SpikingSIM: A Bio-Inspired Spiking Simulator

Junwei Zhao[1], Shiliang Zhang[1,*], Lei Ma[1,2,*], Zhaofei Yu[1], Tiejun Huang[1,2]
[1]*Institute of Digital Media, Peking University, Beijing, China*
[2]*Beijing Academy of Artificial Intelligence, Beijing, China*

2022 IEEE International Symposium on Circuits and Systems (ISCAS) | 978-1-6654-8485-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/ISCAS48785.2022.9937811

***Abstract*—Large-scale neuromorphic dataset is costly to construct and difficult to annotate because of the unique high-speed asynchronous imaging principle of bio-inspired cameras. Lacking of large-scale annotated neuromorphic datasets has significantly hindered the applications of bio-inspired cameras in deep neural networks. Synthesizing neuromorphic data from annotated RGB images can be considered to alleviate this challenge. This paper proposes a simulator to generate simulated spiking data from images recorded by frame cameras. To minimize the deviations between synthetic data and real data, the proposed simulator named SpikingSIM considers the sensing principle of spiking cameras, and generates high-quality simulated spiking data, e.g., the noises in real data are also simulated. Experimental results show that, our simulator generates more realistic spiking data than existing methods. We hence train deep neural networks with synthesized spiking data. Experiments show that, the network trained by our simulated data generalizes well on real spiking data. The source code of SpikingSIM is available at *http://github.com/Evin-X/SpikingSIM*.**

***Index Terms*—Bio-inspired Camera, Simulation, Neuromorphic Computing, Classification**

I. Introduction

Bio-inspired cameras work in different principle with conventional frame cameras [1]. Instead of capturing images at a fixed frame rate, each pixel of bio-inspired cameras simulates a retinal photosensitive cell, i.e., perceiving light and firing spikes asynchronously [2, 3]. Compared with frame cameras, bio-inspired cameras enjoy the advantages of high temporal resolution and high dynamic range [4], making them have been gaining more and more attention in computer vision tasks like detection [5–7], tracking [8–10], recognition [11, 12], optical flow estimation [13–15], and intensity-image reconstruction [16, 17], etc. Current bio-inspired cameras can be classified into two categories, i.e., event cameras and spiking cameras, respectively. Different from the differential sampling principle of event cameras that measure the relative brightness change [18], spiking cameras are based on the integral sampling principle. Each pixel of spiking cameras encodes the brightness intensity over a short time period into spikes, thus reserves more brightness cues than event cameras [19, 20].

Deep learning frameworks have demonstrated remarkable performance in vision tasks. Training deep models requires a huge amount of annotations, while annotating large-scale neuromorphic dataset is expensive and difficult due to the characteristics of asynchronous streams. One of feasible solutions is to design simulators to generate synthetic neuromorphic data from available image/video datasets, as illustrated in Fig. 1. For instance, Rebecq et al. [21] proposed ESIM, the first simulator for event cameras. Based on the ESIM, V2E toolbox [22] was developed as an extension to provide a circuit-level simulation model. Zhu et al. [23] proposed an end-to-end generative model to convert labeled images into events.

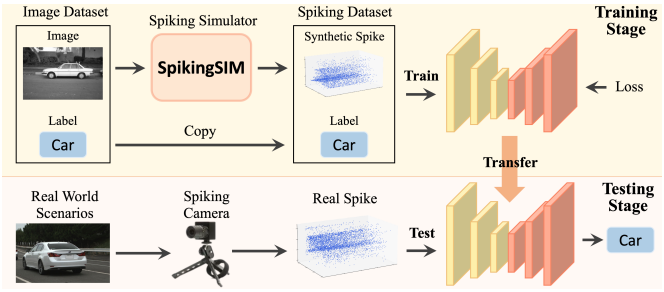


Fig. 1. The SpikingSIM converts existing large scale image datasets into synthetic spiking datasets. Deep networks trained on the synthetic data generalize well to real spiking data. This simulator brings potentials of applying annotations in existing image datasets to neuromorphic vision, and promotes the applications of spiking cameras in high-level vision tasks.

