Supplementary materials for "NeuSpike-Net: High Speed Video Reconstruction via Bio-inspired Neuromorphic Cameras"

1. Network Architecture Details

Our reconstruction network including two encoder paths (motion path and texture path) and one decoder path, which is based on an U-Net like architecture [21]. The event flow and spike flow are first transformed as the size of $32 \times 400 \times 256$ and followed by 3 encoder layers, 2 residual blocks, 3 decoder layers, and a final image prediction layer ($N_E = N_D = 3, N_R = 2$).

For a more intuitive understanding, we use the following operation sequences to represent the network:

$$\mathbf{x}_m^0 = \mathcal{T}_{event}(\mathbf{f}_{event}) \tag{1}$$
$$\mathbf{x}_t^0 = \mathcal{T}_{spike}(\mathbf{f}_{spike}) \tag{2}$$
$$\mathbf{x}_m^{i+1} = \mathcal{E}_m^{i+1}(\mathbf{x}_m^i) \tag{3}$$
$$\mathbf{x}_t^{i+1} = \mathcal{E}_t^{i+1}(\mathbf{x}_t^i) \tag{4}$$
$$\mathbf{r}^{j+1} = \mathcal{R}^j(\mathbf{r}^j), \text{ where } \mathbf{r}^0 = \mathbf{x}_t^{N_E} \oplus \mathbf{x}_m^{N_E} \tag{5}$$
$$\mathbf{d}^{l+1} = \mathcal{D}^{l+1}(\mathbf{d}^l) \tag{6}$$
$$\hat{\mathcal{I}} = \sigma(\mathcal{P}(\mathbf{d}^{N_D})) \tag{7}$$

where $\mathcal{T}_{event}$ and $\mathcal{T}_{spike}$ are the transformers proposed in Section 4.1. $\mathcal{E}_m^i$ and $\mathcal{E}_t^i$ are the encoders of the motion path and texture path, respectively. $\mathcal{R}^j$ is the j-th residual block. $\mathcal{D}^l$ and $\mathcal{P}$ denote the l-th decoder layer and prediction layer, respectively. σ denotes the tanh function, and $\oplus$ the element-wise sum. And we have $0 \le i < N_E$, $0 \le j < N_R$ and $0 \le l < N_D$.

Specifically, $\mathbf{d}^l$ is defined as

$$\mathbf{d}^l = \begin{cases} \mathbf{r}^{N_R}, & \text{if } l = 0 \\ \mathcal{F}(\mathbf{x}_m^{N_R-l}, \mathbf{x}_t^{N_R-l}) \odot \mathbf{d}^{l-1}, & \text{if } 1 \le l < N_R \\ (\mathbf{x}_t^0 \oplus (\mathbf{w} \cdot \mathbf{x}_m^0)) \odot \mathbf{d}^{N_R-1}, & \text{if } l = N_R \end{cases} \tag{8}$$

where $\odot$ denotes the channel-wise concatenation, $\mathcal{F}$ refers to the proposed feature fusion module.

The encoders are strided convolutional layers (stride of 2), with a kernel size of 3. The number of output channel of the first encoder layer is 64, and is doubled for every subsequent encoder layer, i.e. the sequence of output channels is (64, 128, 256). Meanwhile, the size of the output feature is divided by 2 for every encoder layer, i.e. the input size is 400×256, the sizes of the output of each encoder are 200×128, 100×64, and 50×32, respectively.

As shown in Eq. (8), in the first and second encoder layers, the features of motion and texture paths are first fused by a feature fusion module, then symmetric skip connections are used. Meanwhile, the transformed event flow and spike flow are fused by element-wise sum in input layer and skip connect to the last decoder layer. The feature of motion path is multiplied by a weight before the element-wise sum, where the weight is set to 1 on simulated data and the max value of event integration on real data, respectively. The skip connections are based on concatenation. Both residual blocks have a kernel size of 3. Instance normalization is used within the encoders, decoders, and residual blocks (applied before the ReLU activation). In the last image prediction layer, a tanh activation is used instead of ReLU. The decoders are transposed convolution layers, with a kernel size of 3. The number of output channels of the decoders starts at 128, and is divided by two for every subsequent decoder, the channel number of the last decoder is 32. The proposed network can also work with only texture paths and decoders, which can be driven by spike data.

2. Related Works

To simulate some of the properties of the human retina, researchers in the neuromorphic field are committed to developing new bio-inspired vision sensors and the corresponding image reconstruction methods.

Event Camera. To simulate biological vision, one of the most famous artificial silicon retinas is the dynamic vision sensor (DVS) [7, 1]. It is capable of high speed detection and tracking [26, 27, 28]. However, as it only cares about the relative change of luminance intensity, it is very difficult to reconstruct the texture. To solve this problem, some event-based sensors were developed subsequently, by combining DVS and conventional image sensor (DAVIS) [3], or adding an extra photo-measurement circuit such as ATIS [19] and CeleX [10], but there exists a motion mismatch due to the difference of the sampling time resolution between two kinds of heterogeneous circuits.

