NeuSpike-Net: High Speed Video Reconstruction via Bio-inspired Neuromorphic Cameras

Lin Zhu[1,2] Jianing Li[1,2] Xiao Wang[2] Tiejun Huang[1] Yonghong Tian[1,2,*]
[1]Department of Computer Science and Technology, Peking University, Beijing, China
[2]Peng Cheng Laboratory, Shenzhen, China
{linzhu, lijianing, tjhuang, yhtian}@pku.edu.cn wangxiaocvpr@foxmail.com

Abstract

Neuromorphic vision sensor is a new bio-inspired imaging paradigm that emerged in recent years, which continuously sensing luminance intensity and firing asynchronous spikes (events) with high temporal resolution. Typically, there are two types of neuromorphic vision sensors, namely dynamic vision sensor (DVS) and spike camera. From the perspective of bio-inspired sampling, DVS only perceives movement by imitating the retinal periphery, while the spike camera was developed to perceive fine textures by simulating the fovea. It is meaningful to explore how to combine two types of neuromorphic cameras to reconstruct high quality image like human vision. In this paper, we propose a NeuSpike-Net to learn both the high dynamic range and high motion sensitivity of DVS and the full texture sampling of spike camera to achieve high-speed and high dynamic image reconstruction. We propose a novel representation to effectively extract the temporal information of spike and event data. By introducing the feature fusion module, the two types of neuromorphic data achieve complementary to each other. The experimental results on the simulated and real datasets demonstrate that the proposed approach is effective to reconstruct high-speed and high dynamic range images via the combination of spike and event data.

1. Introduction

In recent years, bio-inspired vision sensors have become very attractive in the field of self-driving cars, unmanned aerial vehicles, and autonomous mobile robots [23], due to their significant advantages over conventional frame-based cameras, such as high dynamic range and high temporal resolution [13, 24].

Generally speaking, there are two ways of bio-inspired visual sampling manner: temporal contrast sampling and integral sampling. Among them, dynamic vision sensor (DVS) [9, 1], a.k.a. event camera, is the most well-recognized bio-inspired vision sensor based on temporal contrast sampling, which measures the change of light intensity and outputs high dynamic range events. From the biological point of view, DVS imitates the periphery of the retina [36] which is sensitive to motion. However, it is very difficult to reconstruct the texture from DVS. To solve this problem, some event-based sensors were developed subsequently by combining DVS and a frame-based active-pixel sensor (APS) such as DAVIS [4], or adding an extra photo-measurement circuit such as ATIS [30] and CeleX [15]. However, a mismatch exists due to the difference in the sampling time resolution between two kinds of heterogeneous circuits. Recently, many algorithms were designed to reconstruct images using DVS [33, 31, 25, 34, 38, 44, 6]. There are also some algorithms that combine image and event to reconstruct texture images [28, 29, 27, 33], which can obtain more texture information than only using the DVS signal.

Different from DVS, there are a number of spiking image sensors following the basis of the integrate-and-fire neuron model [43, 19, 7, 35]. Some variants of spiking image sensors such as asynchronous pixel event tricolor vision sensor [22] and near infrared spiking image sensor [3] were proposed. Recently, Dong et al. [10, 47] proposed a spike camera based on fovea-like sampling method, which is with high spatial (250×400) and temporal resolutions (40,000 Hz). Moreover, there is a portable spike camera, a.k.a. Vidar, with a sampling rate of 20,000 Hz. For the spike camera, the spike firing frequency can be used to estimate light intensity [47]. Recently, a fovea-like texture reconstruction framework was proposed to reconstruct images [49]. In addition, some methods based on the spike camera were developed for spike coding [11, 48], tone mapping [16] and motion deblurring [45].

In human vision system [36], peripheral and foveal vision is not independent, but is directly connected [37]. It is biologically plausible to the periphery and fovea are complementary. This motivates us to explore a question: *how to combine the two neuromorphic cameras to reconstruct*

[*] Corresponding author.

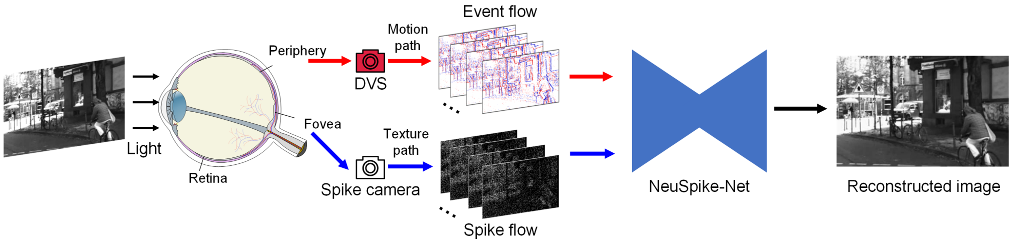

Figure 1. **The motivation of our approach.** Neuromorphic cameras are inspired by the retina. DVS perceives movement by imitating the retinal periphery, while the spike camera was developed to sense fine textures by simulating the fovea. In this work, we combine the spike and event data to achieve effective information complementarity and get better reconstruction quality.

high quality visual image like human vision? In fact, DVS has the ability of high speed and high dynamic range sensing, but it is difficult to perceive the texture information. In contrast, the spike camera has the ability of full texture sampling like a conventional camera, but its dynamic range is greatly affected by the noise. Meanwhile, its sampling ability depends on high light intensity of the scene. In application, it is meaningful to combine the high dynamic range of event data and full texture of spike data to reconstruct high quality images.

In this paper, we combine the two types of neuromorphic data to reconstruct high quality texture images, especially in complex scenes such as high speed and low light. Contributions of this paper are summarized as follows.

1) We first propose the reconstruction network combining spike and event cameras (NeuSpike-Net). According to the characteristics of the neuromorphic data, we explore the learning-based joint reconstruction strategy, which can achieve high quality full texture reconstruction in complex scenes with different light intensities and motion speeds.

2) We propose a neuromorphic data representation to extract useful temporal information hidden in spike and event data. With the help of neuromorphic data representation, the proposed network can effectively learn the features of spike and event data.