Although many simulators have been designed for event cameras, the research of simulating spiking cameras is still at an early stage. Lacking of annotated spiking datasets makes most of current researches focus on low-level vision tasks such as image reconstruction [24–26] and denoising [27]. It is hence appealing to study effective simulator to generate spiking data. Zhao et al. [24] developed a simulator for a spiking camera to generate synthesized spiking streams for image reconstruction. This method is based on an ideal integrate-and-fire model without considering noises. This defect makes it not capable to simulate realistic spiking data, and limits its application in training deep networks for high-level vision tasks.

This paper targets to generate more accurate simulated spiking data based on the working principle of spiking cameras. Specifically, we derive the relation between brightness intensity and the spike generation. Based on the brightness-spike relation, we utilize the grayscale of images to simulate brightness for generating spikes. The source and distribution of noises are also considered to simulate realistic noises in spiking data. We compare the generated spiking data against a real spiking dataset, achieving the similarity of 99%. Besides, as illustrated in Fig. 1, we test the generalization capability of the model trained on synthetic spiking data in classification tasks. On the spike-MNIST dataset collected by using a spiking camera to record the digits from MNIST dataset, the transferred model achieves 96% classification accuracy.

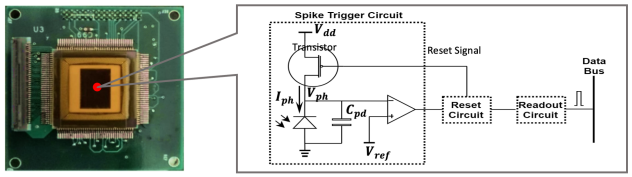
To our best knowledge, it is the first open-source spiking

This work was supported by the National Natural Science Foundation of China (62088102, 62176003, U20B2052, 61936011) and the National Key R&D Program of China (2018YFE0118400, 2021ZD0109802).

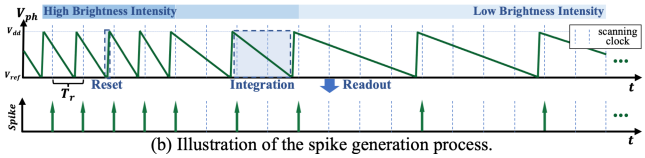
*Corresponding author (slzhang.jdl@pku.edu.cn; lei.ma@pku.edu.cn)

TABLE I
BASIC TECHNICAL PARAMETERS OF THE SENSOR

Parameter	Value	Parameter	Value
Spatial Resolution (WH)	400×250	Operating Clock(CLK)	10MHz
Supply Voltage (V_{dd})	3V	Ref. Voltage (V_{ref})	1V
Capacitance (C_{pd})	15fF	Time Resolution (T_r)	25μs



(a) Chip of a spiking camera and its pixel structure.



(b) Illustration of the spike generation process.

Fig. 2. Illustration of the working principle of spiking cameras.

simulator which considers both the brightness-spike relation and noise simulation. Experiments show that, this simulator provides more realistic synthetic spiking data than existing methods. The deep models trained by the synthetic data generalize well on real spiking data. This demonstrates the effectiveness of our simulation method.

II. Working Principle of Spiking Cameras

We first briefly introduce the working principle of the spiking camera adopted in [25]. Our simulation algorithm is proposed for this camera. The basic parameters of this spiking sensor are summarized in Tab. I. Fig. 2 (a) shows that each pixel of the spiking camera comprises three parts: the spike trigger, reset and readout circuits, respectively. Fig. 2 (b) shows the working principle of the camera, which has three states, i.e., integration, reset, and readout, respectively. In the integration state, the photo-diode in each pixel converts the photocurrent I_{ph} into the voltage V_{ph} on capacitance C_{pd}. As the photo-diode continues to generate electric charges, the voltage V_{ph} will decrease. When V_{ph} reaches the reference voltage V_{ref}, the comparator flips over. Once the reset circuit detects the flip signal, the pixel enters the reset state. The reset circuit then generates a 1-bit signal to reset the photo-diode. After a very short time interval, a new integration stage is resumed. Meanwhile, the 1-bit signal is stored in the readout circuit. In the readout state, the signal stored in the readout circuit is transmitted to the data bus by a scanning clock (40KHz). Thus, the maximum spike firing frequency is 40KHz, which leads to the time resolution of spiking camera is $T_r = 25\mu s$. Afterwards, the readout circuit clears its storage.

