Spike-Based Motion Estimation for Object Tracking Through Bio-Inspired Unsupervised Learning

Yajing Zheng, Zhaofei Yu, *Member, IEEE*, Song Wang, *Senior Member, IEEE*, and Tiejun Huang, *Senior Member, IEEE*

***Abstract*— Neuromorphic vision sensors, whose pixels output events/spikes asynchronously with a high temporal resolution according to the scene radiance change, are naturally appropriate for capturing high-speed motion in the scenes. However, how to utilize the events/spikes to smoothly track high-speed moving objects is still a challenging problem. Existing approaches either employ time-consuming iterative optimization, or require large amounts of labeled data to train the object detector. To this end, we propose a bio-inspired unsupervised learning framework, which takes advantage of the spatiotemporal information of events/spikes generated by neuromorphic vision sensors to capture the intrinsic motion patterns. Without off-line training, our models can filter the redundant signals with dynamic adaption module based on short-term plasticity, and extract the motion patterns with motion estimation module based on the spike-timing-dependent plasticity. Combined with the spatiotemporal and motion information of the filtered spike stream, the traditional DBSCAN clustering algorithm and Kalman filter can effectively track multiple targets in extreme scenes. We evaluate the proposed unsupervised framework for object detection and tracking tasks on synthetic data, publicly available event-based datasets, and spiking camera datasets. The experiment results show that the proposed model can robustly detect and smoothly track the moving targets on various challenging scenarios and outperforms state-of-the-art approaches.**

***Index Terms*— Neuromorphic vision sensor, bio-inspired, unsupervised learning, short-term plasticity, spike-timing-dependent plasticity, motion estimation, spiking camera, high-speed object tracking.**

I. Introduction

HIGH-speed object detection/tracking plays a critical role in autonomous driving and intelligent video analysis. Despite recent advances in object detection/tracking, it is still challenging to deal with uncertainty caused by object occlusions, disappearance, and reappearance. Moreover, a unique challenge posed by high-speed moving objects is that the images captured by digital cameras will be blurry due to insufficient sampling rate (typically 30Hz). Distinct from the conventional frame-based cameras, the neuromorphic vision sensors generate asynchronously binary outputs for all pixels based on the scene radiance change. Two common types of neuromorphic vision sensors are known as event cameras [1], [2], [3], [4] and spiking cameras [5], [6], [7], which apply a sampling mechanism that is similar to the photoelectric conversion process of the retina. Due to their distinctive high temporal resolution (<1 ms), motion information can be recorded more completely and accurately.

Several motion tracking methods [1] for event-based input have been developed in recent years. These methods can distinguish events belonging to different moving objects and have shown the superiority of neuromorphic vision sensors on the high-speed detection task. However, how to smoothly track multiple high-speed moving objects is still a challenging problem. Most of the existing object tracking methods for event cameras are based on motion segmentation or clustering models [8], [9], [10], [11], which learn motion models through time-consuming iterative optimization. Once the model parameters are not well initialized, the clustering results will be poor [12].

Spiking Neural Networks (SNNs) are viewed as the third generation of neural network models, which succeed in modeling behavior and learning potential of the brain [13]. Distinct from traditional Deep Neural Networks (DNNs) [14], the connection weight between neurons is modified by the temporal relationship of spikes [15], [16], [17]. As learning and information transmission is triggered by spikes, SNNs have the advantages of low energy consumption and low-latency, and are highly desirable for neuromorphic computing hardware [18], [19]. The neuromorphic vision sensors convert light signals into electrical signals, yielding spike trains or events as output that can be naturally processed by SNNs. There exist some motion estimation methods for event cameras based on SNNs [20], [21], [22], [23]. They are designed for simple scenes and can not be generalized to complicated cases with multiple moving objects directly. Parameshwara et al. [24] propose the first deep encoder-decoder SNN architecture, *SpikeMS*, for motion segmentation using events as input. Huang et al. [5] propose a spiking network for objects detection and tracking using spiking cameras. However, the motion

Manuscript received 8 March 2022; revised 11 September 2022 and 27 October 2022; accepted 16 November 2022. Date of publication 14 December 2022; date of current version 21 December 2022. This work was supported in part by the National Natural Science Foundation of China under Grant U1803264, Grant 62176003, and Grant 62088102; and in part by the China Postdoctoral Science Foundation under Grant 2022M720238. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Keigo Hirakawa. *(Corresponding author: Zhaofei Yu.)*

Yajing Zheng is with the National Engineering Laboratory for Video Technology, School of Computer Science, Peking University, Beijing 100871, China (e-mail: yj.zheng@pku.edu.cn).

Zhaofei Yu is with the Institute for Artificial Intelligence, Peking University, Beijing 100871, China, and also with the National Engineering Laboratory for Video Technology, School of Computer Science, Peking University, Beijing 100871, China (e-mail: yuzf12@pku.edu.cn).

Song Wang is with the Department of Computer Science and Engineering, University of South Carolina, Columbia, SC 29208 USA (e-mail: songwang@cec.sc.edu).

Tiejun Huang is with the National Engineering Laboratory for Video Technology, School of Computer Science, Peking University, Beijing 100871, China, and also with the Institute for Artificial Intelligence, Peking University, Beijing 100871, China (e-mail: tjhuang@pku.edu.cn).

Digital Object Identifier 10.1109/TIP.2022.3228168

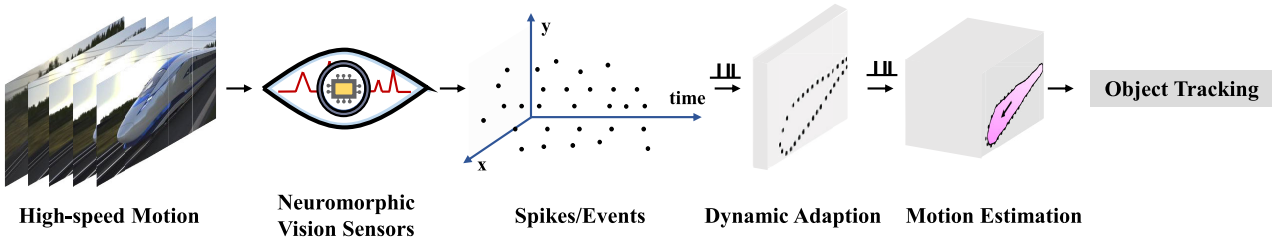

Fig. 1. Architecture of the ODTSnet for neuromorphic vision sensors. The dynamic adaption module filters the spikes/events from neuromorphic vision sensor. Then the motion estimation module extracts motion patterns from the filtered spikes for object detection/tracking.

information is not taken into account, making it easy to fail if spiking cameras have ego-motion.

Motivated by the imbalances between neuromorphic vision sensors and SNN-based tracking algorithms, in this paper, we seek to answer the following questions: How to utilize the spatiotemporal information of spiking cameras based on SNN? How to estimate the motion in challenging scenarios with spikes and tracking multiple objects smoothly? To that end, we propose an unsupervised Spiking Network for Object Detection and Tracking (**ODTSnet**, illustrated in Fig. 1). The ODTSnet framework contains two specific designed modules to directly process the spatiotemporal information of spikes/events: the Dynamic Adaption module and the Motion Estimation module. The proposed framework is inspired by the visual pathways for dynamic vision, which has the short-term adaption property and extracts the motion information through spike-timing-dependent learning. By fully taking advantages of both modules, our models can robustly detect and smoothly track moving objects. Our contributions can be summarized as:

- We propose a novel bio-inspired unsupervised framework based on spiking neural networks to robustly detect and smoothly track moving objects.
- We build short-term plasticity based filters by utilizing the spatiotemporal information of neuromorphic vision sensors, which can filter the redundant signals for efficient motion analysis.
- We propose a motion estimation module based on spike-timing-dependent plasticity, which can perceive motions in various complex scenes.
- We evaluate the models on various challenging neuromorphic datasets, and the proposed model outperforms the state-of-the-art approaches.

II. Related Works

In recent years, many detection and tracking methods based on event cameras have been proposed. In order to maximize the low latency of the event camera and verify the feasibility of the event camera in detecting and tracking tasks, many tracking algorithms initially assumed that the camera was fixed, and events were only generated by moving objects. Therefore, the information of different objects (such as position, size, etc.) can be obtained by clustering events, and then tracking is achieved by updating the parameter information corresponding to different objects [25], [26], [27], [28], [29], [30], [31]. However, most of these algorithms can only track objects with relatively simple shapes, such as lines or circles. Therefore, in order to continuously track objects with variable shapes, many algorithms propose methods of iteratively updating kernel functions/filters [32], [33], [34]. Although these algorithms can achieve better performance when the camera is stationary, in scenes with camera self-motion, the event stream generated by the camera can seriously interfere with the detection and tracking of moving objects.

Therefore, in order to detect independently moving objects and distinguish the event/spikes flow generated by the ego-motion of the camera, some methods distinguish the objects by grouping them with *motion estimation* [8], [9], [10]. For example, Mitrokhin et al. [9] transformed the events into a time image with a four-parameter motion model, and performed a global minimization on the warped image to separate different moving targets. Stoffregen et al. [10] jointly estimated the motion parameters and event-cluster membership through an iterative Contrast-Maximization (CM) algorithm on events sequence, but they need to predefine the number of clusters. Although these methods have shown excellent performance in some challenging scenarios, they usually take a long time to find the optimal motion parameters, and the low temporal latency characteristic of events is not fully utilized in these models. Besides, parameter initialization plays a critical role in finding optima. Otherwise, it may fail to find the motion pattern corresponding to each independent motion object or ego-motion of the camera. Liu et al. [12] proposed an improved global optimal CM method with novel bounding functions. However, it still needs to perform gradient-based optimization on each sequence, and it only tests on rotation scenes. Zhou et al. [11] used a space-time event graph representation to exploit the spatio-temporal nature of events, leading to globally consistent and locally coherent event-based motion segementation results. This method was also designed in the spirit of motion compensation [35], where the initialization of parameters is very time-consuming and is a key factor affecting performance.

In recent years, some deep-learning-based methods are proposed to solve the visual motion problem [36], [37], [38], [39], [40]. In the deep-learning based methods, events within a time-interval are binned and converted to an frame-like input representation to an encoder-decoder network [36], [37]. Using the events within a time-window, Parameshwara et al. segmented multi-object motion by jointly optimizing the

camera ego-motion and motion of independently moving objects. In order to contain clear spatio-temporal motion information, Chen et al. [39] converted the events into a sequence of synchronous Time-Surface with Linear Time Decay (TSLTD) frames. Instead of extracting the motion information globally, Kepple et al. [40] jointly learned visual motion and confidence from the events in spatially local patches. However, it is generally computationally expensive to optimize the parameters of the deep neural networks. Moreover, deep-learning-based object tracking models [41], [42], [43], [44] usually depend on "tracking-by-detection", which need to train detector for each object and require a large amount of labeled data with high cost. Moreover, the texture information that needs to be used to train the detector is primarily lost in neuromorphic vision sensors.

