Capture the Moment: High-Speed Imaging With Spiking Cameras Through Short-Term Plasticity

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

***Abstract*—High-speed imaging can help us understand some phenomena that are too fast to be captured by our eyes. Although ultra-fast frame-based cameras (e.g., Phantom) can record millions of fps at reduced resolution, they are too expensive to be widely used. Recently, a retina-inspired vision sensor, spiking camera, has been developed to record external information at 40, 000 Hz. The spiking camera uses the asynchronous binary spike streams to represent visual information. Despite this, how to reconstruct dynamic scenes from asynchronous spikes remains challenging. In this paper, we introduce novel high-speed image reconstruction models based on the short-term plasticity (STP) mechanism of the brain, termed TFSTP and TFMDSTP. We first derive the relationship between states of STP and spike patterns. Then, in TFSTP, by setting up the STP model at each pixel, the scene radiance can be inferred by the states of the models. In TFMDSTP, we use the STP to distinguish the moving and stationary regions, and then use two sets of STP models to reconstruct them respectively. In addition, we present a strategy for correcting error spikes. Experimental results show that the STP-based reconstruction methods can effectively reduce noise with less computing time, and achieve the best performances on both real-world and simulated datasets.**

***Index Terms*—High-speed reconstruction, motion-dependent, short-term plasticity, spiking cameras.**

I. Introduction

CAPTURING the moment when time flies is not just about creating amazing pictures, but also about extending our knowledge of the world. There are boundless potential applications for high-speed imaging, such as recording the fast-changing processes in physics experiments, studying rapidly moving particles in the chemical reaction, and researching risk in autonomous driving. However, the traditional digital camera records scenes with a constant shutter speed (e.g., 30 fps), which loses much visual information and suffers from motion blur. The most effective way to capture high-speed scenes with traditional digital cameras is to reduce exposure time or reduce the spatial resolution. The maximum imaging speed can reach hundreds of thousands of frames per second for some digital cameras (e.g., Sony IMX400). Despite these, the speed still cannot keep up with the changes in many nanoseconds or microseconds high-speed scenes, not to mention that most of these are actually achieved through interpolation. This has led to the development of ultra-high-speed cameras, such as the Phantom [1], [2], [3], which can record millions of frames per second. However, enormous memory demands are needed to store these images. Moreover, high-speed cameras require specialized sensors that are highly expensive, which cannot be widely used. In addition to ultra-high-speed cameras, some new devices, like single-photon avalanche diodes (SPAD) camera [4], [5]), and the corresponding algorithms [6], [7], [8], [9] were proposed to improve the imaging speed. These technologies can record tens of billions [5] or even trillions [8] of frames per second, yet are cumbersome to operate, and are vulnerable to produce blurred or glow-out images if improperly operated.

Neuromorphic vision sensors have attracted much attention in recent years [10]. Unlike traditional frame-based cameras, which use a global shutter to control the exposure time of all pixels, neuromorphic vision sensors mimic the sampling mechanism of the retina and asynchronously generate spikes/events to represent the radiance change of each pixel. A commonly used neuromorphic vision sensors are dynamic vision sensors (DVS) (also called event cameras), in which events are generated only when the brightness change exceeds a certain threshold [11], [12], [13]. Event cameras have distinctive advantages over traditional frame cameras such as low-latency, low power consumption, and high dynamic range (HDR), which have been applied to optical flow estimation [14], [15], HDR imaging [16], [17], and high frame-rate video synthesis [18], [19]. Despite these, it is difficult for an event camera to reconstruct textures in scenes as visual information of the static scenes is lost. Inspired by the sampling mechanism of primate fovea located in the retina center [20], [21], another retina-inspired camera named

Manuscript received 26 January 2022; revised 18 November 2022; accepted 14 January 2023. Date of publication 18 January 2023; date of current version 5 June 2023. This work was supported in part by the National Natural Science Foundation of China under Grants 62176003, 62088102, and U1803264 and in part by China Postdoctoral Science Foundation under Grant 2022M720238. Recommended for acceptance by W. Heidrich. *(Corresponding author: Zhaofei Yu.)*

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

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).

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).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TPAMI.2023.3237856.

Digital Object Identifier 10.1109/TPAMI.2023.3237856

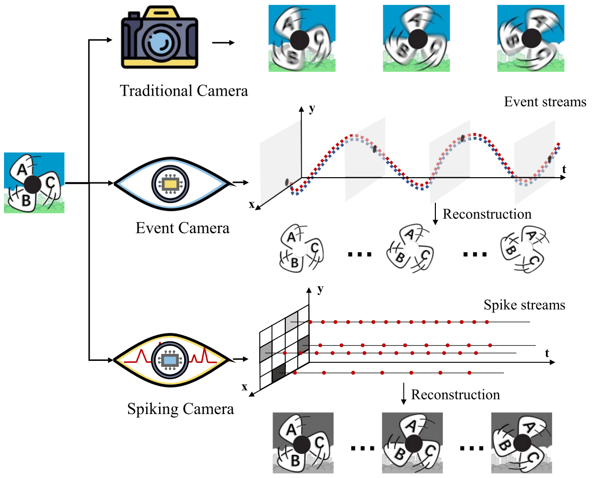

Fig. 1. Illustration of working mechanism for traditional cameras, event cameras, and spiking cameras. Traditional cameras acquire images according to the constant frame rate, event cameras generate asynchronous events for all pixels when brightness changes exceed a certain threshold, and spiking cameras continuously capture photons and generate asynchronous spike for all pixels when the accumulated intensity reaches a predefined threshold.

spiking camera has been developed in recent years [22], [23]. Each pixel of the spiking camera continuously captures photons and generates a spike when the accumulated intensity reaches a predefined threshold. An intuitive illustration for traditional cameras, event cameras, and spiking cameras is shown in Fig. 1. In spiking cameras, the pixels sensed different scene radiance fire spikes with different frequencies; the stronger the radiance, the faster the spikes fire. Compared with event cameras, the spiking camera retains *high-speed spatio-temporal* information for both *moving and static* objects, which is ready-to-use for scene reconstruction.

Recently, some scene reconstruction methods have been proposed by estimating the firing frequency of each pixel, as the photosensitive units that receive different scene radiance will trigger spikes with different frequencies [24], [25], [26], [27]. However, these methods require a predefined length of the time window, which often suffer from the problems of motion blur and low image contrast if the window length is inappropriate [24]. Besides, complex optimization algorithms are utilized to separate the motion and static areas or to align motion pixels so as to make it impossible to reconstruct images in real-time [25], [26]. Therefore, how to estimate firing frequency of each pixel without a predefined time window and reconstruct the texture with high image quality and low latency is still unclear. In addition, due to the influence of dark current and discrete readout spikes, the spiking cameras contain much noise, which will further affect the quality of reconstructed images.

In order to take advantage of the low-latency and low-power consumption of spiking cameras, and obtain high-quality ultra-high-speed reconstructed images without introducing too much computational complexity, we introduce the short-term plasticity (STP) mechanism of the brain [28], [29]. By employing the output spiking streams as the input of spiking neural networks with STP [30], we derive the relationship between the time-varying firing frequency of each pixel and the dynamics of the postsynaptic neuron, and further infer the scene radiance and the pixel value of the reconstructed images. This method is referred to as texture from STP (TFSTP). We analyze the influence of different parameters in STP on the model dynamics, including convergence time and convergence error. In the TFSTP, all pixels are reconstructed using the same set of STP parameters. Hence we need to make a trade-off between removing motion blur and noise, making it inevitable to lose sight of the other. Therefore, we propose a motion-dependent short-time plasticity reconstruction algorithm, termed texture from motion-dependent STP (TFMDSTP). In TFMDSTP, we first detect the motion pixels through the STP based on the characteristic that STP will fluctuate in the changing radiance region, then use STP with different parameter settings to estimate moving and stationary pixels. Experimental results show that the proposed STP-based image reconstruction algorithm can effectively decrease noise, while the motion-dependent algorithm can reduce both noise and motion blur.

This work is an extension of the paper [31] published on CVPR. Compared with the work of the conference version, the contributions of this paper are mainly in the following aspects:

- We propose a strategy to correct the inter-spike-interval (ISI) error, which can effectively reduce the noise of ISI-based reconstruction method, e.g., TFSTP.
- We analyze the effects of different parameter settings on the convergence time and error of the STP model through theory and simulation experiments. Based on this set of analyses, the optimized parameters can be selected for different pixels.
- We extend the previous reconstruction algorithm, TFMSTP, by introducing the STP model in the motion area, which can automatically switch the input to ISI or firing rate according to the characteristics of the motion area.
- We expand experiments with multiple sets of the simulated data, and achieve the best performance on the full-reference image quality assessment.

II. Related Works

A. Event-Based Imaging

It is difficult for event cameras to reconstruct textures in scenes as visual information of the static scenes is lost. Therefore, some hybrid sensors combining event cameras and conventional digital cameras, such as ATIS [32], DAVIS [33], RGB-DAVIS imaging system [34], [35], and Celex [36], were developed in recent years. Based on these sensors, the scene could be reconstructed with a high frame rate and higher quality by directly combining events and frames [36], [37], [38], [39], [40], [41] or warping the events to images [42], [43], [44]. However, the difficulty in achieving reliable temporal synchronization between events and low-rate frames from traditional sensors makes these methods inapplicable in capturing the high-speed scene.

In recent years, generating high-speed and high dynamic range videos with event cameras based on deep neural networks (DNNs) has become a mainstream trend. Inspired by [45], [46], Rebecq et al. [16], [18] trained a recurrent UNet architecture

(E2VID) end-to-end with simulated data. These works were later improved by introducing a temporal consistency loss [47] and achieved the state of the art. Scheerlinck et al. [40] proposed a *FireNet*, which can reduce the model complexity of E2VID by 99% with minor trade-offs in reconstruction quality. Except for using the recurrent architecture, generative adversarial networks (GANs) [48] were used in [49], [50] to generate frames from events. Taking advantage of the low latency and high dynamic range of the event camera, Wang et al. [51] and Mostafavi et al. [52] proposed a learning-based method to solve the super-resolution problem by embedding event streams. Han et al. [35] proposed a "upsampling and luminance fusion network" to obtain high dynamic range images. Nevertheless, the computational cost of DNNs is high and does not leverage the low-power and low-latency of event cameras.

B. High-Speed Imaging Based on Spiking Cameras

Based on the temporal characteristic of spike streams generated by spiking cameras, some reconstruction methods have been proposed [24], [25], [26], [27]. Zhu et al. [24] presented "texture from inter-spike-intervals (TFI)" and "texture from playback (TFP)" to rebuild the scenes according to the firing interval and firing rate, respectively. As there is a trade-off between removing the motion blur and improving the image contrast, the length of the window needs to be carefully defined, which will significantly influence the results. To solve this problem, Zhu et al. [25] proposed a GraphCut-based method to extract the motion area and reconstruct the static and motion area with different methods. Nevertheless, the motion extraction based on GraphCut needs to optimize the motion mask iteratively. Such an energy-based optimized way is time-consuming that diminishes the advantage of the low latency of spiking cameras. Zhao et al. [26] improved the signal-to-noise ratio by utilizing temporal correlations of signals to compensate motion, but it only applied to the scenes with linear motion. Another method is based on deep neural networks. Zhao et al. [27] proposed the Spk2ImgNet, which takes a spike sequence as input and automatically extracts features of different periods to form reference frames and key frames. Besides, they introduced a pyramid deformable alignment (PDA) module to align reference frames to key frames [53], [54]. This method achieves the state-of-the-art results, but it is also time-consuming and does not preserve the low latency benefit of the spiking cameras.

III. Preliminaries

A. Spiking Camera

Fovea-Like Sampling Method: Inspired by the sampling mechanism of primate fovea [20], [21], spiking cameras take advantage of spike sequences to represent the brightness change in the spatial-temporal domain [22], [23]. Specifically, the photosensitive units continuously capture photons and increase the photodiode voltage. When the accumulated intensity exceeds a given threshold, a spike is generated and the photodiode voltage is reset to a predefined reset voltage. This process can be formulated as:

$$\text{A spike is generated at time } t^f \text{ if } \int_{t^{f-1}}^{t^f} I(t)dt \geq \phi, \tag{1}$$

where $I(t)$ denotes the scene radiance, ϕ denotes the predefined threshold, and t^{f-1} represents the firing moment of the last spike. The spikes generated by spiking cameras can be represented by a 3-tuple $\mathcal{S}: \{x, y, t\}$, where $\{x, y\}$ denotes the spatial coordinates of the spikes in the photosensitive units, and t is the spike firing timestamp.