According to the above analysis and Fig. 2, the condition that a pixel emits a spike can be formulated as, i.e.,

$$\frac{1}{C_{pd}} \int_{t}^{t+\Delta t} I_{ph} dt \geq \theta, \tag{1}$$

where Δt is the time period of integration, the firing threshold defined as $\theta = V_{dd} - V_{ref}$, V_{dd} is the supply voltage. We denote $\frac{1}{C_{pd}} \int I_{ph} dt$ as V_p for simplification. Fig. 2 (b) illustrates

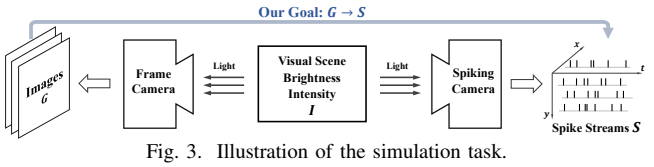


Fig. 3. Illustration of the simulation task.

the generated spike stream. It can be observed that, stronger brightness leads to a faster decreasing V_{ph} and more frequent spikes. Assuming the brightness of a pixel is stable, we can derive the relation between its spike firing frequency f_s and I_{ph} as follows:

$$f_s = \frac{1}{\Delta t} = \frac{I_{ph}}{\theta C_{pd}}, \tag{2}$$

where the maximum f_s equals to the scanning clock of the readout circuit, i.e., 40KHz.

III. Method

A. Formulation

As shown in Fig. 3, suppose we use a spiking camera and a frame camera to record the same visual scene, S and G denote the resulting spiking stream and image, respectively. Our goal is to generate the synthetic spike streams $\hat{S}$ from image G, meanwhile minimizing the differences between $\hat{S}$ and S, i.e.,

$$\hat{S}^* = \arg\min_{\hat{S}} \mathcal{M}(S, \hat{S}), \tag{3}$$

where $\hat{S}$ is generated from a spiking simulator $\mathcal{F}$, which uses the image G as input. $\mathcal{M}(\cdot)$ computes the differences between two spiking streams. Note that two spike streams can be regarded as similar if they have 1) similar spike firing frequency f_s, and 2) similar distributions on spike intervals.

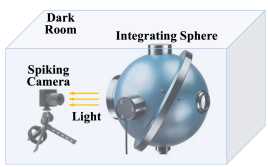
$\mathcal{F}$ is the key of this simulation model. Note that the firing frequency of spikes encodes the brightness cues. To simulate a similar spike firing frequency, $\mathcal{F}$ is expected to build the pixel-wise relation between G and S. In other words, the f_s of a certain pixel in $\hat{S}$ is determined by the corresponding pixel value in G, and larger pixel values corresponds to larger f_s. Sec. IV-B derives the relation based on the working principle of spiking cameras and experimental results.

In real scenarios, the spike generation procedure is affected by noises, making the generated spike streams show varied spike intervals as illustrated in Fig. 2 (b). According to the source of noises, we categorize noises into two types. One comes from the diffuse reflection, the other is inherent noise caused by manufacture, e.g., dark current, mismatch of capacitor and transistor. To make the synthetic spiking data has more realistic distributions on spike intervals, we experimentally study the distribution of noise and add them into the simulation procedure, which is demonstrated in Sec. IV-C.