Improvements in recent years have enabled spiking neural networks to be trained directly through backpropagation [45], [46], [47], which in turn enables the use of deep spiking neural networks to accomplish complex tasks. Parameshwara et al. [24] proposed a novel loss consisting of a binary cross entropy loss and spike loss to end-to-end train the *SpikeMS*, obtaining binary motion-based segmentation.

III. Preliminaries

A. Event Cameras

Event cameras (also called Dynamic Vision Sensors, DVS) are one of the neuromorphic vision sensors, where each pixel independently outputs events when brightness change exceeds a given threshold [1]. The output of event cameras usually takes the form of address event representation (AER) $\xi : \{x, y, t, p\}$, where (x, y) is the event location on the image plane, t is the timestamp, and $p \in \{-1, 1\}$ is the polarity of the event (the sign of the intensity change).

B. Spiking Cameras

Unlike event cameras, spiking cameras continuously capture photons and generate asynchronous spikes for all pixels when the accumulated intensity reaches a predefined threshold [5], [6], [7]. The output of spiking cameras can be represented by a 3-tuple $\mathcal{S} : \{x, y, t\}$, where (x, y) is the spatial coordinates on the image plane, and t is the firing time of the spike. An illustration for event cameras and spiking cameras is shown in Fig. 2. Event cameras only generate events for pixels where brightness changes exceed a certain threshold. In spiking cameras, every pixel will fire spikes according to the input scene radiance, which make it ready-to-use for image reconstruction [5], [48].

C. Spike-Timing-Dependent Plasticity (STDP)

STDP is a local unsupervised learning rule that modifies synaptic connection strength according to the time order of pre- and postsynaptic spikes [49], [50]. The potentiation of synaptic connection (synaptic weight) occurs when the presynaptic neuron fires shortly before the postsynaptic spike while the depression of synaptic connection occurs when the postsynaptic neuron fires shortly before the presynaptic neuron.

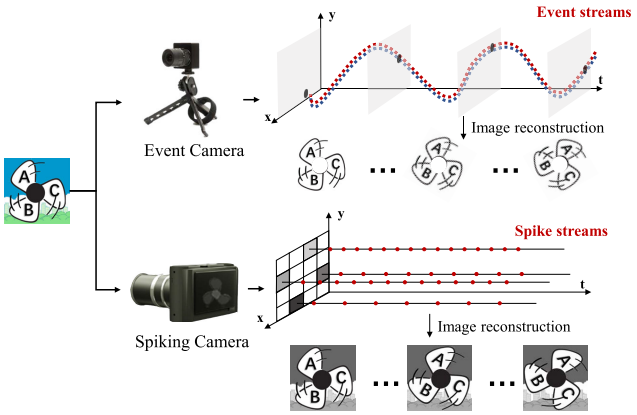

Fig. 2. An illustration of the working mechanism for event cameras and spiking cameras. Event cameras asynchronously generate events for pixels where the brightness change exceeds a certain threshold; spiking cameras generate spikes for every pixel according to the scene radiance. The stronger the scene radiance, the denser the spike streams.

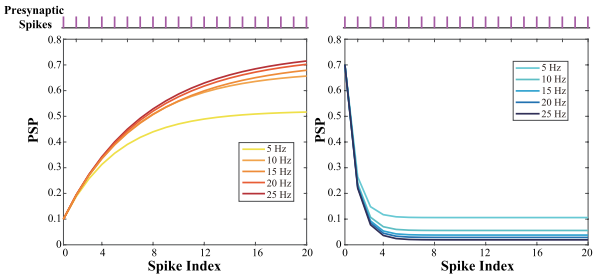

Fig. 3. The dynamic of postsynaptic potential regulated by different types of STP. Left represents short-term facilitation and Right represents short-term depression. The postsynaptic potential will converge to a stable value when receiving input spikes with fixed frequency.

D. Short-Term Plasticity

Distinct from STDP that has a long-term influence on the synaptic weight, short-term plasticity (STP) [51], [52] refers to the short-term change of postsynaptic potential that lasts tens to thousands of milliseconds. When a postsynaptic neuron receives a sequence of spikes from presynaptic neuron, the postsynaptic potential (PSP) changes according to [53]:

$$\begin{aligned} \mathrm{PSP}(t) &= A \cdot x(t) \cdot u(t), \\ \frac{dx(t)}{dt} &= \frac{1 - x(t)}{\tau_D} - u(t^-)x(t^-)\delta(t - t_{sp}), \\ \frac{du(t)}{dt} &= \frac{U - u(t)}{\tau_F} + C[1 - u(t^-)]\delta(t - t_{sp}), \end{aligned} \tag{1}$$

where A is the max amplitude of input efficacy, $x(t)$ and $u(t)$ represent the amount and release probability of neurotransmitters in the axon at time t respectively, $\delta(t)$ is the Dirac delta function. When a postsynaptic neuron receives a spike from presynaptic neuron at time t_{sp}, $x(t)$ decreases by $u(t^-)x(t^-)$ and recovers to 1 with time constant τ_D, while $u(t)$ increases by $C(1 - u(t^-))$ and recovers to baseline release probability U with time constant τ_F. Here C is a constant parameter that effects the change of $u(t)$.

Two types of STP named short-term facilitation and short-term depression have been experimentally observed. They have opposite effects on synaptic strength and can be described by Eq. (1) with different time constants τ_D and τ_F. As illustrated in Fig. 3, if the firing frequency of the presynaptic spikes are

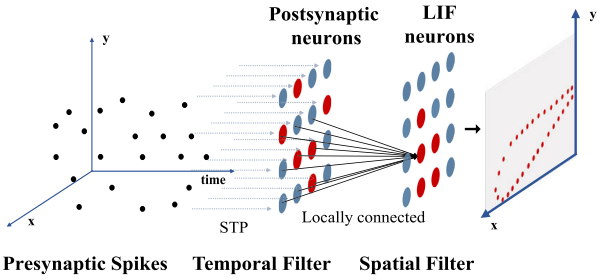

Fig. 4. Structure of the dynamic adaption module. The red dot indicates a spike of a neuron, while the blue dot indicates no spike.

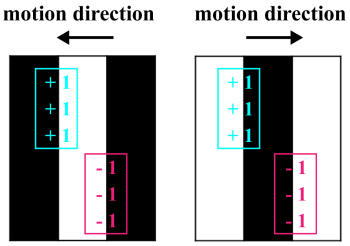

Fig. 5. Illustration of same arrangement of polarities generated by different movements.

fixed, postsynaptic potential will converge to a steady value, no matter what type of PSP.

IV. Methodology

In this section, we will introduce the proposed ODTSnet (Fig. 1) in detail. We first introduce the Dynamic Adaption module to filter redundant spikes/events based on STP and Leaky Integrate-and-Fire (LIF) neurons. Then we present the Motion Estimation module to detect motion underpinning the spikes through a multi-layer spiking neural network and STDP learning rule.

A. Dynamic Adaption Module

Owing to the sampling mechanism of neuromorphic vision sensors, there exist redundant spikes/events when estimating object motion based on the output of neuromorphic vision sensors, which will hamper the subsequent high-level visual tasks, e.g., object detection and tracking. Specifically, event cameras still suffer the severe noise problem under low-light scenarios [54], [55], [56], and spiking cameras also generate spike streams for the background/static part of the scene. To this end, we introduce the dynamic adaption based on STP and LIF neurons to filter redundant spikes and events. The structure of the dynamic adaption module is illustrated in Fig. 4, which composed of temporal filter and spatial filter.

1) Unifying the Output of Spiking Cameras and Event Cameras: Before implementing the STP-based temporal filtering and LIF-based spatial filtering, we need to unify the output of spiking cameras and event cameras. The outputs of spiking cameras are spikes with a frequency of 40,000 Hz, which can be directly inputted to postsynaptic neurons with STP for temporal filtering. For event cameras, as the outputs are sparse events with an asynchronous sampling frequency up to 10^6 Hz [3], a period or a fixed number of events are generally used for analysis. Therefore, at each time step, there might be several events at one location, which will trigger the change of STP several times. To avoid this problem, we need to find the transformation of the events cloud ξ from $\mathbb{R}^3 \to \mathbb{R}^3$ (the polarity of the events is not considered in our methods). The transform method is similar to generating a *time-image* $\mathcal{T}$ [9]. Specifically, at each timestamp, the firing time of the input event is proportional to the average timestamp of the events,

$$\mathcal{T}_{ij} = \frac{1}{C_{ij}} \sum t : t \in \xi_{ij}, \tag{2}$$

where C_{ij} is the number of spikes firing at pixel location (i, j), ξ_{ij} is the set of events firing at location (i, j) in the time window $(t, t + \delta)$. Then the events were represented as $\xi' = \{x, y, \mathcal{T}\}$. The polarity of events is not used in our model. Instead of polarity, the space-time arrangement of events is the key information for estimating motion. The polarity can be used as auxiliary information, yet it is not critical (e.g., motion segmentation without polarity [10]). For example, as illustrated in Fig. 5, the arrangement of polarities generated by leftward and rightward moving gratings are the same.

2) Temporal Filtering With STP: The 3-element tuples $\{x, y, t\}$ of spiking cameras and event cameras are first delivered to the temporal filter (Fig. 4), which is utilized to remove the redundant spikes/events according to their firing patterns. Each postsynaptic neuron corresponds to each input pixel location, and the postsynaptic potential of each neuron is modified by the temporal regularity of spikes/events. If the input spikes or events have a fixed frequency (corresponds to the background or static areas), the postsynaptic potential will converge to a stable state after several spikes arrive (Fig. 3). By taking advantage of the sensitivity of the postsynaptic potential to the release time mode of the input spike streams, the spike streams generated by the background or static areas can be filtered. For the sake of derivation, the dynamics of x and u in Eq. (1) can be rewritten as the following difference equations by integrating between spikes n and $n + 1$ [53]:

$$x_{n+1} = 1 - [1 - x_n(1 - u_n)] \exp\left(-\frac{\Delta t_n}{\tau_D}\right), \tag{3}$$

$$u_{n+1} = U + [u_n + C(1 - u_n) - U] \exp\left(-\frac{\Delta t_n}{\tau_F}\right), \tag{4}$$

where x_n and u_n denote the value of x and u between spikes n and $n + 1$, Δt_n denotes the interval between spikes n and $n + 1$. Similar to [53], we set $C = U$. If the spike rate ρ keeps constant, x and u will converge to their steady-state values $x_\infty(\rho)$ and $u_\infty(\rho)$, that is:

$$x_\infty(\rho) = 1 - [1 - x_\infty(\rho)(1 - u_\infty(\rho))] \exp\left(-\frac{1}{\rho\tau_D}\right), \tag{5}$$

$$u_\infty(\rho) = U + [u_\infty(\rho) + C(1 - u_\infty(\rho)) - U] \exp\left(-\frac{1}{\rho\tau_F}\right). \tag{6}$$

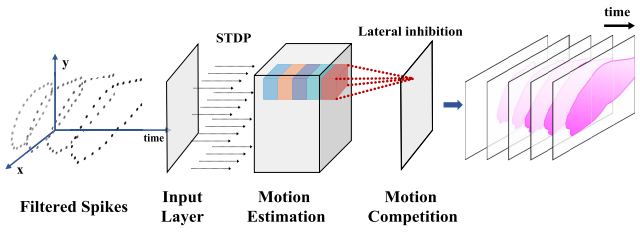

Fig. 6. Structure of the motion estimation module. Motion estimation layer includes motion neurons corresponding to different motion vectors at each pixel location, which are represented by different colors. The motion of each pixel is determined by weighted sum of the motion neurons, whose weights are adjusted by the STDP rule. The lateral inhibition is introduced to the motion competition layer to make motion patterns in the same local region more consistent.