Spike Camera. To explore the different sampling mechanism, there are a number of spiking image sensors fol-

lowing the basis of the integrate-and-fire neuron model [29, 13, 5, 24]. The in-pixel light measurement circuit in ATIS [19] is also a kind of spiking image sensors, but it is driven by DVS circuit. Additionally, some variants of spiking image sensors such as asynchronous pixel event tricolor vision sensor [14] and near infrared spiking image sensor [2] were also proposed in recent years. Recently, Dong et al. [8, 32] 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, also known as the Vidar, with a sampling rate of 20,000 Hz.

Image reconstruction from neuromorphic cameras. Many algorithms were designed to reconstruct texture images using DVS [22, 15]. By transforming the event data into the voxel grid, the image can be reconstructed by the convolutional neural networks [20, 23, 25]. Moreover, more event-based reconstruction tasks such as reconstruction in dark scenes [30] and event-based super resolution reconstruction [4] were proposed in recent years. There are also some algorithms combine APS and DVS to reconstruct texture images [17, 18, 16], which can obtain better texture information than only using the DVS signal.

As for the spike camera, the spike firing frequency or interspike interval reflects the luminance variation which can be used to compute the intensity value [32]. Recently, a fovea-like texture reconstruction framework was proposed to reconstruct images [33]. Further more, some methods based on the spike camera were developed for tone mapping [31] and motion deblurring [11]. In this work, to solve the problem of combination of spike and event data and improve the reconstruction quality, we propose a novel learning-based model to reconstruct high quality texture images in complex scenes.

3. Details of Simulated Spike and Event Dataset

3.1. Noise Analysis

In Section 3.3 of our paper, two typical types of noise have been analyzed. Here we design an experiment to estimate the fixed pattern noise. In the experiment, we use a spike camera to record a completely dark scene (see Figure 1(a)). To make sure no light enters the camera, we cover the spike camera with a lens cap. The spike data recorded by the spike camera is shown in Figure 1(b). Obviously, due to the fixed pattern noise, the camera emits spikes even no light enters. In order to summarize the rule of spike emission at this time, we use 3.6 seconds of spike data to plot the distribution of ISI (inter-spike interval) of all pixels. As shown in Figure 1(c), the ISI distribution can be fitted by a Gaussian distribution with the mean value of 180 and variance of 50. This can guide us to generate more realistic

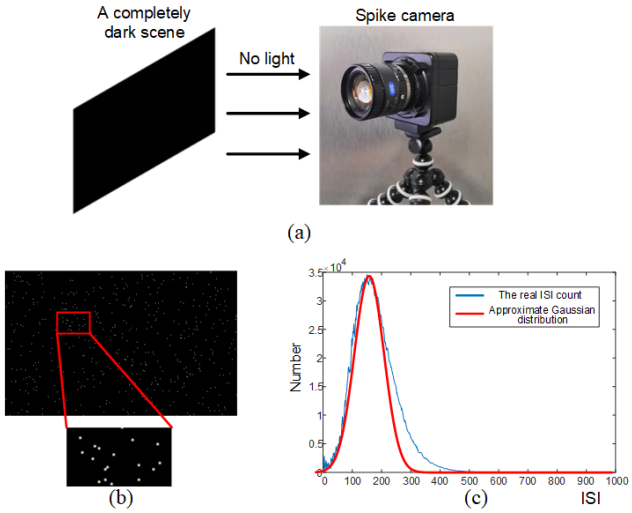

Figure 1. The distribution of fixed pattern noise. (a) The experiment is conducted in a completely dark scene. (b) A spike plane in 1/20,000 second. Spikes are still emitted in the absence of light. (c) The distribution of ISI in this scene, which can be approximately fitted by a Gaussian distribution.

simulation data. To better simulate the spike data, we use Eq. (13) in our paper and the noise matrix N in Algorithm 1 to model the multiple kinds of noise.

3.2. Spike Data Simulator

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. One feasible solution is to train the network using the simulated spike and event data. In our work, we use the videos in Object Tracking Evaluation category of KITTI dataset[9] to generate the simulated data. The process of generating simulated data is shown in Figure 2.

First, the videos are converted into luma frames, then we adopt the Super-SLoMo video interpolation network [12] to increase the frame rate of the video. In our dataset, 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 (Vidar is with 20,000 Hz sampling rate).

Then we generate the event and spike data from the upsampled videos, respectively. The groundtruth image is obtained from the origin videos to ensure they are clear. The event data is generated by the recent DVS simulator V2E [6], while the spike data is generated according to the sampling mechanism of spike camera.