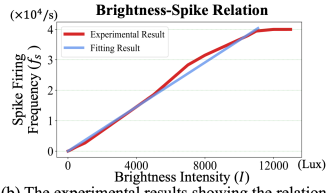
By considering the diffuse noise N_1 and inherent noise N_2, we denote the simulator $\mathcal{F}$, as well as its output $\hat{S}$ as,

$$\hat{S} = \mathcal{F}(G + N_1, T) + N_2, \tag{4}$$

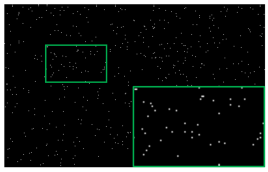
where, T denotes the time of exposure for recording image G, and the simulator $\mathcal{F}(\cdot)$ is derived based on the relation between G and S. The simulated $\hat{S}$ is a spiking stream with time duration of T.



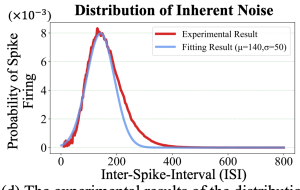
(a) The experimental system: setting the spiking camera in a dark room.



(b) The experimental results showing the relation between f_s and I.



(c) Visualization of the inherent noise.



(d) The experimental results of the distribution of inherent noise.

Fig. 4. The experiments supporting the derivation of the simulation method.

B. Analysis of the Image-Spike Relation

We first determine the relation between G and the brightness intensity I of a scene, then derive the relation between I and the spike firing frequency f_s. Note that the grayscale of G is commonly linearly proportional to I in CCD based frame cameras. We hence could simply formulate the imaging of G as $G = \lambda I$, where λ is a constant.

To simulate the f_s of spikes generated by $\mathcal{F}$, we proceed to derive the relation between f_s and I. As shown in Eq. (2), f_s is linearly associated with I_{ph}. We assume that $I_{ph} = \mathcal{R}(I)$, and derive the relation between f_s and I from Eq. (2) as:

$$f_s = \mathcal{R}(I)/(\theta C_{pd}). \tag{5}$$

Due to the difficulty of measuring the function $\mathcal{R}(\cdot)$, we construct experiments to directly determine the relation between f_s and I, based on experimental settings as shown in Fig. 4 (a). The experimental results in Fig. 4 (b) show that, when the brightness intensity is lower than the turning point I_s (i.e., the 11000 Lux brightness producing the maximum 40KHz spike frequency), f_s is approximately linearly proportional to I. It indicates that I_{ph} is linearly correlated with I as well. The corresponding linear parameter η can be calculated as $1.09e$-13. When I is higher than I_s, f_s keeps the maximum firing frequency $f_{max} = \frac{1}{T_r}$ = 40KHz. According to Fig. 4 (b), the relation between f_s and G can be formulated as:

$$f_s = \begin{cases} (\eta G)/(\lambda\theta C_{pd}), & I \le I_s \\ 1/T_r \quad , & I > I_s \end{cases} \tag{6}$$

We hence use Eq. (6) to simulate f_s for each pixel, given its brightness in image G. This results in a spiking stream with a uniform interval.

C. Analysis of the Noise

In real scenarios, the number of photons that reach the sensing chip through diffuse reflection is randomly distributed over a time period [28], leading to varied spike intervals. Poisson distribution is commonly used for modeling the distribution of random events, and has been employed in modeling image noise [29]. Therefore, we adopt the Poisson model into the spike simulation process to model diffuse noise. We assume

Algorithm 1 The Simulation Algorithm

Input: The luminance intensity G_p of the pixel p on the image G, the simulation time $T = NT_r$.
Output: The simulated spike streams $\hat{S}_p = \{\hat{S}(p,n)\}_{n=1,..,N}$ at location p.

```
Initialize: the spike streams Ŝ_p with each spike Ŝ(p,n) = 0, the noise matrix N_2 ~ N(k|μ,σ)   ▷ N is the Gaussian Model
for each time interval n in range [1, N] do
    Initialize: noise matrix N_1 ~ P(k|T_r,γ)   ▷ P is the Poisson Model
    for each time unit t in range [1, T_r/δt] do
        G_p ← G_p + N_1(t)
        V_p(t) ← V_p(t−1) + (η/(λC_pd)) G_p · δt
        if V_p(t) ≥ θ then Ŝ(p,n) ← 1, V_p(t) ← 0 end if
    end for
    if N_2(n) ≠ 0 then Ŝ(p,n) ← 1 end if
end for
```