By rearranging the Eq. (5) and Eq. (6), we get:

$$x_\infty(\rho) = \frac{1-\exp(-\frac{1}{\rho\tau_D})}{1-[1-u_\infty(\rho)]\exp(-\frac{1}{\rho\tau_D})}, \tag{7}$$

$$u_\infty(\rho) = \frac{U+(C-U)\exp(-\frac{1}{\rho\tau_F})}{1-(1-C)\exp(-\frac{1}{\rho\tau_F})}. \tag{8}$$

As the product of x and u, the postsynaptic potential PSP will also converge to a steady state:

$$\mathrm{PSP}_\infty = A \cdot x_\infty \cdot u_\infty. \tag{9}$$

Therefore, if the firing rate of input spikes varies, the STP dynamics will vary around the steady-state. State change of the input spikes/events can be detected by evaluating the STP dynamics, e.g. x, u or postsynaptic potential PSP.

Specifically, we use the facilitation dominated STP model with $\tau_D = 0.05\ s$, $\tau_F = 0.5\ s$, $U = C = 0.15$, and detect the redundant spikes based on the difference between PSP_n and PSP_{n+1}, where n is the spike index. Postsynaptic spike is fired according to the change value of x,

$$\mathcal{I}_{ij} = \begin{cases} 1, & |\mathrm{x}_{n+1} - \mathrm{x}_n| \geq \vartheta \\ 0, & |\mathrm{x}_{n+1} - \mathrm{x}_n| < \vartheta, \end{cases} \tag{10}$$

where $\mathcal{I}_{ij}$ is the indicator representing whether postsynapse fire or not, which equals to 1 when the absolute value of change exceeds the predefined threshold ϑ.

3) Spatial Filtering: In addition to filtering the input spikes/events according to their temporal regularity, spatial filtering is also introduced to remove noise. The spatial filtering is conducted implicitly by regarding the filtered spikes obtained above ($\mathcal{I}$) as afferent spikes to leaky integrate-and-fire (LIF) neurons [57]. The sub-threshold dynamics of a LIF neuron can be described as:

$$\tau_m \frac{dv(t)}{dt} = -[v(t) - v_{rest}] + RI(t), \tag{11}$$

where $v(t)$ represents the membrane potential of the LIF neuron at time t, τ_m denotes the membrane time constant, v_{rest} is the resting potential, R is the resistance value, $I(t)$ is the input current. As the spikes generated by each neuron in the temporal filter layer transmit to the eight-connected adjacent LIF neuron (Fig. 4), the input current $I(t) = \sum_{\boldsymbol{x}} \mathcal{I}_{\boldsymbol{x}}$, with $\boldsymbol{x}$ denoting the location of neuron connected to the LIF. After integrating the input spike, the state of the LIF neuron is updated according to Eq. (11) and a predefined threshold θ:

$$\lim_{\delta\to 0,\delta>0} v(t+\delta) = \begin{cases} v_{rest}, & v(t) \geq \theta \\ v(t), & v(t) < \theta. \end{cases} \tag{12}$$

When the membrane potential v exceeds a certain threshold θ, the LIF neuron will fire a spike, and the membrane potential v goes back to the resting value v_{rest}. Because the LIF neuron is locally connected to its pre-layer neurons and only be activated when the integrated voltage exceeds the threshold, random noise events/spikes will be filtered out.

4) Advantages Over Using Simpler Filter: The STP-based temporal filter implicitly and smoothly counts the spikes. We can filter out redundant spikes without choosing an interval explicitly to process spikes in advance. If it is simply filtered by counting the spike frequency, it need to pre-define a counting window. If the time window is too long, fast-motion will results in more recent timestamps covering all of the older ones and make motion indistinguishable. However, small counting window will bring about significant noise, which make it impossible to effectively filter the spikes corresponding to non-moving objects. Similarly, LIF-based spatial filter not only counts the spikes in a local area at the current moment, it also retains the influence of the input at the previous moment (the first term of Eq. (11)). Simply spatial averaging can also achieve the effect of spatial filtering, but it does not retain the continuity of timing.

B. Motion Estimation Module

We propose a multi-layer feedforward spiking network to estimate the motion of objects, which is shown in Fig. 6.

1) Motion Estimation (M1): In this layer, we set 8×4 neurons at each pixel corresponding to eight motion directions and four motion speeds. The motion pattern $\boldsymbol{m}$ of each spike is determined by the weighted sum of the motion neurons:

$$\boldsymbol{m} = \sum_{k=1}^{32} w^k \boldsymbol{v}^k, \tag{13}$$

where w^k and $\boldsymbol{v}^k$ refers to the synaptic weight and the motion vector of the k-th motion neuron, respectively. In order to distinguish this layer from the module name, we will call this layer **M1** layer in the following.

At each pixel location, there are 32 weights corresponding to different motion patterns, which will be modified according to the motion pattern of each pixel. Here, we propose an unsupervised learning algorithm to update the synaptic weights of the motion neuron by taking advantage of the STDP learning rule, the principle of which is explained in Fig. 7.

Fig. 7 shows the postsynaptic spikes of downward and leftward motion cells, respectively. The postsynaptic spikes at timestamp $t+1$ are obtained by warping filtered spikes at timestamp $t+1$ with different motion neurons. The spikes shown in the yellow and purple squares are results based on the speed of 1 pixel/timestamp downward and leftward motion neuron, separately. The synaptic weights of motion neurons are

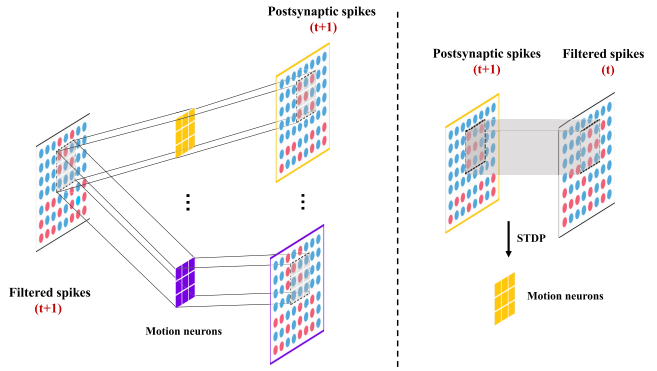

Fig. 7. Unsupervised motion learning with rectified STDP learning rule [Eq. (14)]. Red dot indicates a spike, while blue dot indicates no spike at that location. The yellow and purple squares represent downward and leftward motion neurons, respectively.

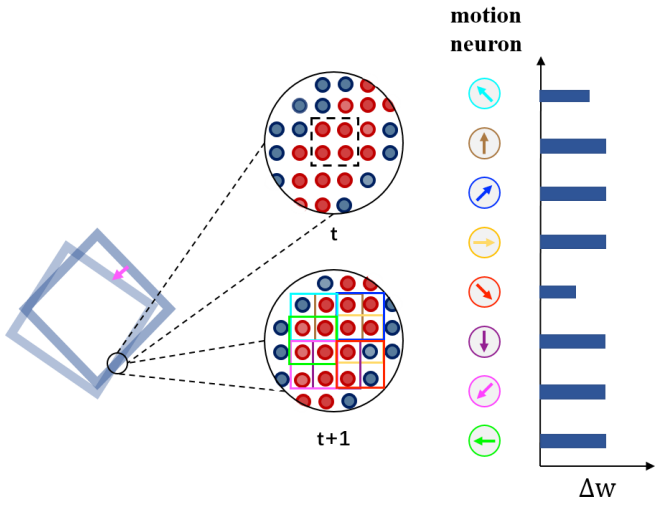

Fig. 8. Illustration of motion confusion in an edge with dense firing spikes. Except for the upper left and lower right motion neurons, the weight increments of other neurons are the same. In the M1 layer, the motion of the spikes in the square will be determined as the upper left with Eq. 13.

updated by comparing the location of all firing spikes between postsynaptic spikes at timestamp $t+1$ and filtered spikes of input layer at timestamp t. Regarding the firing spikes of input layer at timestamp t as presynaptic spikes, the synaptic weight w_i^k of the k-th motion cell at location i (we simplify the subscript as 1-dimension for brief description) is updated according to the following rectified STDP learning rule:

$$\Delta w_i^k = \eta \frac{\kappa * \boldsymbol{A_i}}{N_i}, \tag{14}$$

$$\boldsymbol{A_i} = A_+ \Gamma(t_{k,j}^{pre} - t_{k,j}^{post}) - A_-(1 - \Gamma(t_{k,j}^{pre} - t_{k,j}^{post})). \tag{15}$$

Here η is the learning rate of the synaptic weight, j refers to the neighbor neurons of neuron i. t^{pre} and t^{post} represent the timestamp of the filtered spikes and postsynaptic spikes, respectively. N_i is the number of firing spikes in the neighbor region of neuron i, which is computed by $N_i = \sum_{j=1}^{j \in Nei(i)} \Gamma(t_{k,j}^{post} - t)$. $\Gamma(x)$ is the indicator function that equals to 1 only when $x = 0$. The A_+ and A_- are parameters specifying the amount of change with long-term potentiation and long-term depression, respectively. κ refers to a convolutional kernel that has a two-dimensional Gaussian distribution with mean $\mu = 1$. After updating the weights of motion cells using Eq. 14, M1 layer obtains the motion vector (u, v) of each spikes with Eq. 13.