The details of the generator are summarized in Algorithm 1. The input parameters include the motion scales S, the light intensity scales L. Specifically, the motion scales S refer to the number of upsampled frames used to gen-

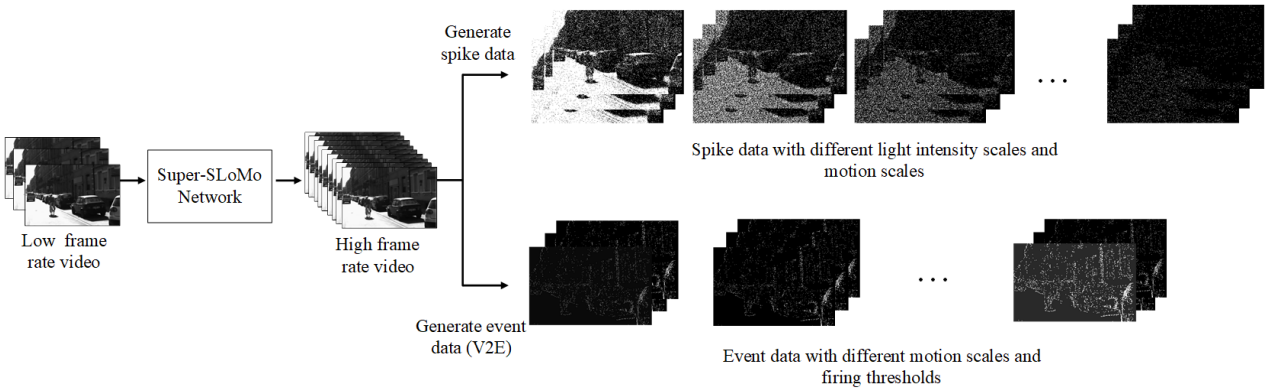

Figure 2. The pipeline of generating simulated spike and event data. The input low frame rate video (30 FPS) is upsampled to about 20,000 FPS. Based on the high frame rate video, the spike data and event data are generated according to Algorithm 1.

erate spike data with length T. In our experiment, we set $S = \{16, 32, 64, 128, 256\}$ and $T = 256$. Different motion speeds can be simulated by changing S. For example, if we set $S = 16$, then the generator will use 16 upsampled frames to generate 256 spike planes, which is equivalent to slow motion. If we set $S = 256$, then fast motion is simulated. The light intensity scales L change the spike frequency by adjusting the integral contribution of each pixel gray value. We set $L = \{1/2, 1/4, 1/8, 1/16, 1/32\}$ in our dataset. By setting a small L, the generated spikes can be more sparse to simulate the scene with weak light intensity. For the event data, we also change the contrast threshold to simulate different light intensities. Thus, during the training process, a simulated event sequence corresponds to 25 simulated spike sequences (5 different light scales times 5 different motion scales) and a groundtruth image.

Algorithm 1 Simulated Spike and Event Data Generator.

Input: A frame f_{ori} in original video and its timestamp $t_{f_{ori}}$, the temporal upsampled video V_{up}, the light intensity scales L, the motion scales S and the output spike data length T.
Output: The simulated spike data $\mathbf{f}_{spike} \in \mathbb{R}^{m \times n \times T}$, asynchronous event data $\mathbf{f}_{event}$ and groundtruth $\mathbf{g} \in \mathbb{R}^{m \times n}$.

Initialize: The noise matrix $N \in \mathbb{R}^{m \times n}$ subject to $\mathcal{N}(0, 1)$, the integrator $I \in \mathbb{R}^{m \times n}$
for each motion scale S_j **in** S **do**
  $F \leftarrow S_j$ frames in V_{up} according to $t_{f_{ori}}$
  for each light scale L_i **in** L **do**
    $I \leftarrow N \cdot \phi$
    for each frame F_k **in** F **do**
      for iter **in** range($\lfloor T/S_j \rfloor$) **do**
        $I \leftarrow I + F_k \cdot L_i$
        Generating spike plane $f_{plane} \in \mathbb{R}^{m \times n}$ by comparing I and ϕ
        $\mathbf{f}_{spike} \leftarrow \mathbf{f}_{spike} \cup f_{plane}$
        $I \leftarrow I - f_{plane} \cdot \phi$
      end for
    end for
    Output the simulated spike data $\mathbf{f}_{\text{spike}}$ with light scale L_i and motion scale S_j
    Clear $\mathbf{f}_{spike}$
  end for
  Generating event data using S_j frames by V2E simulator.
  Output the simulated event data $\mathbf{f}_{event}$ with motion scale S_j.
  Output the groundtruth $\mathbf{g} \leftarrow f_{ori}$ with motion scale S_j.
end for
end

4. Details of Real World Dataset

4.1. Spike and Event Calibration

Inspired by [11], 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. To ensure the consistency of spike and event data both in the temporal and spatial domain, a calibration step is needed. The details of the two neuromorphic cameras registration are described as follows:

[**Spatial calibration (SC)**] First, to ensure the SC parameters are unchanged, we fix the beam splitter and camera lens through the customized adapter rings. Then, a checkerboard is captured by the camera system to make a full view. Since this scene is static, we directly use the APS images and the reconstructed textures (using TFI method) from Vidar to calculate the SC parameters (i.e., a homography matrix). An affine transformation is performed to connects two sets of coordinates as $[x_i^s, y_i^s, 1]^T = \mathbf{H}[x_i^e, y_i^e, 1]]^T$.

