Boosting Spike Camera Image Reconstruction from a Perspective of Dealing with Spike Fluctuations

Rui Zhao\textsuperscript{1,2} Ruiqin Xiong\textsuperscript{1,2}+ Jing Zhao\textsuperscript{1,2} Jian Zhang\textsuperscript{3} Xiaopeng Fan\textsuperscript{4} Zhaofei Yu\textsuperscript{1} Tiejun Huang\textsuperscript{1,2} \\
\textsuperscript{1}School of Computer Science, Peking University \hspace{1cm} \textsuperscript{2}National Key Laboratory for Multimedia Information Processing, Peking University \hspace{1cm} \textsuperscript{3}School of Electronic and Computer Engineering, Peking University \hspace{1cm} \textsuperscript{4}School of Computer Science and Technology, Harbin Institute of Technology \\
ruizhao@stu.pku.edu.cn \hspace{1cm} \{rqxiong, jzhaopku, zhangjian.sz, yuzf12, tjhuang\}@pku.edu.cn \hspace{1cm} fxp@hit.edu.cn

Abstract

As a bio-inspired vision sensor with ultra-high speed, spike cameras exhibit great potential in recording dynamic scenes with high-speed motion or drastic light changes. Different from traditional cameras, each pixel in spike cameras records the arrival of photons continuously by firing binary spikes at an ultra-fine temporal granularity. In this process, multiple factors impact the imaging, including the photons’ Poisson arrival, thermal noises from circuits, and quantization effects in spike readout. These factors introduce fluctuations to spikes, making the recorded spike intervals unstable and unable to reflect accurate light intensities. In this paper, we present an approach to deal with spike fluctuations and boost spike camera image reconstruction. We first analyze the quantization effects and reveal the unbiased estimation attribute of the reciprocal of differential of spike firing time (DSFT). Based on this, we propose a spike representation module to use DSFT with multiple orders for fluctuation suppression, where DSFT with higher orders indicates spike integration duration between multiple spikes. We also propose a module for inter-moment feature alignment at multiple granularities. The coarser alignment is based on patch-level cross-attention with a local search strategy, and the finer alignment is based on deformable convolution at the pixel level. Experimental results demonstrate the effectiveness of our method on both synthetic and real-captured data. The source code and dataset are available at https://github.com/ruizhao26/BSF.

1. Introduction

Vision technology has undergone remarkable advancements recently. Machine vision in scenes with high-speed motion or drastic light changes is a key challenge in emerging applications such as autonomous driving [14], unmanned aerial vehicles [72], and assistant referees in sports [18]. Traditional digital cameras typically record scenes with a frame rate of 30 Hz \(~\sim\) 120 Hz, which is inadequate to fulfill the demands of these applications.

Neuromorphic cameras (NeurCams) are a new kind of bio-inspired vision sensor designed to handle the above-mentioned challenges. NeurCams record light intensity at an ultra-high temporal resolution. One kind of NeurCams are event cameras [22, 29, 35]. They employ a differential imaging model, in which each pixel records the scene by outputting events whenever the change in light intensity in the logarithmic domain surpasses a certain threshold.

Different from event cameras, spike cameras [8, 18] employ an integral imaging model. Each pixel accumulates photons from scenes independently. Whenever the accumulation of a pixel reaches a predefined threshold, it fires a spike and restarts the accumulation. The reading out of spikes is at a high temporal resolution of 40 kHz. Thus, spike cameras can describe light intensities in a very fine temporal granularity by reporting each pixel’s...
status of receiving photons at a very high frequency. Based on these characteristics, spike cameras can handle scenes with high-speed motion or drastic light changes. Recently, many tasks have been researched for spike cameras, including image reconstruction [9, 10, 60, 64, 69, 73], coding [8, 12, 74], object tracking [21, 70, 79], optical flow estimation [17, 48, 55, 62, 67], and depth estimation [44, 57].

Reconstructing clear images from spikes at an arbitrary time is a key task for spike cameras. However, the recording of photons is affected by multiple factors. First, the arrival of photons follows a Poisson process. Second, the circuits introduce thermal noise. Third, the spike reading is controlled by a clock signal, which introduces quantization effects. These factors introduce fluctuations and randomness to spikes, i.e., even when the light intensity is constant, the integration period of each spike changes over time. Thus, reconstructing clear images from spikes is challenging.

Several spike camera image reconstruction (SCIR) methods have been proposed from different perspectives. These methods can be roughly categorized into filtering-based methods (FM) [61, 65, 73], neuronic-models-based method (NMM) [69, 71, 75, 77], and deep-learning-based method (DLM) [3, 64]. Many of them consider spatial-temporal information aggregation to obtain clear images. However, the fluctuations in spikes are not fully explored.

