performs worse than GCPNet-TFI and GCPNet-TFP. In particular, methods with TFI input perform better than methods with TFP input, as TFP is easy to bring motion blur. With limited temporal information, the single-frame TENet-based methods don't perform well. In contrast, the single-frame method MPRNet with 3DRI for demosaicing previously achieves better performance. As there is no module to handle motion like GCPNet-based methods, the performance of the multi-

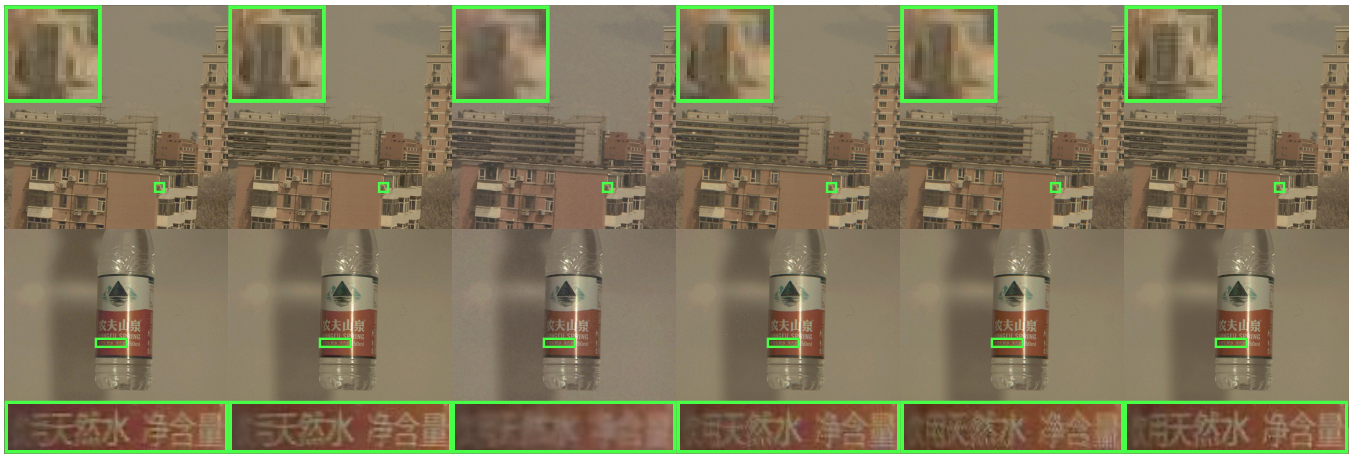

Figure 5: Visual comparison on real-world captured Bayer-pattern spike streams from BSS. The first spike stream is captured by a fast-moving spike camera. The second one records a fast-rotating water bottle. Please enlarge for better visualization.

frame method FastDVDNet-3DRI is also limited.

Visualization Comparison. Fig. 5 shows the visual results on real-world captured spike streams. The first sample is captured outdoors, whose motion comes from the camera shake. The motion of the second sample which is captured indoors is caused by object shake. Compared to FastDVDNet-based and MPRNet-based methods, our method and GCPNet-based methods, which contain modules for alignment, can suppress more motion blur. Compared to GCPNet-TFI with more noise and GCPNet-TFP with more blur, our method achieves better visual quality.

Ablation Study

As shown in Table 2, we perform an ablation study to demonstrate the effectiveness of our CSR, SNE, and CCP modules in SJDD-Net. To verify the effect of the multi-scale 3D encoder in CSR, we set the kernel size of the three 3D convolution paths to the same in (A). Without offset sharing, we use independent deformable convolution for each color channel in (B). We also remove the color mask from the input of offset computation in (C). By comparing these cases with the final model (F), we find the design in CSR brings some performance improvements. In (D), we remove SNE to show the effectiveness of the noise features input for CCP. After that, we use a single-head encoder for CCP in (E), which verifies the effectiveness of the multi-head encoder design of CCP. Besides the proposed modules, we

Index	PSNR	SSIM	Parameters	Running Time
(A)	30.99dB	0.8850	3.771M	0.042s
(B)	29.58dB	0.8465	3.776M	0.044s
(C)	31.16dB	0.8880	3.769M	0.042s
(D)	30.83dB	0.8846	**2.364M**	0.039s
(E)	31.11dB	0.8914	3.495M	**0.038s**
(F)	**32.01dB**	**0.9077**	3.776M	0.044s

Table 2: Ablation study on DSPK with GN ($\sigma = 15$).

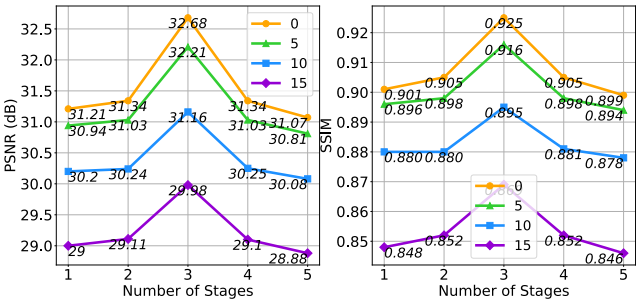

Figure 6: Ablation study of the number of iteration stages K on RSPK, with different GN ($\sigma \in \{0, 5, 10, 15\}$).

also conduct an ablation study of the number of iteration stages K under different noise level settings. According to the results in Fig. 6, K is set to 3 in our final method.