Texture Reconstruction from Inter-spike-interval (TFI): Based on the sampling mechanism of spiking cameras, the photosensitive units receive different scene radiance will trigger spikes with different frequencies. The inter-spike-interval decreases as the scene radiance increases. Therefore, the pixel value (proportional to scene radiance) can be estimated by the interval between two neighboring spikes:

$$\hat{P}_{TFI} = \frac{C}{\Delta t}, \tag{2}$$

where C refers to the maximum dynamic range of the spiking camera, and Δt represents the inter-spike-interval.

Texture reconstruction from Playback (TFP): The TFP method infers the pixel value by collecting the spikes in a moving time window. By counting these spikes, we have

$$\hat{P}_{TFP} = \frac{N_w}{w} \cdot C, \tag{3}$$

where C is the maximum dynamic range of the spiking camera, w is the size of the time window, and N_w is the total number of spikes collected in the time window.

B. Short-Term Plasticity (STP)

Short-term plasticity (STP) refers to the short-term change of synaptic strength, which is usually between tens to thousands of milliseconds [28], [29]. STP is sensitive to firing frequency of the presynaptic spikes and can transiently adjust postsynaptic potential (PSP) amplitude accordingly. When a postsynaptic neuron receives a sequence of action potentials (spikes) from a presynaptic neuron, the PSP changes according to:

$$\mathrm{PSP}(t) = A \cdot R(t) \cdot u(t), \tag{4}$$

where A denotes the maximum voltage value that an action potential can trigger on a postsynaptic neuron, $R(t)$ denotes the remaining number of available neurotransmitters in the axon at time t, and $u(t)$ denotes the release probability of neurotransmitter in the axon at time t. The following ordinary differential equations define the dynamics of $R(t)$ and $u(t)$:

$$\frac{dR(t)}{dt} = \frac{1 - R(t)}{\tau_D} - u(t^-)R(t^-)\delta(t - t_{sp}), \tag{5}$$

$$\frac{du(t)}{dt} = \frac{U - u(t)}{\tau_F} + C[1 - u(t^-)]\delta(t - t_{sp}). \tag{6}$$

Here $\delta(t)$ represents Dirac delta function, C is a constant parameter that influences the change of $u(t)$. (5) illustrates that the amount of neurotransmitters $R(t)$ decreases by $u(t^-)R(t^-)$ when a presynaptic spike releases at time t_{sp}, and recovers to

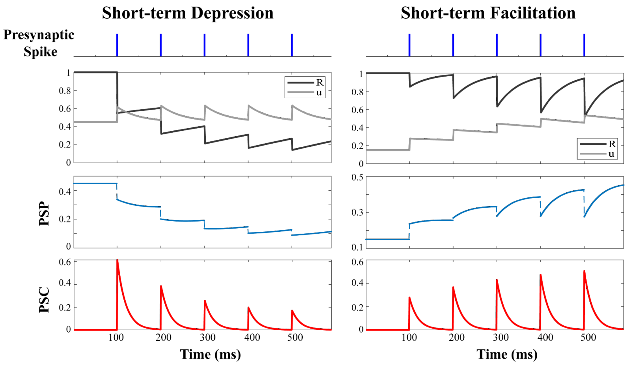

Fig. 2. The postsynaptic potential (PSP) and postsynaptic current (PSC) generated by STP with received spike streams from a presynaptic neuron. Left: The short-term depression dominated model, the parameters are $\tau_D =$ 750 ms, $\tau_F = 50$ ms, $U = 0.45$, $C = 0.3$. Right: The short-term facilitation dominate model, the parameters are $\tau_D = 50$ ms, $\tau_F = 750$ ms, $U =$ 0.15, $C = 0.15$.

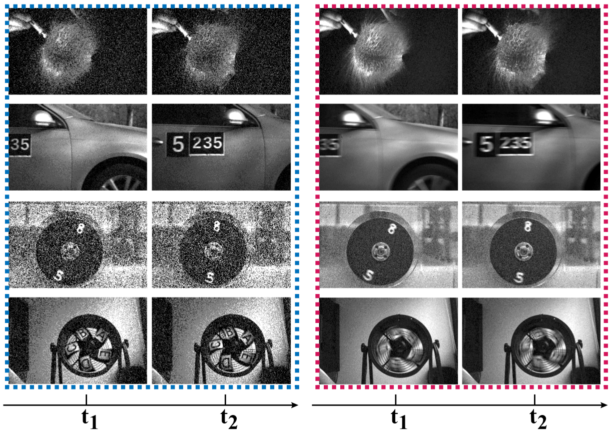

Fig. 3. Example results of the TFP [24] with different length of time window. Images in the blue dotted box are recovered with $w = 8$, and images in the red dotted are recovered with $w = 32$.

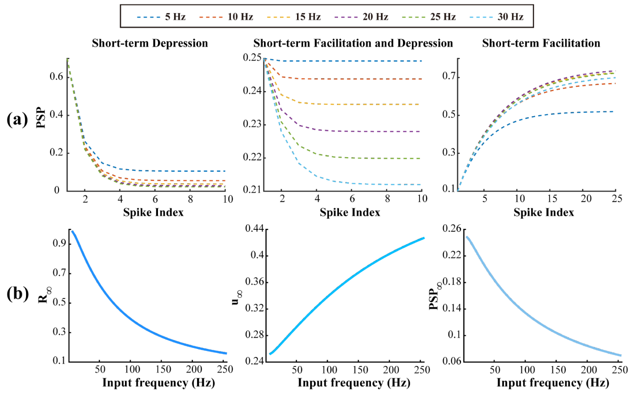

Fig. 4. (a) The dynamic of PSP regulated by STP. The dotted lines with different colors refer to spike trains with different frequencies, from 5 Hz to 30 Hz. The short-term facilitation and depression model has a mixture of properties of both short-term depression and short-term facilitation. (b) The steady value of PSP, the number of neurotransmitters R, and the release probability u with respect to different input frequencies. The results are obtained with a short-term facilitation and depression model.

1 with a depression time constant τ_D. Note, the notation t^- denotes that these functions should be computed in the limit approaching the spike release time from below. (6) indicates that the release probability $u(t)$ increases by $C[1 - u(t^-)]$ once a presynaptic spike fires, and decays back to baseline release probability U with facilitation time constant τ_F. Similar to PSP, the postsynaptic current (PSC) is formulated by:

$$\frac{d\text{PSC}(t)}{dt} = -\frac{\text{PSC}(t)}{\tau_s} + A \cdot R(t^-) \cdot u(t) \cdot \delta(t - t_{sp}). \quad (7)$$

Intuitively, the dynamics of $R(t)$ and $u(t)$ (5) and (6) can be seen as two low-pass filters of the input spikes, and their cutoff frequencies are inversely proportional to time constants τ_D and τ_F. There are two types of STP named short-term depression and short-term facilitation, respectively. Short-term depression and short-term facilitation have opposite effects on synaptic efficacy, which are illustrated in the middle and bottom of Fig. 2. Through changing the four parameters $\text{STP}^{\boldsymbol{\theta}} = \{\tau_D, \tau_F, U, C\}$, STP can have forms being either short-term depression dominated or short-term facilitation dominated. STP has effects on information transformation and network dynamics, including temporal filtering [55], [56], gain control [57], [58], induction of instability or mobility of network state [59].

IV. Methodology

A. Overview of the Method

Previous works mainly reconstruct the scenes by estimating the firing frequency of each pixel [24], [25], [26]. Fig. 3 illustrates the reconstruction results of TFP [24] with different time windows. One can find that a short time window leads to lower contrast and less motion blur, while the long one has higher contrast and more motion blur. Thus, it requires an appropriate predefined time window to estimate the firing frequency accurately, to make the texture relatively high contrast, and avoid motion blur.

To mitigate the weakness of the setting of the time window, we set up the STP model at each pixel of the spiking cameras to record the temporal regularity of spikes implicitly. The dynamics of PSP regulated by STP is shown in Fig. 4(a). It can be find that PSP will converge to a steady value if the firing frequency of the input spike streams is fixed, no matter what type of STP is used.

Moreover, the steady value of PSP, the number of vesicles R, and the release probability u are all monotonically increasing functions of firing frequency (shown in Fig. 4(b)). As mentioned above, the firing frequency of the spike streams triggered by each photosensitive unit is proportional to the received scene radiance. Intuitively, we can estimate the scene radiance and the pixel value of the reconstructed images, as well as detect the motion area based on PSP. The details of our approach are presented in the following sections.

B. Texture Reconstruction Through STP

1) Estimation of the Firing Frequency With STP: By setting up the STP model at each pixel of the spiking cameras to record the output spike stream, we derive the equation between the

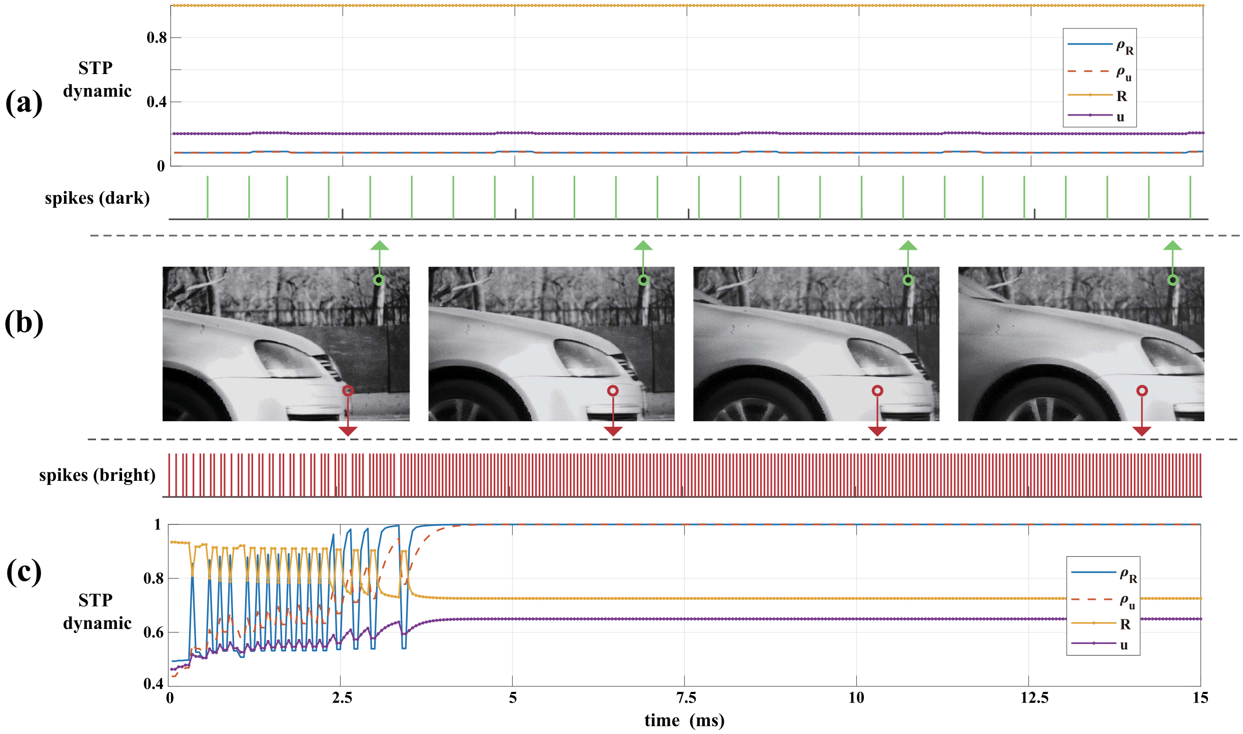

Fig. 5. Difference between STP dynamics of dark and bright area. (a) Spike raster and the corresponding STP dynamics of dark area (green circle in (b)). (b) Scenes reconstruction via Algorithm 1. (c) Spike raster and STP dynamics of bright area (red circle in (b)).

