Super-Resolution Reconstruction from Bayer-Pattern Spike Streams

Yanchen Dong[1], Ruiqin Xiong[1]*, Jian Zhang[2], Zhaofei Yu[3], Xiaopeng Fan[4], Shuyuan Zhu[5], Tiejun Huang[1,3]
[1]School of Computer Science, Peking University
[2]School of Electronic and Computer Engineering, Peking University
[3]Institute for Artificial Intelligence, Peking University
[4]School of Computer Science and Technology, Harbin Institute of Technology
[5]University of Electronic Science and Technology of China
yanchendong@stu.pku.edu.cn, {rqxiong, zhangjian.sz, yuzf12, tjhuang}@pku.edu.cn,
fxp@hit.edu.cn, eezsy@uestc.edu.cn

Abstract

Spike camera is a neuromorphic vision sensor that can capture highly dynamic scenes by generating a continuous stream of binary spikes to represent the arrival of photons at very high temporal resolution. Equipped with Bayer color filter array (CFA), color spike camera (CSC) has been invented to capture color information. Although spike camera has already demonstrated great potential for high-speed imaging, its spatial resolution is limited compared with conventional digital cameras. This paper proposes a Color Spike Camera Super-Resolution (CSCSR) network to super-resolve higher-resolution color images from spike camera streams with Bayer CFA. To be specific, we first propose a representation for Bayer-pattern spike streams, exploring local temporal information with global perception to represent the binary data. Then we exploit the CFA layout and sub-pixel level motion to collect temporal pixels for the spatial super-resolution of each color channel. In particular, a residual-based module for feature refinement is developed to reduce the impact of motion estimation errors. Considering color correlation, we jointly utilize the multi-stage temporal-pixel features of color channels to reconstruct the high-resolution color image. Experimental results demonstrate that the proposed scheme can reconstruct satisfactory color images with both high temporal and spatial resolution from low-resolution Bayer-pattern spike streams. The source codes are available at https://github.com/csycdong/CSCSR.

1. Introduction

With the quick development of high-speed vision applications such as autonomous driving and unmanned aerial vehicles, the demand for cameras that can capture high-speed motion and respond quickly is increasing. Most conventional digital cameras use a certain exposure time window to accumulate photoelectric signals and create a snapshot image, which requires the scene to be still during the exposure interval. Due to the exposure time, conventional cameras suffer from motion blur when capturing high-speed scenes. To be specific, a given point of moving objects can move during exposure time, bringing undesired motion blur to the image. Therefore, it's hard for conventional digital cameras to meet the requirements of high-speed imaging.

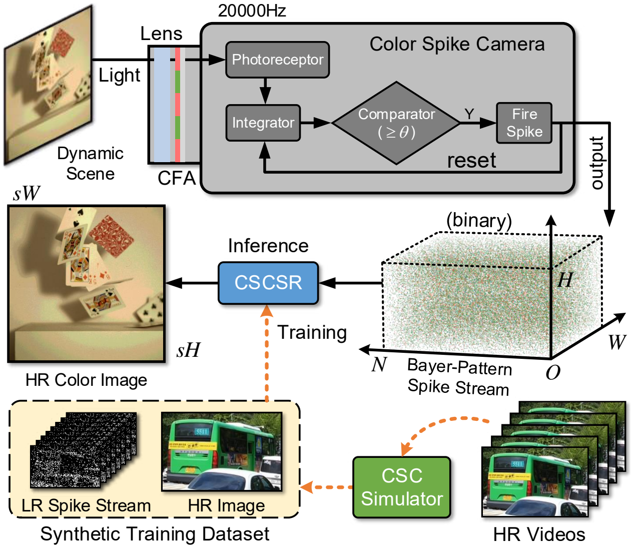

Figure 1. Pipeline of super-resolution reconstruction from Bayer-pattern spike streams. The super-resolution scale is denoted as s.

Mimicking the structure of human vision, neuromorphic cameras with ultra-high temporal resolution (*e.g.*, 20,000Hz) show great potential in capturing high-speed scenes. A well-known type of neuromorphic camera called

[*]Corresponding author.

event camera [19, 21, 24] is sensitive to motion in dynamic scenes, which can record *relative* light intensity changes. To record more texture details, *another* type of neuromorphic camera called spike camera [8, 9, 49] captures *absolute* light intensity via an “integrate-and-fire” mechanism. As the first-generation spike camera can only reproduce gray-level signals, color spike camera, which employs the Bayer-pattern color filter array (CFA), has been invented to record dynamic scenes with color information and meanwhile enjoy the benefits of high-speed imaging.

Although capable of capturing color information like conventional cameras, the spatial resolution of color spike cameras is limited compared to recent digital cameras, which is a trade-off for high temporal resolution and low latency. Therefore, we aim to reconstruct high-resolution (HR) color images from low-resolution (LR) Bayer-pattern spike streams, achieving both high temporal and spatial resolution with a color spike camera. An intuitive way is to combine color spike camera reconstruction methods [10] with image super-resolution (SR) algorithms [7, 18, 20]. However, information among temporal neighboring frames that is essential for SR will be degraded in the first reconstruction stage of this scheme. Besides, high-speed motion and quantization noise within the Bayer-pattern spike streams also bring challenges to the SR task. As a result, how to improve the spatial resolution as well as the imaging quality for color spike cameras is worth studying.

In this paper, we propose a Color Spike Camera Super-Resolution (CSCSR) network to super-resolve dynamic scenes from Bayer-pattern spike streams. To represent the binary data with color and motion information, we first develop a Bayer-pattern Spike Stream Representation (BSSR) to explore local temporal information with global perception. As the pixels of a single frame are limited for spatial super-resolution, we propose a Motion-guided Super-Resolution (MSR) module to search temporal pixels of each color channel according to the Bayer-pattern color layout and sub-pixel level motion jointly estimated from color channels. In most cases, there are errors in the estimated motion, causing an undesired impact on the temporal pixel search. Therefore, a Feature Refinement (FR) module is designed based on the residual between a temporal average reference and the temporal pixels, resulting in refined temporal-pixel features. Finally, we further extract features and jointly utilize multi-stage features of each color channel to restore the final HR image, achieved by our Color Correlation-based Reconstruction (CCR) module.

To train and evaluate our models, we design a simulator based on the mechanism of color spike cameras, which can generate data pairs for SR reconstruction. Besides, we also capture some real-world Bayer-pattern spike streams for further evaluation. Experimental results on both synthetic and real-world captured data show that our method can reconstruct HR color images from LR Bayer-pattern spike streams, with good texture details. The main contribution of our work can be summarized as follows:

- We develop a color spike camera super-resolution network to reconstruct high-resolution color images from low-resolution Bayer-pattern spike streams, which is the first attempt to the best of our knowledge.
- To reconstruct details beyond the sensor resolution, we explore the sub-pixel level motion of each color channel and propose a motion-guided super-resolution module to collect temporal pixels according to the color layout.
- Experiments demonstrate that our method outperforms all the existing methods, achieving better quantitative and qualitative results in color spike camera super-resolution.