Fig. 5. Visualization of the reconstructed images from a spiking dataset.

that the probability of k noise photons arriving at a pixel in a time unit δt ($\delta t = \frac{2}{CLK}$) is γ, and the time period is T_r. Thus, the probability distribution function of the Poisson model can be formulated as:

$$\mathcal{P}(k|T_r,\gamma) = \frac{(\gamma T_r)^k}{k!} e^{-\gamma T_r}. \quad (k = 0, 1, 2...) \tag{7}$$

The inherent noise leads to fired spikes in the absence of light, as shown in Fig. 4 (c). To investigate the distribution of the inherent noise, we cover the spiking camera with a lens cap, then record the emitted spiking data in a time period (e.g., 1s). We count the Inter-Spike Interval (ISI) of the recorded data and draw the results in Fig. 4 (d). The results show that the distribution of ISI can be approximately fitted by a Gaussian model (ISI=k, mean(μ)=140, and std(σ)=50) denoted as,

$$\mathcal{N}(k|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{1}{2}\cdot(\frac{k-\mu}{\sigma})^2}, \quad (k = 0, 1, 2...) \tag{8}$$

which is hence used to amend the distribution of spike intervals based on the computed f_s in Eq. (6).

D. Implementation of the Simulation Method

Eq. (6) and Eqs. (7, 8) make it possible to simulate $\mathcal{F}$ and noises N_1 and N_2. The RGB images are first converted to YUV images, which are hence used for the simulation. Eq. (6) establishes the relation between each pixel value and its spike firing frequency. Eq. (7) and Eq. (8) add noise during the simulation for improving the data reality. The asynchronous imaging principle allows us to compute the spiking stream for each pixel independently. We take a pixel p as an example to demonstrate the simulation, as illustrated in Algorithm 1.

IV. Experiments

A. Validity of the Simulation Method

Implementation Details: We perform experiments on the spiking dataset adopted by [17, 24], which is captured by a spiking camera. The spike streams in this dataset record the real world scenes including gallery, lake, flower, street, etc. Firstly, we reconstruct intensity maps from raw spikes through

TABLE II
THE RESULTS OF VALIDATION EXPERIMENTS.

Metric	Method	Simulation time: 2.5ms				Simulation time: 5ms				Simulation time: 7.5ms				Simulation time: 10ms			
		Gallery	Lake	Flower	Street	Gallery	Lake	Flower	Street	Gallery	Lake	Flower	Street	Gallery	Lake	Flower	Street
$\mathcal{M}_1$	PSS [30]	0.589	0.611	0.632	0.785	0.642	0.667	0.701	0.842	0.718	0.736	0.758	0.857	0.745	0.762	0.786	0.887
	IFM [24]	0.594	0.627	0.653	0.781	0.654	0.691	0.727	0.825	0.731	0.745	0.763	0.843	0.763	0.776	0.797	0.883
	Ours	**0.991**	**0.993**	**0.993**	**0.994**	**0.993**	**0.995**	**0.995**	**0.996**	**0.995**	**0.995**	**0.996**	**0.997**	**0.995**	**0.996**	**0.996**	**0.997**
$\mathcal{M}_2$	PSS [30]	24.583	23.494	20.361	18.896	7.188	7.134	7.015	5.872	3.357	3.316	3.233	2.578	1.889	1.828	1.437	0.791
	IFM [24]	22.744	21.381	19.963	19.117	7.109	7.011	6.897	6.132	3.293	3.276	3.132	2.733	1.837	1.801	1.343	0.957
	Ours	**0.081**	**0.079**	**0.057**	**0.044**	**0.073**	**0.066**	**0.045**	**0.031**	**0.070**	**0.062**	**0.041**	**0.026**	**0.069**	**0.057**	**0.035**	**0.023**

* (1)Larger(smaller) values of $D_1(D_2)$ indicates the larger similarity between S and $\hat{S}$. (2) $D_1 \in [0, 1]$, $D_2 \in [0, +\infty)$.