3) We propose to simulate multi-scale spike data, which considers the various noises existing in spike camera. The simulated dataset is generated for network training and testing by simulating different light intensities and motion speeds. Moreover, we build a hybrid cameras system to collect real world datasets to test the effectiveness of the model.

2. The Motivation of Our Approach

In this section, we analyze the interaction of fovea and peripheral in retina (Section 2.1), and the sampling principle of two types of neuromorphic cameras (Section 2.2). The relationship between the spike data and event data is further analyzed (Section 2.3), and the noise distribution of the spike camera is discussed in Section 2.4.

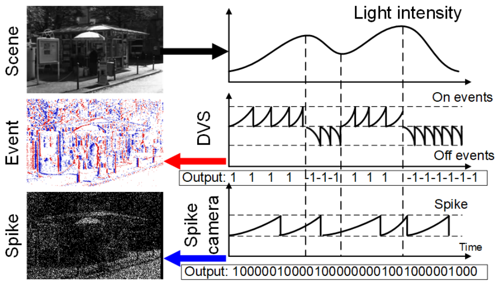
Scene
Event
Spike
Light intensity
On events
Off events
DVS
Spike camera
Spike
Time
Output: 1 1 1 1 -1-1-1 1 1 1 -1-1-1-1-1-1
Output: 10000010000100000000100100000 1000

Figure 2. **The sampling mechanism of DVS and spike camera.**

2.1. Sampling Mechanisms of Human Retina

In human vision system, the retina is an important part that receives the light and perceives the scene. The center of the retina, a.k.a. fovea, is used for scrutinizing highly detailed objects, and the peripheral vision is optimized for perceiving coarser motion information [39]. Research in the last decade has shown that peripheral and foveal vision is not independent in human vision, but is directly connected [37]. Humans use peripheral vision to select regions of interest and foveate them by saccadic eye movements for further scrutiny [41]. In fact, as stated in [37], peripheral and foveal inputs interact and influence each other to better perceive the scene. There is an integration process to combine the foveal and peripheral information, which is beneficial to perception [37]. Recently, researchers in the neuromorphic field developed several bio-inspired vision sensors to simulate the properties of the human retina. In this paper, motivated by the perception mechanism of the retina, we explore how to combine two kinds of bio-inspired cameras (the fovea-like spike camera and the peripheral-like DVS) to reconstruct high quality images (see Figure 1).

2.2. Bio-inspired Neuromorphic Cameras

For spike camera, each pixel independently accumulates luminance intensity which inputs from an analog-to-digital converter (ADC), and generates a spike if the ADC value exceeds the dispatch threshold ϕ [47]:

$$\int_0^T I dt \geq \phi, \tag{1}$$

where I refers to the luminance intensity (usually measured by photocurrent in the circuit). Then the accumulator is reset and all the charges on it are drained. At different pixels, the accumulation speed of the luminance intensity is different. As shown in Figure 2, the greater the light intensity, the more frequent the spikes are emitted. For a spike camera, a pixel continuously measures the light intensity and emits a spike train with 40,000 Hz. At a certain sampling time, the states ("1" or "0") of all pixels form a spike plane.

The Dynamic Visual Sensor (DVS) [9, 1] tracks the light intensity changes at each pixel, and fires asynchronous events whenever the log intensity changes over a dispatch threshold θ (see Figure 2):

$$|\log(I_{t+1}) - \log(I_t)| \geq \theta. \tag{2}$$

Since each pixel individually responds to the light intensity changes, DVS does not have a fixed sampling rate. For a pixel (x, y), if an event occurs at time t, the event is represented as a four-dimensional tuple $e = (t, x, y, p)$ where p denotes the polarity of the event ("+1" for light intensity increase and "-1" for decrease). This representation is called Address Event Representation (AER), and is the standard format used by event-based sensors.

Generally speaking, the event camera has high dynamic sensing ability to moving objects, but it can't record texture. Spike camera has the ability of full texture sampling, but its dynamic range is not as high as that of an event camera. Therefore, this work explores how to effectively combine the two cameras to achieve complementarily.

2.3. Relationship between Spike and Event Data

Although the sampling mechanisms are different, spike and event cameras both record the change of light intensity. Based on the light intensity information hidden in the data, the relationship between the spike and the event data is analyzed to guide the development of our model. Considering that Eq. (1) can be simplified as $It \geq \phi$, for spike camera, the average intensity of the pixel in this period can be estimated by

$$\overline{I} = \frac{\phi}{t_{\text{ISI}}}, \tag{3}$$

where ϕ denotes the dispatch threshold, and t_{ISI} exactly corresponds to the inter-spike interval (ISI).

For the event camera, we first map the event sequence into a continuous-time function which incorporate the statistical description. An event sequence with N spike firing times $\{t_i \in \mathcal{T} | i = 1, 2, ..., N\}$ can be described by a sum of Dirac delta functions

$$e(t) = \sum_i p_i \delta(t - t_i), \tag{4}$$

where p_i refers to the polarity described in Eq. (4). The Dirac delta function has the following property $\delta(t) = \begin{cases} 0, t \neq 0 \\ \infty, t = 0 \end{cases}$, with $\int \delta(t)dt = 1$.

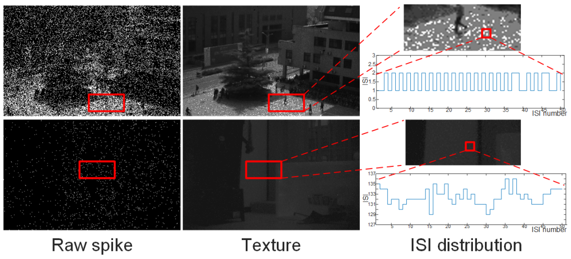

Figure 3. **The noise analysis of spike camera on bright and dark scenes.** Top: the bright scene. Bottom: the dark scene. From the ISI distribution, we can see that the bright scene suffer from the noise type 1 (analyzed in Section 2.4), while the dark scene main influenced by the fixed pattern noise.