2. Related Work

2.1. Reconstruction

Event Camera Reconstruction. Event camera [19, 21, 24] is a kind of neuromorphic camera that monitors relative light intensity changes, making it promising in capturing motion. With the high temporal resolution, reconstructing intensity images with an event camera is an active topic. Extended Kalman Filter [16] is the early attempt at event camera reconstruction, which is based on photometric constancy. Bardow et al. [1] estimate intensity as well as optical flow via the primal-dual algorithm. Inspired by the success of deep learning, E2VID [25] and FireNet [26] were presented with promoting reconstruction performance.

Spike Camera Reconstruction. Spike camera [8, 9, 33, 41, 49] is another type of neuromorphic camera. Different from event cameras, spike camera produces a continuous spike stream to record the dynamic scenes by accumulating photons and firing spike signals. To reconstruct images from spike streams, some exploration [11, 39, 40, 44] has been performed. Zhu et al. [48] proposed to utilize the number of spike signals within a temporal window (TFP) and neighboring spike intervals (TFI). For better performance, some work [37, 43, 47] tried to design reconstruction networks for spike cameras. In addition, there are also attempts at hybrid input [5, 32, 50] for reconstruction. Focusing on the newly invented color spike camera, 3DRI [10] was proposed for scene reconstruction with color information.

2.2. Super-Resolution

Image and Video Super-Resolution. In the last ten years, many researchers have been devoted to learning to super-resolve images. Dong et al. [7] proposed SRNet as the first SR network, bringing great performance gains. In recent years, Liang et al. [18] have achieved significant improvement based on Transformer [28]. Besides the progress in image SR, there are also some video SR work [2, 3, 14, 15] proposed to restore HR frames from LR video frames.

Event Camera Super-Resolution. Event camera SR includes upsampling from LR events to HR events [12, 17] and HR image reconstruction from LR events [6, 29, 31]. Li et al. [17] designed a two-stage scheme to solve the spatial SR problem of spatiotemporal events. Then a deep neural framework EventZoom [12] was proposed for event SR. To restore SR images from events directly, Choi et al. [6] proposed a network E2SRI and trained models using synthetic data. In contrast, another deep network EventSR [29] was implemented without ground truth HR images.

Spike Camera Super-Resolution. As a trade-off of temporal resolution, the spatial resolution of spike cameras is limited. To restore images with both high temporal and spatial resolution, there is some research about spike camera super-resolution. Zhao et al. [42] exploited relative motion and derived the relationship between light intensity and each spike for super-resolution. Xiang et al. [34] proposed an end-to-end network VidarSR to reconstruct HR images from LR spike streams. Besides, another network SpikeSR-Net [45] is designed based on the observation model of spike camera, achieving state-of-the-art performance.

3. Preliminary

3.1. Color Spike Camera

Color spike camera (CSC) is a recently implemented neuromorphic sensor with $H \times W$ pixels that work independently. To capture color information of dynamic scenes, color filter array is employed on the sensor. As a result, each pixel of the sensor corresponds to one of the three colors, red, green and blue, according to the Bayer-pattern color layout.

As shown in Fig. 1, CSC is a neuromorphic vision sensor that mimics the structure of human vision. Conventional digital cameras usually use a certain time window for exposure to accumulate photoelectric signals and compact them into a snapshot. In contrast, CSCs accumulate photons and fire spikes continuously to record dynamic scenes, resulting in a Bayer-pattern spike stream. As shown in Fig. 2, every individual pixel denoted as (x, y) within the sensor accumulates photons continuously and converts the instantaneous intensity into an electric signal. When the accumulated electric signals reach a predetermined threshold θ, the pixel triggers a flag indicating firing a spike. Then the pixel will be reset and continue to accumulate photons after the flag is checked, starting a new *accumulating and firing* process. The accumulated electric signal of a certain pixel (x, y) at arbitrary time point t can be formulated as

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

where η denotes the photoelectric conversion rate of the sensor, $\mathbf{I}_C(\tau, x, y)$ denotes the instantaneous intensity of a certain color at time τ, and $C \in \{R, G, B\}$ denotes the color determined by the Bayer-pattern color layout.

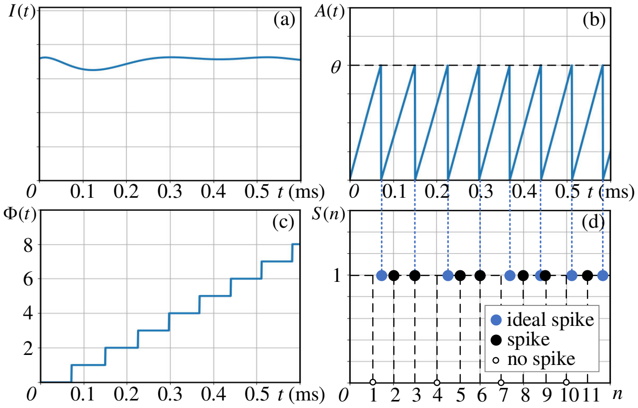

Figure 2. Illustration of the spike camera model. (a) The instantaneous intensity of a certain pixel. (b) The accumulated electric signal of the pixel. (c) The number of spikes that have been fired at the pixel. (d) The spike stream by the pixel, controlled by a clock.

Ideally, the pixels on the sensor would be reset immediately after firing a spike. The number of spikes that have been fired before arbitrary time point t can be written as

$$\Phi(t, x, y) = \lfloor \frac{1}{\theta} \int_0^t \mathbf{I}_C(\tau, x, y) d\tau \rfloor. \tag{2}$$

Then the read-out spike signal at the n-th ($n \in \mathbb{N}^+$) frame can be formulated as follows:

$$\mathbf{S}_n(x, y) = \begin{cases} 1, & \Phi(t_n, x, y) - \Phi(t_{n-1}, x, y) > 0 \\ 0, & \Phi(t_n, x, y) - \Phi(t_{n-1}, x, y) = 0 \end{cases}, \tag{3}$$

where t_n denotes the time point according to the n-th spike frame. However, the flags of firing spikes are checked and reset at discrete time points in real hardware implementation, bringing undesired quantization errors. To be specific, CSCs can only check them at discrete time points with a fixed sensor period T, which is controlled by a clock. If the flag of the pixel (x, y) is set up between time points $t = n \cdot T$ and $t' = (n - 1) \cdot T$, we can read out the spike signal $\mathbf{S}_n(x, y)$ as 1. Otherwise, we have $\mathbf{S}_n(x, y) = 0$. After N times of the *checking and resetting* process, CSC outputs a binary Bayer-pattern spike stream $\{\mathbf{S}_i\}_{i=1}^N$ with a spatial-temporal shape of $N \times H \times W$.