time-varying firing frequency of each pixel and the dynamics of postsynaptic neuron. For the sake of derivation, and numerical implementation with simplicity and efficiency, the dynamics of $R(t)$ and $u(t)$ (5) and (6) can be rewritten as the following difference equations by integrating between spikes n and $n+1$:

$$R_{n+1} = 1 - [1 - R_n(1-u_n)] \exp\left(-\frac{\Delta t_n}{\tau_D}\right), \tag{8}$$

$$u_{n+1} = U + [u_n + C(1-u_n) - U] \exp\left(-\frac{\Delta t_n}{\tau_F}\right), \tag{9}$$

where R_n and u_n denote the value of R and u between spikes n and $n+1$, Δt_n denotes the interval between spikes n and $n+1$. Similar to [60], we set $C = U$. If the spike rate ρ keeps constant, R and u will converge to their steady-state values $R_\infty(\rho)$ and $u_\infty(\rho)$:

$$R_\infty(\rho) = \frac{1 - \exp(-\frac{1}{\rho\tau_D})}{1 - [1 - u_\infty(\rho)] \exp(-\frac{1}{\rho\tau_D})}, \tag{10}$$

$$u_\infty(\rho) = \frac{U + (C-U)\exp(-\frac{1}{\rho\tau_F})}{1 - (1-C)\exp(-\frac{1}{\rho\tau_F})}. \tag{11}$$

Conversely, assuming that the spike rate ρ keeps constant and R and u have already converged to their steady-state values, we can estimate ρ from R and u separately through (10) and (11):

$$\rho_R = -\frac{1}{\tau_D \ \ln\left(\frac{1-R}{1-R(1-u)}\right)}, \tag{12}$$

$$\rho_u = -\frac{1}{\tau_F \ \ln\left(\frac{u-U}{C-U+u(1-C)}\right)}. \tag{13}$$

As the firing frequency of each pixel is proportional to the scene radiance, the estimated pixel value is a weighted average of ρ_R and ρ_u:

$$\hat{P}_{stp} \propto w_1 \cdot \rho_R + w_2 \cdot \rho_u. \tag{14}$$

By varying the weighted parameter $\boldsymbol{w} = \{w_1, w_2\}$, we can control the contribution of ρ_R and ρ_u to the constructed image.

Fig. 5 compares the STP dynamics $\{\rho_R, \rho_u, R, u\}$ for dark and bright area with different scene radiance. In the dark area without moving objects (green circle in Fig. 5(b)), the spikes generated by a spiking camera have a nearly constant frequency. Hence the corresponding STP dynamics will converge to the steady value rapidly. When the car in the scenes moves across the bright areas (red circle in Fig. 5(b)) between 0–3.5 ms, there are some small-range fluctuations of the STP dynamics, but the overall trend is still converging towards the corresponding state of the bright place.

2) Spike Interval Correction.: The spiking camera utilizes the row scanner to read out the spike streams, and the time resolution (minimum sampling time) is $T = 25$ μs. If the true interval is not a multiple of 25 μs, spike intervals from the raw spike data will jump between two adjacent integers, which causes the salt-and-pepper noise in the reconstructed image. An example is illustrated in Fig. 6. To reduce the noise, we need to correct the possibly wrong spike intervals to better reflect the changes in light intensity. We propose the following scheme to

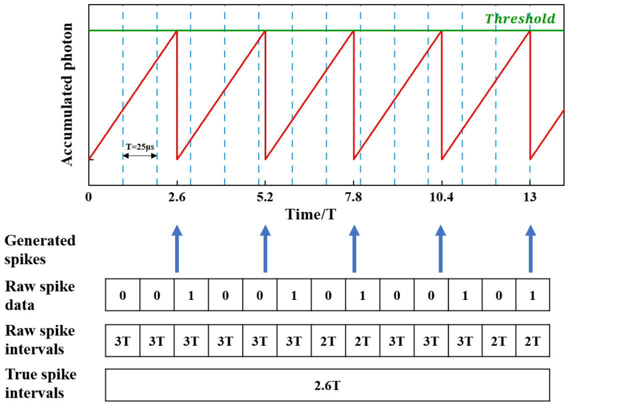

Fig. 6. Illustration of spike interval correction. In this example, as the true interval, 2.6 T, is not a multiple of T, the spike intervals from the raw spike data will switch between 2 T and 3 T, which differ from the true spike interval. If we replace each interval with the average of five neighboring intervals, all intervals will be corrected to 2.6 T.

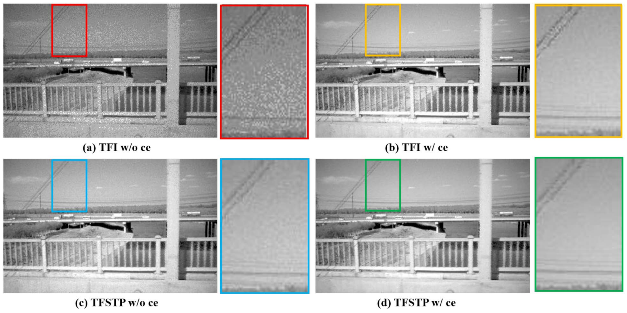
(a) TFI w/o ce
(b) TFI w/ ce
(c) TFSTP w/o ce
(d) TFSTP w/ ce

Fig. 7. The reconstruction images of the "viaduct-bridge" dataset. (a) reconstructed through TFI, without spike interval correction. (b) reconstructed through TFI, with spike interval correction. (c) reconstructed through TFSTP, without spike interval correction. (d) reconstructed through TFSTP, with spike interval correction. To the right of each reconstructed image is the closeup.

TABLE I
PARAMETER SETTINGS OF THE SHORT-TERM PLASTICITY MODELS. $T = 25\,\mu$s IS THE TEMPORAL RESOLUTION OF SPIKING CAMERA

STP type	$\tau_D\ (T)$	$\tau_F\ (T)$
Strong facilitation	$0.02\mathcal{A}$	$1.7\mathcal{A}$
Weak facilitation	$0.05\mathcal{A}$	$0.5\mathcal{A}$
Facilitation and depression	$0.2\mathcal{A}$	$0.2\mathcal{A}$
Weak depression	$0.5\mathcal{A}$	$0.05\mathcal{A}$
Strong depression	$1.7\mathcal{A}$	$0.02\mathcal{A}$

Algorithm 1: Texture From STP (TFSTP).

```
Input: Spike streams $\mathcal{S}_{ij}$.
Output: Estimated pixel value $\hat{P}_{ij}$.
1: Initialize the parameters of STP, $\{\tau_D, \tau_F, U, C\}$, $R$
   and $u$, and the weight parameter $\mathbf{w}$.
2: Compute inter-spike-intervals $\Delta t_n$.
3: Detect the wrong intervals and correct wrong intervals
   to $\Delta t'_n$ using (15) and (16).
4: Update $R_{n+1}$ and $u_{n+1}$ using (8) and (9).
5: Estimated the firing frequency $\rho_R$ and $\rho_u$ using (12)
   and (13).
6: Estimate the pixel value using (14).
```

detect wrong intervals:

$$\Delta t_n^{ij} \text{ is wrong } \iff \max\{\Delta t_k^{ij} \mid k \in (n - n_r, n + n_r)\}$$
$$- \min\{\Delta t_k^{ij} \mid k \in (n - n_r, n + n_r)\} = 1, \qquad (15)$$

where Δt_n^{ij} denotes the interval between nth spike and $n + 1$th spike on location (i, j), and n_r is used to decide how many neighboring intervals of Δt_n^{ij} are used to infer whether Δt_n^{ij} is wrong or not. When the difference between the maximum value and the minimum value of the five intervals equals 1, Δt_n^{ij} needs to be corrected. Here we propose a moving average interval to correct this error interval:

$$\Delta t_n^{ij\prime} = \frac{\sum_{k=n-n_r}^{n+n_r} \Delta t_k^{ij}}{2n_r + 1}. \qquad (16)$$

In this work, we set $n_r = 2$, i.e., five intervals are used to correct Δt_n^{ij}. The steps of this method (texture from short-term plasticity, *TFSTP*) are summarized in Algorithm 1.

Fig. 7 shows four qualitative comparison results of TFI and TFSTP, with and without spike interval correction. It can be seen that the images reconstructed through TFSTP are less noisy than TFI. Besides, spike interval correction can also reduce the noise of the reconstructed images. When TFSTP and spike interval correction are combined, the reconstructed images achieve significantly lower noise, higher contrast, and higher dynamic range than those using TFI.

C. STP Model Analysis

In order to further improve the quality of reconstructed images, we analyze the dynamic characteristics of STP here, including the steady state of R and u, convergence speed and noise after convergence. We used both theoretical analysis and simulated data analysis to explore the relationship between the properties of STP model and two time constants τ_D and τ_F in (8) and (9).

Theoretically, τ_D and τ_F can change arbitrarily in $\mathbb{R}^+ \times \mathbb{R}^+$, but we do not discuss this arbitrary change in the following analysis. Here we discuss two types of change in τ_D and τ_F:

1. The magnitude of τ_D and τ_F. We will use a scale factor $\mathcal{A}$ to make τ_D and τ_F change proportionally.
2. The ratio of τ_D and τ_F, which will determine the dynamics of STP.

Specifically, we explore five types of STP, whose ratio of τ_D and τ_F are shown in Table I. In the following, we will explore the dynamic properties of these five types of STP in relation to the scale factor $\mathcal{A}$. For brief description, we use $T = 25\,\mu$s, the sampling temporal resolution of the spiking camera, as the time unit.

1) Theoretical Analysis: For steady values of R and u shown in (10) and (11), by simple monotonicity analysis, we can deduce that when we set $C = U < 1$, R_∞ is a monotone decreasing function of $\rho\tau_D$ and u_∞ is a monotonically increasing function

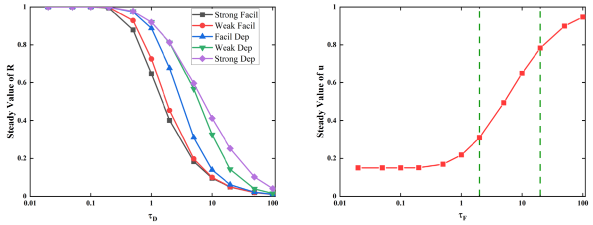

Fig. 8. The steady value of R (left) and u (right). Note that the steady value of R is related to both τ_D and τ_F, so different STP type leads to different steady value of R even if ρ_D is kept unchanged, which contributes to the five different curves in the left subfigure. Nevertheless, the steady value of u has no relation to τ_D, hence the right subfigure has only one curve.

of $\rho\tau_F$. The diagram of the steady value and the $\rho\tau$ in Fig. 8 also demonstrates the monotonicity of R_∞ and u_∞.

For convergence speed, since the convergence of R and u requires constant spike rate, we can safely assume that spike rate ρ and spike interval $\Delta t = \frac{1}{\rho}$ keep constant, then from (8) we have:

$$R_{n+1} - R_\infty = (1 - u_\infty) \exp\left(-\frac{\Delta t}{\tau_D}\right)(R_n - R_\infty) \quad (17)$$

Hence R_n converges Q-linearly with rate $r_R = (1 - u_\infty)\exp(-\frac{\Delta t}{\tau_D})$. Since R only takes value between [0,1], we have $R_0 - R_\infty < 1$, so it takes at most $\frac{\ln \epsilon}{\ln r_R}$ steps for R_n to converge with error no more than ϵ. The convergence time $\mathcal{T}_R(\epsilon)$ for R with error ϵ is:

$$\mathcal{T}_R(\epsilon) = \frac{\ln \epsilon}{\ln r_R} \cdot \Delta t = \frac{\ln \epsilon^{-1}}{\frac{\ln(1-u_\infty)^{-1}}{\Delta t} + \frac{1}{\tau_D}} \quad (18)$$

Using $\Delta t = \frac{1}{\rho}$, we have:

$$\mathcal{T}_R(\epsilon) = \frac{\ln \epsilon^{-1}}{\rho \ln(1 - u_\infty)^{-1} + \frac{1}{\tau_D}} \quad (19)$$

By (19), when $\mathcal{A}$ increases, i.e., τ_D and τ_F change proportionally, the first term in the denominator increases since u_∞ increases with τ_F, but the second term $\frac{1}{\tau_D}$ decreases, resulting the nonmonotonicity of $\mathcal{T}_R(\epsilon)$. When STP changes toward a more facilitative type, i.e., τ_D decreases and τ_F increases, both the first term and the second term in the denominator increases, so convergence time $\mathcal{T}_R(\epsilon)$ decreases and vice versa.