Assuming that the positive ("+1") and negative ("-1") events are triggered according to same threshold θ, and θ does not change between t and $t + 1$, according to Eq. (2), we have the follow expression

$$\log(I_{t+1}) - \log(I_t) = \theta \int_t^{t+1} e(s)ds. \tag{5}$$

Considering that the light intensities I_{t+1} and I_t can be represented by the ISI of spike train as described in Eq. (3), the trigger threshold θ is expressed as

$$\theta = \frac{\log(t_{\text{ISI}_1}/t_{\text{ISI}_2})}{\int_t^{t+1} e(s)ds}, \tag{6}$$

where t_{ISI_1} and t_{ISI_2} denote the ISI at t and $t + 1$ in spike train, respectively. Therefore, in ideal conditions, we can obtain the dispatch threshold θ according to Eq. (7). For any time $t_i > t$, the light intensity L_{t_i} can be estimated by

$$L_{t_i} = \exp(\log(I_t) + \theta \int_t^{t_i} e(s)ds). \tag{7}$$

However, there are large temporal noises in both spike and event cameras, which has a great impact on image reconstruction. We will analyze it in Section 2.4. Despite the influence of noise, Eq. (7) can help us to better design the representation of spike data and event data for our network.

2.4. Noise Analysis of Neuromorphic Data

The performance of image reconstruction from the neuromorphic cameras is heavily affected by noise. For event cameras, the image quality is directly affected by the dispatch threshold which changes per pixel (due to manufacturing mismatch), and also due to dynamic effects (incident light, time, etc.) [12, 2]. Moreover, the temporal noise becomes significant in the conditions of low event threshold, high bandwidth, and low light intensity [8].

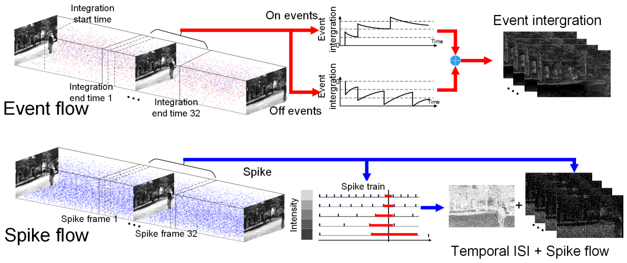

Figure 4. **Spike and event data representations.** For spike flow, we use a temporal ISI map and $N_c - 1$ spike planes as the input of the motion path. Meanwhile, the event flow is transformed into event integration according to the timestamp of the spike plane as the motion path input (see Eq. (8)). The temporal information can be effective explored by our spike and event representation.

For spike camera, if there is no noise exist, the texture image can be accurately and quickly reconstructed by using the TFI method [47]. However, the existence of temporal noise also has a great impact on image reconstruction, mainly including the following two types: 1) For the constant light intensity, the ISI may be inconsistent due to the readout and reset time delay in the circuit. For example, the accurate ISI of the spike train at a certain time should be 2, but the readout ISI may fluctuate between 1 and 2 due to noise in the temporal domain. This is especially harmful to image reconstruction under high light intensity (see Figure 3). 2) Under the low light intensity, the influence of 1) becomes smaller due to the long interval of the emission. The main noise is the fixed pattern noise (e.g., dark current noise). At this time, the noise will actively emit the spike due to the noise, which leads to the mismatch between the ISI and the real light intensity, and also limits the maximum ISI range. An analysis of fixed pattern noise can be found in our supplementary material.

3. Methodology

In this section, we first design neuromorphic data representations for spike and event data (Section 3.1). The principle is to extract more useful temporal information. The image reconstruction network is introduced in Section 3.2 and the feature fusion module for spike and event data is detailed in Section 3.3. Finally, a multi-scale neuromorphic data simulation method is introduced in Section 3.4.

3.1. Adaptive Neuromorphic Data Representation

Eq. (7) represents the relationship of the two types of neuromorphic data. The light intensity at the subsequent time can be estimated by the ISI of the initial time and the integration of the event data. According to the sampling principle of the spike camera, ISI represents the integration time for a pixel to reach one emission, which contains temporal information in this period. On the other hand, since the spike camera outputs in the form of spike planes, the spikes can naturally be inputted to the network, where each spike plane is used as a channel. For the representation of spike data, as shown in Figure 4, we propose to use one channel of ISI and N-1 channels of spike planes as the input of texture path, with a total of N channels.

The event data is discrete in spatio-temporal domain because of the asynchronous nature. Eq. (7) can guide us to design the representation of event data. Based on the timestamp of spike planes, the asynchronous events are needed to be transformed into effective two-dimensional features suitable for network input. In our work, inspired by the integrate-and-fire (IF) model [21], we design an event integration model to extract temporal information from the event flow and transform it into 2-D feature. The membrane potential $V(t)$ is defined as

$$V(t) = \int_0^t (e(s) + \exp(\frac{-(t-s)}{\tau})e(s))ds, \tag{8}$$

where τ is the time constant to control the decay rate. As shown in Figure 4, we calculate the accumulation of positive and negative events respectively, and then add them together to get the final feature map as the motion path input.

3.2. Network Architecture

Our neural network is a fully convolutional network that accommodates both event flow and spike flow. Figure 5 clarifies the architecture of the network. The NeuSpike-Net has two encoders and a decoder, so it is a variant of the U-shaped model [32]. The event flow and spike flow are first transformed as the size of $N_c \times W \times H$ and followed by N_e encoder layers, N_r residual blocks, N_d decoder layers, and a final image prediction layer. The number of channels is doubled after each encoder layer.

In the encoder, there are two input paths: motion path and texture path. The two paths have the same encoder structure, we use skip connections between symmetric encoder and decoder layers. The motion path is designed to extract more useful features from the event data because the event flow responds to moving objects. The texture path captures the texture information hidden in the spike flow. The motion and texture features are fused by a feature fusion module in other encoder layers (see Section 3.3). The prediction layer performs a depthwise convolution followed by a sigmoid layer to produce an image prediction. We use $N_c = 32$, $N_e = N_d = 3$ and $N_r = 2$ in our model.