[**Temporal calibration (TC)**] A coarse TC step is first realized using a synchronization script to trigger the sampling software of two cameras simultaneously. Then, a fine TC step is needed to ensure the time accuracy in microseconds for high-speed scenes: We transform the event and spike data into event frames (e.g., 500 μs events for the driving scene) and texture images (with 20,000 Hz). By manually

Table 1. The details of our real world spike and event dataset.

	Scene	Seq. number	Time length	Spike number	Event number	Description
Outdoor	Driving1	2	2 × 2 s	204411034	9848112	Low light
	Driving2	2	2 × 2 s	742923817	3783720	HDR/normal
	Walking	1	2 s	572464038	10663676	HDR/low light
	Person	1	2 s	643442363	5294827	Low light
	Roof	1	1.4 s	1270146167	335494	High light
High-speed	Rotation (light condition1)	5	5 × 0.5s	92266915	12307410	High speed (five speeds from 500 - 2600 RPM)
	Rotation (light condition2)	5	5 × 0.5s	77361295	11052540	High speed (five speeds from 500 - 2600 RPM)

comparing them, fine-tuning of the timestamp is performed to achieve the fine TC.

The spatio-temporal accuracy is sufficient for image reconstruction after SC and TC steps.

4.2. Real World Dataset

We construct a real world 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). The details of the dataset are shown in Table 1. More results on the real world dataset can be found in this document and our supplementary video. The dataset will be released later.

5. More Results and Discussions

5.1. Additional Results

The number of parameters and runtime are shown in Table 2. To test the performance under noise conditions, we conduct experiments on simulated data ('0001' of KITTI dataset). For event data, we add the shot noise via V2E simulator. For spike data, we adjust the standard deviation σ to simulate different FPN. We test three different noise levels: small (0.001 HZ shot noise for event data, σ=10 for spike data), medium (10Hz, σ=50), high (100Hz, σ=90). The noise greatly affects the reconstruction of E2VID and TFI, especially at high noise levels. Our method is less affected by noise.

Table 2. The inference time comparison on GPU and CPU.

Resolution	GPU (ms)			CPU (ms)		
	E2VID	FireNet	Ours	E2VID	FireNet	Ours
240×180	6.15	2.21	10.31	116.14	25.57	68.35
346×260	11.13	3.48	17.53	207.34	51.39	120.34
400×250	14.33	4.02	22.49	227.14	55.84	140.12
640×480	28.46	10.25	51.55	630.41	210.6	473.46

* Test on NVIDIA Titan Xp GPU and Intel 2.2GHz E5-2630 CPU.
* Number of parameters: E2VID 10700k, FireNet 38k, Ours 5985k.

In order to compare our method with state-of-the-art spike and event-based image reconstruction methods, we conduct experiments on our simulated dataset. We compare our method with TFP [32] (including four different reconstruction window sizes), TFI [32], SNM [33], FireNet [23], and E2VID [20]. The former three methods are based on spike data, while the latter two methods are based on event data. There are two experiments on simulated data: we first fix the motion speed scale $S = 128$ and change the light intensity scales L from 1/32 to 1/2 (see Figure 3). Then, we fix L to 1/4, and change S from 16 to 256. The comparing results are presented in Figure 3 and 4. It clearly shows that our method can combine the advantages of spike and event data, and thus generate higher quality images with clear motion details and less noise.

Table 3. Quantitative comparison on different noise levels.

Noise level	PSNR			SSIM		
	E2VID	TFI	Ours	E2VID	TFI	Ours
Small	14.76	21.76	26.07	0.6463	0.6911	0.8578
Medium	13.68	21.03	25.84	0.5381	0.6395	0.8503
High	11.03	18.56	25.48	0.3068	0.5259	0.8009

* E2VID: event, TFI: spike, Ours: spike+event.

Figure 5 provides results on the real dataset. Our method performs well in HDR and low light scenes. Compared with other methods, our method performs well by combining spike and event data.

We also test spike version of our network (the encoder only contains texture path) on the PKU-Spike-Recon dataset [33]. This dataset is captured under ideal light conditions for the spike camera, i.e. high light intensity. Our method has a good performance on these scenes. As shown in Figure 6, our method achieves better image quality in this dataset. The motion details are better reconstructed while the noise is suppressed.