Similarly, we can get the convergence time of u with error ϵ:

$$\mathcal{T}_u(\epsilon) = \frac{\ln \epsilon^{-1}}{\rho \ln(1 - C)^{-1} + \frac{1}{\tau_F}} \quad (20)$$

In (20) shows that $\mathcal{T}_u(\epsilon)$ is a monotonically increasing function of τ_F, so both decreasing $\mathcal{A}$ and making STP type more depressive can decrease the convergence time of u, and vice versa.

2) Simulated Data Analysis: In addition to theoretical analysis, we use a simulated spike data composed of several long spike intervals $\Delta t_1 = 20\ T$ followed by several short spike intervals $\Delta t_2 = 5\ T$ to further analyze the convergence time and noise of R and u in different STP settings. Since the spike interval is inversely proportional to the scene radiance, this setting can simulates a pixel moving from dark area to bright area. Besides,

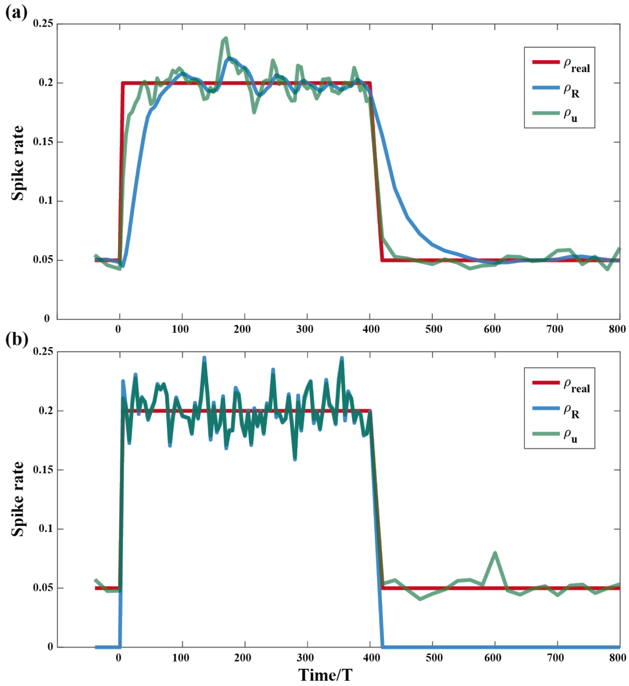

Fig. 9. A demonstration of our simulated data. In this figure, the spike interval change from $\Delta t_1 = 20\ T$ to $\Delta t_2 = 5\ T$ at $T_0 = 0$, and change from $\Delta t_2 = 20\ T$ to $\Delta t_2 = 5\ T$ at $T_1 = 400\ T$. The red, blue, and green lines represent the actual firing rate and those estimated from R and u of the STP model, respectively.

we also introduce some gaussian noise to the simulated data simulating the noise in real spike data.

For each STP parameter setting, we compare ρ_R and ρ_u with the real spike rate ρ_{real} to get the convergence time and noise of R and u. Fig. 9 shows the spike firing rate estimated by the STP model against the real spike frequency during the change in the spike interval. In Fig. 9(a), ρ_R has a shorter convergence time but higher noise, while ρ_u have a longer convergence time but lower noise, and in Fig. 9(b), ρ_u only needs one spike to switch to different states that close to the real firing rate.

In our analysis, we define T_0 as the moment that spike interval turn to short, i.e., the pixel moves into the bright area. The convergence time $\mathcal{T}_R^e(\epsilon)$ and $\mathcal{T}_u^e(\epsilon)$ are the difference between T_0 and the first time after T_0 that ρ_R and ρ_u get into the ϵ-neighborhood of ρ_{real}:

$$\begin{aligned} \mathcal{T}_R^e(\epsilon) &= \min\{t : t > T_0 \wedge |\rho_R(t) - \rho_{real}| \leq \epsilon\} - T_0, \\ \mathcal{T}_u^e(\epsilon) &= \min\{t : t > T_0 \wedge |\rho_u(t) - \rho_{real}| \leq \epsilon\} - T_0. \end{aligned} \quad (21)$$

Besides, the noise after convergence $\mathcal{N}_R^e$ and $\mathcal{N}_u^e$ are defined as the mean square error between ρ_R, ρ_u and ρ_{real} after spike intervals have turned to short and R and u has converged:

$$\begin{aligned} \mathcal{N}_R^e &= \frac{1}{T_{\max} - T_{\min}} \sum_{t=T_{\min}+T}^{T_{\max}} (\rho_R - \rho_{real})^2, \\ \mathcal{N}_R^e &= \frac{1}{T_{\max} - T_{\min}} \sum_{t=T_{\min}+T}^{T_{\max}} (\rho_u - \rho_{real})^2, \end{aligned} \quad (22)$$

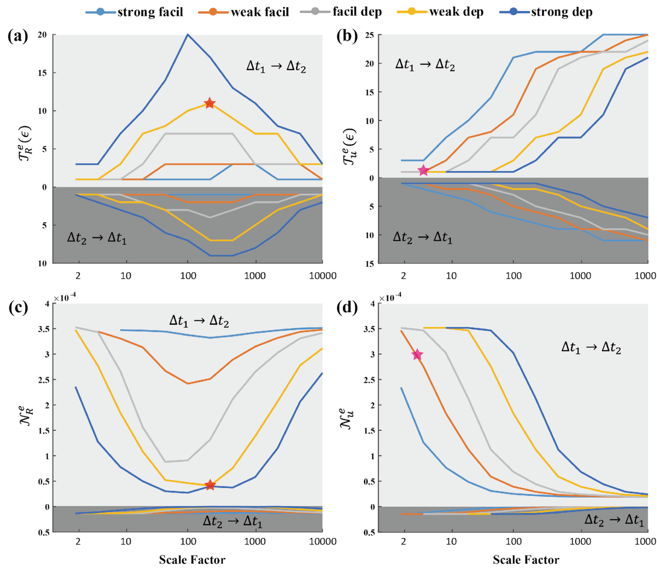

Fig. 10. Illustration of dynamics of the STP models under different settings. (a)–(b) The number of spikes required for the convergence of R and u against the change of the scale factor $\mathcal{A}$. (c)–(d) Relationship between the noise after R and u convergent and the scale factor $\mathcal{A}$. The light gray shaded area represents those when inter-spike interval changes from $\Delta t_1 = 20\ T$ to $\Delta t_2 = 5\ T$, while the dark shaded areas denote the cases from $\Delta t_2 = 5\ T$ to $\Delta t_1 = 20\ T$.

where $T_{\min}$ must be chosen after R and u have converged, and $T_{\max}$ is the maximum simulation time. Here, we choose $T_{\min} = 200\ T$ and $T_{\max} = 1000\ T$ in our simulation.

As shown in Fig. 10, the convergence time of R increases and then decreases as the scale factor $\mathcal{A}$ increases, but it keeps decreasing monotonically when STP type changes from strong depression to strong facilitation, which is consistent with the theoretical analysis in this section. Nevertheless, the noise after convergence of R almost shows an opposite trend to convergence time of R, first decreasing and then increasing as $\mathcal{A}$ increases and keeps increasing as STP type turns more facilitative. Convergence time of u shows a monotonically increasing trend as u increases, i.e., $\mathcal{A}$ increases or STP type turns more facilitative, which also agrees with the theoretical analysis. As for noise of u, it monotonically decreases as $\mathcal{A}$ increases or STP type turns more facilitative.

D. Texture Construction From Motion-Dependent STP

For most scenes, the TFSTP method proposed in Section IV-B works pretty well. Nevertheless, for high-speed scenes with limited illumination, it may suffer from motion blur caused by rare spikes in the dark area, and not timely updated STP status. Interestingly, we find that STP can also be used to detect the motion area, and propose another texture reconstruction method (texture from motion-dependent short-term plasticity, *TFMDSTP*). Specifically, we first analyze the change of STP status on each pixel to detect and extract the motion area and then reconstruct the motion and static area via STP with different parameters, separately.

1) Motion Determination: If there exists motion in the area, the STP value will vary around the steady value corresponding to the scene radiance (Fig. 5). Therefore, it is able to detect the motion area by evaluating the STP dynamics, e.g., R, and u, which updates according to (8) and (9). The reconstruction process of TFMDSTP is shown in Fig. 11, it begins with motion determination via STP. We use the change of u at the pixel within a short time Δt to determine whether a pixel belongs to the motion area or not:

$$\mathcal{M}_{i,j,t} = \begin{cases} 1, & |u(i,j,t) - u(i,j,t-\Delta t)| \geq \theta \\ 0, & |u(i,j,t) - u(i,j,t-\Delta t)| < \theta \end{cases}, \tag{23}$$

where $\mathcal{M}_{i,j,t}$ denotes whether pixel(i,j) belongs to the motion area at time t, and $\theta = 0.01$ is a predefined threshold.

2) Area Refinement: Except for finding the motion pixels, places along the moving trajectory of objects are also regarded as motion areas in our methods, which is achieved by feeding the $\mathcal{M}_{i,j}$ as the input voltage to a locally connected network consisted of leaky integrate-and-fire neurons. The membrane potential $v(t)$ of this neuron changes according to:

$$\tau_m \frac{dv(t)}{dt} = -[v(t) - v_{rest}] + I_{i,j}(t), \tag{24}$$

where τ_m is the membrane time constant, and v_{rest} is the resting potential. The current $I_{i,j}$ of the neuron is the integrated result of neurons that locally connect to it, which is calculated as $I_{i,j} = \sum_{\boldsymbol{x}} \mathcal{M}_{\boldsymbol{x}}$ ($\boldsymbol{x}$ is the location of neurons in the corresponding and 8-neighborhood of neuron at (i,j)). In our cases, τ_m equals the minimum sampling time ($T = 25\ \mu$s) of the spiking cameras. The leaky integrate-and-fire neuron will release spikes when the membrane potential exceeds a certain threshold ϑ, and the membrane potential is reset to the resting potential. The state of the leaky integrate-and-fire neuron is changed as:

$$\chi_{i,j} = \begin{cases} 1, & v \geq \vartheta \\ 0, & v < \vartheta \end{cases} \tag{25}$$

After that, areas with $\chi = 0$ are regarded as static pixels while those ones with $\chi = 1$ refer to motion pixels. In Fig. 12, we show some example results obtained when detecting the brightness change with our proposed method. With continuous input of spike streams, the STP dynamic of each pixel gradually converges to a steady state, and only the motion area has state change (i.e., $\mathcal{M}_{i,j} = 1$ in Fig. 11).