3.2. Bayer-Pattern Spike Stream to Raw Images

To reconstruct color images from the Bayer-pattern spike stream, an intuitive way is to convert it to raw images for demosaicing. To infer the raw images, we can employ some spike camera reconstruction methods that don't fuse values of spatial neighboring pixels of multiple colors, such as the spike interval-based method (TFI [48]). Specifically, the spike interval of two temporally neighboring spikes that covers the n-th spike frame can be formulated as

$$\begin{aligned} \Psi_n(x, y) = & \min\{k | \mathbf{S}_i(x, y) = 1, k \geq n\} \\ & - \max\{k | \mathbf{S}_i(x, y) = 1, k < n\}. \end{aligned} \tag{4}$$

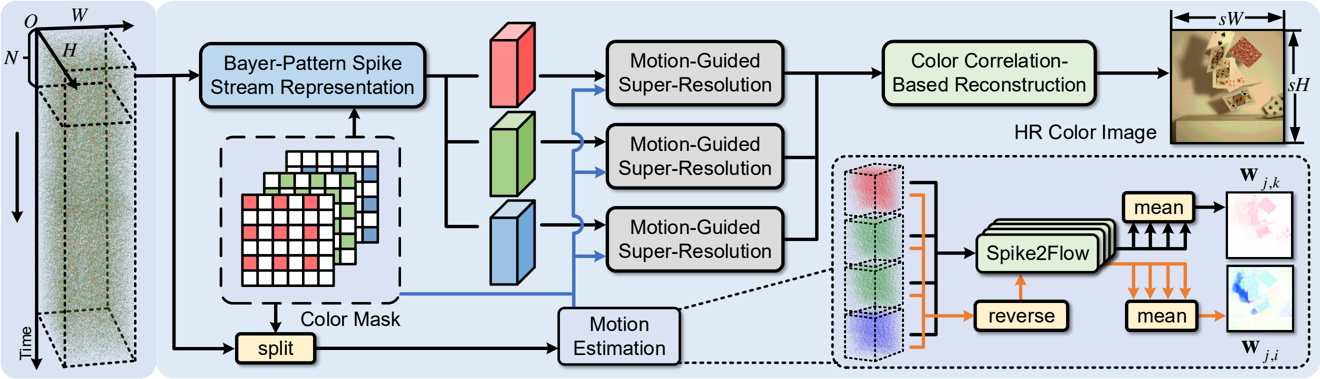
O
W
N
H
Time
Bayer-Pattern Spike Stream Representation
Color Mask
split
Motion Estimation
Motion-Guided Super-Resolution
Motion-Guided Super-Resolution
Motion-Guided Super-Resolution
Color Correlation-Based Reconstruction
HR Color Image
sW
sH
Spike2Flow
reverse
mean
mean
$\mathbf{W}_{j,k}$
$\mathbf{W}_{j,i}$

Figure 3. The overall architecture of CSCSR network. First, the Bayer-pattern spike stream representation (BSSR) is employed to represent the spike stream clip $\{\mathbf{S}_i\}_{i=1}^{N}$. Then the features are passed to the motion-guided super-resolution (MSR) module based on the temporal neighboring pixels of each color. Finally, the HR color image is restored by our color correlation-based reconstruction (CCR) module.

Then the n-th Bayer-pattern raw image can be estimated according to the intervals, which can be written as

$$\mathbf{R}_n(x,y) = \frac{\delta}{\Psi_n(x,y)}, \tag{5}$$

where δ denotes the maximum dynamic range. As to a spike number-based method (TFP [48]), the raw image $\mathbf{R}_n(x,y)$ can also be inferred based on the spike number within a temporal window with an odd window length w:

$$\mathbf{R}_n(x,y) = \frac{\delta}{w}\sum_{i=n+1}^{n+w} \mathbf{S}_{i-\frac{w+1}{2}}(x,y). \tag{6}$$

4. Method

4.1. Overall Architecture

To reconstruct HR color images from LR Bayer-pattern spike stream, we develop a CSCSR network whose overall architecture is shown in Fig. 3. The input of the network is a clip of the spike stream with a shape of $N \times H \times W$, denoted as $\{\mathbf{S}_i\}_{i=1}^{N}$. To represent color and motion information within the binary input, we first pass it to our proposed Bayer-pattern spike stream representation (BSSR) module. Considering the motion consistency of color channels, we split out four sequences according to the color layout and jointly estimate sub-pixel level motion from them. Guided by the motion information, temporal neighboring pixels of each color channel are searched in the motion-guided super-resolution (MSR) module, resulting in HR temporal-pixel features. Finally, the temporal-pixel features of the three color channels are fused to reconstruct the final HR image, exploiting the correlation of color channels.

4.2. Bayer-Pattern Spike Stream Representation

The spike signals in the Bayer-pattern spike stream mean the arrival of a certain photon amount for a certain color channel, which may suffer from quantization errors as introduced in Sec. 3.1. Considering the motion and color information within the binary data, how to represent it appropriately is worth studying. As shown in Fig. 4, we propose a Bayer-pattern spike stream representation module to represent the information contained in the data.

To keep the color information, we first perform element-wise multiplication between each frame of the spike sequence and the mask of each color, splitting the data into three sequences. For the sequence of each color channel, we develop an encoder to extract the temporal features and retain the relative motion. To be specific, we employ a sliding temporal window to extract the features of each time point. As the impact of quantization errors can be reduced by more temporal information, we extract the features from the whole clip serving as an additional global perception (GP) to the features of each time point. With the temporal window sliding, features of each color are obtained.

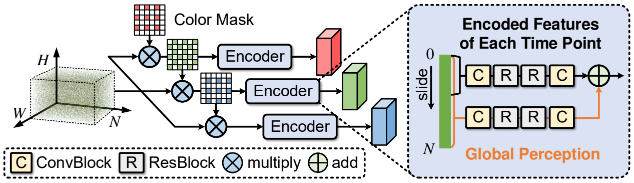

Figure 4. Illustration of the Bayer-pattern spike stream representation module, with a temporal sliding window in encoders. In the figure, “ConvBlock” denotes a 3 × 3 convolution layer followed by a ReLU activation function. “ResBlock” refers to [13].