Conclusion

We propose a JDD network for spike cameras, considering the noise of the binary spike camera data. To learn the latent and motion-aligned representation from the input spike stream, we propose CSR consisting of an offset-sharing DCN module to align temporal features of color channels. Then we develop SNE to estimate noise features of the Bayer-pattern spike stream. Besides, we design the CCP module to utilize color correlation for better reconstruction. Experiments demonstrate that our method is promising in restoring color images from Bayer-pattern spike streams.

Acknowledgements

This work was supported in part by the National Key R&D Program of China under Grant 2021YFF0900501, and in part by the National Natural Science Foundation of China under Grants 62072009, 22127807, U22B2035.

References

Bardow, P.; Davison, A. J.; and Leutenegger, S. 2016. Simultaneous optical flow and intensity estimation from an event camera. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 884–892.

Byun, J.; Cha, S.; and Moon, T. 2021. Fbi-denoiser: Fast blind image denoiser for poisson-gaussian noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5768–5777.

Chang, Y.; Zhou, C.; Hong, Y.; Hu, L.; Xu, C.; Huang, T.; and Shi, B. 2023. 1000 FPS HDR Video With a Spike-RGB Hybrid Camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 22180–22190.

Chen, S.; Duan, C.; Yu, Z.; Xiong, R.; and Huang, T. 2022. Self-supervised mutual learning for dynamic scene reconstruction of spiking camera. IJCAI.

Condat, L.; and Mosaddegh, S. 2012. Joint demosaicking and denoising by total variation minimization. In *2012 19th IEEE International Conference on Image Processing*, 2781–2784. IEEE.

Dong, S.; Huang, T.; and Tian, Y. 2017. Spike Camera and Its Coding Methods. In *Data Compression Conference (DCC)*, 437.

Dong, S.; Zhu, L.; Xu, D.; Tian, Y.; and Huang, T. 2019. An Efficient Coding Method for Spike Camera Using Inter-Spike Intervals. In *Data Compression Conference (DCC)*, 568.

Dong, Y.; Zhao, J.; Xiong, R.; and Huang, T. 2022. 3D Residual Interpolation for Spike Camera Demosaicing. In *2022 IEEE International Conference on Image Processing (ICIP)*, 1461–1465. IEEE.

Foi, A.; Trimeche, M.; Katkovnik, V.; and Egiazarian, K. 2008. Practical Poissonian-Gaussian noise modeling and fitting for single-image raw-data. *IEEE transactions on image processing*, 17(10): 1737–1754.

Gharbi, M.; Chaurasia, G.; Paris, S.; and Durand, F. 2016. Deep joint demosaicking and denoising. *ACM Transactions on Graphics (ToG)*, 35(6): 1–12.

Guo, S.; Liang, Z.; and Zhang, L. 2021. Joint denoising and demosaicking with green channel prior for real-world burst images. *IEEE Transactions on Image Processing*, 30: 6930–6942.

Hirakawa, K.; and Parks, T. W. 2006. Joint demosaicing and denoising. *IEEE Transactions on Image Processing*, 15(8): 2146–2157.

Hu, L.; Zhao, R.; Ding, Z.; Ma, L.; Shi, B.; Xiong, R.; and Huang, T. 2022. Optical flow estimation for spiking camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17844–17853.

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

Khashabi, D.; Nowozin, S.; Jancsary, J.; and Fitzgibbon, A. W. 2014. Joint demosaicing and denoising via learned nonparametric random fields. *IEEE Transactions on Image Processing*, 23(12): 4968–4981.

Kiku, D.; Monno, Y.; Tanaka, M.; and Okutomi, M. 2013. Residual interpolation for color image demosaicking. In *IEEE International Conference on Image Processing (ICIP)*, 2304–2308.

Kim, H.; Handa, A.; Benosman, R.; Ieng, S.-H.; and Davison, A. J. 2008. Simultaneous mosaicing and tracking with an event camera. *J. Solid State Circ*, 43: 566–576.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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

Lichtsteiner, P.; Posch, C.; and Delbruck, T. 2008. A 128 $times$128 120 dB 15mu s latency asynchronous temporal contrast vision sensor. *IEEE journal of solid-state circuits*, 43(2): 566–576.

Litzenberger, M.; Posch, C.; Bauer, D.; Belbachir, A. N.; Schon, P.; Kohn, B.; and Garn, H. 2006. Embedded vision system for real-time object tracking using an asynchronous transient vision sensor. In *2006 IEEE 12th Digital Signal Processing Workshop & 4th IEEE Signal Processing Education Workshop*, 173–178. IEEE.