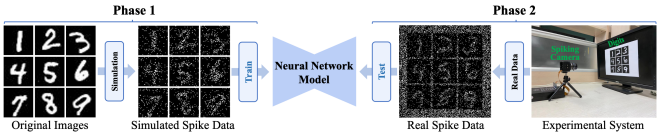


Fig. 6. Testing the generalization of the model trained by simulated data.

the Texture-From-Window (TFW) method [17], as visualized in Fig. 5. Given a spike stream with a time length T, the intensity of each pixel can be calculated as $\frac{n}{N}$ according to TFW, where n is the number of spikes, and $N = \frac{T}{T_r}$ is the number of time intervals. Then, we use those intensity maps to simulate spike streams via SpikingSIM, and calculate the similarity between the simulated spiking data and real spiking data. Note that the time length of simulated spikes is as same as the time length of the raw data used for reconstructing images. In addition, we compare the SpikingSIM with recent related works. Lee et al. [30] provided a spike generation approach for converting images into Poisson-distributed Spike Streams (PSS). Zhao et al. [24] developed a simulator for a spiking camera based on an Integrate-and-Fire Model (IFM).

Evaluation Metrics: We adopt two metrics to calculate the similarity between the simulated data and real data. The first metric is to measure the similarity of spike firing frequency of each pixel. Assuming that a stream of spikes can be represented as a matrix with the dimension of $N \times M$ ($M = W \times H$), the spike firing frequency of the pixel p can be calculated as $f_p(\cdot) = \frac{n}{N}$, where n is the number of spikes. Thus, this metric is represented as:

$$\mathcal{M}_1(S, \hat{S}) = \frac{1}{M} \sum_{p=1}^{M} (1 - \frac{|f_p(S) - f_p(\hat{S})|}{f_p(S)}). \tag{9}$$

The second metric uses Kullback-Leibler Divergence (KLD) to further measure the distribution differences between simulated data and real data. Assuming that the probability of the pixel p firing (not firing) spikes can be denoted as $P(S_p = 1)$ ($P(S_p = 0)$), the KLD metric can be formulated as:

$$\mathcal{M}_2(S, \hat{S}) = \sum_{p=1}^{M} \sum_{i=0,1} P(S_p = i) \log_2 \frac{P(S_p = i)}{P(\hat{S}_p = i)}. \tag{10}$$

Experimental Results: The evaluation results are summarized in Tab. II. The values of $\mathcal{M}_1$ show that SpikingSIM achieves 99% similarity with real data in spike firing frequency. The metric of $\mathcal{M}_2$ shows that our simulated data is quite similar with the original data in spike distribution. It is also clear that, our simulation method significantly outperforms the other methods. The remarkable performance can be attributed to the derivation of brightness-spike relation and the simulation of noises through experiments.

B. Experiments on Model Generalization

We further test the generalization ability of neural networks trained on our synthesized dataset. As shown in Fig. 6, we convert the image samples in MNIST dataset [31] into spike streams, and use synthesized data to train a LeNet-5 [31] (model structure: 32C5-AP2-64C5-AP2-FC256-FC10). We hence construct a real spiking dataset, named spike-MNIST for testing. Specifically, we randomly select 100 samples from the test set of MNIST dataset, and use a spiking camera to record 1ms of those digits. The LeNet-5 trained with the SpikingSIM achieves the classification accuracy of 96% on spike-MNIST dataset. As a comparison, we also adopt the synthetic data generated from the PSS [30] and IFM [24] to train models, which achieve the accuracy of 87% and 91%, respectively. This result demonstrates that our simulation method is more effective in generating annotated spiking data for model training. This is mainly benefited from its capability to generate more realistic spiking data.