3) Texture Estimation: Although the TFP method with short sliding windows can effectively decrease motion artifacts caused by the untimely update of the inter-spike intervals, the short-window statistics also make noise inevitably introduced. Therefore, in order to take into account the reduction of noise and motion artifacts, we also introduce the STP model for pixel estimation in the motion area. For CCD/CMOS-based vision sensors, the main factors related to motion blur are motion speed, exposure time, image size, and field of view (FOV) [61]. The relationship between motion blur within a pixel and these factors is:

$$\text{Blur in Pixels} = (\text{Line Speed} \cdot \text{Exposure Time}) \cdot \left(\frac{\text{Image Size}}{\text{FOV}}\right). \tag{26}$$

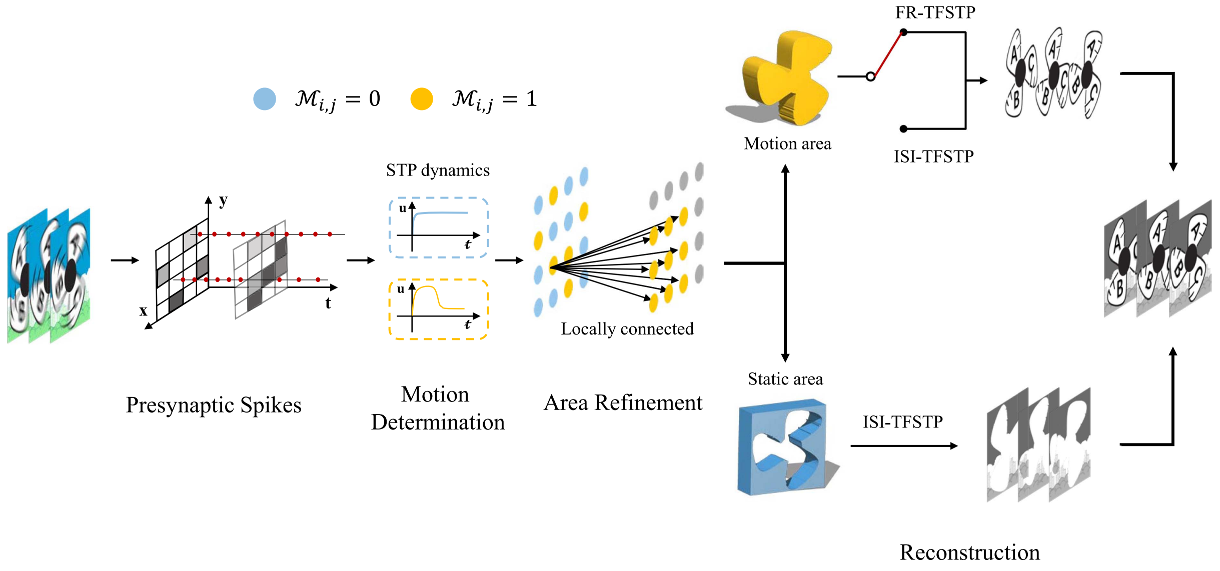

Fig. 11. Illustration of the reconstruction process based on distinguishing the motion and static pixel through STP.

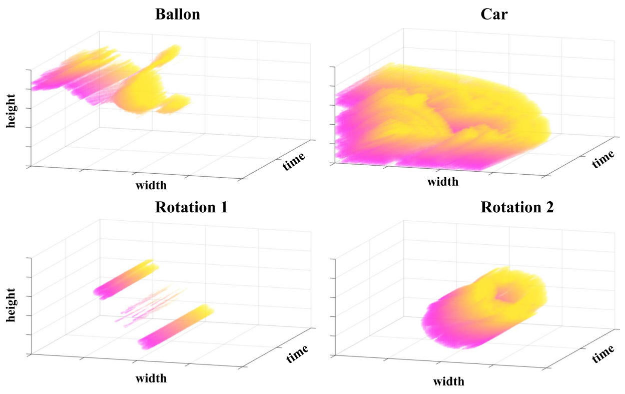

Fig. 12. Examples of the motion areas obtained during the process of the TFMDSTP.

Since speed of the moving objects is uncontrollable and the image resolution is fixed, we usually shorten the exposure time or increase the FOV to reduce the pixel blur. The effective way to shorten the exposure time is to increase the illumination, so as to shorten the time required to update the pixels. Increasing the FOV will increase the imaging angle and reduce moving pixels. Inspired by this relationship, if one wants to capture high-speed moving objects, it needs to accumulate enough photons in a short time, and the smaller the proportion of pixels of the moving object, the better.

Therefore, in the TFMDSTP, in order to reduce motion artifacts, the input of the STP model in the motion area will be determined by the motion area and the average firing rate of motion pixels. If the area of the motion pixels is large and the firing rate in the area is very low, the reciprocal of the firing rate accumulated in the local short window will be used as the input of the STP model; for other cases, the input of the STP model is still the inter-spike interval. The switched threshold between firing-rate-based TFSTP (*FR-TFSTP*) and ISI-based TFSTP (*ISI-TFSTP*) in the motion area is shown in the Fig. 13. When the motion area occupies over 10% of the image plane and the firing rate in this area is lower than 0.125, we will use FR-TFSTP to estimate the motion pixels (orange shading area). In other cases, the motion pixels will be reconstructed by the ISI-TFSTP method (green shading area).

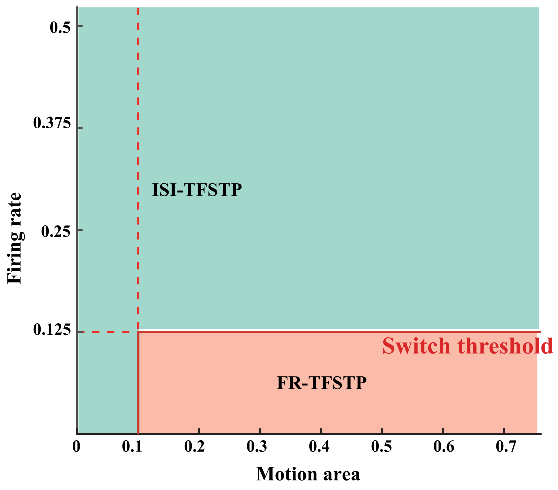

Fig. 13. Illustration of the switch threshold between ISI-TFSTP and FR-TFSTP.

In TFMDSTP, We use three sets of parameters based on the analysis of STP in Section IV-C for motion detection, motion and stationary pixels estimation, respectively. For motion detection, we want to be able to detect changes in the input sensitively by the state of the STP, such as the change in the steady-state value of u in Fig. 8; for motion pixels, we want the STP to converge quickly to the states corresponding to different motion pixels, i.e., the shorter the convergence time, the better; for stationary pixels, we want the noise to be as less as possible. The detailed

Algorithm 2: Texture from Motion-Dependent STP (TFMDSTP).

```
Input: Spike streams $\mathcal{S}_{i,j}$.
Output: pixel value $\hat{P}'_{i,j}$, motion state $chi_{i,j}$.
Initialize the three sets of parameters of STP (STP$^{\boldsymbol{\theta}1}$: static pixels estimation, STP$^{\boldsymbol{\theta}2}$: motion pixels estimation, STP$^{\boldsymbol{\theta}3}$: motion detection).
Obtain the corrected intervals $\Delta t'_n$ using (15) and (16).
Update $R_1, u_1, R_2, u_2, u_3$ using (8) and (9).
Use the change of $u_3$ to obtain $\mathcal{M}_{i,j,t}$ with (23).
Refine the motion area and get $\chi_{i,j}$ with (24) and (25).
Compute the area and average firing rate of motion pixels.
Determine the input of $STP^{\theta 2}$.
Use $R_1$ of STP$^{\boldsymbol{\theta}1}$ to estimate static pixel $\hat{P}'_{\chi=0}$ with (12), and use $u_2$ of STP$^{\boldsymbol{\theta}2}$ to estimate motion pixel $\hat{P}'_{\chi=1}$ with (13).
```

parameter settings will be given in the experimental section. All steps of *TFMDSTP* are summarize in Algorithm 2.

V. Experiments

In this section, in order to better compare the reconstruction methods for spiking cameras, we will first introduce six non-reference image quality metrics to quantitatively compare the pros and cons of different methods, and compare the reconstruction results qualitatively. Then, simulated spike data with higher spatial resolution, including ego-motion of cameras and moving objects, will be used to compare the reconstruction images with reference.

A. Parameters Selection

Based on the analysis of convergence time and noise after convergence of R and u in Section IV-C, we select parameters based on the priority of refactoring for different regions. For static area, low noise is more important than short convergence time, so we choose $\tau_D^1 = 100\ T$ and $\tau_F^1 = 10\ T$ (i.e., $\mathcal{A} = 200$ and weak depression, shown in the two red stars in Fig. 10), and use ρ_R for static area reconstruction, which has an almost lowest noise and acceptable convergence time. For motion area, short convergence time is more important than low noise, therefore we choose $\tau_D^2 = 0.25\ T$ and $\tau_F^2 = 2.5\ T$ (i.e., $\mathcal{A} = 5$ and weak facilitation, shown in the two magenta stars in Fig. 10), and use ρ_u for static area reconstruction. For motion detection, since we need to detect the change in R or u at each pixel, we choose an appropriate τ_D or τ_F to make steady value of R or u change smoothly as spike rate ρ changes. As shown in Fig. 8, when $\rho\tau_F$ is in range [2,20] (highlighted by two dark green dash lines in Fig. 8), steady value of u almost changes linearly as $\rho\tau_F$ changes logarithmically. In custom spike datasets, most of the spike intervals lies in $[2\ T, 20T]$, which indicates that the spike rate mostly lies in $[0.05/T, 0.5/T]$. Hence we choose $\tau_F^3 = 40\ T$ and use ρ_u, so that $\rho\tau_F$ will mostly lie in range [2,20], which is most suitable for motion detection. However, in the TFSTP method, in order to considerate both moving and static pixels, the parameters of TFSTP are set as STP$^{\boldsymbol{\theta}0}$: $\{\tau_D = 1\ T, \tau_F = 10\ T, C = U = 0.15\}$.

B. Real-World Scenarios

The real-world scenarios contains eight sequences captured by the spike camera with a sampling rate of 40,000 Hz, which can be divided into two categories: high-speed scenes with the object's motion (Class A) and high-speed scenes with camera's ego-motion (Class B) [25]. Class A includes "Balloon," "Car," "Rotation1," "Rotation2" and "Rotation2x". Among them, "Balloon" records a balloon filled with water being punctured by a needle, "Car" describes a car traveling at a speed of 100 km/h, "Rotation1" describes a disk with 2000 rpm (revolutions per minute), "Rotation2" and "Rotation2x" depicts a rotating fan with 2600 rpm. Class B includes "Forest," "Railway," "Train" and "Viaduct-bridge"(V-b). These four sequences are recorded by a spiking camera in a high railway with a speed of 350 km/h.

As shown in Fig. 14, compared with other methods, the reconstruction results of our method are less noisy, and they also effectively retain the texture information of high-speed moving objects. Furthermore, to quantitatively verify the effectiveness of the proposed methods on real-world data, we employ three no-reference image quality assessment (NR-IQA) metrics, namely BIQI [62], two-dimensional (2-D) entropy [63], standard deviation (STD). 2-D entropy uses both the gray value of a pixel and its local average gray value to evaluate the amount of information carried by the image, and a larger 2-D entropy means more information. Standard deviation evaluates the contrast of the image, and a larger standard deviation means higher contrast. BIQI considers JPEG quality, JP2K quality, noise, motion blur, and fast fading of the image. Different from the former two metrics, a lower BIQI score indicates higher image quality. The quantitative comparison results are reported in Table II. As shown in Table II, our methods achieve better results than other methods in almost all three metrics, which is consistent with the subjective observation in Fig. 14. TFSTP can achieve the best performance results with these no-reference image quality indicators on most sequences. However, as described in Section. V-A, for scenes with high-speed moving objects and static background, the selection of parameters in different regions during reconstruction has different tendencies. In TFSTP, only a set of STP parameters are used for reconstruction, and we need to make some trade-offs between removing the noise in the static area and the motion blur in the dynamic area. Except for the three methods of TFMDSTP, TFMSTP, and TFP, other methods all have motion artifacts on the two sequences of Rotation2 and Rotation2x.