5.2. Discussions

Generally speaking, with the ideal illumination, the spike camera has the ability to record full texture information with high speed. However, the spike camera very relies on light intensity. If the light intensity is insufficient, the spike emitted by the spike camera will be very sparse, resulting in the reduction of effective signal (most signals are "0") and the increase of noise. Therefore, a key problem of spike camera image reconstruction is how to better reconstruct under different light intensities. As a well-known neuromorphic camera, DVS make up for the shortage of spike camera. DVS has the ability of motion sensitive and high dynamic range sampling. Moreover, event cameras are less dependent on the light intensity, which is exactly what spike camera lacks. Meanwhile, the ability of full texture sampling of spike camera also solves the problem that the event camera is difficult to obtain texture.

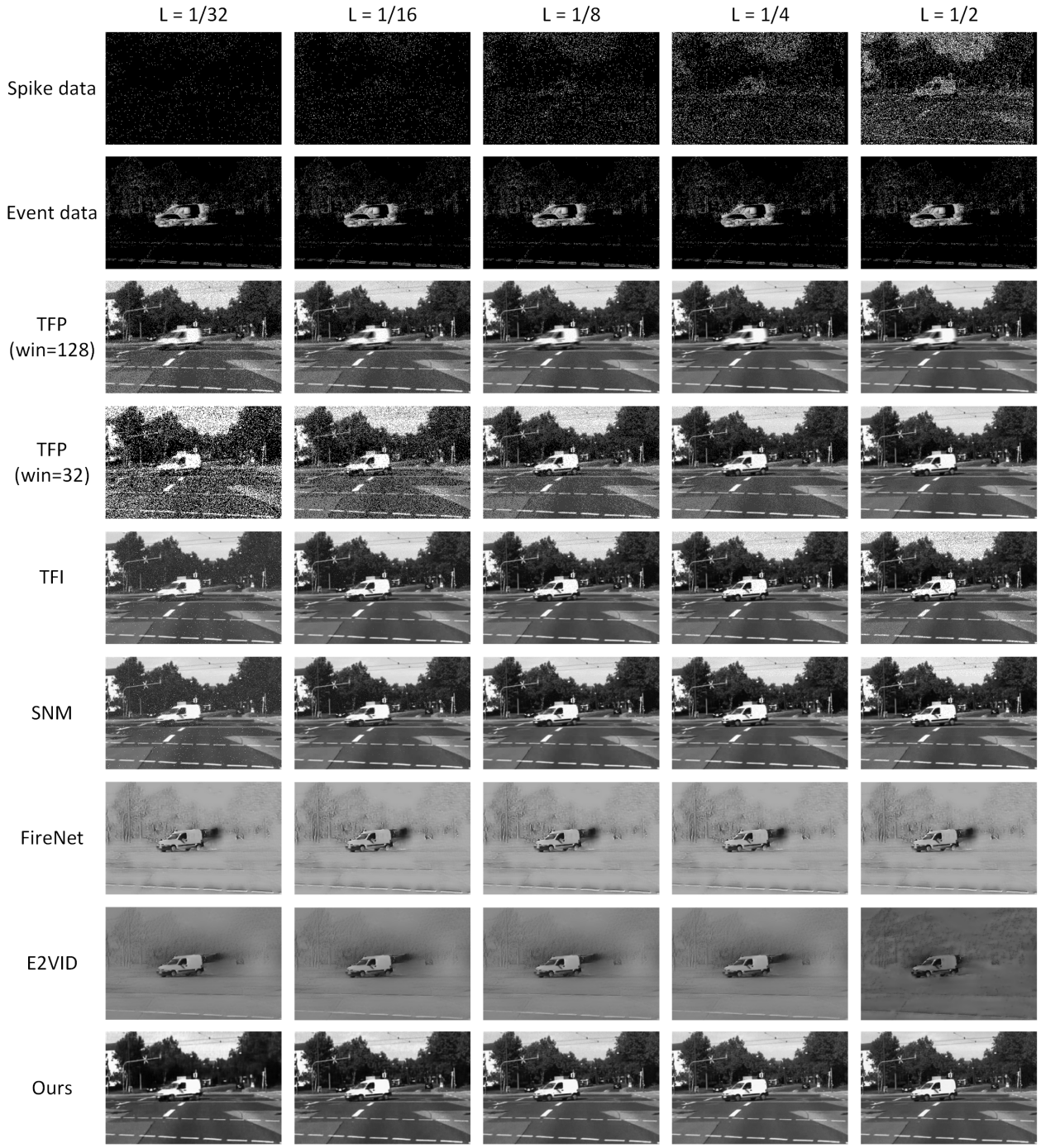

Figure 3. The reconstruction results on the simulated data with different light intensities. We fix the motion speed scale $S = 128$ and change the light intensity scales L from 1/32 to 1/2. The results show that the spike-based methods are sensitive to light intensity while the event-based methods are more robust. However, the event-based methods are difficult to reconstruct the background texture. The result of TFP is affected by noise when the light intensity is small. This is because it is difficult to collect enough spikes in the window for reconstruction when the light intensity is weak. On the contrary, the noise is obvious in the results of TFI when the light intensity is high. Our method uses both spike and event data to reconstruct high quality texture images.