C. Discussions

Data Augmentation: The SpikingSIM can also be adopted to convert video datasets to spiking data. This allows to use the massive video annotations in computer vision tasks, e.g., motion detection, tracking and action recognition, in spiking neural network training, which is potential to alleviate the scarcity of neuromorphic dataset and annotations.

Virtual Camera in Synthetic Scenarios: Certain scenarios with high speed motions, such as car collisions, are crucial for computer vision tasks such as autonomous driving, but are hard to acquire data. Most of existing approaches collect datasets for such scenarios through simulation platforms, such as CARLA [32]. Our model can be integrated into the simulation platforms as an embedded module, i.e., playing the role of a virtual camera. The advantage of this approach is that both the scenario parameters (e.g., brightness condition, car speed, etc.) and camera parameters (e.g., camera viewpoint, spacial and temporal resolution, etc.) are adjustable, thus enjoys plenty of flexibility and can provide comprehensive settings to study the applications of spiking cameras in such applications.

V. CONCLUSION

Insufficient neuromorphic annotations have severely limited the applications of spiking cameras in deep neural networks. This work provides an effective solution by proposing a simulator for spiking cameras. This simulator generates high-quality synthetic spiking data from conventional image datasets. We verify that the deep neural networks trained on synthetic data generalize well on real spiking data. Moreover, our released codes will facilitate the neuromorphic community.

REFERENCES

[1] D. Gehrig, M. Gehrig, J. Hidalgo-Carrió, and D. Scaramuzza, "Video to events: Recycling video datasets for event cameras," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3586–3595.

[2] G. Gallego, T. Delbruck, G. M. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis *et al.*, "Event-based vision: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 1, pp. 154–180, 2020.

[3] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza, "High speed and high dynamic range video with an event camera," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 6, pp. 1964–1980, 2019.

[4] A. Lakshmi, A. Chakraborty, and C. S. Thakur, "Neuromorphic vision: From sensors to event-based algorithms," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 9, no. 4, p. e1310, 2019.

[5] J. Manderscheid, A. Sironi, N. Bourdis, D. Migliore, and V. Lepetit, "Speed invariant time surface for learning to detect corner points with event-based cameras," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 245–10 254.

[6] X. Peng, B. Zhao, R. Yan, H. Tang, and Z. Yi, "Bag of events: An efficient probability-based feature extraction method for aer image sensors," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 4, pp. 791–803, 2016.

[7] H. Liu, D. P. Moeys, G. Das, D. Neil, S.-C. Liu, and T. Delbrück, "Combined frame-and event-based detection and tracking," in *IEEE International Symposium on Circuits and systems*. IEEE, 2016, pp. 2511–2514.

[8] H. Li and L. Shi, "Robust event-based object tracking combining correlation filter and cnn representation," *Frontiers in neurorobotics*, vol. 13, p. 82, 2019.

[9] D. Gehrig, H. Rebecq, G. Gallego, and D. Scaramuzza, "Eklt: Asynchronous photometric feature tracking using events and frames," *International Journal of Computer Vision*, vol. 128, no. 3, pp. 601–618, 2020.

[10] H. Leow and K. Nikolic, "Machine vision using combined frame-based and event-based vision sensor," in *IEEE International Symposium on Circuits and Systems*. IEEE, 2015, pp. 706–709.

[11] R. Xiao, H. Tang, Y. Ma, R. Yan, and G. Orchard, "An event-driven categorization model for aer image sensors using multispike encoding and learning," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 9, pp. 3649–3657, 2019.

[12] B. R. Pradhan, Y. Bethi, S. Narayanan, A. Chakraborty, and C. S. Thakur, "N-har: A neuromorphic event-based human activity recognition system using memory surfaces," in *IEEE International Symposium on Circuits and Systems*. IEEE, 2019, pp. 1–5.