2) Motion Competition (MC): In the areas with dense spikes, the motion learning of each neuron would be affected by neighboring spikes, and different motion neurons may have the same synaptic weight and will cause *motion confusion* (illustrated in Fig. 8.) To correct confuse motion patterns in a local region, we introduce a motion competition layer based on the Winner-Take-All (WTA) mechanism, where the most frequent motion vector will dominate the motion pattern of the region. In other words, after extracting the motion pattern through the rectified STDP learning rule, lateral inhibition between motion neurons is introduced in a region that is larger than the neighboring size defined in Eq. (14) and Eq. (15). If there are multiple winners, all the motion cells will be inhibited. The aperture problem [58] is alleviated incidentally owing to the larger region under consideration in WTA. To find the winner motion neuron for each region, we use the motion map to quantify the motion orientation of the individual spike in the M1 and MC layers. Motion vectors are quantified into eight motion orientations $(0, \pi/4, \pi/2, \ldots, 2\pi)$. In M1 layer, motion map $M1 \equiv (p_{ij}) \in \{1, \ldots, 8\}$ for each spatial location (i, j), where the motion mode p_{ij} is obtained by:

$$p_{ij} = \lfloor \frac{\tan^{-1}(v/u) + \pi}{2\pi/8} \rfloor + 1 \tag{16}$$

With such motion map, we can calculate the number C_{ij}^k $(k \in \{1, 2, \ldots, 8\})$ of each motion pattern in each spatial neighborhood field where lateral inhibition is performed. Finding the motion pattern with the largest number of occurrences in each pixel location field $p'_{ij} = \arg\max_k C_{ij}^k$, and obtain a new motion map of MC layer $Mc \equiv (p'_{ij})$.

By comparing the motion map $M1$ and Mc, finding the position (i, j) where the motion pattern does not match, and replace the motion vector of this position from (u_{ij}, v_{ij}) to the $(u_{i'j'}, v_{i'j'})$ which has the maximum value $C_{i'j'}^{p'_{ij}}$ in the local region.

C. Object Tracking

To track different objects utilizing the motion patterns and spike pixel location without prior knowledge about the number or shape of moving objects, we use the density-based spatial clustering of applications with noise (DBSCAN) algorithm [59]. After applying the DBSCAN algorithm, we calculate the mean pixel location and average velocity of spike points that belong to the same cluster, to represent the center location $\boldsymbol{x}_c$ and velocity $\boldsymbol{x}_v$ of the detecting object. The DBSCAN distance function is constructed as follows:

$$D_{i,j}(\boldsymbol{p}, \boldsymbol{m}) = w_p \left\| \boldsymbol{p_i} - \boldsymbol{p_j} \right\| + w_m \left\| \boldsymbol{m_i} - \boldsymbol{m_j} \right\|, \tag{17}$$

where $\boldsymbol{p}$ denotes the spike position, $\boldsymbol{m}$ is the estimated motion pattern, and w_p, w_m are the weight parameters to control the

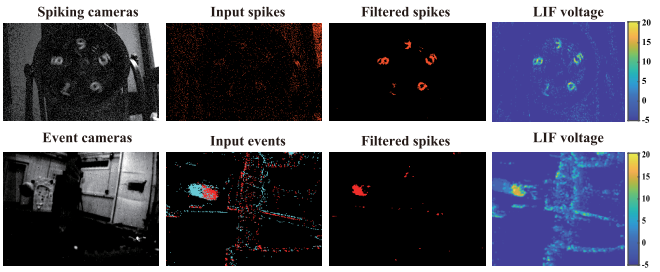

Fig. 9. Example results of the Dynamic Adaption module. The first row represents the scene of a high-speed rotating fan, which is recorded by spiking camera. The second row represents the scene of a ball that flies across a pillar, which is recorded by event camera. The scene of the spiking camera in the first column is reconstructed with TFP algorithm [62]. Images have been brightened for visualization.

effect of each individual part. Since the scale of pixel location and motion velocity are completely different, we utilize standardized Euclidean distance to balance the contribution of these variables. After obtaining the distance matrix of the spikes, we can cluster them according to the distance and a predefined neighborhood threshold ϵ.

Based on the estimated location and velocity, object assignment is solved optimally with Hungarian algorithm [60]. In order to smoothly tracking the moving objects, we apply Kalman filters to correct the detection results, in which the state vector including the estimated location $\boldsymbol{x}_c$ and velocity $\boldsymbol{x}_v$ of each cluster. Kalman filters [61] are quite functional for predicting and correcting the states (e.g., location and velocity) of objects through efficient linear processing, which have been widely used in object tracking.

In our tracking model, the state vector used in Kalman filter is $\boldsymbol{x}_k = \{\boldsymbol{x}_{c,k}, \boldsymbol{x}_{v,k}\}$, where $\boldsymbol{x}_c$ and $\boldsymbol{x}_v$ are the center position and velocity vector of each cluster obtained above, k is the processing time step. Based on the high temporal resolution of the neuromorphic vision sensors, we assume a constant velocity model as followed,

$$\boldsymbol{x}_{c,k} = \boldsymbol{x}_{c,k-1} + \boldsymbol{x}_{v,k-1}\Delta t, \tag{18}$$

$$\Delta t = t_k - t_{k-1}. \tag{19}$$

With such definition, the linear Kalman filter can be formulated as:

$$\hat{\boldsymbol{x}}_k = \mathbf{A}_k\hat{\boldsymbol{x}}_{k-1} + \boldsymbol{v}_k, \tag{20}$$

$$\boldsymbol{z}_k = \boldsymbol{H}\hat{\boldsymbol{x}}_k + \boldsymbol{w}_k, \tag{21}$$

$$\mathbf{A}_k = \begin{bmatrix} \boldsymbol{I}_{4\times4} & \Delta t \cdot \boldsymbol{I}_{4\times4} \\ 0_{4\times4} & \boldsymbol{I}_{4\times4} \end{bmatrix}, \tag{22}$$

$$\mathbf{H} = \begin{bmatrix} \boldsymbol{I}_{4\times4} & 0_{4\times4} \end{bmatrix}, \tag{23}$$

where $\mathbf{A}$ is the transition matrix, $\mathbf{H}$ is the measurement matrix, $\boldsymbol{v} \sim \mathcal{N}(0, \boldsymbol{Q})$ and $\boldsymbol{w} \sim \mathcal{N}(0, \boldsymbol{R})$ are the Gaussian distribution of process noise and measurement noise, respectively.

V. Experiments

In this section, we evaluate the performance of ODTSnet for object tracking tasks on the synthetic data and real-world data. We visualize the effect of each module, and compare the proposed model with state-of-the-art approaches. Furthermore, an ablation study is also conducted to analyze the performance of the Dynamic Adaption module and the Motion Estimation module individually.

A. Datasets

To test the generality of our models on the tracking task, we use synthetic dataset, publicly available event-based dataset, and spiking dataset, which include lots of challenging situations that will hamper the tracking of multiple high-speed objects. In addition to the event-based dataset which is public available, the synthetic dataset and the real-world spiking dataset are build specifically for this work.

1) Synthetic Dataset: The synthetic dataset includes two cross-moving characters "A" and "C", which is called "crossAC" in the following.

2) Spiking Dataset: To evaluate the performance of ODTSnet for continuously tracking multiple high-speed targets, we construct two ultra-high-speed scenarios using spiking cameras. One is called Spiking Rotating Digits Dataset (SRD), which uses a spiking camera to still shoot multiple high-speed rotating digits (2400 revolutions per minute) during the day, and the other is called Spiking Rotation Translation Character (SRTC), which uses a spiking camera to move horizontally indoors at night to shoot multiple high-speed rotating characters.

3) Event-Based Dataset: The Extreme Event Dataset (EED) [9] is collected by a DAVIS event camera in real-world scenes [2]. The EED comprises several challenging scenarios, including occluded moving objects, small fast-moving objects, and light-changing environments. Except for using the EED to evaluate the tracking performance, we also use event sequences of clockwise and counterclockwise rotating disk [63] to evaluate the performance of the proposed motion estimation modules.

Details of the datasets used in the following experiments is reported in Tab. I.

B. Parameter Selection

In the ODTSnet, hyper-parameter settings of the STP model and motion neurons are critical to the tracking results. In the following, we will discuss how to choose the parameter of the STP model and the influence of motion neuron number on the performance of the motion estimation module.

1) Time Constant of STP: The selection of time constants in the STP model is mainly to distinguish different input spike patterns. The parameters should ensure that there is a strong monotonic correspondence between different steady-state of STP (x and u) and input sequences. Monotonic analysis of Eq. (7) and Eq. (8) demonstrates that u_∞ will monotonically increase with the increase of $\rho\tau_F$, and x_∞ will monotonically decrease with the increase of $\rho\tau_D$. Fig. 10 shows the changes of u_∞ and x_∞ with $\rho\tau_F$ and $\rho\tau_D$ respectively in Eq. (7) and Eq. (8) under five types of STP when $U = C = 0.15$. The ratio of the time constant τ_D and τ_F of five types of STP are shown in Tab. II.

TABLE I
SUMMARY OF THE DATASETS

Data type	Dataset	Spatial resolution	# Seq	# Events/Spikes	Ground truth	Application
Synthetic dataset	crossAC	120x120	6	6244∼25798	trajectory, bounding box, motion vector	detection, tracking
Event-based dataset	DVSFLOW16	240x180	3	248515∼478409	optical flow	motion estimation
	EED	190x180	7	116604∼1302121	bouding box	detection
Spiking dataset	SRD	400x250	1	4889185	trajectory, bounding box	detection, tracking
	SRTC	400x250	1	1841534		

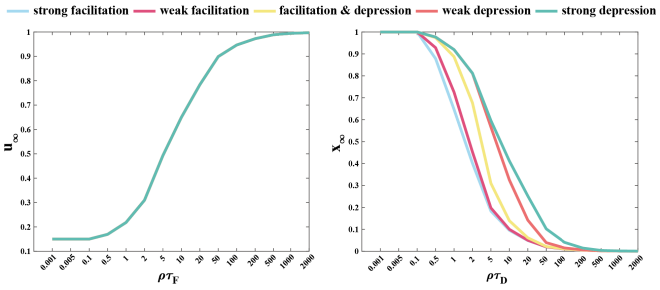

Fig. 10. The steady value of u (left) and x (right) against different $\rho\tau_F$ and $\rho\tau_D$, respectively.

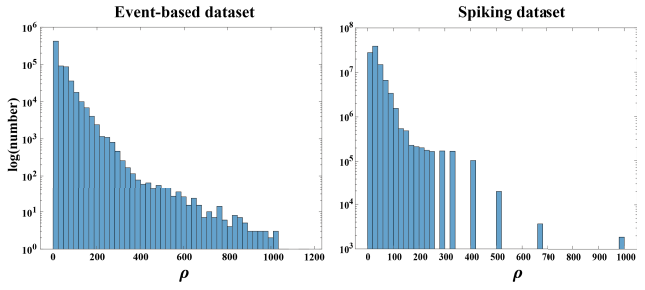

Fig. 11. Distribution of the spike firing rate of the event-based and spike-based datasets.