Fig. 15 shows the firing rate and the size of moving area in the real scenes. The motion area indicates that there is an area where the gray value changes. The motion area indicates an area where the gray value changes. For the dataset of Class B and the Car sequence,the motion area is much larger. However, these scenes are shot outdoors with sufficient light (sunlight). Hence, the high spike firing rate can ensure that even if the gray value of the moving area keeps changing, the spike firing state of the moving pixel

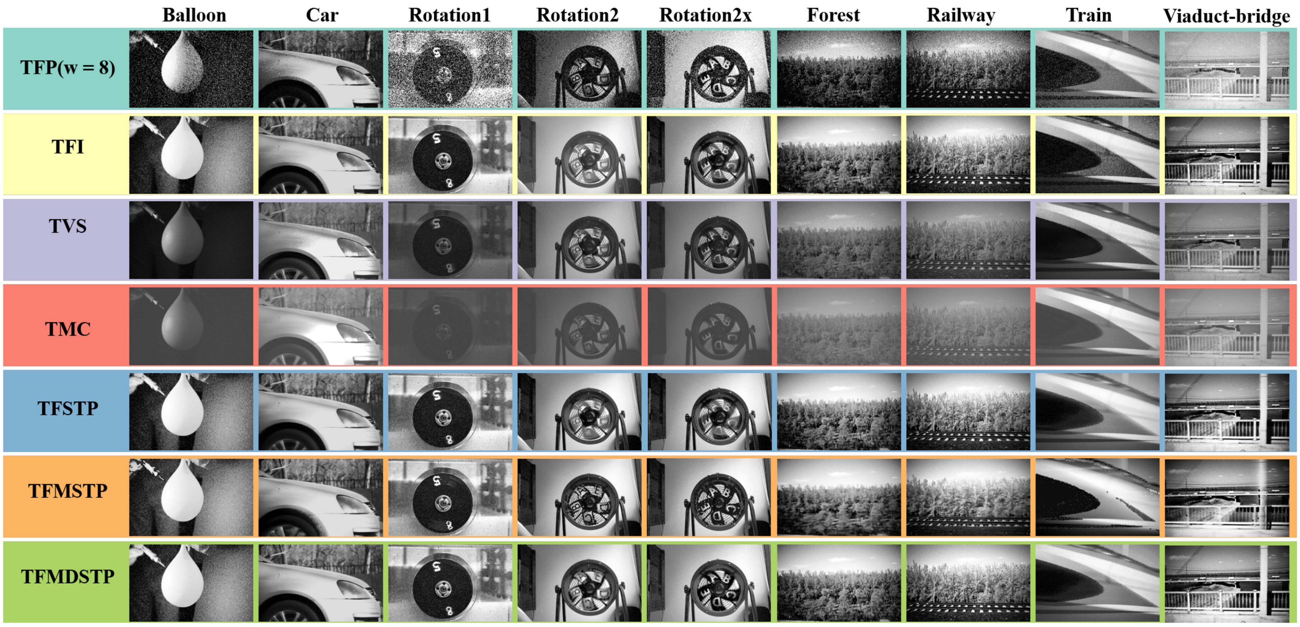

Fig. 14. Reconstruction results of TFP ($w = 8$), TFI [24], TVS [25], TMC [26], TFSTP, TFMSTP and TFMDSTP.

TABLE II
COMPARISON AMONG DIFFERENT RECONSTRUCTION METHODS. DOWNWARD ARROW DENOTES "THE LOWER THE BETTER," AND UPWARD ARROW DENOTES "THE HIGHER THE BETTER"

Metric	Method	Class A(object motion)					Class B(camera's ego motion)				average
		Balloon	Car	Rotation1	Rotation2	Rotation2x	Forest	Railway	Train	V-b	
2-D entropy ↑	TFP [24]	3.88	5.53	4.00	4.80	11.41	5.30	5.33	6.03	6.01	5.81
	TFI [24]	13.30	11.53	13.75	12.73	**13.20**	13.28	13.62	12.51	13.39	13.03
	TVS [25]	9.75	9.58	11.10	11.05	11.22	11.35	11.29	10.60	10.46	10.71
	TMC [26]	7.11	9.64	7.33	8.43	8.38	9.96	10.43	9.79	10.53	9.07
	TFSTP	**13.45**	**13.45**	**13.82**	**12.91**	12.89	**13.44**	**13.71**	**12.86**	**13.64**	**13.35**
	TFMSTP	13.01	12.01	13.12	12.48	11.66	12.62	12.86	11.28	12.89	12.44
	TFMDSTP	13.31	11.74	13.77	12.54	12.62	13.38	13.67	12.79	13.51	13.04
STD ↑	TFP [24]	46.41	46.09	**127.45**	**87.39**	69.27	51.77	68.45	59.82	56.99	68.18
	TFI [24]	77.88	70.96	74.76	74.27	71.37	73.19	72.79	71.62	71.47	73.15
	TVS [25]	31.83	58.34	27.04	37.74	52.89	46.94	37.82	66.49	36.31	43.93
	TMC [26]	17.94	47.01	7.05	20.36	21.45	23.04	24.48	46.44	27.70	26.16
	TFSTP	**77.95**	72.16	74.25	74.79	75.82	73.78	73.77	73.80	73.58	74.43
	TFMSTP	74.39	**74.84**	74.29	77.23	**77.76**	73.31	**74.56**	**77.16**	**74.01**	**75.28**
	TFMDSTP	77.92	71.91	74.31	73.55	73.66	**73.93**	73.88	73.82	73.58	74.06
BIQI ↓	TFP [24]	61.61	53.76	75.93	85.61	86.72	62.19	66.17	58.23	55.11	67.26
	TFI [24]	54.24	29.78	52.99	20.41	25.72	25.18	27.76	**23.77**	36.67	32.95
	TVS [25]	**37.26**	28.40	52.40	23.89	36.06	45.51	36.62	34.30	29.46	35.99
	TMC [26]	38.14	32.95	69.06	36.29	36.60	52.13	51.08	38.49	31.54	42.92
	TFSTP	54.30	31.56	52.70	**16.40**	24.72	**17.68**	**15.96**	26.88	30.45	30.07
	TFMSTP	45.07	**25.82**	**46.71**	17.83	22.95	32.81	22.30	24.73	**29.70**	**29.77**
	TFMDSTP	54.29	31.31	52.43	18.01	**22.51**	19.61	22.38	24.18	31.93	30.74

can be updated to the pixel (e.g., ISI). Therefore, using ISI-based reconstruction can effectively capture the high-speed motion of these five scenes. For the Class A scene that only contains object motion, except for the Car sequence, they are all shot indoors, so the firing rate of these four sequences is relatively low. However, the motion speed in Balloon and Rotation1 is slower, and the proportion of motion pixels is small. Using ISI-based methods can also effectively remove motion blur. In the rotation2 and rotation2x sequences, the area of motion is relatively larger, and the amount of spike emitted is also low, resulting in the use of ISI to update and still have motion artifacts. Therefore, for rotation2 and rotation2x that fall in the switched zone (gray area in Fig. 15), the TFMDSTP method will automatically select the FR-TFSTP to reconstruct the motion area, while other sequences use ISI-TFSTP.

Besides, as shown in Fig. 14, although the use of short window TFP can effectively remove some motion artifacts (TFMSTP), the difference between the dynamic range of the moving area and the static area is too large because the length of the spike-integration window is too short. As a result, there is a larger grayscale gap at the boundary of the motion area, and there is also more noise. The boundary between moving and static regions of TFMDSTP is smoother. Fig. 16 shows some results of the four methods of TFI, TFSTP, TFMSTP and TFMDSTP

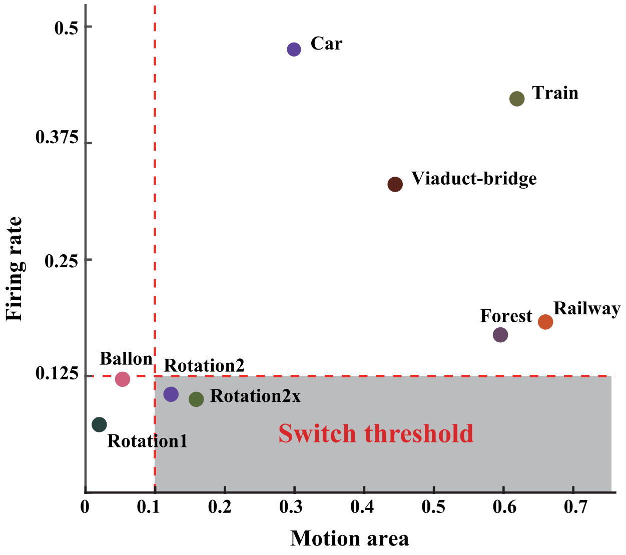

Fig. 15. Firing rate and motion area of the real-world spike sequences. The gray shaded area indicates the switched zone to use the FR-TFSTP in motion area.

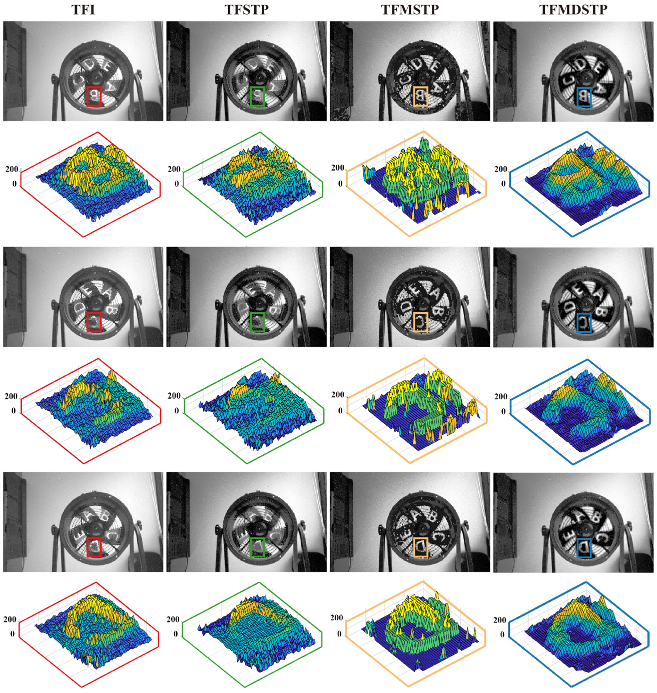

Fig. 16. Comparison among different reconstruction methods on the scenes of “Rotation2x”. Below each row of the reconstructed images is the zoomed-in 3D surface view of characters “B,” “C” and “D”.

on the rotation2x sequence. It can be seen that the TFMDSTP is less noisy than other methods, while maintaining low motion blur.

In addition to comparing with other methods, we quantitatively measure the image quality gains brought by the improvement strategies proposed in this work. In the comparison of the above results, the ISI-based methods (TFI, TFSTP, TFMSTP and TFMDSTP) are all added with corrected error spike intervals. In this work, we compare six non-reference image quality assessment metrics, of which the better the image quality, the lower the BIQI, NIQE [64], BRISQUE [65] and PIQE [66] indicators, and the higher the STD and 2-D entropy. The non-reference quality evaluation results of TFP, TFI, TFSTP, and TFMDSTP with and without error correction are shown in Fig. 17. Fig. 17(a)–(f) show the comparison results of indicators on different sequences, and the dots with different colors indicate the method that obtains the best result on the sequence. It can be seen that as different assessments have different emphasis on image quality evaluation, different optimal methods will appear on different measurements. However, TFSTP and TFMDSTP achieve the best results for most sequences (the dark green and dark blue points in the figure appear more frequently). Note that TFP shows abnormally high standard deviation on dataset “Rotation1” and “Rotation2,” which is caused by the high noise in the reconstruction image (see the first row of Fig. 14). Fig. 17(g) shows the average value of different indicators in all sequences. The red star indicates the method to obtain the best indicator result, and the blue star indicates the method to obtain the second best result. On PIQE, TFMDSTP obtains the best results, and TFSTP sub-optimal results. Other indicators are that TFSTP achieves the best results, and TFMDSTP achieves sub-optimal results. In addition, the error correction has also significantly improved the reconstruction methods based on spike interval (TFI, TFSTP and TFMDSTP), of which the quality of TFI has been improved the most. The corresponding numerical results are in Table S1.

C. Simulated Scenarios

In the results of the previous section, we find that when evaluating the reconstructed image quality of the real-world dataset, the results of the no-reference image quality assessment metrics have great volatility, making it difficult to evaluate the quality of the results comprehensively. For example, in Fig. 16, the results of TFMDSTP on rotation2x is obviously the best, but instead, TFSTP achieves the best performance with BIQI and STD, and TFI achieves the best on NIQE, BRISQUE, PIQE and 2-D entropy. Therefore, in this section, we use the data generated by the spiking camera simulator to evaluate the quality of the reference image. The simulation scene mainly uses various sky or aerial view of city as the background, plus some suspended foreground objects, such as chairs, helicopters with complex textures, etc. The resolution of the generated data is 800×500, which is four times the 400×250 of the real data sequence. In this section, we mainly compare the results of the TFP, TFI, TMC [26], TFSTP and TFMDSTP. In addition, we focus on comparing the impact of the improvement strategy proposed in this work on image quality, that is, error correction and the TFMDSTP.