References

[1] Patrick Lichtsteiner an Christoph Posch and Tobi Delbruck. A 128 $\times$ 128 120 db 15 μs latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits*, 43(2):566–576, 2008. 1

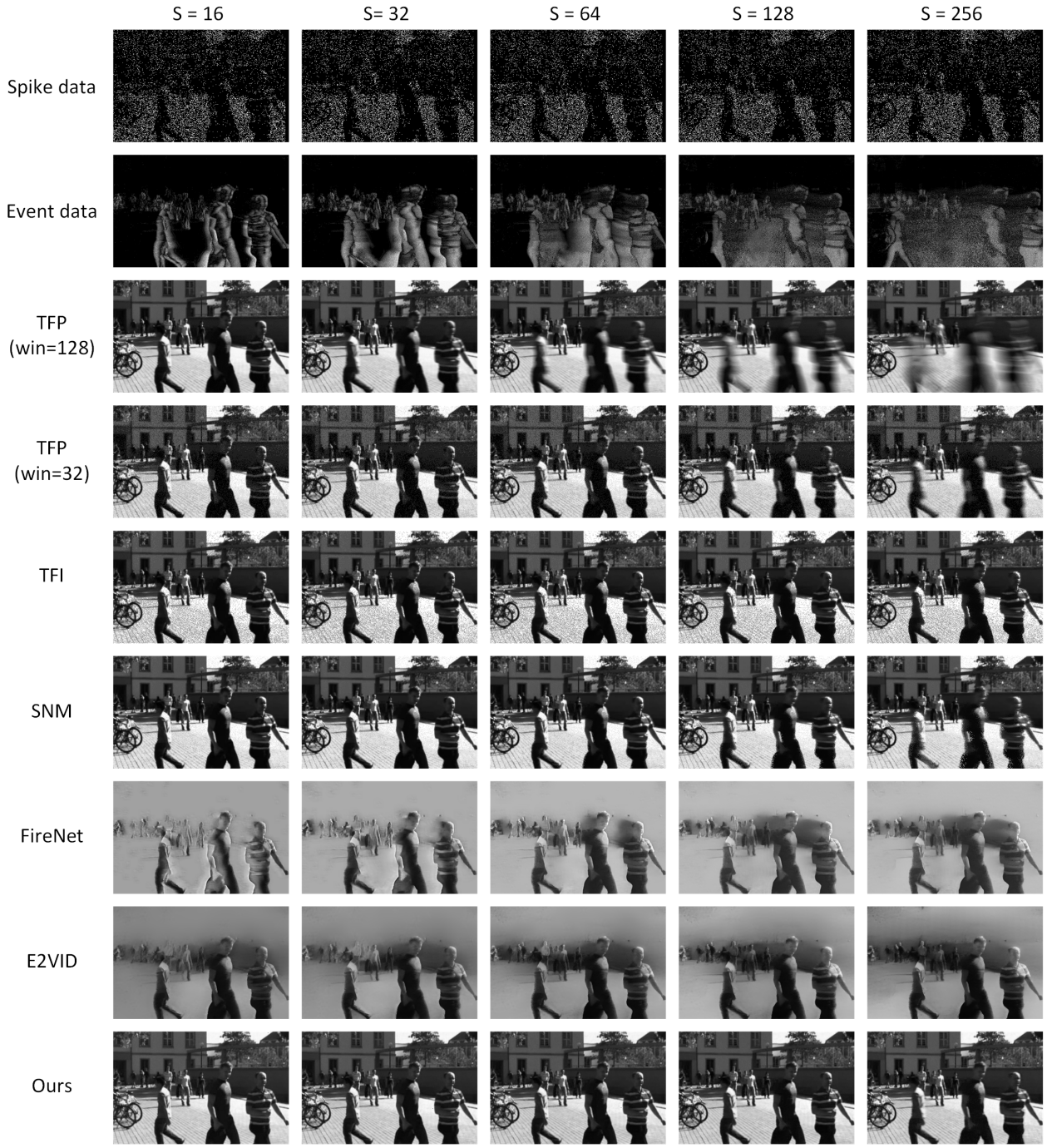

Figure 4. The reconstruction results on the simulated data with different motion speeds. We fix the light intensity scale L to 1/4 and change the motion speed scale S from 16 to 256. The results show that the event-based methods perform better when the motion is larger. The results of TFP suffer from motion blur in large motion speed scales. TFI and SNM can reconstruct high speed motion, but are affected by the noise.