Liu, L.; Jia, X.; Liu, J.; and Tian, Q. 2020. Joint demosaicing and denoising with self guidance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2240–2249.

Malvar, H. S.; He, L.-w.; and Cutler, R. 2004. High-quality linear interpolation for demosaicing of Bayer-patterned color images. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 3, 485–488.

Monno, Y.; Kiku, D.; Tanaka, M.; and Okutomi, M. 2017. Adaptive residual interpolation for color and multispectral image demosaicking. *Sensors*, 17(12): 2787.

Munda, G.; Reinbacher, C.; and Pock, T. 2018. Real-time intensity-image reconstruction for event cameras using manifold regularisation. *International Journal of Computer Vision*, 126: 1381–1393.

Nah, S.; Baik, S.; Hong, S.; Moon, G.; Son, S.; Timofte, R.; and Mu Lee, K. 2019. NTIRE 2019 challenge on video deblurring and super-resolution: Dataset and study. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 1996–2005.

Pont-Tuset, J.; Perazzi, F.; Caelles, S.; Arbeláez, P.; Sorkine-Hornung, A.; and Van Gool, L. 2017. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*.

Posch, C.; Matolin, D.; and Wohlgenannt, R. 2010. A QVGA 143 dB dynamic range frame-free PWM image sensor with lossless pixel-level video compression and time-domain CDS. *IEEE Journal of Solid-State Circuits*, 46(1): 259–275.

Qian, G.; Wang, Y.; Gu, J.; Dong, C.; Heidrich, W.; Ghanem, B.; and Ren, J. S. 2022. Rethinking Learning-based Demosaicing, Denoising, and Super-Resolution Pipeline. In *2022 IEEE International Conference on Computational Photography (ICCP)*, 1–12. IEEE.

Rebecq, H.; Ranftl, R.; Koltun, V.; and Scaramuzza, D. 2019a. Events-to-video: Bringing modern computer vision to event cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3857–3866.

Rebecq, H.; Ranftl, R.; Koltun, V.; and Scaramuzza, D. 2019b. High speed and high dynamic range video with an event camera. *IEEE transactions on pattern analysis and machine intelligence*, 43(6): 1964–1980.

Scheerlinck, C.; Barnes, N.; and Mahony, R. 2019. Continuous-time intensity estimation using event cameras. In *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part V*, 308–324. Springer.

Scheerlinck, C.; Rebecq, H.; Gehrig, D.; Barnes, N.; Mahony, R.; and Scaramuzza, D. 2020. Fast image reconstruction with an event camera. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 156–163.

Wei, K.; Fu, Y.; Yang, J.; and Huang, H. 2020. A physics-based noise formation model for extreme low-light raw denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2758–2767.

Xia, L.; Ding, Z.; Zhao, R.; Zhang, J.; Ma, L.; Yu, Z.; Huang, T.; and Xiong, R. 2023a. Unsupervised Optical Flow Estimation with Dynamic Timing Representation for Spike Camera. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Xia, L.; Zhao, J.; Xiong, R.; and Huang, T. 2023b. SVFI: spiking-based video frame interpolation for high-speed motion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 2910–2918.

Zamir, S. W.; Arora, A.; Khan, S.; Hayat, M.; Khan, F. S.; Yang, M.-H.; and Shao, L. 2021. Multi-stage progressive image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 14821–14831.

Zhao, J.; Xie, J.; Xiong, R.; Zhang, J.; Yu, Z.; and Huang, T. 2021a. Super resolve dynamic scene from continuous spike streams. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2533–2542.

Zhao, J.; Xiong, R.; and Huang, T. 2020. High-speed motion scene reconstruction for spike camera via motion aligned filtering. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, 1–5.

Zhao, J.; Xiong, R.; Liu, H.; Zhang, J.; and Huang, T. 2021b. Spk2ImgNet: Learning to Reconstruct Dynamic Scene from Continuous Spike Stream. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 11996–12005.

Zhao, J.; Xiong, R.; Xie, J.; Shi, B.; Yu, Z.; Gao, W.; and Huang, T. 2021c. Reconstructing Clear Image for High-Speed Motion Scene With a Retina-Inspired Spike Camera. *IEEE Transactions on Computational Imaging*, 8: 12–27.

Zhao, J.; Xiong, R.; Zhang, J.; Zhao, R.; Liu, H.; and Huang, T. 2023. Learning to super-resolve dynamic scenes for neuromorphic spike camera. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 3579–3587.

Zhao, R.; Xiong, R.; Zhao, J.; Yu, Z.; Fan, X.; and Huang, T. 2022. Learning optical flow from continuous spike streams. *Advances in Neural Information Processing Systems*, 35: 7905–7920.

Zheng, Y.; Zheng, L.; Yu, Z.; Shi, B.; Tian, Y.; and Huang, T. 2021. High-speed Image Reconstruction through Short-term Plasticity for Spiking Cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6358–6367.

Zhu, L.; Dong, S.; Li, J.; Huang, T.; and Tian, Y. 2020. Retina-like visual image reconstruction via spiking neural model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1438–1446.