In our network, the loss function is defined as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{l_2} + \lambda \mathcal{L}_{PL}, \tag{9}$$

where $\mathcal{L}_{\ell_2}$ is the ℓ_2 loss $\frac{1}{T}\sum_{i=1}^{T} \|\mathcal{I}_i^* - \mathcal{I}_i^g\|^2$, $\mathcal{I}_i^*$ and $\mathcal{I}_i^g$ denote the generated texture image and the ground truth im-

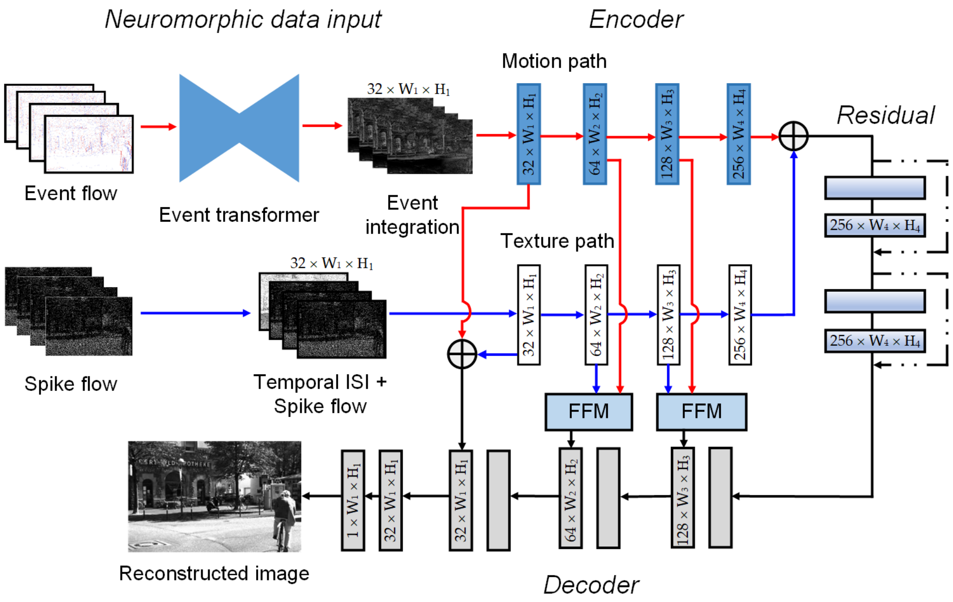

Figure 5. **NeuSpike-Net architecture.** Our network contains two encoder paths correspond to event and spike flow respectively. The two paths have the same encoder structure, we use skip connections between symmetric encoder and decoder layers. Besides, FFM are applied to fuse the features of spike and event flow in each encoder layer. The prediction layer performs a depthwise convolution followed by a sigmoid layer to produce an image prediction.

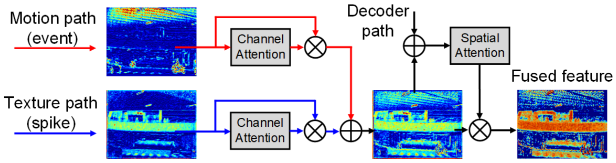

Figure 6. **The feature fusion module.** For better viewing, the feature maps are obtained by adding up all channels. The feature map is based on the driving scene in Fig.10. The motion path highlights the motion and HDR parts while the texture path pays more attention to the texture information of the scene.

age, respectively. $\mathcal{L}_{PL}$ is the perceptual loss [18]

$$\mathcal{L}_{PL} = \frac{1}{W_{i,j}H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} (\phi_{i,j}(\mathcal{I}^*)_{x,y} - \phi_{i,j}(\mathcal{I}_g)_{x,y})^2, \quad (10)$$

where $\phi_{i,j}$ is the feature map obtained by the j-th convolution (after activation) before the i-th maxpooling layer $W_{i,j}$ and $H_{i,j}$ are the dimensions of the feature maps.

3.3. Feature Fusion Module

Humans use peripheral vision to select regions of interest and foveate them for further scrutiny [41]. In our model, the motion path (peripheral) captures more high dynamic range and motion information, and the texture path (fovea) preserves more detailed texture information. Inspired by the human vision, we propose an attention-based feature fusion module (FFM) to extract more useful features of two encoder paths. Figure 6 shows the visualized intermediate feature maps of the FFM. Since each channel of the input spike and event data represents the temporal information of the neuromorphic data, the channels of the feature contain the temporal information to some extent. Thus, the channel attention is applied to texture and motion features to highlight more useful temporal information of the two paths. Given the intermediate feature maps of two paths $\mathbf{F_m} \in \mathbb{R}^{C \times H \times W}$ and $\mathbf{F_t} \in \mathbb{R}^{C \times H \times W}$ as input, FFM first infers a 1D channel attention map $\mathbf{M_c} \in \mathbb{R}^{C \times 1 \times 1}$

$$\mathbf{F_m^{'}} = \mathbf{M_c(F_m)} \otimes \mathbf{F_m} \text{ and } \mathbf{F_t^{'}} = \mathbf{M_c(F_t)} \otimes \mathbf{F_t}, \quad (11)$$

where $\otimes$ denotes element-wise multiplication, $\mathbf{M_c}(\cdot)$ denotes the channal attention module[42]. By introducing spatial attention, the final fused feature is obtained by

$$\mathbf{F_{fuse}} = \mathbf{M_s}((\mathbf{F_m^{'}} \oplus \mathbf{F_t^{'}}) \oplus \mathbf{F_d}) \otimes (\mathbf{F_m^{'}} \oplus \mathbf{F_t^{'}}), \quad (12)$$

where $\oplus$ denotes element-wise sum, $\mathbf{M_s}(\cdot)$ denotes the spatial attention module [26], and $\mathbf{F_d}$ denotes the feature map of decoder.

3.4. Multi-scale Training Neuromorphic Data

Our network requires training data including event flow, spike flow and corresponding ground truth images. However, the ground truth images are usually difficult to obtain. Thus, we propose to train the network on the simulated multi-scale neuromorphic data[1].

High Frame Rate Video Preparation Inspired by the DVS simulator V2E [8], the videos are first converted into luma

[1] Please see our supplementary material for more details about the simulated data.