TABLE II
PARAMETER SETTINGS OF DIFFERENT TYPES OF SHORT-TERM PLASTICITY MODELS. T IS THE TEMPORAL RESOLUTION OF CAMERAS

STP type	τ_D (T)	τ_F (T)
Strong facilitation	0.02	1.7
Weak facilitation	0.05	0.5
Facilitation and depression	0.2	0.2
Weak depression	0.5	0.05
Strong depression	1.7	0.02

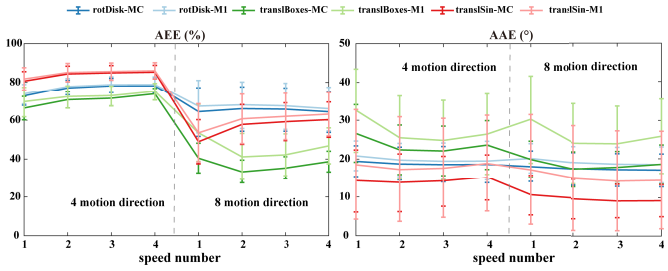

Fig. 12. The effect of motion neuron settings on performance of optical flow using event-based dataset DVSFLOW16.

As can be seen in the Fig. 10, when $\rho\tau_F$ and $\rho\tau_D$ are in the range of $[0.1, 50]$, both u_∞ and x_∞ have a relatively larger gradient, which is conducive to filtering the input with varying input frequency (that is, corresponding motion area). According to the gradient of u_∞, it is slightly slower at both very small and big firing rates, and the gradient in the middle is larger. If filtering is performed according to the difference of u, the region where the spike firing rate is relatively centered will be retained. As STP changes from enhanced to suppressed, x_∞ gradually transitions from the state of "*low firing rate high gradient, high firing rate low gradient*" to "*low firing rate low gradient, high firing rate high gradient*".

We count the firing rates for the event camera and spiking camera datasets. The distribution of the spike firing rate is shown in Fig. 11. The firing rate of the event camera and spiking camera data presents a long-tailed distribution. Therefore, in order to find out motion spikes with low spike firing rates, we choose to use the enhanced STP model, and filter spikes by detecting the change of x. On this basis, we choose $\tau_D = 0.05T$ so that changes in the range of $\rho \in [2, 1000]$ can be detected sensitively. It corresponds to the weak facilitation STP model in the Fig. 10 ($\tau_F = 0.5T$).

2) Motion Neuron in Motion Estimation Module: In the M1 layer of ODTSNet, the motion estimation is based on the weighted superposition of motion cells according to Eq. 13. We quantify the entire motion space with motor neurons. Since the phase plane is a grid-like spatial arrangement, the motion orientation can be evenly divided into four or eight different orientations. The speed setting is based on the characteristics of the high temporal resolution of the neuromorphic camera, assuming the spike shift will not exceed four pixel positions.

We make an ablation study with different motion neuron settings on the DVSFLOW16 dataset [63]. Performance of motion estimation is assessed by the Average Angular Error (AAE) and Average End-Point Errors (AEE). As shown in Fig. 12, using 8 motion orientations is obviously better than 4 motion orientations. Besides, due to the low latency advantage of the event camera, the performance of motion estimation results changes more gently with the number of speed codes increases. Therefore, to ensure the accuracy of motion estimation, we set the neurons to 8×4 motion neurons corresponding to eight motion orientations and four motion speeds.

C. Evaluation of Modules

Fig. 9 illustrates examples of the filtered spikes obtained by the dynamic adaption module, and the corresponding voltage of the LIF neurons obtained by Eq. (11) and Eq. (12).

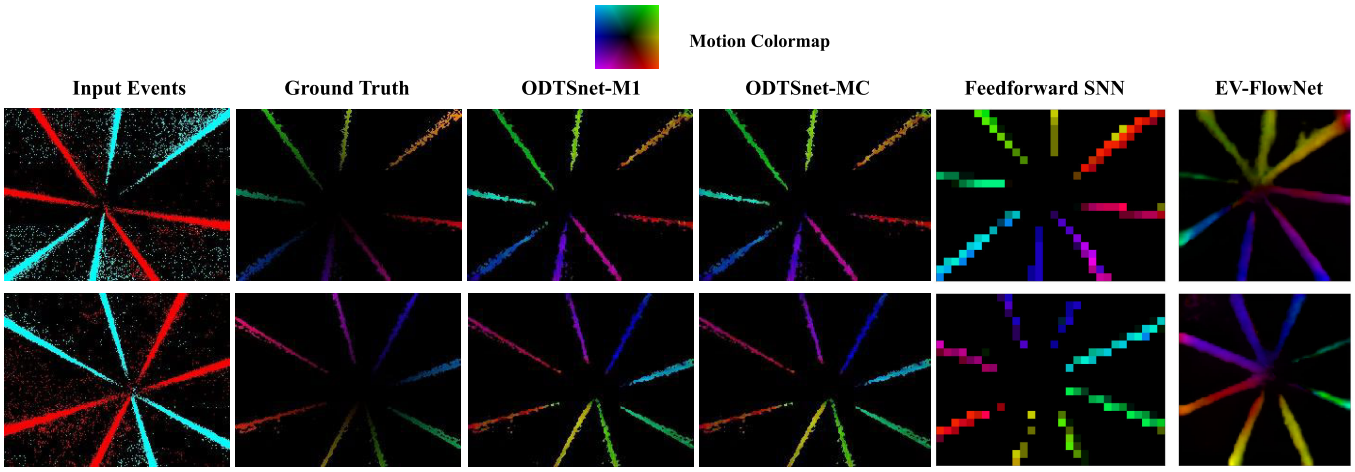

Fig. 13. Visual comparison of the motion estimation on real event streams. Motion vector is encoded as the same colormap used in EV-FlowNet.. From top to down, event sequences of clockwise and counterclockwise rotating disk. The second column shows the filtered results of the dynamic adaption module. The third column illustrates the outputs of the **M1** layer, which are rectified by the winner-take-all circuits in the **MC** layer (fourth column). The last two columns are results of feedforward SNN [21] and EV-Flownet [64] respectively.

TABLE III
QUANTITATIVE COMPARISON WITH EV-FLOWNET ON EVENT CAMERA OPTICAL FLOW DATASET DVSFLOW16 [63]

Method	Rotation Disk		Translation Box		Translation Sinusoidal	
	AEE (%)	AAE	AEE (%)	AAE	AEE (%)	AAE
EV-FlowNet [64]	99.46	36.00	29.28	7.21	56.97	6.20
ODTSnet-M1 (Ours)	65.95	18.49	46.81	25.88	63.14	14.55
ODTSnet-MC (Ours)	64.39	17.04	38.66	18.56	60.66	9.67

Obviously, the filtered spikes are clearer than the input spikes/events, making the subsequent tasks more focused on analyzing the moving object. Furthermore, we compare the proposed motion estimation module with other existing methods. EV-FlowNet [64] and the feedforward SNN [21] are state-of-the-art event-based optical flow estimation methods, which are based on conventional DNNs and SNNs, respectively. The qualitative evaluations for the rotating disk of DVSFLOW16 dataset [63] are shown in Fig. 13, where different colors referred to different motion directions are used to visualize the estimated motion (applies only to the last four columns).

These results show that the aperture problem is solved by incorporating motion competition in the local region, and the motion estimated by **MC** layer in a local region is more consistent than the estimated results of **M1** layer. As can be seen in the fourth to sixth columns, the results obtained by the ODTSnet-MC layer are comparable to those of the EV-FlowNet and feedforward SNN methods. In addition to qualitatively comparing the results of motion estimation, we also compare the quantitative evaluation results of optical flow estimation on DVSFLOW16 [63] with EV-FlowNet. As reported in Tab. III, the proposed motion estimation module can also obtain comparable quantitative results with EV-FlowNet, and the results of MC layer are better than M1 layer.

The qualitative motion estimation results of EED, SRD and synthetic data are shown in Fig. 14, Fig. 15 and Fig. 16, respectively. In Fig. 14, there is a ball that flies across a

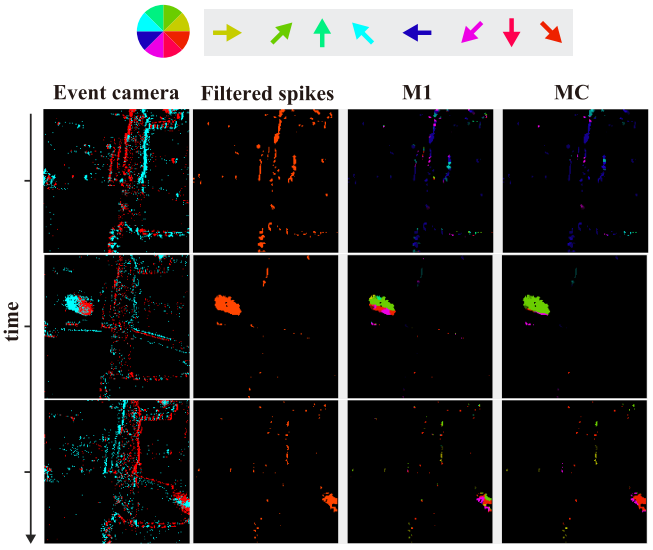

Fig. 14. Motion estimation results of the scenes in Extreme Event Datase (EED). Different colors of the color circle in the first row referred to different motion directions are used to visualize the estimated motion. The scenes are the same as the second row of Fig. 13, in which a ball flies across a pillar.

pillar. In the last two columns, the spikes generated by the ego-motion of cameras are filtered by the dynamic adaption module, leaving the spikes of the high-speed flying ball. For the spike sequences shown in Fig. 15, even though the shape of digits is more complicated than the edges of the rotating-disk, our model can still estimate the motion accurately. For example, the digit "5" in the second row of Fig. 15 is shown with magenta and bright red colors, which denotes the left-down and downward direction, respectively. The proposed model also can handle the scenario in the synthetic data named "crossAC" with overlapping translation trajectories ("C" move from left-top toward the right-down direction and "A" move from left-down to the right-top, which are shown in Fig. 16).

TABLE IV
COMPARISON WITH THE STATE-OF-THE-ART METHODS ON THE EED DATASETS

	Methods	Fast moving drone	Multiple objects	Lighting variation	What is background?	Occlusions	Average
Deep neural networks	SiamFC [41]	**100**	-	94.7	83.3	0	-
	ECO [42]	**100**	-	93.4	75	33.3	-
	SiamRPN++ [43]	94.1	-	50	83.3	16.7	-
	ATOM [44]	**100**	-	92.1	91.7	50	-
	ECO-E [42]	88.2	-	80.3	0	33.3	-
	RMRNet-TS [39]	11.8	-	6.6	0	0	-
	RMRNet [39]	**100**	-	94.7	8.3	83.3	-
	Kepple [40]	**100**	93.3	97.4	**100**	**100**	98.14
Optimized-based	SOFAS [65]	88.89	46.15	0	22.08	80	47.42
	E-MS [8]	30.7	-	32.1	36	35.3	-
	Mitrokhin [9]	92.78	87.32	84.52	89.21	90.83	88.93
	Stoffregen [10]	96.3	96.77	80.51	**100**	92.31	93.18
Spiking neural networks	ODTSNet	**100**	**97.33**	**98.7**	**100**	**100**	**99.21**

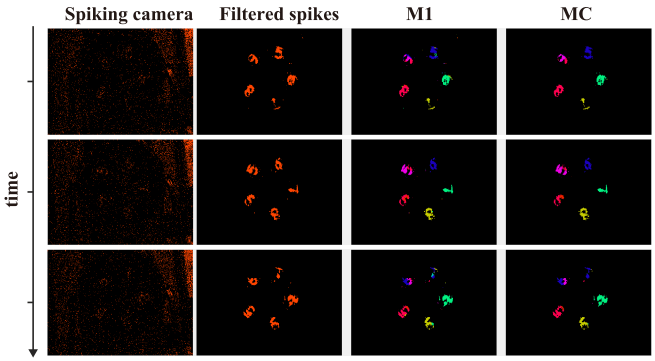

Fig. 15. Motion estimation results of the counterclockwise rotating digits sequence recorded by spiking cameras.