In this paper, we propose a deep neural network that is boosted from a perspective of dealing with spike fluctuations to obtain clear images. We first analyze the characteristics of the quantization effects of spikes. The differential of spike firing time (DSFT) [67] measures the duration of spike integral to which a spatial-temporal point belongs. We reveal the reciprocal of DSFT can offer an unbiased estimation of light intensities under quantization effects when the input is stable. Based on this attribute, we propose a multi-order DSFT fusion (MODF) module for spike representation. The MODF extracts features from the reciprocal of DSFT to pursue unbiased light information. Further considering photons’ Poisson arrival, we extend DSFT to higher orders. Higher-order DSFT represents intervals of multiple spikes, describing stabilized light intensities in a longer temporal scale. The MODF uses multi-order DSFT to approximate light intensities with higher fidelity. It extracts features from the reciprocal of DSFT with multiple orders and fuses them in pursuit of reducing the influence of fluctuations in spikes.

Further considering the impact of the dynamic changes of light and motions, we propose a multi-granularity alignment (MGA). It aligns features in a pyramidal fashion. At each level, the feature is aligned by coarse-granularity alignment (CA) and fine-granularity alignment (FA). The CA aligns the reference features towards the key features using a patch-level cross-attention with a local search strategy. Based on the initial patch-level alignment of CA, the FA aligns features at pixel level using deformable convolutions. We synthesize a benchmark dataset that incorporates multiple light intensities and Poisson effects to evaluate SCIR methods. Our method achieves state-of-the-art performance on both synthetic and real-captured data. Our key contributions can be summarized as follows.

1. We analyze the characteristics of spikes under the quantization effects and reveal the unbiased estimation attributes of the reciprocal of DSFT.
2. We propose a network with multi-order DSFT fusion (MODF) and multi-granularity alignment (MGA) modules. These two modules contribute to our model for obtaining more clear images from spikes.
3. Experiments conducted on synthetic and real-captured data demonstrate that the proposed method achieves state-of-the-art performance for SCIR.

2. Related Work

Image reconstruction for spike cameras.

Filtering-based methods. TFP and TFI [73] use the firing rate in a temporal window and the firing interval of a spike to represent the light intensity, respectively. Zhao et al. [61, 65] propose to align the preliminary reconstructed images with optical flow [3] and fuse aligned images using a temporal auto-regressive model [47] for long-term filtering. MGSR [63] predicts super-resolved images from spikes based on the mapping of coordinates with different scales based on optical flow.

Neuronic-model-based methods. SNM [75, 77] is a neuronic model with three-layer leaky integrate and fire (LIF) neurons. The model is trained based on the spiking-timing-dependent plasticity (STDP) mechanism [2]. TFP and TFI [69, 71] use the short-term plasticity (STP) mechanism [27] to construct the relationship between binary spikes and light intensities based on postsynaptic potentials.


Image reconstruction for other emerging cameras.

Event cameras record changes in light intensities in the logarithmic domain using polar events. There are optimization-based [1, 30] and deep-learning-based methods such as E2Vid [36, 37], FireNet [39], and ETNet [46]
proposed for event-based image reconstruction. Since event cameras mainly record moving objects, reconstructing images with only events may cause errors of estimation for global contrast. Recently, more works [23, 32, 33, 40, 41] use traditional images to offer light information of static regions to help event cameras reconstruct clear images.

Quantum image sensor (QIS) [26] is a kind of photon-counting sensor. It can be divided into CCD / CMOS (CIS) QIS [25] and single-photon avalanche detector (SPAD) QIS [54]. QIS aims at photon-level detection and can work well in ultra-low-light scenes with very few photons [6, 11, 13, 34, 38].

3. Working Mechanism of Spike Camera

The working mechanism of spike cameras is shown in Fig. 2. Each pixel of the spike camera comprises three main components: a photon receptor, an integrator, and a comparator. The incoming photons are captured by the photon receptor and accumulated by the integrator. Whenever the number of accumulated photons reaches a predefined threshold \( \theta \), a spike is fired, and the integrator is reset. Suppose \( L = L(x, t) \) is the expected number of arrival photons at a pixel area per unit time, where \( x = (x, y) \) is spatial coordinate and \( t \) is time stamp. The accumulation in the integrator can be formulated as:

\[
A(x, t) = \int_0^t \alpha P(L(x, \tau)) \, d\tau \mod \theta,
\]

where \( A \) is the accumulation, \( \alpha \) is the quantum conversion coefficient of photons, and \( P \) means Poisson sampling:

\[
P(X = k; J) = \frac{e^{-J} J^k}{k!}, \quad k \in \mathbb{Q},
\]

where \( P \) means probability, and \( J \) is the expected number of arrival photons. \( \mathbb{Q} \) means the natural number set. The thermal noises are omitted here.