4.3. Motion-Guided Super-Resolution

Joint Motion Estimation. Though the spatial pixels are limited, some pixels from the temporal domain are available for SR, which makes it essential to analyze the motion within the Bayer-pattern spike clip. Therefore, we employ the spike camera optical flow method Spike2Flow [46]. To apply the method to the Bayer-pattern spike stream, we split it into four single-channel spike sequences and estimate motion from each channel as shown in Fig. 3, resulting in op-

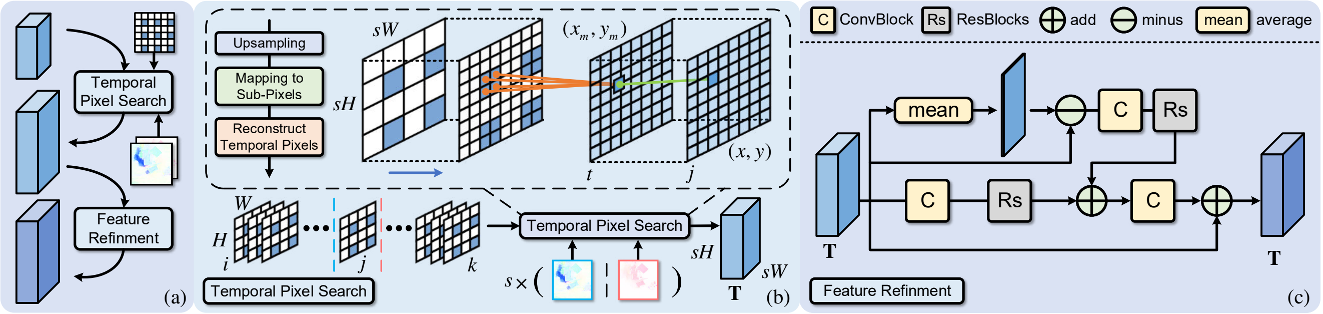

Figure 5. Illustration of the motion-guided super-resolution (MSR) module, taking the blue channel as an example. (a) Overview of the MSR module. (b) Motion-guided temporal pixel search, with three steps. (c) Structure of the feature refinement (FR) module.

tical flows $\{\mathbf{w}_{j,k}^C\}, C \in \{R, G_1, G_2, B\}$, from the middle time point j to the last time point k. By reversing the spike sequences, we then estimate optical flows $\{\mathbf{w}_{j,i}^C\}$ from the middle time point j to the first time point i. Though coming from different intensities, the color channels of the dynamic scene share the same motion. For more robust motion estimation, we jointly consider the color channels and obtain the final sub-pixel level optical flows as

$$(\mathbf{w}_{j,i}, \mathbf{w}_{j,k}) = \left(\frac{1}{4}\sum_{C} \mathbf{w}_{j,i}^C, \frac{1}{4}\sum_{C} \mathbf{w}_{j,k}^C\right). \tag{7}$$

Temporal Pixel Search. To provide features for the final reconstruction of the middle time point, we propose to fill the gap of spatial pixels from LR to HR based on the known temporal pixels at non-middle time points, considering the motion and color layout within the spike stream clip.

The positions of known spatial pixels of a certain color can be inferred according to the Bayer-pattern color mask. Denoting the SR scale as s, the target spatial resolution is $sH \times sW$. In the target, one originally known spatial position corresponds to s^2 neighboring positions, resulting in an upsampled color mask as shown in Fig. 5 (b). To get the extra known pixels after upsampling, we apply bilinear interpolation to the features, resulting in the super-resolved features $\mathbf{F}$. Utilizing the estimated optical flows $(\mathbf{w}_{j,i}, \mathbf{w}_{j,k})$, we can infer the new spatial position of every pixel at the middle time point j when moving to a certain time point t. To be specific, the optical flow $\mathbf{w}_{j,i}$ can be formulated as the pixel offset along x direction and y direction, which can be written as $\left[\mathbf{U}_j^i, \mathbf{V}_j^i\right]$. Considering the spatial upsampling, the offsets should be multiplied by the scale s. Since the time span from time point i or k to j is extremely small (0.025ms × N), we assume the motion is linear. As a result, after moving to a certain time point t, the new position of a certain original position (x, y) at the middle time point j can be written as follows:

$$\begin{cases} \left(x + s \cdot \dfrac{t-j}{i-j}\mathbf{U}_j^i(x,y), y + s \cdot \dfrac{t-j}{i-j}\mathbf{V}_j^i(x,y)\right), & t < j, \\ \left(x + s \cdot \dfrac{t-j}{k-j}\mathbf{U}_j^k(x,y), y + s \cdot \dfrac{t-j}{k-j}\mathbf{V}_j^k(x,y)\right), & t > j. \end{cases} \tag{8}$$

According to Eqn. (8), we can map all the pixels at the middle time point to their new positions at any other time point. However, the new positions are usually sub-pixel. Thus we are going to reconstruct the temporal feature value at the sub-pixel position by the values of known integer-pixels according to the upsampled color mask. In particular, the value of the new sub-pixel position (x_m, y_m) at time point t can be calculated based on the 4 nearest known pixels $\{(x_i, y_i)\}_{i=1}^4$, which can be formulated as follows:

$$\mathbf{T}(t,x,y) = \sum_{i=1}^{4} \frac{\phi_i}{\sum_{i=1}^4 \phi_i}\mathbf{F}(t, x_i, y_i), \tag{9}$$

where $\mathbf{T}$ denotes the temporal-pixel features, (x, y) denotes the original position at the middle time point, and ϕ_i denotes the weight of each known value, which can be written as

$$\phi_i = \frac{1}{\sqrt{(x_m - x_i)^2 + (y_m - y_i)^2}}. \tag{10}$$

After the temporal pixel search process, we can obtain the temporal-pixel features of each color channel for the following HR reconstruction of the middle time point.

Feature Refinement. In most cases, there are errors in the estimated motion, causing an undesired impact on the temporal pixel search. Therefore, we propose a feature refinement as shown in Fig. 5 (c). By averaging the input temporal-pixel features $\mathbf{T}$ temporally, a rough temporal average result of the channel as a reference can be obtained. Then we compute residuals between the reference and $\mathbf{T}$ to represent the pixel fluctuation of the frames. At last, we extract deeper features from the residuals and the based $\mathbf{T}$, and add them together for refined temporal-pixel features $\hat{\mathbf{T}}$.

4.4. Color Correlation-Based Reconstruction

With the temporal-pixel features, we are going to reconstruct the final HR image. Despite having different intensities, there is a correlation between color channels. Thus we can restore the image by integrating the features, achieving mutual information complementation of color channels.