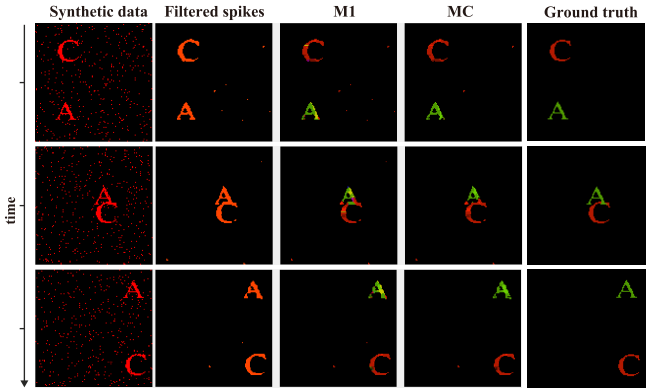

Fig. 16. Motion estimation results of the translation characters synthetic sequences "crossAC" with salt and pepper noise (noise level = 0.06).

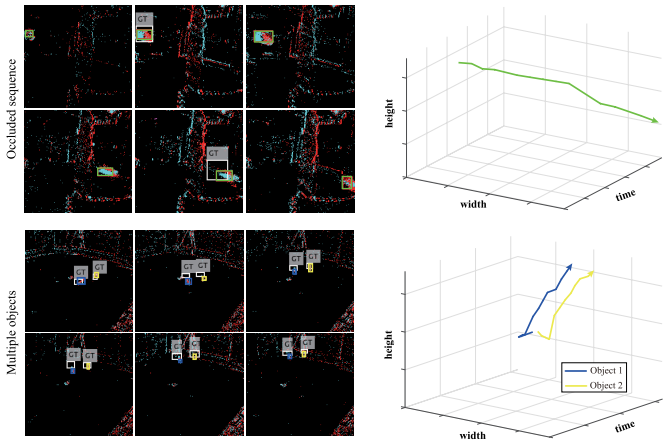

Fig. 17. Tracking examples on the EED dataset. Left and right: predicted location of the moving objects, and the corresponding trajectory.

D. Comparison With the State-of-the-Art

We perform quantitative evaluation of object detection on the event dataset EED [9] and the spiking dataset (SRD and SRTC). As the EED is recorded using a moving DAVIS camera, we need to distinguish the events generated by independent moving objects and the cameras' ego-motion. Some tracking examples on the scenes of "Occluded sequence" and "Multiple objects" are shown in Fig. 17. As the ground truths (denoted by the bounding boxes with "GT" label) are acquired by hand labeling on the RGB frames of digit cameras, the interval between two consequent GT is relatively long. However, our method can locate the moving objects during the long interval, and get smoother tracking results.

The quantitative results of object detection on the EED are reported in Tab. IV. Similar to [39], we utilize the evaluation criteria of Average Robustness (AR). The previous approaches can be divided into two categories. One is implemented with deep neural networks [39], [40], [41], [42], [43], [44], and the other is based on clustering events [8], [9], [10], [65]. From Tab. IV, it can be seen that our method outperforms the previous approaches and achieve the best performance on all sequences. We achieve an average AR of 99.21%.

In the spiking camera datasets, We annotate the movement trajectories of each moving object on the spiking camera dataset, enabling us to evaluate metrics for multi-object tracking. As shown in Fig. 18, compared to the tracking framework (SVS) proposed by Huang et al. [5], ODTSnet can robustly detect and track moving objects relatively continuously.

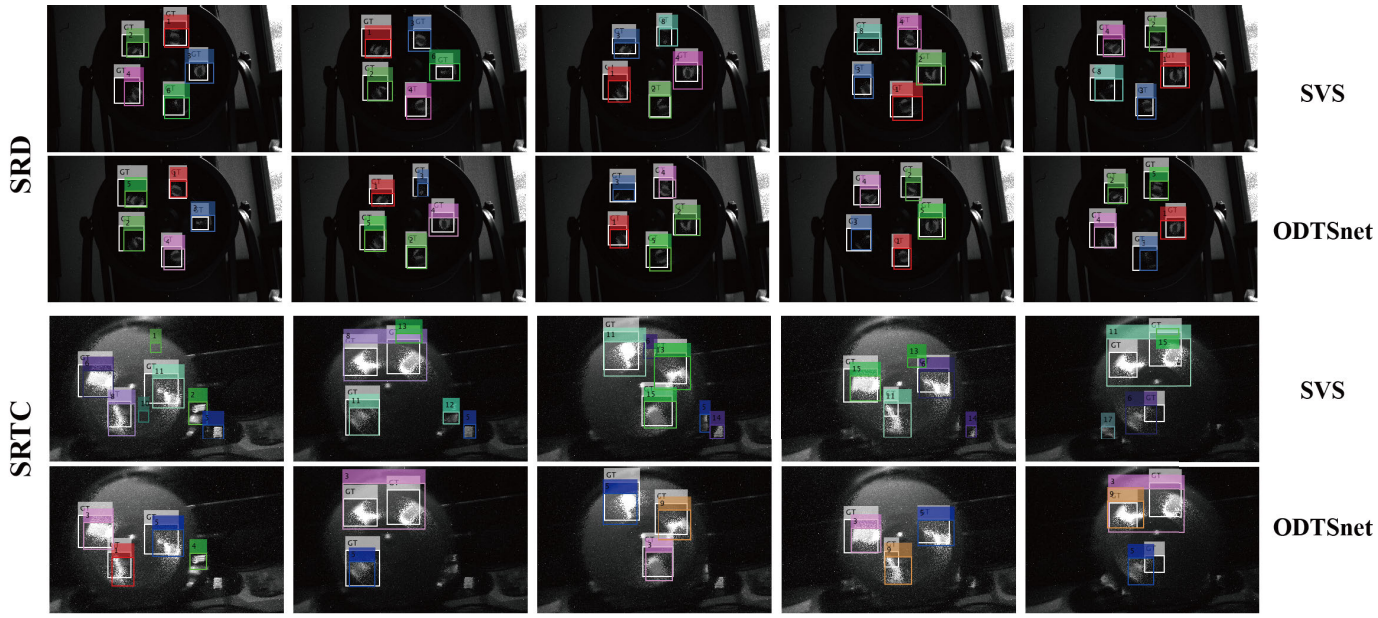

Fig. 18. Examples of tracking result on the spiking datasets using ODTSnet. The estimated bounding boxes and trajectories are encoded with different colors, and the white boxes with label "GT" denote the ground truth.

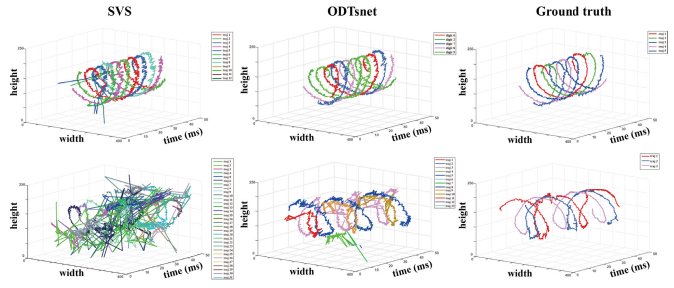

Fig. 19. Illustration of tracking results on the spiking dataset. From top to down: the predcited trajectories of moving objects of SRD and SRTC sequence.

TABLE V

QUANTITATIVE COMPARISON OF MULTI-TARGET TRACKING RESULTS IN SPIKING DATASETS

Dataset	Method	FN rate (%)	FP rate (%)	IDSW	MOTP	MOTA
SRD	SVS	4.98	6.58	9	0.7962	0.8821
	ODTSnet	2.18	2.18	0	0.8041	0.9564
SRTC	SVS	5.59	96.08	66	0.855	-0.0442
	ODTSnet	9.18	18.31	40	0.8428	0.7084

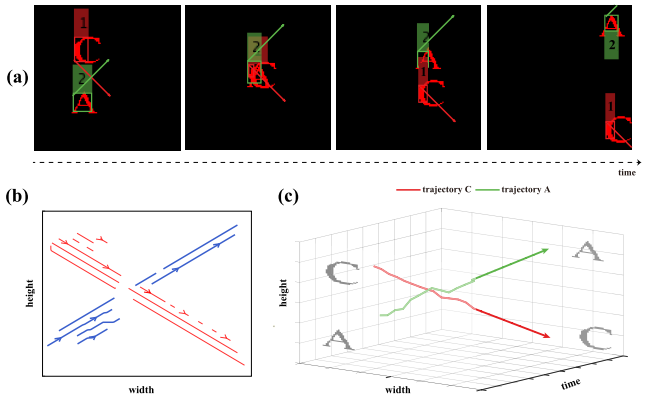

Fig. 20. Illustration of the proposed methods on the scenes of two high-speed cross moving characters, "A" and "C". **(a)** The estimated location and velocity of each moving object. The locations of the two moving characters are highlighted by bounding boxes with tracker ID "1" and "2" while the velocities are visualized with arrows. **(b)** Motion flows of the estimated motion. **(c)** The predicting trajectories of characters "A" and "C".

Fig. 19 shows the corresponding predicted moving trajectories of each object against the ground truth. It can be seen from the tracking results that both SVS and ODTSnet can smoothly predict the moving trajectories of the five rotating digits in the "SRD" sequence. For the motion trajectories of three characters in the "SRTC" sequence with both rotational and translational motions, the ODTSNet algorithm with motion estimation module can continuously estimate the trajectories of three moving objects, but the SVS algorithm will produce a large number of misjudgments trajectory. This suggests that object clustering relying only on spatial location is susceptible to background spikes. Quantitative comparison of multi-target tracking results is reported in Tab. V.

Compared to the tracking results on SRD sequence, ODTSnet produces many false positives and missed targets on the SRTC sequence. The main reason is that DBSCAN clustering algorithm is sensitive to parameter Settings. Once there are noise points between two objects, DBSCAN algorithm can easily treat these two objects as one object. When the predicted tracking box has a large deviation and the object is identified as a new category, the Kalman filter will assign a new tracker to it, resulting in the change of object ID.