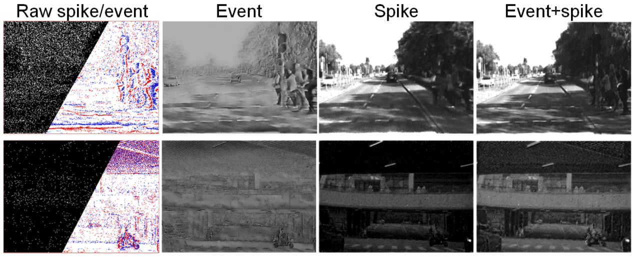

Figure 7. **Effect of different input modalities.** Column 1: the input spike and event data. Column 2-4: the reconstructed images from event data, spike data, and event+spike data, respectively.

frames, then we adopt the Super-SLoMo video interpolation network [17] to increase the frame rate of the video. We use the videos in Object Tracking Evaluation of KITTI dataset[14] to generate the simulated data. The average upsampling ratio is 750. The original 30 FPS videos are upsampled to about 22,500 FPS, which is similar to the sampling frequency of a spike camera. We use the images in original videos as the groundtruth to ensure they are clear. The upsampled video is used to generate spike and event data according to the timestamps of the groundtruth images.

Table 1. Effect of different event/spike representations.

Representation	PSNR	SSIM
ES + SR w/o TISI	26.31	0.8134
Voxel + SR w/o TISI	26.74	0.8159
ML + SR w/o TISI	27.86	0.8204
IF + SR w/o TISI	27.65	0.8224
ES + SR	29.35	0.9129
Voxel + SR	29.44	0.9021
ML + SR	30.16	0.9110
IF + SR (Ours)	**30.31**	**0.9234**

Table 2. Effect of different input modalities.

Modality	PSNR	SSIM
Event	12.85	0.5198
Spike	26.84	0.8389
Event + spike (Ours)	**27.40**	**0.8510**

Multi-scale Spike and Event Data The way to generate basic spike data is intuitive: first, we set up an accumulator for each pixel. Each input image contributes to the accumulator according to the pixel gray value multiplied by a light intensity scale. If the accumulated value exceeds the emission threshold, spike "1" is fired, otherwise "0" is output.

To better simulate real-world scenarios, we generalize the simulated data to a multi-scale form, including different noises, light conditions, and motion speeds. Different light conditions are simulated by adjusting the light intensity scale to control the dense of generated spikes. Meanwhile, different motion speeds are simulated by adjusting the integral times of each frame. The corresponding event data is generated by V2E using the same upsampled frames to simulate different motions. Also, the contrast threshold is adjusted to simulate different light intensities.

Temporal Noise Simulation We consider simulating the noise distribution (see Section 2.4) in real spike data. To simulate the noise under high light intensity, we add a random matrix to the initial accumulator. For simulating the fixed pattern noise in low light conditions, we first generate a matrix $N \in \mathbb{R}^{m\times n}$ that follows Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$, where $m \times n$ denotes the spatial resolution of the data. Then the length of the spike interval is constrained by

$$T_{x,y} = \begin{cases} Y_{x,y} & \text{if } \; Y_{x,y} \leq N_{x,y} \\ N_{x,y} & \text{if } \; Y_{x,y} > N_{x,y} \end{cases}, \tag{13}$$

where $Y_{x,y}$ is the original simulated spike interval at pixel (x, y), and $T_{(x,y)}$ denotes the constrained spike interval. According to statistics on real spike data, the mean μ and standard deviation σ are set to 180 and 50, respectively.

4. Experiment

4.1. Training Details

Our network[2] is trained on an NVIDIA 2080 Ti GPU. We adopt the batch size of 8 and Adam optimizer [20] in the training process. The network is trained for 60 epochs, with a learning rate of 10^{-4}. The weight λ of perceptual loss is set as 0.01. During training, input images are randomly flipped horizontally and vertically (with 0.5 probability) and cropped to 400×256. As stated in Section 3.4, the simulated data (spike data, event data, and groundtruth images) is with 5 different light scales and 5 different motion speeds. We use all the origin frames in the video "0000" to generate training data, and the random frames in videos "0000"-"0019" to generate test data. In total, we use 1,183 data to train the network and 745 data to test the performance.

4.2. Effect of Different Neuromorphic Inputs

Different Event/Spike Representations To evaluate the effect of different event and spike representations, we conduct experiments on the simulated data. For the event representation, we test event stacking (ES) [40], voxel grid (Voxel) [46], Matrix-LSTM (ML) [5] and the proposed representation (Eq. (8)). For the spike data, we compare the effect of two representations: the proposed spike representation without temporal ISI (SR w/o TISI) and the spike flow with temporal ISI (SR). ML is a learn-based representation for event data, and voxel grid is widely used in DVS reconstruction. As the quantitative results shown in Table 1, IF + SR performs better than other representations because it is designed as the sampling mechanism described in Eq. (7) which is more suitable to our framework. By introducing the proposed representation, the temporal components of event and spike data are effectively explored by our framework.

[2] Project page: https://sites.google.com/view/retina-recon/

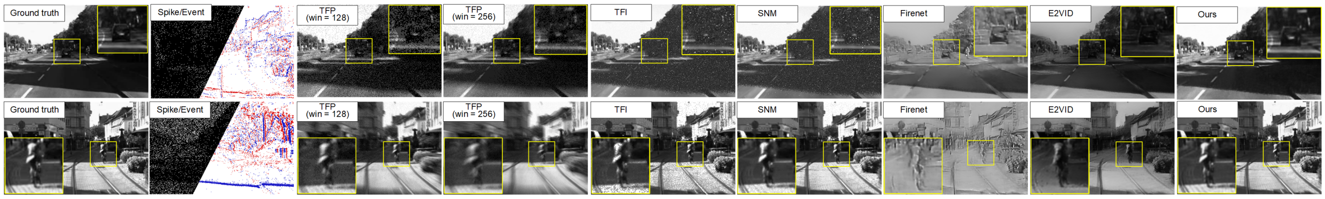

Figure 8. **Quantitative results on the simulated dataset.** Top: the results under simulated low light scene. Bottom: the results under simulated high speed scene. TFP (win=n) means the reconstruction window size is set as n spike planes. Other methods use their default configurations. The results show that our method can reconstruct the car and person clearly while eliminating the noise completely.