Fig. 18 shows the reconstruction results of these four methods on the generated data and the corresponding ground-truth. Fig. 18(a) is the result of simulating the 80FPS frame-based camera to collect the same scene, in which blur degree can reflect the movement speed of different objects and cameras. Fig. 18(d)–(f) are results of TFI, TFSTP and TFMDSTP based on

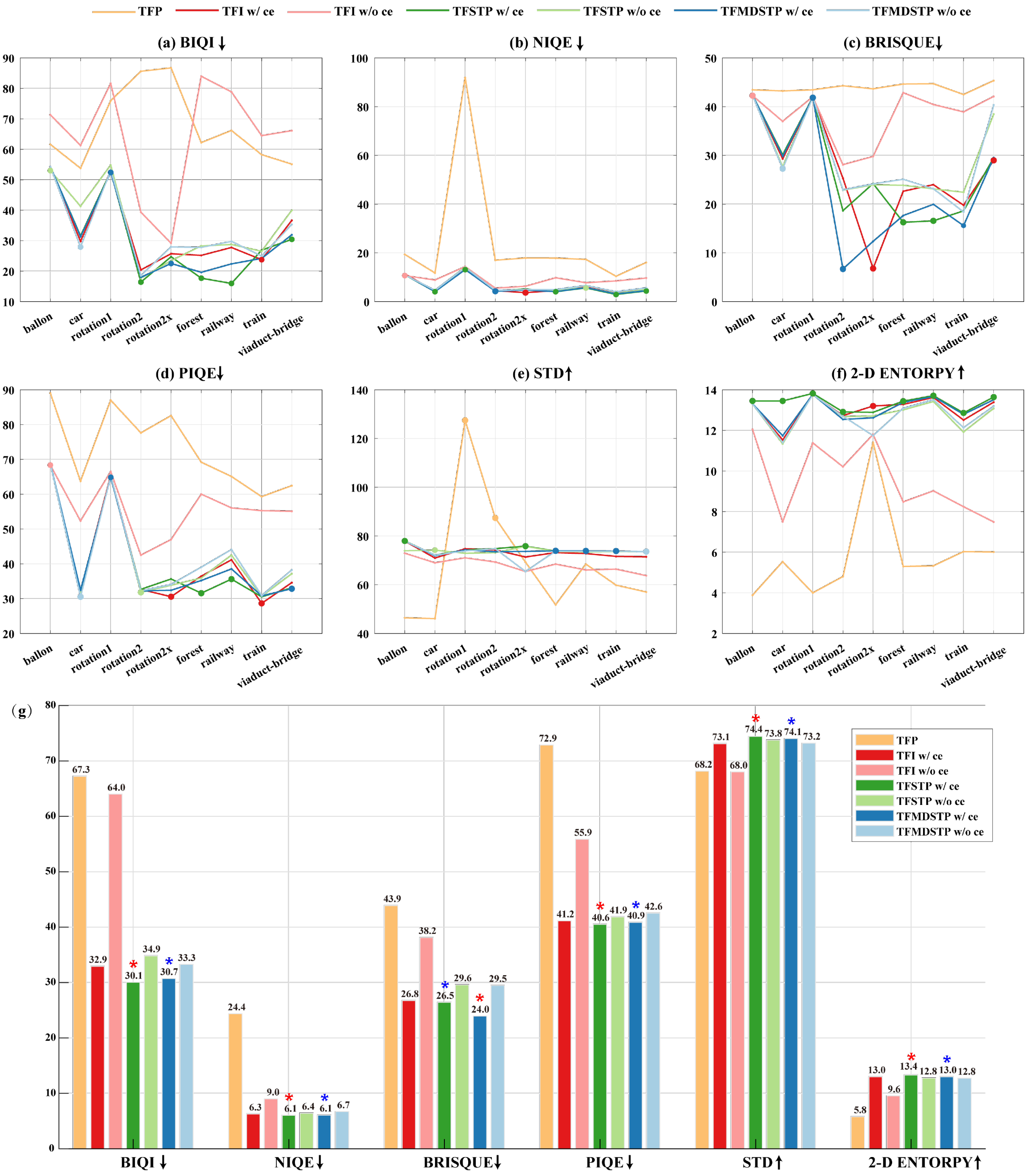

Fig. 17. Comparison among different methods with no-reference image quality assessment. Different color curves in (a)–(f) correspond to different methods as indicated by the legend at the top. The different colored dots in (a)–(f) indicate the method to obtain the best performance. (g): the average value of different indicators on all sequences. The red star indicates the optimal result, and the blue star indicates the sub-optimal result. The better the image quality, the lower the BIQI, NIQE, BRISQUE and PIQE, and the higher the STD and 2-D ENTROPY.

corrected error spike intervals. It can be seen that the result of TFI reconstruction is still more noisy than TFSTP and TFMDSTP. And TFMDSTP can take into account the removal of noise and the blur of high-speed moving objects. Table III shows the PSNR and SSIM results of several methods on the generated data, where TFMDSTP achieves the best results on each sequence and the proposed error correction can effectively tune up the performance.

D. Computational Complexity

Here we evaluate the computational complexity of our methods. For comparison, we consider the problem of reconstructing a K-frame video with a size $H \times W$. For each pixel, the TFSTP method only needs to update R, u, ρ_R, ρ_u when a spike generates at that pixel. Therefore, it needs at most K updates for each pixel. The time complexity of the TFSTP method is $O(HWK)$. Note that writing a $H \times W \times K$ video into the memory also takes $O(HWK)$ time, so our reconstruction method has reached the minimum asymptotic time complexity in theory.

Different from TFSTP, the TFMDSTP method needs extra steps to determine whether a pixel belongs to the motion area or not with (23). However, this only takes constant time for each pixel. It does not affect the asymptotic time complexity. In comparison, the GraphCut-based method (TVS) in [25] takes at least $O(H^3\ W^3)$ time to implement graph cut for each frame,

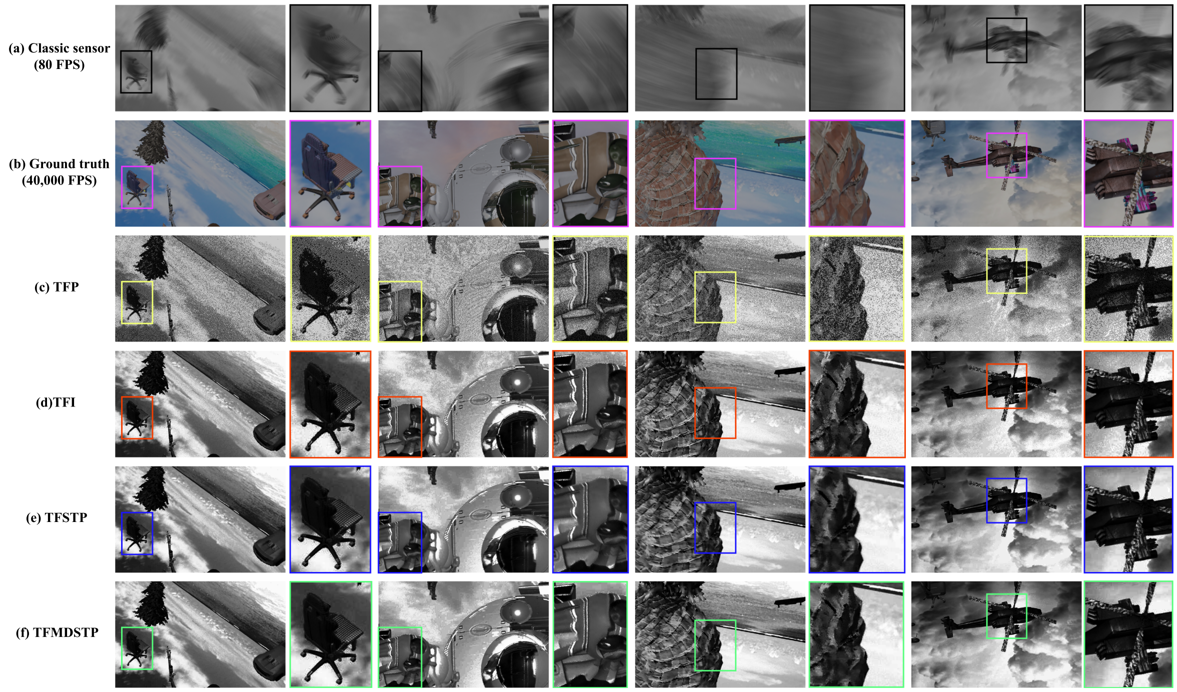

Fig. 18. Comparison of different methods on the synthesis dataset.

TABLE III
REFERENCED QUANTITATIVE EVALUATION USING SIMULATED DATA

Metric	Method	#4	#9	#12	#35	#38	#51	Average
PSNR ↑	TFP	13.32	12.15	14.40	13.38	12.67	12.42	13.06
	TFI	27.29	25.90	27.90	22.88	24.36	24.70	25.51
	TFI w/o ce	15.74	17.47	15.02	15.73	18.89	15.62	16.41
	TMC	21.03	21.05	21.32	19.56	24.19	19.77	21.15
	TFSTP	28.87	26.27	31.87	23.65	23.50	24.49	26.44
	TFSTP w/o ce	25.23	23.07	25.64	20.68	23.25	22.16	23.34
	TFMDSTP	**29.94**	**27.12**	**33.13**	**25.02**	**24.77**	**25.82**	**27.63**
	TFMDSTP w/o ce	25.37	23.21	25.62	20.67	24.00	22.10	23.50
SSIM ↑	TFP	0.1187	0.1136	0.1237	0.1271	0.1017	0.1962	0.1302
	TFI	0.7227	0.7189	0.7024	0.6228	0.7463	0.7690	0.7137
	TFI w/o ce	0.2293	0.4241	0.2346	0.3640	0.4008	0.4168	0.3449
	TMC	0.4267	0.6005	0.4363	0.5320	0.6084	0.5689	0.5288
	TFSTP	0.8304	0.7544	0.8922	0.7383	0.7756	0.8212	0.8020
	TFSTP w/o ce	0.6025	0.6204	0.5914	0.5310	0.6680	0.6557	0.6115
	TFMDSTP	**0.8398**	**0.7599**	**0.8958**	**0.7549**	**0.7999**	**0.8332**	**0.8139**
	TFMDSTP w/o ce	0.5964	0.6172	0.5857	0.5184	0.6718	0.6496	0.6065

thus it takes at least $O(H^3\ W^3\ K)$ time to reconstruct a K-frame video. The OF-based method (TMC) in [26] takes $O(H \cdot W \cdot K \cdot T \cdot iter)$ time to reconstruct a K-frame video.[1] Therefore, our methods achieve a significantly lower time complexity than other methods.

VI. Conclusion

In this paper, we propose novel bio-inspired image reconstruction methods for spiking cameras. The proposed methods are able to infer the scene radiance and the pixel value of the reconstructed images. We analyze the impact of parameters settings on the convergence time and error of STP, and design a motion-dependent reconstruction method based on this analysis, which can jointly reduce the motion blur and background noise. The theoretical analysis and experimental results show that our methods can reconstruct high-quality images with low computational complexity.

The binary stream output by the spiking camera is similar to the action potential that transmits information in the brain. The STP model we adopted here is to simulate the dynamic characteristics of synapses in neuroscience. It has good

[1] T denotes the size of the time window used in their method, and $iter$ denotes the number of iterations in computing the optical flow.

convergence properties and can gain control over the input spike streams. The Short-term plasticity is input-specific [67] and can memory previous states that are robust against perturbations. If using a sliding window to average the spike frequency, it will face the same problem as the TFP reconstruction algorithms. Too short a window is susceptible to noise spikes, making the appearance of noise as a region of motion. However, too long a window will make the changes hard to be perceived sensitively, similar to the motion blur in TFP. Moreover, as seen from the analysis in Section IV-C, we can adjust the dynamic properties exhibited by the STP model to different requirements. Unlike black-box models such as deep neural networks, the parameters of our proposed model have clearer meaning and more flexible adjustability.