The tracking results of the dataset "crossAC" are shown in Fig. 20. Even if the characters "A" and "C" are occluded during the movement, our algorithm still can track them smoothly.

TABLE VI
ABLATION STUDIES OF OUR PROPOSED DYNAMIC ADAPTION MODULE AND MOTION ESTIMATION MODULE OVER THE EVENT-BASED DATASETS (EED) AND SPIKING DATASETS (SRD AND SRTC)

Module		ODTSnet1	ODTSnet2	ODTSnet3	ODTSnet4
Dynamic adaption		✗	✓	✗	✓
Motion estimation		✗	✗	✓	✓
fast moving drone	AR	11.11	96.27	96.3	**100**
lighting variation	AR	7.79	54.55	94.81	**98.7**
what is background?	AR	7.14	24.43	7.14	**100**
occlusions	AR	0	53.85	92.31	**100**
SRD	AR	0	84.68	5.31	**97.82**
	MOTA	-100	72.72	-30.79	**95.64**
SRTC	AR	0	38.33	9.47	**81.79**
	MOTA	-100	18.94	-24.57	**70.84**

TABLE VII
ROBUSTNESS OF THE PROPOSED MODULES AGAINST NOISE INPUTS

Module		ODTSnet1		ODTSnet2		ODTSnet3		ODTSnet4	
Dynamic adaption		✗		✓		✗		✓	
Motion estimation		✗		✗		✓		✓	
Data	noise level	AR	MOTA	AR	MOTA	AR	MOTA	AR	MOTA
CrossAC	0.02	28	0	96	74	84	48	**100**	**84**
	0.04	8	0	98	78	64	20	**100**	**86**
	0.06	8	0	**100**	62	56	4	**100**	**78**
	0.08	0	0	96	50	50	0	**96**	**76**
	0.1	8	0	60	8	48	0	**86**	**56**

E. Ablation Study

To evaluate each component of the proposed ODTSnet, we split and recombine the modules to build different models. The performance on the EED and SRD datasets is shown in Tab. VI. For "ODTSnet1" and "ODTSnet2" that do not use the motion estimation module, the state vector of the Kalman filters only contains the center position of each cluster.

Except for using the AR metric, we also use the Multi-object tracking accuracy (MOTA) [66] to measure continuity of the trajectories obtained by the models. As can be seen from Tab. VI, without the dynamic adaption and motion estimation module, the model "ODTSnet1" cannot work. By adding the dynamic adaption module ("ODTSnet2") or motion estimation module ("ODTSnet3"), the multi-object tracking performance improves on all sequences. The model with both the dynamic adaption and motion estimation modules ("ODTSnet4") outperforms either the dynamic adaption module only ("ODTSnet2") or the motion estimation module only ("ODTSnet3"), demonstrating the complementariness of the two proposed modules.

As reported in the Tab. VI, the "ODTSnet2" with the dynamic adaption module can achieve better performance than the models "ODTSnet1" and "ODTSnet3". However, it only uses the spatial information to cluster the results and corrects the tracking results with the typical constant velocity Kalman filter. Thus, the tracking results of the model "ODTSnet2" are more sensitive to the wrong detection results, resulting in the miss of objects or switch of tracking ID (shown in Fig. 21). In addition, on the SRTC dataset, the detection and tracking performance of ODTSnet4 is significantly better than that of ODTSnet2. This means that it is effective to use motion estimation module to assist clustering when the camera has ego-motion.

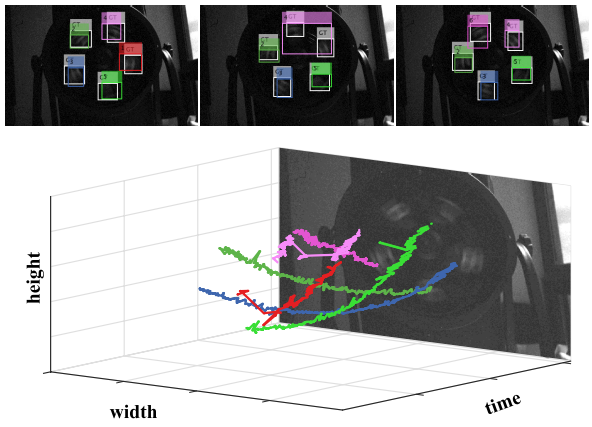

Fig. 21. Qualitative results of tracking the multiple high-speed rotating characters with "ODTSnet2.".

Robustness Against Noise: To further test the robustness of the models, we add different levels of *salt and pepper* noise into the synthetic data "crossAC" and evaluate the tracking performance. The results are reported in Tab. VII. It can be found that the full model "ODTSnet4" is the most robust network. On the contrary, the model "ODTSnet1" cannot work as the dynamic adaption and motion estimation modules are removed. With the dynamic adaption module to filter the noise, the "ODTSnet2" can achieve comparable performance when the noise level is low. However, as the two characters will shade each other at some time, clustering objects without the motion information make the characters indistinguishable and lead some targets to be missed. The tracking performance of "ODTSnet3" is unstable because of the noise. Except for the model "ODTSnet4", all the other models fail to track the cross-moving characters when the noise level gradually increases.

F. Limitations

The main innovation of this paper is to propose a tracking framework that can be used for both event cameras and spiking cameras. In order to achieve robust tracking of multiple targets in various scenarios, we propose a Dynamic Adaption filtering module based on short-term plasticity and LIF neurons, and a Motion Estimation module based on STDP and WTA circuits. These two modules require no training and can adjust neuron/synaptic states online based on incoming spikes/event streams. In the proposed Dynamic Adaption and Motion Estimation modules, all processes between each location can be parallelized, and the time complexity is $O(1)$.

However, in the tracking module, we still use the traditional tracking algorithm (DBSCAN, and Kalman filter). The total average time complexity of the DBSCAN clustering algorithm is $O(n \log n)$ (the worst time complexity is $O(n^2)$), and the space complexity is $O(n)$, where n refers to the number of points to be clustered. The time and space complexity of the Kalman filter is $O(K)$, where K represents the number of trackers. Therefore, the running delay of the whole framework is mainly proportional to the

number of spikes that need to be clustered after filtering. And the DBSCAN algorithm is also sensitive to parameter settings (to ensure fairness, the parameters used in all ablation experiments are the same). Therefore, in the current overall detection and tracking framework, we mainly ensure the performance of detection and tracking, and the low-latency advantages of neuromorphic cameras have not been reflected yet. In future work, we will replace the current traditional algorithms with a tracking module based on spiking neurons to realize the tracking framework of the full-spiking neuron pathway.

VI. Conclusion

In this paper, we present a novel bio-inspired unsupervised motion estimation model for neuromorphic vision sensors. We propose a dynamic adaption module based on short-term plasticity, enabling eliminate the noise and redundant signals by utilizing the spatiotemporal information of input events/spikes. Further, the spiking neural networks-based motion estimation module is introduced to perceive the motion of various complex scenes.

Although the spike-based model can process events and spikes asynchronously, the current implementation still uses the form of discrete and synchronous processing of each spike on the traditional processor. Therefore, at the beginning of our model, we need to reduce the event/spike streams into a "frame" -like expression. The output of the spiking camera is a two-dimensional spike array at each timestamp so that it can be directly fed into the network. However, the output of the event camera is a sparse stream of events, and the time resolution is usually above the millisecond level. It is time-consuming to update the state of the model according to the temporal resolution of the event camera. Besides, the spatiotemporal information carried by the discrete event points is minimal, which is not conducive to the downstream vision task. All the existing event algorithms use sliding windows with constant events or duration to convert the event flow into a frame-like expression, and our approach here is similar.

The EV-Flownet and feedforward SNN can also provide motion information to segment moving objects and backgrounds, but their respective limitations make them unable to detect objects in the complex task of EED. EV-Flownet unable to predict optical flow in fast-motion scenes due to the overlapping of dense event streams. The feedforward SNN will downsample the input when predicting optical flow, which makes small objects cannot be detected. Therefore, not every motion-estimation method can well assist the object detection task, especially in the complex and extreme scenes in the EED dataset. The experiment results show that without iterative optimizing or training with labeled data, the proposed model can track multi-objects smoothly and outperform the state-of-the-art algorithms on challenging datasets.

References

[1] G. Gallego et al., "Event-based vision: A survey," 2019, *arXiv:1904.08405*.

[2] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck, "A 240×180 130 db 3 μs latency global shutter spatiotemporal vision sensor," *IEEE J. Solid-State Circuits*, vol. 49, no. 10, pp. 2333–2341, Sep. 2014.

[3] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128×128 120 dB 15 μs latency asynchronous temporal contrast vision sensor," *IEEE J. Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, Jan. 2008.

[4] T. Delbruck, B. Linares-Barranco, E. Culurciello, and C. Posch, "Activity-driven, event-based vision sensors," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2010, pp. 2426–2429.

[5] T. Huang et al., "1000x faster camera and machine vision with ordinary devices," 2022, *arXiv:2201.09302*.

[6] S. Dong, T. Huang, and Y. Tian, "Spike camera and its coding methods," in *Proc. Data Compress. Conf. (DCC)*, Apr. 2017, p. 437.

[7] S. Dong, L. Zhu, D. Xu, Y. Tian, and T. Huang, "An efficient coding method for spike camera using inter-spike intervals," in *Proc. Data Compress. Conf. (DCC)*, Mar. 2019, p. 568.

[8] F. Barranco, C. Fermuller, and E. Ros, "Real-time clustering and multi-target tracking using event-based sensors," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 5764–5769.

[9] A. Mitrokhin, C. Fermüller, C. Parameshwara, and Y. Aloimonos, "Event-based moving object detection and tracking," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1–9.

[10] T. Stoffregen, G. Gallego, T. Drummond, L. Kleeman, and D. Scaramuzza, "Event-based motion segmentation by motion compensation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7244–7253.

[11] Y. Zhou, G. Gallego, X. Lu, S. Liu, and S. Shen, "Event-based motion segmentation with spatio-temporal graph cuts," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Nov. 12, 2021, doi: 10.1109/TNNLS.2021.3124580.

[12] D. Liu, A. Parra, and T.-J. Chin, "Globally optimal contrast maximisation for event-based motion estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6349–6358.

[13] W. Maass, "Networks of spiking neurons: The third generation of neural network models," *Neural Netw.*, vol. 10, no. 9, pp. 1659–1671, 1997.

[14] Y. LeCun, Y. Bengio, and G. E. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, Dec. 2015.

[15] Q. Yu, R. Yan, H. Tang, K. C. Tan, and H. Li, "A spiking neural network system for robust sequence recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 3, pp. 621–635, Apr. 2015.

[16] 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 Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3649–3657, Sep. 2020.

[17] Q. Yu, S. Song, C. Ma, J. Wei, S. Chen, and K. C. Tan, "Temporal encoding and multispike learning framework for efficient recognition of visual patterns," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 8, pp. 3387–3399, Aug. 2022.