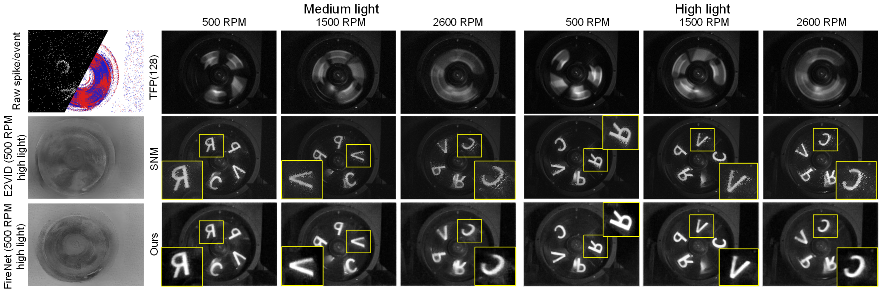

Figure 9. **Quantitative results of ultra high speed scenes in real-world dataset.** The scene depicts a high speed fan with speeds from 500 to 2600 RPM, we compare our method with four methods: TFP (win=128), SNM, FireNet, and E2VID. The event-based method is difficult to reconstruct images in this scene because the events are too dense. For spike-based methods, there are artifacts in the results of SNM and TFP.

Table 3. Quantitative results on simulated data

Scene	**Method**	***TFP (128)**	***TFP (256)**	*TFI	*SNM	†FireNet	†E2VID	*†**Ours**
Normal	PSNR	28.08	28.81	27.06	29.60	10.91	12.82	**30.31**
	SSIM	0.7038	0.7887	0.7750	0.8416	0.4640	0.5181	**0.9234**
Complex	PSNR	22.70	22.58	23.38	26.44	11.78	12.85	**27.40**
	SSIM	0.5648	0.6104	0.7063	0.7722	0.5247	0.5198	**0.8510**

* Spike-based methods. † Event-based methods.

Different Input Modalities The valid the effect of two types of neuromorphic data, we conduct an ablation experiment on the complex scenes in simulated data. We design a network using the texture path of our framework to reconstruct images that only uses spike data. Meanwhile, the event-based approach E2VID is used to test the performance of only event input. The results are shown in Table 2 and Figure 7. Moreover, the results on real dataset also prove the effectiveness of combining spike and event data (see Figure 10 and Table 5). The event-based methods are difficult to estimate the texture, while the spike-based methods are rely on light intensity thus hard to reconstruct the HDR details. By combining spike and event data, the texture and HDR information can be reconstructed effectively.

4.3. Effect of Different Network Structures

Network Structure. We compare different network architectures for finding the best hyperparameters. Table 4 reports the result of replacing the FFM (our default architecture) by the element-wise sum. Moreover, rows 3-6 demonstrate that our model performs best with 3 encoders and 2 residual blocks.

Loss functions. We use the $\mathcal{L}_{l_2} + \mathcal{L}_{PL}$ loss by default, but have evaluated many alternative loss functions. As shown in Table 4 (rows 7-9), $\mathcal{L}_{l_1} + \mathcal{L}_{PL}$ has larger SSIM than our default model, but the PSNR is lower. Other loss functions do not show advantages in the results.

4.4. Evaluation on the Simulated Dataset

Experiment Settings To evaluate the proposed network, we conduct experiments on both simulated and real-world datasets. We compare our method with other eight state-of-the-art methods, including three spike-based reconstruction methods (TFP [47], TFI [47] and SNM [49]) and 5 event-based reconstruction methods (MF [25], HF [33], CF [33], FireNet [34], E2VID [31]). Among them, FireNet and E2VID are the learning-based methods, and CF uses both event and frame to reconstruct image.

Results Figure 8 shows the qualitative results of two typical scenes including low light and high light scenes. The results show that the proposed model is effective to handle these scenes. Table 3 shows the quantitative evaluation of our method on normal scenes and complex scenes (e.g., high speed and low light), respectively. The results show that our method is better than other methods, especially SSIM. In summary, the qualitative and quantitative results demonstrate that our method can reconstruct high quality images by fusing spike and event data.

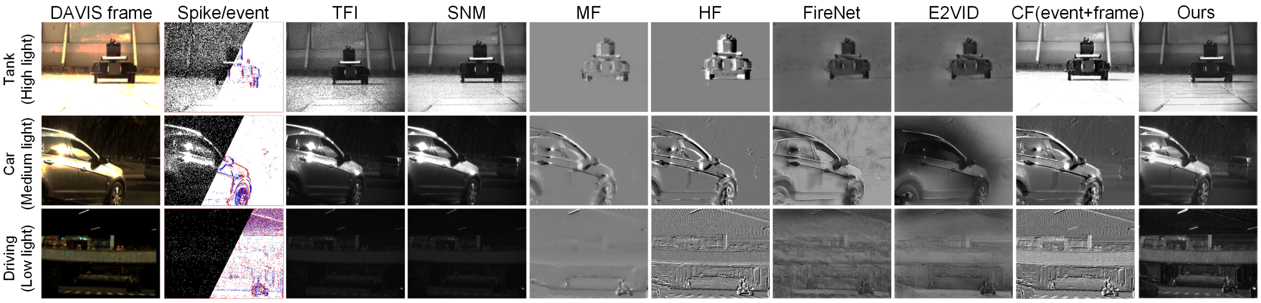

Figure 10. **Quantitative results of the outdoor scenes in real data.** The scenes are with different light intensities that can be estimated by the firing density of the raw spike data. Top: a tank under high light intensity. Middle: a car under medium light intensity. Bottom: a driving scene under low light intensity. Our method performs better by combining event and spike data.

Table 4. Effect of different network structures and losses.

Condition	PSNR	SSIM
1. Our default model	30.31	0.9234
2. FFM → Element-wise sum	29.74	0.9159
3. Encoders: 3 → 2	30.26	0.9104
4. Encoders: 3 → 4	29.59	0.9229
5. Residual blocks: 2 → 1	29.79	0.9243
6. Residual blocks: 2 → 3	29.80	0.9076
7. Loss: $\mathcal{L}_{l_2} + \mathcal{L}_{PL} \rightarrow \mathcal{L}_{l_1}$	29.68	0.9104
8. Loss: $\mathcal{L}_{l_2} + \mathcal{L}_{PL} \rightarrow \mathcal{L}_{l_2}$	29.57	0.9084
9. Loss: $\mathcal{L}_{l_2} + \mathcal{L}_{PL} \rightarrow \mathcal{L}_{l_1} + \mathcal{L}_{PL}$	28.79	0.9297

Table 5. Quantitative results on real world data.