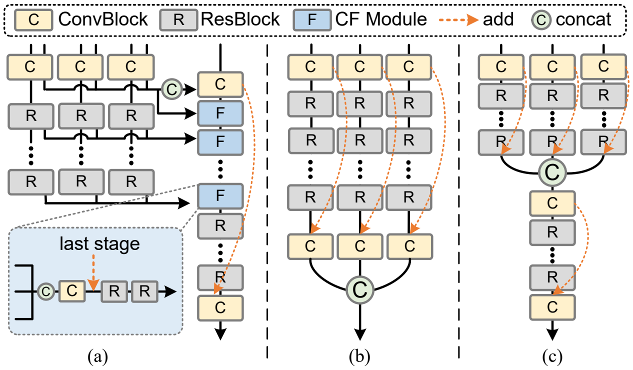

Figure 6. Illustration of three structures of the reconstruction module. (a) Our proposed color correlation-based reconstruction (CCR) module. (b) A reconstruction module considering no color correlation. (c) An intuitive design considering color correlation.

As shown in Fig. 6 (a), we design a reconstruction module utilizing multi-stage features of color channels. In particular, we first apply residual blocks to extract deeper features from the features of each color channel. To fuse the multi-color features of each stage for better aggregation, we develop a color fusion (CF) module to encode the features of color channels with the output of the previous CF stage. After several stages of CF, we employ residual blocks in the end to generate the final image from the fused features.

5. Experiments

5.1. Color Spike Camera Simulator

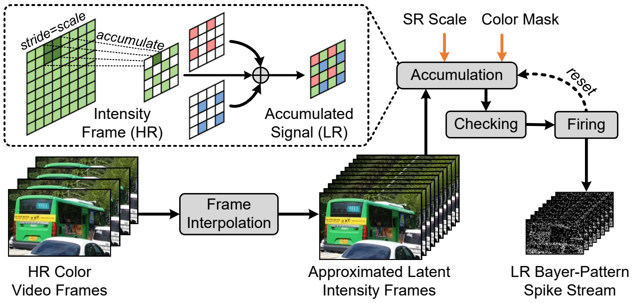

Figure 7. The pipeline of the color spike camera simulator, with the green channel as an example of signal accumulation.

To train models, a large amount of LR spike stream-HR color image pairs are required. However, it's hard to collect corresponding high-quality HR images, especially in scenes with high-speed motion. Inspired by the success of gray-scale spike camera simulators [43, 45], we also developed a simulator for color spike cameras. The pipeline of our simulator is shown in Fig. 7. To generate spike streams, we regard the input video as the dynamic scene to be recorded. As the frame rate of the input video is usually limited, the temporal information is not enough for the simulator of CSC with ultra-high temporal resolution to generate Bayer-pattern spike streams. To address this issue, we use a frame interpolation method [27] to generate latent intensity frames between the original video frames. To generate spike signals, we follow the mechanism introduced in Sec. 3.1 to accumulate light intensity from the latent frames and compare the accumulated value with the threshold periodically for firing spikes. As a result, a Bayer-pattern spike stream is generated from the input video. In particular, the pixel count used to accumulate intensity signals is determined by SR scales. For example, we accumulate signals of 4 pixels to generate the signal of one target pixel in the ×2 case. Considering the CFA, the 4 pixels come from one of the channels according to the Bayer-pattern color layout.

5.2. Experimental Settings

Implementation Details. In our implementation, we randomly crop the Bayer-pattern spike frames into patches with spatial resolution 96 × 96 for training, with the RGGB Bayer pattern. We set the number of spike frames N to 41 and the batch size to 4. The length w and stride s of the sliding temporal window are set to 11 and 3. We employ ℓ_1 loss function and use Adam optimizer with an initial learning rate of 10^{-4}, which will decay to 0.8 times after every 7500 iterations. Besides, we employ the PyTorch framework and train the models via an NVIDIA RTX3090 GPU.

Training and Evaluation Datasets. Using our developed CSC simulator, we generated the training dataset based on 240 scenes for training from REDS (120fps) [23], resulting in 5 × 240 = 1200 clips for each super-resolution scale (×2, ×3 and ×4). Then we generate 5 × 30 = 150 clips for evaluation based on 30 videos of the REDS evaluation dataset. To demonstrate the generalization to non-REDS-based data, we also generated evaluation datasets from two video SR datasets, Vid4 [22] and Vimeo [35]. Besides, we also captured some real-world Bayer-pattern spike streams to verify the performance further. The details of the synthetic evaluation datasets can be found in Table 1.

Source	Scenes	Samples	Resolution	Scales
REDS [23]	30	150	720×1280	2, 3, 4
Vid4 [22]	4	4	-	2, 3, 4
Vimeo [35]	7824	7824	448×256	4

Table 1. Details of the three synthetic evaluation datasets.

Comparison Methods. As there are no SR methods for CSCs, we first choose the state-of-the-art gray-scale spike camera SR networks SpikeSR-Net [45] and VidarSR [34] for comparison. To adapt the gray-scale methods to the task, we adopt two strategies: using Bayer-pattern spike streams for end-to-end training and handling each color channel separately. We denote the networks with the second strategy as SpikeSR-Net* and VidarSR*. Then we combine a CSC reconstruction method 3DRI [10] and an image/video SR method SwinIR [18]/BasicVSR [3], resulting in another two comparison methods 3DRI+SwinIR and 3DRI+BasicVSR.