[2] 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. 2

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

[4] Jonghyun Choi, Kuk-Jin Yoon, et al. Learning to super

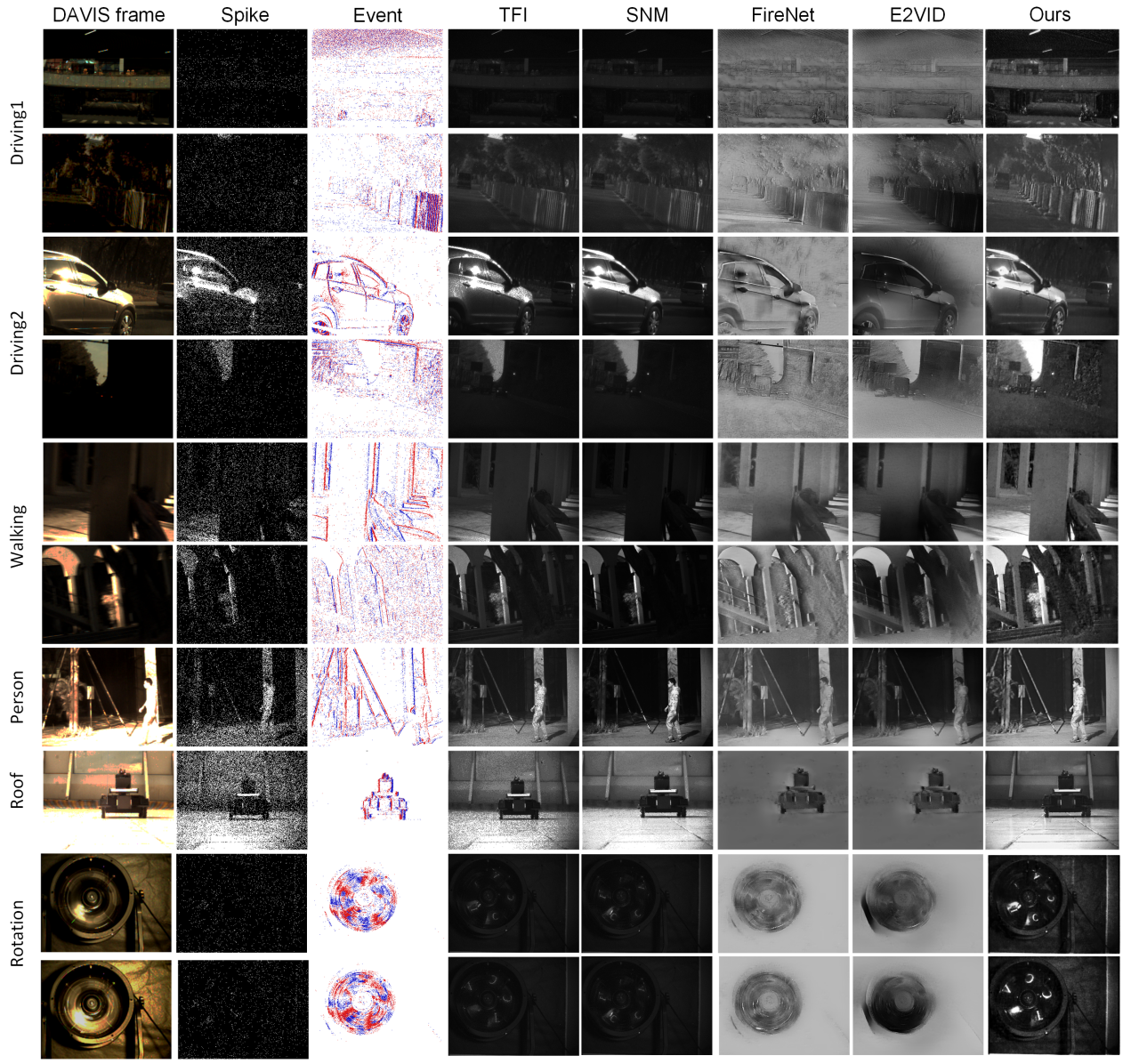

Figure 5. The reconstruction results on our real world dataset.

resolve intensity images from events. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2768–2776, 2020. 2

[5] 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. 2

[6] Tobi Delbruck, Yuhuang Hu, and Zhe He. V2e: From video frames to realistic dvs event camera streams, 2020. 2