[18] F. Akopyan et al., "TrueNorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 10, pp. 1537–1557, Oct. 2015.

[19] M. Davies et al., "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, Jan. 2018.

[20] G. Orchard, R. Benosman, R. Etienne-Cummings, and N. V. Thakor, "A spiking neural network architecture for visual motion estimation," in *Proc. IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, Oct. 2013, pp. 298–301.

[21] F. Paredes-Valles, K. Y. W. Scheper, and G. C. H. E. D. Croon, "Unsupervised learning of a hierarchical spiking neural network for optical flow estimation: From events to global motion perception," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 2051–2064, Aug. 2020.

[22] M. Gehrig, S. B. Shrestha, D. Mouritzen, and D. Scaramuzza, "Event-based angular velocity regression with spiking networks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 4195–4202.

[23] C. Lee, A. K. Kosta, A. Z. Zhu, K. Chaney, K. Daniilidis, and K. Roy, "Spike-FlowNet: Event-based optical flow estimation with energy-efficient hybrid neural networks," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 366–382.

[24] C. M. Parameshwara, S. Li, C. Fermuller, N. J. Sanket, M. S. Evanusa, and Y. Aloimonos, "SpikeMS: Deep spiking neural network for motion segmentation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 3414–3420.

[25] T. Delbruck and P. Lichtsteiner, "Fast sensory motor control based on event-based hybrid neuromorphic-procedural system," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2007, pp. 845–848.

[26] T. Delbrück and M. Lang, "Robotic goalie with 3 ms reaction time at 4% CPU load using event-based dynamic vision sensor," *Front. Neurosci.*, vol. 7, p. 223, Nov. 2013.

[27] M. Litzenberger et al., "Estimation of vehicle speed based on asynchronous data from a silicon retina optical sensor," in *Proc. IEEE Intell. Transp. Syst. Conf.*, Jun. 2006, pp. 653–658.

[28] M. Litzenberger et al., "Embedded vision system for real-time object tracking using an asynchronous transient vision sensor," in *Proc. IEEE 12th Digit. Signal Process. Workshop, 4th IEEE Signal Process. Educ. Workshop*, Sep. 2006, pp. 173–178.

[29] D. Drazen, P. Lichtsteiner, P. Häfliger, T. Delbrück, and A. Jensen, "Toward real-time particle tracking using an event-based dynamic vision sensor," *Experim. Fluids*, vol. 51, no. 5, pp. 1465–1469, 2011.

[30] Z. Ni, C. Pacoret, R. Benosman, S. Ieng, and S. Regnier, "Asynchronous event-based high speed vision for microparticle tracking," *J. Microsc.*, vol. 245, no. 3, pp. 236–244, 2012.

[31] Z. Ni, S.-H. Ieng, C. Posch, S. Régnier, and R. Benosman, "Visual tracking using neuromorphic asynchronous event-based cameras," *Neural Comput.*, vol. 27, no. 4, pp. 925–953, 2015.

[32] X. Lagorce, C. Meyer, S.-H. Ieng, D. Filliat, and R. Benosman, "Asynchronous event-based multikernel algorithm for high-speed visual features tracking," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 8, pp. 1710–1720, Aug. 2014.

[33] D. R. Valeiras, X. Lagorce, X. Clady, C. Bartolozzi, S.-H. Ieng, and R. Benosman, "An asynchronous neuromorphic event-driven visual part-based shape tracking," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 12, pp. 3045–3059, Mar. 2015.

[34] A. Glover and C. Bartolozzi, "Robust visual tracking with a freely-moving event camera," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 3769–3776.

[35] G. Gallego, H. Rebecq, and D. Scaramuzza, "A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3867–3876.

[36] N. J. Sanket et al., "EVDodgeNet: Deep dynamic obstacle dodging with event cameras," 2019, *arXiv:1906.02919*.

[37] A. Mitrokhin, C. Ye, C. Fermuller, Y. Aloimonos, and T. Delbruck, "EV-IMO: Motion segmentation dataset and learning pipeline for event cameras," 2019, *arXiv:1903.07520*.

[38] C. M. Parameshwara, N. J. Sanket, C. Deep Singh, C. Fermüller, and Y. Aloimonos, "MOMS with Events: Multi-object motion segmentation with monocular event cameras," 2020, *arXiv:2006.06158*.

[39] H. Chen, D. Suter, Q. Wu, and H. Wang, "End-to-end learning of object motion estimation from retinal events for event-based object tracking," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 10534–10541.

[40] D. R. Kepple, D. Lee, C. Prepsius, V. Isler, and I. Memming, "Jointly learning visual motion and confidence from local patches in event cameras," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 500–516.

[41] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional Siamese networks for object tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 850–865.

[42] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ECO: Efficient convolution operators for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6638–6646.

[43] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "SiamRPN++: Evolution of Siamese visual tracking with very deep networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4282–4291.

[44] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ATOM: Accurate tracking by overlap maximization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4660–4669.

[45] J. H. Lee, T. Delbruck, and M. Pfeiffer, "Training deep spiking neural networks using backpropagation," *Frontiers Neurosci.*, vol. 10, p. 508, Nov. 2016.

[46] F. Zenke and S. Ganguli, "SuperSpike: Supervised learning in multilayer spiking neural networks," *Neural Comput.*, vol. 30, no. 6, pp. 1514–1541, Jun. 2018.

[47] S. B. Shrestha and G. Orchard, "SLAYER: Spike layer error reassignment in time," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1412–1421.

[48] Y. Zheng, L. Zheng, Z. Yu, B. Shi, Y. Tian, and T. Huang, "High-speed image reconstruction through short-term plasticity for spiking cameras," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 6358–6367.

[49] S. Song, K. D. Miller, and L. F. Abbott, "Competitive Hebbian learning through spike-timing-dependent synaptic plasticity," *Nature Neurosci.*, vol. 3, no. 9, pp. 919–926, 2000.

[50] G. Bi and M.-M. Poo, "Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and post-synaptic cell type," *J. Neurosci., Off. J. Soc. Neurosci.*, vol. 18, pp. 10464–10472, Jan. 1999.

[51] M. V. Tsodyks and H. Markram, "The neural code between neocortical pyramidal neurons depends on neurotransmitter release probability," *Proc. Nat. Acad. Sci. USA*, vol. 94, no. 2, pp. 719–723, 1997.

[52] M. Tsodyks, K. Pawelzik, and H. Markram, "Neural networks with dynamic synapses," *Neural Comput.*, vol. 10, no. 4, pp. 821–835, 1998.

[53] R. P. Costa, P. J. Sjöström, and M. C. W. van Rossum, "Probabilistic inference of short-term synaptic plasticity in neocortical microcircuits," *Frontiers Comput. Neurosci.*, vol. 7, p. 75, Jun. 2013.

[54] Y. Hu, S.-C. Liu, and T. Delbruck, "v2e: From video frames to realistic DVS events," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 1312–1321.

[55] 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 *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1609–1619.

[56] Z. Ding et al., "Spatio-temporal recurrent networks for event-based optical flow estimation," in *Proc. AAAI Conf. Artif. Intell.*, 2022, vol. 36, no. 1, pp. 525–533.

[57] W. Gerstner, W. M. Kistler, R. Naud, and L. Paninski, *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition*. Cambridge, U.K.: Cambridge Univ. Press, 2014.

[58] K. Nakayama and G. H. Silverman, "The aperture problem—I. Perception of nonrigidity and motion direction in translating sinusoidal lines," *Vis. Res.*, vol. 28, no. 6, pp. 739–746, Jan. 1988.

[59] M. Ester et al., "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discovery Data Mining*, 1996, vol. 96, no. 34, pp. 226–231.

[60] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Res. Logist.*, vol. 2, nos. 1–2, pp. 83–97, Mar. 1995.

[61] X. Li, K. Wang, W. Wang, and Y. Li, "A multiple object tracking method using Kalman filter," in *Proc. IEEE Int. Conf. Inf. Autom.*, Jun. 2010, pp. 1862–1866.

[62] L. Zhu, S. Dong, T. Huang, and Y. Tian, "A retina-inspired sampling method for visual texture reconstruction," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2019, pp. 1432–1437.

[63] B. Rueckauer and T. Delbruck, "Evaluation of event-based algorithms for optical flow with ground-truth from inertial measurement sensor," *Frontiers Neurosci.*, vol. 10, p. 176, Sep. 2016.

[64] A. Zihao Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "EV-FlowNet: Self-supervised optical flow estimation for event-based cameras," 2018, *arXiv:1802.06898*.

[65] T. Stoffregen and L. Kleeman, "Simultaneous optical flow and segmentation (SOFAS) using dynamic vision sensor," 2018, *arXiv:1805.12326*.

[66] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *EURASIP J. Image Video Process.*, vol. 2008, pp. 1–10, May 2008.

Yajing Zheng received the B.S. degree from Sichuan University, Sichuan, China, in 2017, and the Ph.D. degree from the School of Computer Science, Peking University, Beijing, China, in 2022. Her research interests include neuroscience, brain-inspired computing, machine learning, and spiking neural networks.

Zhaofei Yu (Member, IEEE) received the B.S. degree from the Hong Shen Honors School, College of Optoelectronic Engineering, Chongqing University, Chongqing, China, in 2012, and the Ph.D. degree from the Automation Department, Tsinghua University, Beijing, China, in 2017. He is currently an Assistant Professor with the Institute for Artificial Intelligence, Peking University, Beijing. His current research interests include artificial intelligence, brain-inspired computing, and computational neuroscience.

Song Wang (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana–Champaign (UIUC), Champaign, IL, USA, in 2002. He was a Research Assistant with the Image Formation and Processing Group, Beckman Institute, UIUC, from 1998 to 2002. In 2002, he joined the Department of Computer Science and Engineering, University of South Carolina, Columbia, SC, USA, where he is currently a Professor. His current research interests include computer vision, image processing, and machine learning. He is also serving as the Publicity Chair/the Web Portal Chair of the Technical Committee of Pattern Analysis and Machine Intelligence of the IEEE Computer Society and an Associate Editor for IEEE TRANSACTION ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTION ON MULTIMEDIA, and *Pattern Recognition Letters*. He is a member of the IEEE Computer Society.

Tiejun Huang (Senior Member, IEEE) received the bachelor's and master's degrees in computer science from the Wuhan University of Technology, Wuhan, in 1992 and 1995, respectively, and the Ph.D. degree in pattern recognition and intelligent system from the Huazhong (Central China) University of Science and Technology, Wuhan, China, in 1998. He is currently a Professor with the School of Electronic Engineering and Computer Science, Peking University, Beijing, China, where he is also the Director of the Institute for Digital Media Technology. His research interests include video coding, image understanding, digital right management, and digital library. He has authored or coauthored over 100 peer-reviewed papers and three books. He is a member of the Board of Director for Digital Media Project, the Advisory Board of the IEEE Computing Society, and the Board of the Chinese Institute of Electronics.