Scale	Method	REDS-based Dataset			Vid4-based Dataset			Vimeo-based Dataset		
		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
×2	TFI [48] +TSCNN [4]	29.24dB	0.8173	0.3126	25.06dB	0.7786	0.3196	-	-	-
	TFP[48] +TSCNN	27.43dB	0.7565	0.3671	24.60dB	0.7666	0.3142	-	-	-
	TFI+Real-RawVSR [36]	31.03dB	0.8627	0.2478	26.79dB	0.8422	0.2508	-	-	-
	TFP+Real-RawVSR	30.81dB	0.8535	0.2601	27.10dB	0.8564	0.2329	-	-	-
	3DRI [10] +SwinIR [18]	31.13dB	0.8809	0.2191	26.57dB	0.8668	0.2100	-	-	-
	3DRI+BasicVSR [3]	31.60dB	0.8912	0.2052	27.03dB	0.8748	0.2021	-	-	-
	VidarSR [34]	30.81dB	0.8991	0.1884	26.44dB	0.8911	0.1791	-	-	-
	VidarSR*	30.27dB	0.8430	0.2631	25.28dB	0.8106	0.2538	-	-	-
	SpikeSR-Net [45]	32.38dB	0.8913	0.2099	28.34dB	0.8957	0.1833	-	-	-
	SpikeSR-Net*	29.66dB	0.8249	0.2861	24.62dB	0.7854	0.2721	-	-	-
	CSCSR (ours)	**33.39dB**	**0.9121**	**0.1764**	**28.77dB**	**0.9110**	**0.1612**	-	-	-
×3	TFI+TSCNN	27.22dB	0.7505	0.3842	22.91dB	0.6609	0.4058	-	-	-
	TFP+TSCNN	26.55dB	0.7176	0.4137	23.22dB	0.6800	0.3808	-	-	-
	TFI+Real-RawVSR	28.68dB	0.7942	0.3292	24.00dB	0.7225	0.3474	-	-	-
	TFP+Real-RawVSR	28.66dB	0.7883	0.3344	24.39dB	0.7407	0.3326	-	-	-
	3DRI+SwinIR	28.38dB	0.8136	0.2977	23.97dB	0.7534	0.3160	-	-	-
	3DRI+BasicVSR	28.65dB	0.8208	0.2905	24.08dB	0.7568	0.3135	-	-	-
	VidarSR	29.36dB	0.8414	0.2679	24.66dB	0.8009	0.2816	-	-	-
	VidarSR*	27.53dB	0.7670	0.3554	22.78dB	0.6742	0.3680	-	-	-
	SpikeSR-Net	29.79dB	0.8270	0.2970	25.33dB	0.7915	0.2989	-	-	-
	SpikeSR-Net*	26.73dB	0.7367	0.3908	22.12dB	0.6323	0.4067	-	-	-
	CSCSR (ours)	**29.92dB**	**0.8513**	**0.2653**	**25.68dB**	**0.8240**	**0.2761**	-	-	-
×4	TFI+TSCNN	25.98dB	0.7010	0.4310	21.69dB	0.5722	0.4659	24.77dB	0.7304	0.3554
	TFP+TSCNN	26.55dB	0.7176	0.4137	22.53dB	0.6240	0.4177	**25.56dB**	0.7587	0.3557
	TFI+Real-RawVSR	27.27dB	0.7391	0.3851	22.53dB	0.6240	0.4177	25.12dB	0.7492	0.3244
	TFP+Real-RawVSR	27.32dB	0.7347	0.3885	22.81dB	0.6408	0.3634	25.15dB	0.7493	0.3324
	3DRI+SwinIR	26.94dB	0.7544	0.3602	22.47dB	0.6533	0.3901	25.21dB	0.7423	0.3218
	3DRI+BasicVSR	26.94dB	0.7570	0.3605	22.52dB	0.6507	0.3973	25.05dB	0.7401	0.3204
	VidarSR	27.56dB	0.7692	0.3505	22.69dB	0.6712	0.3868	25.19dB	0.7565	0.3098
	VidarSR*	26.51dB	0.7135	0.4135	21.58dB	0.5880	0.4395	24.64dB	0.7352	0.3423
	SpikeSR-Net	28.28dB	0.7755	0.3517	23.59dB	0.6983	0.3727	25.27dB	0.7611	0.3119
	SpikeSR-Net*	25.93dB	0.6913	0.4403	21.01dB	0.5451	0.4718	24.46dB	0.7240	0.3593
	CSCSR (ours)	**28.77dB**	**0.7963**	**0.3253**	**23.79dB**	**0.7223**	**0.3555**	25.34dB	**0.7660**	**0.2967**

Table 2. Quantitative comparison. **Red** and blue indicate the best and the second-best performance, respectively. For each scene of Vimeo [35], there is a sequence of LR frames but only one corresponding HR frame (×4). Thus only the dataset of ×4 scale is generated for Vimeo.

It's also intuitive to combine methods of estimating raw images from Bayer-pattern spike streams (Sec. 3.2) and joint demosaicing and super-resolution (JDSR) methods. Thus we employ TFI or TFP [48] to infer raw images, followed by an image JDSR method TSCNN [4] or a video JDSR method Real-RawVSR [36] to reconstruct color images. In addition, we employ PSNR, SSIM [30] and LPIPS [38] (lower means better) as quantitative metrics for comparison.

5.3. Comparative Results

Quantitative Results. As shown in Table 2, we conduct comparative experiments on the three SR scales. It can be found that our method achieves the best quantitative results in most experiments. The state-of-the-art spike camera SR methods VidarSR and SpikeSR-Net achieve competitive performance. However, VidarSR* and SpikeSR-Net* that handle each color channel separately don't perform as well as VidarSR and SpikeSR-Net, due to the lack of consideration of color correlation. With the temporal information, the multi-frame method 3DRI+BasicVSR performs better than the single-frame method 3DRI+SwinIR in most cases. This is also true for the multi-frame JDSR method RawVSR and the single-frame JDSR method TSCNN. Besides, with the increase of the SR scale, TFP-based methods show growing competitiveness to TFI-based methods.

Visual Results. Fig. 8 presents the visual comparison (×4) on the synthetic data. The visual quality of results by our proposed method is better than the other methods, with less motion blur and more accurate color. There are artifacts in the results produced by VidarSR* and SpikeSR-Net*, which demonstrates the importance of the consideration of color correlation. In addition, the multi-frame methods show better visual quality. To demonstrate model generalization, we conduct experiments on real-world captured Bayer-pattern spike streams as shown in Fig. 9. In contrast, our method can restore better textures and details.

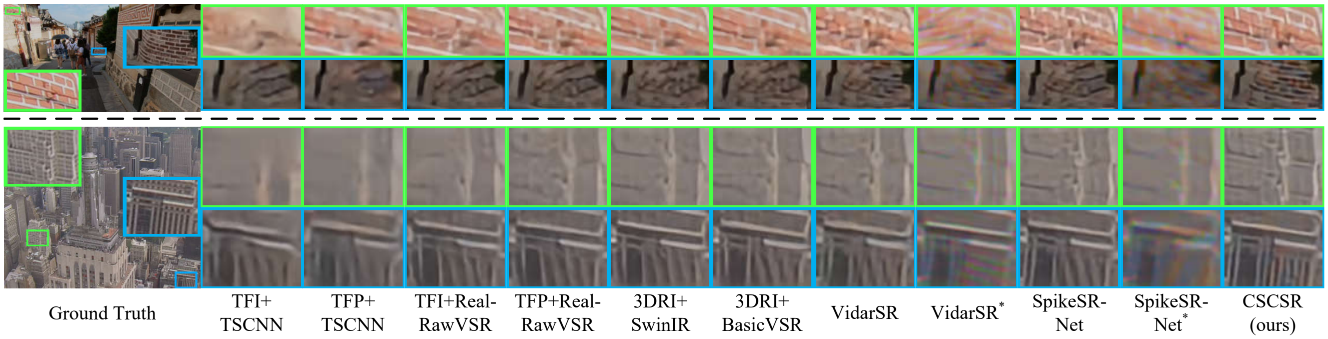

Figure 8. Visual comparison (×4) on the synthetic Bayer-pattern spike streams. The first sample comes from the REDS-based dataset, while the second one comes from the Vid4-based dataset. Please enlarge the figure for better comparison.