[7] 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

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

[9] 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. 2

[10] 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

[11] 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. 2, 3

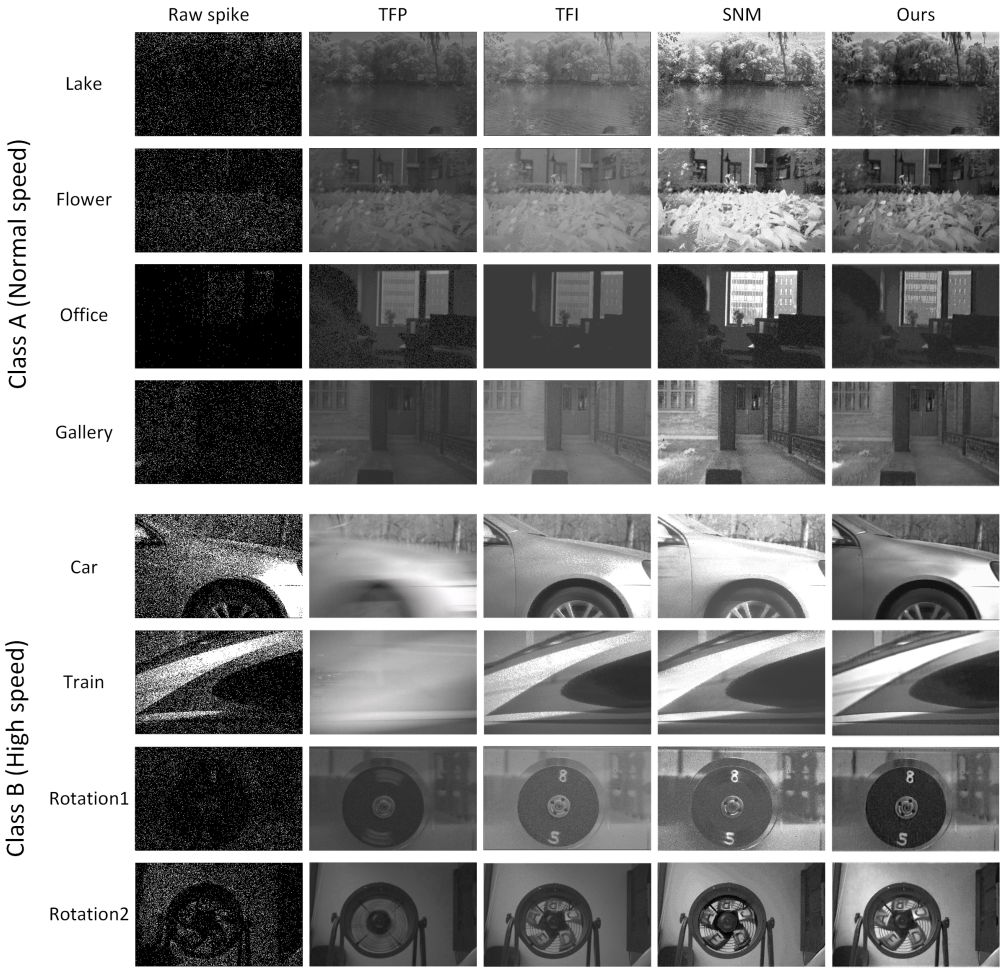

Figure 6. The results on PKU-Spike-Recon dataset [33]. The dataset consists of normal speed and high speed scenes under high light intensities. Our method can handle all these scenes. Compared with TFP, TFI ,and SNM, our method is effective in reconstructing high-speed motion and removing noise.

[12] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9000–9008, 2018. 2

[13] 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. 2

[14] 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. 2

[15] 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. 2

[16] 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. 2

[17] 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. 2

[18] 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. 2

[19] 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, 2

[20] 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, 4

[21] 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. 1

[22] 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. 2

[23] 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. 2, 4

[24] 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. 2

[25] 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*, pages 534–549. Springer, 2020. 2

[26] Xiao Wang, Jianing Li, Lin Zhu, Zhipeng Zhang, Zhe Chen, Xin Li, Yaowei Wang, Yonghong Tian, and Feng Wu. Visevent: Reliable object tracking via collaboration of frame and event flows, 2021. 1

[27] Xiao Wang, Xiujun Shu, Zhipeng Zhang, Bo Jiang, Yaowei Wang, Yonghong Tian, and Feng Wu. Towards more flexible and accurate object tracking with natural language: Algorithms and benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13763–13773, 2021. 1

[28] Xiao Wang, Jin Tang, Bin Luo, Yaowei Wang, Yonghong Tian, and Feng Wu. Tracking by joint local and global search: A target-aware attention-based approach. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–15, 2021. 1

[29] Woodward Yang. A wide-dynamic-range, low-power photosensor array. In *Proceedings of IEEE International Solid-State Circuits Conference ISSCC*, pages 230–231, 1994. 2

[30] 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*, pages 666–682. Springer, 2020. 2

[31] Jin 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. 2

[32] 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. 2, 4

[33] 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. 2, 4, 8