Method	MSE	SSIM	2D-entropy	
	Outdoor	Outdoor	High speed	Outdoor
TFI (spike)	0.0842	0.3396	8.8037	7.4797
SNM (spike)	0.0785	0.4362	8.8594	8.5884
E2VID (event)	0.1014	0.4167	8.4820	8.9180
MF (event)	0.1281	0.4062	8.8895	5.5215
HF (event)	0.1347	0.3845	8.9459	7.5335
CF (event + frame)	**0.0526**	0.4787	9.7556	**10.5473**
Ours (spike)	0.0810	0.4682	9.4246	10.1517
Ours (spike+event)	0.0741	**0.5046**	**9.8747**	10.4897

4.5. Evaluation on the Real World Dataset

Experiment Settings Inspired by [16], we build a hybrid camera system (see Figure 11) consisting of a spike camera (Vidar), an event camera (DAVIS 346) and a beam splitter. Two cameras can record the same scene through the beam splitter. The detail of this system can be found in our supplementary material. We construct a real dataset including 15 sequences with different light conditions, which consists of 5 outdoor scenes and 10 ultra high speed fan scenes (the fan with speeds from 500 RPM to 2600 RPM).

Results Figure 9 shows the results of ultra high speed scenes. In this scenario, due to the event data are too dense, the event-based methods are difficult to reconstruct satisfactory results. With the help of spike data, our method can reconstruct clear letters under different light intensities and motion speeds Figure 10 shows the results on outdoor scenes with different light conditions. Compared with SNM (spike-based) and CF (event+frame), our method can reconstruct HDR scenes clearly with less noise, which shows the advantages of the combination of event and spike. In "Tank", the noise analyzed in Section 2.4 can be eliminated completely. For the medium and low light scenes "Car" and "Driving", our method can take advantage of the combination of two types of data to reconstruct more HDR details. The quantitative evaluation are shown in Table 5. We use APS images as the groundtruth image to evaluate MSE and SSIM. Moreover, since APS cannot record high-speed scenes, we introduce a non-reference metric 2D-entropy to evaluate all real-world data. Note that CF uses the APS frames as the initial state, thus it performs better MSE and 2D-entropy on outdoor scenes. Our method performs well in all quantitative results (the introduction of events has greatly improved SSIM). For more experimental results, please refer to our supplementary materials.

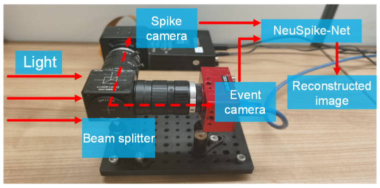

Figure 11. **The hybrid neuromorphic cameras system.**

5. Conclusion

In this work, we propose to combine the high dynamic range of event data and the full texture sampling of spike data to reconstruct high quality visual images, especially in high speed and different light intensities scenes. To this end, the NeuSpike-Net is proposed to handle these two types of neuromorphic data. An effective representation of spike and event data is proposed to extract temporal information. Furthermore, a feature fusion module is designed to effectively fuse the two types of neuromorphic data. Our network is trained by a multi-scale simulated neuromorphic dataset. To test the performance, we also build a hybrid neuromorphic cameras system to record the real-world dataset. Extensive evaluations on simulated and real datasets show that the proposed approach achieves superior performance than various existing spike-based and event-based methods.

Acknowledgments. This work was supported in part by the National Natural Science Foundation of China under Contract 62027804, Contract 61825101 and Contract 62088102.

References

[1] Patrick Lichtsteiner an 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. 1, 3

[2] R Baldwin, Mohammed Almatrafi, Vijayan Asari, and Keigo Hirakawa. Event probability mask (epm) and event denoising convolutional neural network (edncnn) for neuromorphic cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1701–1710, 2020. 3

[3] Juan Antonio Lenero Bardallo, Jose-Maria Guerrero-Rodriguez, Ricardo Carmona-Galan, and Angel Rodriguez-Vazquez. On the analysis and detection of flames with an asynchronous spiking image sensor. *IEEE Sensors Journal*, 18(16):6588–6595, Aug 2018. 1

[4] Christian Brandli, Raphael Berner, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck. A 240 × 180 130 *db* 3 μs latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits*, 49(10):2333–2341, 2014. 1

[5] Marco Cannici, Marco Ciccone, Andrea Romanoni, and Matteo Matteucci. A differentiable recurrent surface for asynchronous event-based data. In *European Conference on Computer Vision (ECCV)*, pages 136–152. Springer, 2020. 6

[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. 1

[7] Eugenio Culurciello, Ralph Etienne-Cummings, and K-wabena A Boahen. A biomorphic digital image sensor. *IEEE Journal of Solid-State Circuits*, 38(2):281–294, 2003. 1

[8] Tobi Delbruck, Yuhuang Hu, and Zhe He. V2e: From video frames to realistic dvs event camera streams, 2020. 3, 5

[9] Tobi Delbrück, Bernabe Linares-Barranco, Eugenio Culurciello, and Christoph Posch. Activity-driven, event-based vision sensors. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 2426–2429, 2010. 1, 3

[10] Siwei Dong, Tiejun Huang, and Yonghong Tian. Spike camera and its coding methods. In *Data Compression Conference (DCC)*, pages 437–437, 2017. 1

[11] 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)*, pages 568–568. IEEE, 2019. 1

[12] Guillermo Gallego, Tobi Delbruck, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew Davison, Joerg Conradt, Kostas Daniilidis, and Davide Scaramuzza. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2020. 3

[13] Daniel Gehrig, Henri Rebecq, Guillermo Gallego, and Davide Scaramuzza. Asynchronous, photometric feature tracking using events and frames. In *European Conference on Computer Vision (ECCV)*, pages 750–765, 2018. 1