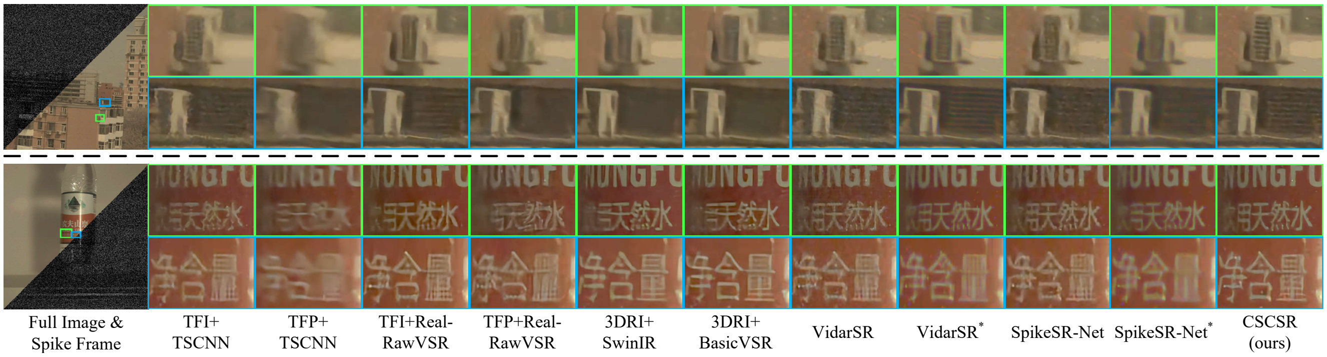

Figure 9. Visual comparison (×4) on the real-world Bayer-pattern spike streams. The first spike stream sample is captured by a fast-moving CSC. The second one records a fast-rotating water bottle. Please enlarge the figure for better comparison.

Case	Setting Description	×2	×3	×4
(A)	Removing BSSR module	33.04	29.57	28.57
(B)	Removing GP from BSSR	33.17	29.74	28.68
(C)	Interval-based representation	32.05	28.19	28.06
(D)	Independent motion estimation	33.22	29.55	28.59
(E)	Assembled motion estimation	30.24	27.08	26.66
(F)	Removing MSR module	32.66	29.42	28.39
(G)	Removing FR from MSR	33.07	29.70	28.56
(H)	Reconstruction module (b)	30.19	27.36	26.22
(I)	Reconstruction module (c)	33.24	29.69	28.66
(J)	Our final network	**33.39**	**29.92**	**28.77**

Table 3. Ablation study on the REDS-based evaluation dataset. Greener blocks represent higher PSNR(dB) performance.

5.4. Ablation Study

To verify the effectiveness of our methodology, we perform ablation studies on our modules and motion estimation strategy. Case (J) is our final CSCSR network. First, we remove the BSSR module in (A) to demonstrate the proposed representation. Then we remove the additional GP branch from BSSR in (B). We also replace BSSR with the interval-based representation [46, 48] in (C). To demonstrate our joint motion estimation strategy, we employ independent motion estimation for each color channel in (D) and an optical flow assembly strategy in (E). To be specific, the optical flows of color channels are assembled according to the Bayer pattern. After that, we implement studies on the MSR module. In (F), MSR is removed from the network. In (G), the FR module is removed from MSE to verify its effectiveness. Finally, (H) and (I) are the studies on the proposed reconstruction module, where the module is replaced by the structures in (b) and (c) of Fig. 6, with the same number of residual blocks. The study results are shown in Table 3.

6. Conclusions

In this paper, we present a deep network to restore HR color images from LR Bayer-pattern spike streams. To represent the binary data, we develop a representation that utilizes local temporal information with global perception. Then we propose to collect temporal pixels for spatial super-resolution according to the color layout and sub-pixel level motion. To reduce the impact of motion estimation errors, we design a feature refinement module based on residuals. Finally, multi-stage temporal-pixel features of each color channel are jointly considered, resulting in the final HR color image. Experimental results on both synthetic and real-world captured data demonstrate our performance.

Acknowledgments

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

References

[1] Patrick Bardow, Andrew J Davison, and Stefan Leutenegger. Simultaneous optical flow and intensity estimation from an event camera. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 884–892, 2016. 2

[2] Jose Caballero, Christian Ledig, Andrew Aitken, Alejandro Acosta, Johannes Totz, Zehan Wang, and Wenzhe Shi. Real-time video super-resolution with spatio-temporal networks and motion compensation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4778–4787, 2017. 2

[3] Kelvin CK Chan, Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. BasicVSR: The search for essential components in video super-resolution and beyond. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4947–4956, 2021. 2, 6, 7

[4] Kan Chang, Hengxin Li, Yufei Tan, Pak Lun Kevin Ding, and Baoxin Li. A two-stage convolutional neural network for joint demosaicking and super-resolution. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(7):4238–4254, 2021. 7

[5] Yakun Chang, Chu Zhou, Yuchen Hong, Liwen Hu, Chao Xu, Tiejun Huang, and Boxin Shi. 1000 FPS HDR video with a spike-rgb hybrid camera. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22180–22190, 2023. 2

[6] Jonghyun Choi, Kuk-Jin Yoon, et al. Learning to super resolve intensity images from events. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2768–2776, 2020. 3

[7] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2014. 2

[8] Siwei Dong, Tiejun Huang, and Yonghong Tian. Spike camera and its coding methods. In *Data Compression Conference (DCC)*, page 437, 2017. 2

[9] Siwei Dong, Lin Zhu, Daoyuan Xu, Yonghong Tian, and Tiejun Huang. An efficient coding method for spike camera using inter-spike intervals. In *Data Compression Conference (DCC)*, page 568, 2019. 2

[10] Yanchen Dong, Jing Zhao, Ruiqin Xiong, and Tiejun Huang. 3D residual interpolation for spike camera demosaicing. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 1461–1465. IEEE, 2022. 2, 6, 7

[11] Yanchen Dong, Jing Zhao, Ruiqin Xiong, and Tiejun Huang. High-speed scene reconstruction from low-light spike streams. In *2022 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, pages 1–5. IEEE, 2022. 2

[12] Peiqi Duan, Zihao W Wang, Xinyu Zhou, Yi Ma, and Boxin Shi. EventZoom: Learning to denoise and super resolve neuromorphic events. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12824–12833, 2021. 3

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 4

[14] Younghyun Jo, Seoung Wug Oh, Jaeyeon Kang, and Seon Joo Kim. Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3224–3232, 2018. 2

[15] Armin Kappeler, Seunghwan Yoo, Qiqin Dai, and Aggelos K Katsaggelos. Video super-resolution with convolutional neural networks. *IEEE Transactions on Computational Imaging*, 2(2):109–122, 2016. 2