The experimental results show that under the condition of sufficient lighting, high-quality reconstruction results can also be obtained by using TFI with the proposed error correction mechanism if the raw spike data is corrected. Therefore, we believe that in future work, we can dig deeper into the spatiotemporal information of the spike sequences to obtain higher-quality reconstructed images without too much computation.

REFERENCES

[1] D. Bradley, P. Bell, O. Landen, J. Kilkenny, and J. Oertel, “Development and characterization of a pair of 30–40 ps X-ray framing cameras,” *Rev. Sci. Instrum.*, vol. 66, no. 1, pp. 716–718, 1995.

[2] J. Itatani, F. Quéré, G. L. Yudin, M. Y. Ivanov, F. Krausz, and P. B. Corkum, “Attosecond streak camera,” *Phys. Rev. Lett.*, vol. 88, no. 17, 2002, Art. no. 173903.

[3] Phantom high speed, 2021. [Online]. Available: https://www.phantomhighspeed.com

[4] F. Guerrieri, S. Tisa, and F. Zappa, “Fast single-photon imager acquires 1024pixels at 100 kframe/s,” in *Sensors, Cameras, Syst. Industrial/Scientific Appl. X, vol. 7249, Int. Soc. Opt. Photon.*, 2009, Art. no. 72490U.

[5] S. Ma, S. Gupta, A. C. Ulku, C. Bruschini, E. Charbon, and M. Gupta, “Quanta burst photography,” *ACM Trans. Graph.*, vol. 39, no. 4, pp. 79:1–79:16, 2020.

[6] D. Liu, J. Gu, Y. Hitomi, M. Gupta, T. Mitsunaga, and S. K. Nayar, “Efficient space-time sampling with pixel-wise coded exposure for high-speed imaging,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 2, pp. 248–260, Feb. 2014.

[7] C. Deng et al., “Sinusoidal sampling enhanced compressive camera for high speed imaging,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 4, pp. 1380–1393, Apr. 2019.

[8] K. Dorozynska, “Frequency recognition algorithm for multiple exposures: Snapshot imaging using coded light,” Ph.D. dissertation, Lund University, Lund, Sweden, 2020.

[9] J. N. Martel, L. K. Mueller, S. J. Carey, P. Dudek, and G. Wetzstein, “Neural sensors: Learning pixel exposures for HDR imaging and video compressive sensing with programmable sensors,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 7, pp. 1642–1653, Jul. 2020.

[10] G. Indiveri and R. Douglas, “Neuromorphic vision sensors,” *Science*, vol. 288, no. 5469, pp. 1189–1190, 2000.

[11] L. Patrick, C. Posch, and T. Delbruck, “A 128x 128 120 dB 15 μ s latency asynchronous temporal contrast vision sensor,” *IEEE J. Solid-state Circuits*, vol. 43, no. 2, pp. 566–576, Feb. 2008.

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

[13] G. Gallego et al., “Event-based vision: A survey,” 2019, *arXiv:1904.08405*.

[14] 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 Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3867–3876.

[15] 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 Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 2051–2064, Aug. 2019.

[16] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza, “High speed and high dynamic range video with an event camera,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 6, pp. 1964–1980, Jun. 2021.

[17] M. Mostafavi, L. Wang, and K.-J. Yoon, “Learning to reconstruct hdr images from events, with applications to depth and flow prediction,” *Int. J. Comput. Vis.*, vol. 129, no. 4, pp. 900–920, 2021.

[18] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza, “Events-to-video: Bringing modern computer vision to event cameras,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3852–3861.

[19] P. Duan, Z. Wang, B. Shi, O. Cossairt, T. Huang, and A. Katsaggelos, “Guided event filtering: Synergy between intensity images and neuromorphic events for high performance imaging,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 8261–8275, Nov. 2022.

[20] H. Wässle, “Parallel processing in the mammalian retina,” *Nature Rev. Neurosci.*, vol. 5, no. 10, pp. 747–757, 2004.

[21] R. H. Masland, “The neuronal organization of the retina,” *Neuron*, vol. 76, no. 2, pp. 266–280, 2012.

[22] S. Dong, T. Huang, and Y. Tian, “Spike camera and its coding methods,” in *Proc. Data Compression Conf.*, 2017, p. 437.

[23] 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 Compression Conf.*, 2019, pp. 568–568.

[24] 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*, 2019, pp. 1432–1437.

[25] L. Zhu, S. Dong, J. Li, T. Huang, and Y. Tian, “Retina-like visual image reconstruction via spiking neural model,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1435–1443.

[26] J. Zhao, R. Xiong, and T. Huang, “High-speed motion scene reconstruction for spike camera via motion aligned filtering,” in *Proc. IEEE Int. Symp. Circuits Syst.*, 2020, pp. 1–5.

[27] J. Zhao, R. Xiong, H. Liu, J. Zhang, and T. Huang, “Spk2imgNet: Learning to reconstruct dynamic scene from continuous spike stream,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 11996–12005.

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

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

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

[31] 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.*, 2021, pp. 6358–6367.

[32] C. Posch, D. Matolin, and R. Wohlgenannt, “An asynchronous time-based image sensor,” in *Proc. IEEE Int. Symp. Circuits Syst.*, 2008, pp. 2130–2133.

[33] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck, “A 240× 180 130 dB 3 s latency global shutter spatiotemporal vision sensor,” *IEEE J. Solid-State Circuits*, vol. 49, no. 10, pp. 2333–2341, Oct. 2014.

[34] 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 Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2426–2429.

[35] J. Han et al., “Neuromorphic camera guided high dynamic range imaging,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1727–1736.

[36] M. Guo, J. Huang, and S. Chen, “Live demonstration: A. 768× 640 pixels 200Meps dynamic vision sensor,” in *Proc. IEEE Int. Symp. Circuits Syst.*, 2017, Art. no. 1.

[37] C. Brandli, L. Muller, and T. Delbruck, “Real-time, high-speed video decompression using a frame-and event-based DAVIS sensor,” in *Proc. IEEE Int. Symp. Circuits Syst.*, 2014, pp. 686–689.

[38] L. Pan, C. Scheerlinck, X. Yu, R. Hartley, M. Liu, and Y. Dai, “Bringing a blurry frame alive at high frame-rate with an event camera,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6813–6822.

[39] L. Pan, R. Hartley, C. Scheerlinck, M. Liu, X. Yu, and Y. Dai, “High frame rate video reconstruction based on an event camera,” 2019, *arXiv: 1903.06531*.

[40] C. Scheerlinck, N. Barnes, and R. Mahony, “Continuous-time intensity estimation using event cameras,” in *Proc. Asian Conf. Comput. Vis.*, 2018, pp. 308–324.

[41] Z. W. Wang, W. Jiang, K. He, B. Shi, A. Katsaggelos, and O. Cossairt, “Event-driven video frame synthesis,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops*, 2019, pp. 4320–4329.

[42] H.-C. Liu, F.-L. Zhang, D. Marshall, L. Shi, and S.-M. Hu, “High-speed video generation with an event camera,” *Vis. Comput.*, vol. 33, no. 6/8, pp. 749–759, 2017.

[43] P. Shedligeri and K. Mitra, “Photorealistic image reconstruction from hybrid intensity and event-based sensor,” *J. Electron. Imag.*, vol. 28, no. 6, 2019, Art. no. 063012.

[44] L. Pan, M. Liu, and R. Hartley, “Single image optical flow estimation with an event camera,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1669–1678.

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

[46] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in *Proc. Int. Conf. Med. Image Comput. Comput.- Assist. Intervention*, 2015, pp. 234–241.

[47] W.-S. Lai, J.-B. Huang, O. Wang, E. Shechtman, E. Yumer, and M.-H. Yang, “Learning blind video temporal consistency,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 170–185.

[48] I. Goodfellow et al., “Generative adversarial nets,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[49] S. Pini, G. Borghi, and R. Vezzani, “Learn to see by events: Color frame synthesis from event and RGB cameras,” in *Proc. Int. Joint Conf. Comput. Vis. Imag. Comput. Graph. Theory Appl.*, 2020, pp. 37–47.

[50] L. Wang et al., “Event-based high dynamic range image and very high frame rate video generation using conditional generative adversarial networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10073–10082.

[51] 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 *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8315–8325.

[52] J. Choi et al., “Learning to super resolve intensity images from events,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2768–2776.

[53] J. Dai et al., “Deformable convolutional networks,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 764–773.

[54] X. Zhu, H. Hu, S. Lin, and J. Dai, “Deformable convnets V2: More deformable, better results,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9308–9316.

[55] M. S. Goldman, P. Maldonado, and L. Abbott, “Redundancy reduction and sustained firing with stochastic depressing synapses,” *J. Neurosci.*, vol. 22, no. 2, pp. 584–591, 2002.

[56] M. A. Bourjaily and P. Miller, “Dynamic afferent synapses to decision-making networks improve performance in tasks requiring stimulus associations and discriminations,” *J. Neuriophysiol.*, vol. 108, no. 2, pp. 513–527, 2012.

[57] D. L. Cook, P. C. Schwindt, L. A. Grande, and W. J. Spain, “Synaptic depression in the localization of sound,” *Nature*, vol. 421, no. 6918, pp. 66–70, 2003.

[58] L. Abbott and W. G. Regehr, “Synaptic computation,” *Nature*, vol. 431, no. 7010, pp. 796–803, 2004.

[59] Y. Igarashi, M. Oizumi, and M. Okada, “Theory of correlation in a network with synaptic depression,” *Phys. Rev. E*, vol. 85, no. 1, 2012, Art. no. 016108.

[60] R. P. Costa, P. J. Sjostrom, and M. C. Van Rossum, “Probabilistic inference of short-term synaptic plasticity in neocortical microcircuits,” *Front. Comput. Neurosci.*, vol. 7, 2013, Art. no. 75.

[61] Smart vision lights, 2013. [Online]. Available: http://bit.ly/2kpuWZC

[62] A. Moorthy and A. Bovik, “A two-step framework for constructing blind image quality indices,” *IEEE Signal Process. Lett.*, vol. 17, no. 5, pp. 513–516, 2010.

[63] L. Xi, L. Guosui, and J. Ni, “Autofocusing of ISAR images based on entropy minimization,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 35, no. 4, pp. 1240–1252, Oct. 1999.

[64] A. Mittal, R. Soundararajan, and A. C. Bovik, “Making a “completely blind” image quality analyzer,” *IEEE Signal Process. Lett.*, vol. 20, no. 3, pp. 209–212, Mar. 2012.

[65] A. Mittal, A. K. Moorthy, and A. C. Bovik, “No-reference image quality assessment in the spatial domain,” *IEEE Trans. Image Process.*, vol. 21, no. 12, pp. 4695–4708, Dec. 2012.

[66] N. Venkatanath, D. Praneeth, M. C. Bh, S. S. Channappayya, and S. S. Medasani, “Blind image quality evaluation using perception based features,” in *Proc. IEEE 21st Nat. Conf. Commun.*, 2015, pp. 1–6.

[67] L. F. Abbott, J. Varela, K. Sen, and S. Nelson, “Synaptic depression and cortical gain control,” *Science*, vol. 275, no. 5297, pp. 221–224, 1997.

Yajing Zheng received the BS degree from Sichuan University, Sichuan, China, in 2017, and the PhD 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 network.

Lingxiao Zheng is currently working toward the undergraduation degree in school of EECS, Peking University. He is currently advised by prof. Tiejun Huang, and his research interest is visual information processing and neuromorphic computing.

Zhaofei Yu (Member, IEEE) received the BS degree from the Hong Shen Honors School, College of Optoelectronic Engineering, Chongqing University, Chongqing, China, in 2012 and the PhD 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.

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 PhD degree in pattern recognition and intelligent system from Huazhong (Central China) from the 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 area includes video coding, image understanding, digital right management, and digital library. He has authored or co-authored more than 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.

Song Wang (Senior Member, IEEE) received the PhD 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.