[13] F. Paredes-Vallés, K. Y. Scheper, and G. C. de Croon, "Unsupervised learning of a hierarchical spiking neural network for optical flow estimation: From events to global motion perception," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 8, pp. 2051–2064, 2019.

[14] M. Almatrafi, R. Baldwin, K. Aizawa, and K. Hirakawa, "Distance surface for event-based optical flow," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 7, pp. 1547–1556, 2020.

[15] M. T. Aung, R. Teo, and G. Orchard, "Event-based plane-fitting optical flow for dynamic vision sensors in fpga," in *IEEE International Symposium on Circuits and Systems*. IEEE, 2018, pp. 1–5.

[16] D. P. Moeys, C. Li, J. N. Martel, S. Bamford, L. Longinotti, V. Motsnyi, D. S. S. Bello, and T. Delbruck, "Color temporal contrast sensitivity in dynamic vision sensors," in *IEEE International Symposium on Circuits and Systems*. IEEE, 2017, pp. 1–4.

[17] L. Wang, T.-K. Kim, and K.-J. Yoon, "Eventsr: From asynchronous events to image reconstruction, restoration, and super-resolution via end-to-end adversarial learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8315–8325.

[18] T. Delbrück, B. Linares-Barranco, E. Culurciello, and C. Posch, "Activity-driven, event-based vision sensors," in *IEEE International Symposium on Circuits and Systems*. IEEE, 2010, pp. 2426–2429.

[19] S. Dong, T. Huang, and Y. Tian, "Spike camera and its coding methods," in *Data Compression Conference*. IEEE, 2017, pp. 437–437.

[20] T. Huang, Y. Zheng, Z. Yu, R. Chen, Y. Li, R. Xiong, L. Ma, J. Zhao, S. Dong, L. Zhu *et al.*, "1000x faster camera and machine vision with ordinary devices," *arXiv preprint arXiv:2201.09302*, 2022.

[21] H. Rebecq, D. Gehrig, and D. Scaramuzza, "Esim: an open event camera simulator," in *Conference on Robot Learning*. PMLR, 2018, pp. 969–982.

[22] Y. Hu, S.-C. Liu, and T. Delbruck, "V2e: From video frames to realistic dvs events," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2021, pp. 1312–1321.

[23] A. Z. Zhu, Z. Wang, K. Khant, and K. Daniilidis, "Eventgan: Leveraging large scale image datasets for event cameras," in *IEEE International Conference on Computational Photography)*. IEEE, 2021, pp. 1–11.

[24] J. Zhao, R. Xiong, H. Liu, J. Zhang, and T. Huang, "Spk2imgnet: Learning to reconstruct dynamic scene from continuous spike stream," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11 996–12 005.

[25] J. Han, C. Zhou, P. Duan, Y. Tang, C. Xu, C. Xu, T. Huang, and B. Shi, "Neuromorphic camera guided high dynamic range imaging," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1730–1739.

[26] L. Zhu, S. Dong, J. Li, T. Huang, and Y. Tian, "Retina-like visual image reconstruction via spiking neural model," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1438–1446.

[27] Z. W. Wang, P. Duan, O. Cossairt, A. Katsaggelos, T. Huang, and B. Shi, "Joint filtering of intensity images and neuromorphic events for high-resolution noise-robust imaging," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1609–1619.

[28] H. W. Jensen, S. R. Marschner, M. Levoy, and P. Hanrahan, "A practical model for subsurface light transport," in *Proceedings of the annual conference on Computer graphics and interactive techniques*, 2001, pp. 511–518.

[29] S. W. Hasinoff, "Photon, poisson noise." pp. 608–610, 2014.

[30] C. Lee, S. S. Sarwar, P. Panda, G. Srinivasan, and K. Roy, "Enabling spike-based backpropagation for training deep neural network architectures," *Frontiers in neuroscience*, vol. 14, p. 119, 2020.

[31] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[32] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the Annual Conference on Robot Learning*, 2017, pp. 1–16.