Spike cameras read spike arrays out at an ultra-high speed of up to 40 kHz. Their outputs can be formulated as \( S \in \mathbb{B}^{H \times W \times T} \), where \( \mathbb{B} \) means the binary domain. The threshold \( \theta \) is configured to ensure that no more than one spike can be fired within any spike-reading interval.

Figure 2. Working mechanism of spike cameras.

4. Methods

4.1. Overall Architecture

The overall architecture of the proposed method is shown in Fig. 3. We segment 5 spike sub-streams \( \{S_t\}_{t=0}^{2} \) in the temporal axis from the spike stream \( S \in \mathbb{B}^{H \times W \times T} \) to reconstruct the scene at moment \( t_0 \), where \( i \) is the index indicating the time. \( S_i \) is centered at moment \( t_i \): \( S_i(x) = \{S(x, t)\}_{t=t_i-w_h}^{t_i+w_h} \), where \( w_h \) is the window radius. We refer to \( t_0 \) as key moment and refer to \( \{t_i\}_{i=1,2} \) as reference moments. These five spike sub-streams are extracted into representations \( \{R_i\}_{i=1,2} \) through the multi-order DSFT fusion (MODF) module, where DSFT is differential of the spike firing time [67]. Then the representations are encoded into features \( \{F_i\}_{i=1,2} \). Features at reference moments are then aligned to the key moment by the multi-granularity alignment (MGA) module. The reconstructed image is obtained by fusing the key feature and aligned reference features using several convolutional layers.

4.2. Multi-order DSFT Fusion

The purpose of spike representation is to extract initial light information from spikes. In the imaging process of spike cameras, the recording of photons has randomness due to the Poisson process of photons’ arrival, making the time for the number of photons to reach the threshold exhibit randomness. Besides, since the spike readout is controlled by a clock signal, the reading time and firing time are usually slightly different, introducing quantization effects. In short, the periods of spikes are unstable and do not directly reflect the light intensity, i.e., the spikes are fluctuating.

To extract stable light intensities under the impacts of the above-mentioned fluctuation factors of spikes, we design a multi-order DSFT fusion module based on three propositions as follows. Note that we focus on the processing of a single spike sub-stream in this subsection, we omit the subscript index \( i \) that indicates the time.

**Proposition 1**: Using DSFT as input. The concept of differential of spike firing time (DSFT) is proposed in Spike2Flow [67]. As shown in Fig. 4, the DSFT of each
point in the 3D coordinate represents the duration of the spike integration period it belongs to. In other words, the DSFT measures the time interval between the previous and the next spike. In this paper, the version of DSFT mentioned above is named (1,1)-order DSFT \( D_{\text{SFT}}^{(1,1)} \).

In binary spikes, “1” represents light intensities in a time process but not a time point. Thus, the binary spikes cannot reflect light intensities at the reading moment in a simple way. The light intensity at the reading moment of each “1” can be different. In DSFT, the value represents the current light intensity of each point in the 3D coordinate. Thus, DSFT offers more relative information about light than binary spikes, and we use DSFT to pursue light intensities contained in physical reality.

**Proposition 2: Processing DSFT in reciprocal domain.**

In this part, we analyze the DSFT and quantization effects. In the analyses of this proposition, we omit the thermal noise and assume the photons’ arrival is constant. Suppose the firing threshold of the spike camera is \( \theta \), the spike-reading time interval is \( T_{\text{r}} \), and the incoming rate of photons is \( \zeta \). During a spike-reading time interval, the number of photons reaching a pixel area is \( L = \zeta T_{\text{r}} \). We can infer the light intensity through the ratio of \( \theta \) and \( D_{\text{SFT}}^{(1,1)} \): \( \bar{L} = \theta / D_{\text{SFT}}^{(1,1)} \). When \( \theta \mod L = 0 \), the \( D_{\text{SFT}}^{(1,1)} \) is stable. However, as shown in Fig. 5, when \( \theta \mod L \neq 0 \), even when the photon’s arrival is constant, the \( D_{\text{SFT}}^{(1,1)} \) has different values up and down around \( \theta / L \). This is the value instability attribute of DSFT, which contributes to the fluctuations of spikes. To handle such issues caused by quantization effects, we propose the following theorem and design a spike representation module based on the theorem.