[16] Hanme Kim, Ankur Handa, Ryad Benosman, Sio-Hoi Ieng, and Andrew J Davison. Simultaneous mosaicing and tracking with an event camera. *J. Solid State Circ*, 43:566–576, 2008. 2

[17] Hongmin Li, Guoqi Li, and Luping Shi. Super-resolution of spatiotemporal event-stream image. *Neurocomputing*, 335: 206–214, 2019. 3

[18] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. SwinIR: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1833–1844, 2021. 2, 6, 7

[19] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128 × 128 120 db 15 μs latency asynchronous temporal contrast vision sensor. *IEEE journal of solid-state circuits*, 43 (2):566–576, 2008. 2

[20] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *IEEE conference on computer vision and pattern recognition workshops (CVPRW)*, pages 136–144, 2017. 2

[21] Martin Litzenberger, Christoph Posch, D Bauer, Ahmed Nabil Belbachir, P Schon, B Kohn, and H Garn. 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*, pages 173–178. IEEE, 2006. 2

[22] Ce Liu and Deqing Sun. On bayesian adaptive video super resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(2):346–360, 2013. 6

[23] Seungjun Nah, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, Radu Timofte, and Kyoung Mu Lee. NTIRE 2019 challenge on video deblurring and super-resolution: Dataset and study. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 0–0, 2019. 6

[24] Christoph Posch, Daniel Matolin, and Rainer Wohlgenannt. 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, 2010. 2

[25] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. Events-to-video: Bringing modern computer vision to event cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3857–3866, 2019. 2

[26] Cedric Scheerlinck, Henri Rebecq, Daniel Gehrig, Nick Barnes, Robert Mahony, and Davide Scaramuzza. Fast image reconstruction with an event camera. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 156–163, 2020. 2

[27] Hyeonjun Sim, Jihyong Oh, and Munchurl Kim. XVFI: extreme video frame interpolation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14489–14498, 2021. 6

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[29] Lin Wang, Tae-Kyun Kim, and Kuk-Jin Yoon. EventSR: From asynchronous events to image reconstruction, restoration, and super-resolution via end-to-end adversarial learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8315–8325, 2020. 3

[30] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 7

[31] Zihao W. Wang, Peiqi Duan, Oliver Cossairt, Aggelos Katsaggelos, Tiejun Huang, and Boxin Shi. Joint filtering of intensity images and neuromorphic events for high-resolution noise-robust imaging. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1609–1619, 2020. 3

[32] Lujie Xia, Jing Zhao, Ruiqin Xiong, and Tiejun Huang. SVFI: Spiking-based video frame interpolation for high-speed motion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2910–2918, 2023. 2

[33] Lujie Xia, Ziluo Ding, Rui Zhao, Jiyuan Zhang, Lei Ma, Zhaofei Yu, Tiejun Huang, and Ruiqin Xiong. Unsupervised optical flow estimation with dynamic timing representation for spike camera. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[34] Xijie Xiang, Lin Zhu, Jianing Li, Yixuan Wang, Tiejun Huang, and Yonghong Tian. Learning super-resolution reconstruction for high temporal resolution spike stream. *IEEE Transactions on Circuits and Systems for Video Technology*, 2021. 3, 6, 7

[35] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *Int. J. Comput. Vis*, 127:1106–1125, 2019. 6, 7

[36] Huanjing Yue, Zhiming Zhang, and Jingyu Yang. Real-rawvsr: Real-world raw video super-resolution with a benchmark dataset. In *European Conference on Computer Vision*, pages 608–624. Springer, 2022. 7

[37] Jiyuan Zhang, Shanshan Jia, Zhaofei Yu, and Tiejun Huang. Learning temporal-ordered representation for spike streams based on discrete wavelet transforms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 137–147, 2023. 2

[38] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018. 7

[39] Yiyang Zhang, Ruiqin Xiong, and Tiejun Huang. Spike signal reconstruction based on inter-spike similarity. In *2022 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, pages 1–5. IEEE, 2022. 2

[40] Jing Zhao, Ruiqin Xiong, and Tiejun Huang. High-speed motion scene reconstruction for spike camera via motion aligned filtering. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, 2020. 2

[41] Jing Zhao, Ruiqin Xiong, Rui Zhao, Jin Wang, Siwei Ma, and Tiejun Huang. Motion estimation for spike camera data sequence via spike interval analysis. In *2020 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, pages 371–374. IEEE, 2020. 2

[42] Jing Zhao, Jiyu Xie, Ruiqin Xiong, Jian Zhang, Zhaofei Yu, and Tiejun Huang. Super resolve dynamic scene from continuous spike streams. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2533–2542, 2021. 3

[43] Jing Zhao, Ruiqin Xiong, Hangfan Liu, Jian Zhang, and Tiejun Huang. Spk2ImgNet: Learning to reconstruct dynamic scene from continuous spike stream. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11996–12005, 2021. 2, 6

[44] Jing Zhao, Ruiqin Xiong, Jiyu Xie, Boxin Shi, Zhaofei Yu, Wen Gao, and Tiejun Huang. Reconstructing clear image for high-speed motion scene with a retina-inspired spike camera. *IEEE Transactions on Computational Imaging*, 8:12–27, 2021. 2

[45] Jing Zhao, Ruiqin Xiong, Jian Zhang, Rui Zhao, Hangfan Liu, and Tiejun Huang. Learning to super-resolve dynamic scenes for neuromorphic spike camera. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3579–3587, 2023. 3, 6, 7

[46] Rui Zhao, Ruiqin Xiong, Jing Zhao, Zhaofei Yu, Xiaopeng Fan, and Tiejun Huang. Learning optical flow from continuous spike streams. *Advances in Neural Information Processing Systems*, 35:7905–7920, 2022. 4, 8

[47] Rui Zhao, Ruiqin Xiong, Jian Zhang, Zhaofei Yu, Shuyuan Zhu, Lei Ma, and Tiejun Huang. Spike camera image reconstruction using deep spiking neural networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 2023. 2

[48] Lin Zhu, Siwei Dong, Tiejun Huang, and Yonghong Tian. A retina-inspired sampling method for visual texture reconstruction. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1432–1437. IEEE, 2019. 2, 3, 4, 7, 8

[49] Lin Zhu, Siwei Dong, Tiejun Huang, and Yonghong Tian. Hybrid coding of spatiotemporal spike data for a bio-inspired camera. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(7):2837–2851, 2021. 2

[50] Lin Zhu, Jianing Li, Xiao Wang, Tiejun Huang, and Yonghong Tian. NeuSpike-Net: High speed video reconstruction via bio-inspired neuromorphic cameras. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2400–2409, 2021. 2