[14] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 6

[15] Menghan Guo, Jing Huang, and Shoushun Chen. Live demonstration: A 768× 640 pixels 200meps dynamic vision sensor. In *International Symposium on Circuits and Systems (ISCAS)*, pages 1–1. IEEE, 2017. 1

[16] Jin Han, Chu Zhou, Peiqi Duan, Yehui Tang, Chang Xu, Chao Xu, Tiejun Huang, and Boxin Shi. Neuromorphic camera guided high dynamic range imaging. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1, 8

[17] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9000–9008, 2018. 6

[18] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision (ECCV)*, pages 694–711. Springer, 2016. 5

[19] Zaven Kalayjian and Andreas G. Andreou. Asynchronous communication of 2d motion information using winner-takes-all arbitration. *Analog Integrated Circuits and Signal Processing*, 13(1):103–109, 1997. 1

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[21] Christof Koch and Idan Segev. *Methods in neuronal modeling: from ions to networks*. MIT press, 1998. 4

[22] Juan Antonio Leero-Bardallo, D. H. Bryn, and Philipp H-fliger. Bio-inspired asynchronous pixel event tricolor vision sensor. *IEEE Transactions on Biomedical Circuits and Systems*, 8(3):345–357, June 2014. 1

[23] Martin Litzenberger, Christoph Posch, D Bauer, Ahmed N-abil Belbachir, P Schon, B Kohn, and H Garn. Embedded vision system for real-time object tracking using an asynchronous transient vision sensor. In *Digital Signal Processing Workshop-signal Processing Education Workshop*, pages 173–178, 2006. 1

[24] Ana I Maqueda, Antonio Loquercio, Guillermo Gallego, Narciso García, and Davide Scaramuzza. Event-based vision meets deep learning on steering prediction for self-driving cars. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5419–5427, 2018. 1

[25] Gottfried Munda, Christian Reinbacher, and Thomas Pock. Real-time intensity-image reconstruction for event cameras using manifold regularisation. *International Journal of Computer Vision*, 126(12):1381–1393, 2018. 1, 7

[26] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, et al. Attention u-net: Learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999*, 2018. 5

[27] Liyuan Pan, Cedric Scheerlinck, Xin Yu, Richard Hartley, Miaomiao Liu, and Yuchao Dai. Bringing a blurry frame alive at high frame-rate with an event camera. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6820–6829, 2019. 1

[28] Stefano Pini, Guido Borghi, and Roberto Vezzani. Learn to see by events: Color frame synthesis from event and rgb cameras. In *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2019. 1

[29] Stefano Pini, Guido Borghi, Roberto Vezzani, and Rita Cucchiara. Video synthesis from intensity and event frames. In *International Conference on Image Analysis and Processing*, pages 313–323. Springer, 2019. 1

[30] Christoph Posch, Daniel Matolin, and Rainer Wohlgenannt. An asynchronous time-based image sensor. *IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 2130–2133, 2008. 1

[31] 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. 1, 7

[32] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 4

[33] Cedric Scheerlinck, Nick Barnes, and Robert Mahony. Continuous-time intensity estimation using event cameras. In *IEEE Asian Conference on Computer Vision (ACCV)*, pages 308–324. Springer, 2018. 1, 7

[34] Cedric Scheerlinck, Henri Rebecq, Daniel Gehrig, Nick Barnes, Robert Mahony, and Davide Scaramuzza. Fast image reconstruction with an event camera. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 156–163, 2020. 1, 7

[35] Chen Shoushun and Amine Bermak. Arbitrated time-to-first spike cmos image sensor with on-chip histogram equalization. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 15(3):346–357, 2007. 1

[36] Raunak Sinha, Mrinalini Hoon, Jacob Baudin, Haruhisa Okawa, Rachel OL Wong, and Fred Rieke. Cellular and circuit mechanisms shaping the perceptual properties of the primate fovea. *Cell*, 168(3):413–426, 2017. 1

[37] Emma EM Stewart, Matteo Valsecchi, and Alexander C Schütz. A review of interactions between peripheral and foveal vision. *Journal of Vision*, 20(12):2–2, 2020. 1, 2

[38] Timo Stoffregen, Cedric Scheerlinck, Davide Scaramuzza, Tom Drummond, Nick Barnes, Lindsay Kleeman, and Robert Mahony. Reducing the sim-to-real gap for event cameras. In *European Conference on Computer Vision (ECCV)*, pages 534–549. Springer, 2020. 1

[39] Matteo Toscani, Karl R Gegenfurtner, and Matteo Valsecchi. Foveal to peripheral extrapolation of brightness within objects. *Journal of vision*, 17(9):14–14, 2017. 2

[40] Lin Wang, Yo-Sung Ho, Kuk-Jin Yoon, et al. Event-based high dynamic range image and very high frame rate video generation using conditional generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10081–10090, 2019. 6

[41] Christian Wolf and Alexander C Schütz. Trans-saccadic integration of peripheral and foveal feature information is close to optimal. *Journal of vision*, 15(16):1–1, 2015. 2, 5

[42] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *European conference on computer vision (ECCV)*, pages 3–19, 2018. 5

[43] Woodward Yang. A wide-dynamic-range, low-power photosensor array. In *Proceedings of IEEE International Solid-State Circuits Conference ISSCC*, pages 230–231, 1994. 1

[44] Song Zhang, Yu Zhang, Zhe Jiang, Dongqing Zou, Jimmy Ren, and Bin Zhou. Learning to see in the dark with events. In *European Conference on Computer Vision (ECCV)*, pages 666–682. Springer, 2020. 1

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

[46] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised event-based optical flow using motion compensation. In *European Conference on Computer Vision (ECCV)*, pages 711–714. Springer, 2018. 6

[47] Lin Zhu, Siwei Dong, Tiejun Huang, and Yonghong Tian. A retina-inspired sampling method for visual texture reconstruction. In *International Conference on Multimedia and Expo (ICME)*, pages 1432–1437, 2019. 1, 2, 4, 7

[48] 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. 1

[49] Lin Zhu, Siwei Dong, Jianing Li, Tiejun Huang, and Yonghong Tian. Retina-like visual image reconstruction via a spiking neural model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1438–1446, 2020. 1, 7