**Theorem 4.1.** Suppose the symbolic definition is the same as above and \( \theta \mod L \neq 0 \). When the photons’ arrival is constant, The (1,1)-order DSFT has only two values \( \{ [\theta / L], [\theta / L] \} \) and its distribution is as follows:

\[
\begin{align*}
\Pr \{ D_{\text{SFT}}^{(1,1)} = [\theta / L] \} &= p_1 = ( [\theta / L] - \theta / L ) \cdot [\theta / L] / \theta / L \\
\Pr \{ D_{\text{SFT}}^{(1,1)} = [\theta / L] \} &= p_2 = ( \theta / L - [\theta / L] ) \cdot [\theta / L] / \theta / L
\end{align*}
\]

where \( \Pr \{ \} \) means probability.

The proof of the above theorem is in Sec. 7 of the supplementary material (abbreviated as supp hereafter). Eq. (3) shows that when photons’ arrival is constant, the \( D_{\text{SFT}}^{(1,1)} \) has only two values: the ceiling value and the floor value of \( \theta / L \). This distribution indicates the estimated \( \bar{L} = \theta / D_{\text{SFT}}^{(1,1)} \) fluctuate. Although the \( D_{\text{SFT}}^{(1,1)} \) is unstable when \( \theta \mod L \neq 0 \), we find that \( 1 / D_{\text{SFT}}^{(1,1)} \) is an unbiased estimation of \( L / \theta \) according to Theorem 4.1:

\[
E \left( \frac{1}{D_{\text{SFT}}^{(1,1)}} \right) = \frac{1}{\theta / L} \cdot p_1 + \frac{1}{\theta / L} \cdot p_2 = \frac{L}{\theta}. \tag{4}
\]

Through the reciprocal of \( D_{\text{SFT}}^{(1,1)} \), the unbiased estimation of \( L \) can be obtained. Note that since \( D_{\text{SFT}}^{(1,1)} \) is not constant, \( E(D_{\text{SFT}}^{(1,1)}) \) is a biased estimation for \( \theta / L \) according to harmonic mean inequality (introduced in Sec. 7 of supp). Thus, we process the DSFT in its reciprocal domain to pursue an unbiased estimation of light intensities.

**Proposition 3: Fusing DSFT with multiple orders.**

Considering Poisson noises, motion, and light change, the constant photon-arrival assumption is practically limited in reality. In pursuit of stable light information under these factors that contribute to spike fluctuations, we expanded the (1,1)-order DSFT to \( \{ n_1, n_2 \} \)-order DSFT:

\[
D_{\text{SFT}}^{(n_1, n_2)}(x, t) = T_{\text{next}}^{(n_2)}(x, t) - T_{\text{prev}}^{(n_1)}(x, t)
\]

\[
= \min \left\{ \tau \mid \sum_{k=t+1}^{\tau} S(x, k) = n_2, \tau > t \right\} - \max \left\{ \tau \mid \sum_{k=t}^{\tau} S(x, k) = n_1, \tau \leq t \right\}, \tag{5}
\]

where \( T_{\text{next}}^{(n_2)}(x, t) \) is the time stamp of the \( n_2 \)-th next spike at pixel \( x \) and time stamp \( t \). \( T_{\text{prev}}^{(n_1)}(x, t) \) is the time stamp of the \( n_1 \)-th previous spike, where spike at \( x \) and \( t \) is counted into \( T_{\text{prev}}^{(n_1)}(x, t) \) if \( S(x, t) = 1 \). The \( D_{\text{SFT}}^{(n_1, n_2)}(x, t) \) represents the spike interval between the \( n_1 \)-th previous spike and the \( n_2 \)-th next spike. DSFT with higher order can suppress the Poisson effects to stabilize fluctuations. Thus, we use DSFT with multiple orders to obtain stable light information.
Based on the above discussion, we propose a multi-order DSFT fusion (MODF) module. As shown in Fig. 6. We use $\{D^{(n_1, n_2)}_{SFT}\}_{n_1, n_2 \in \{1, 2\}}$ as input. Similar to spike sub-streams, all the DSFT streams have a length of $2w_t + 1$. All the DSFT streams are taken reciprocal and normalized by multiplying with interval numbers $\xi_l = n_l + n_2 - 1$ of the corresponding order. Then the reciprocal DSFT are extracted to be features through a weight-shared feature extractor $\Phi$. Compared with DSFT with higher orders, $D^{(1, 1)}_{SFT}$ undergoes less motion blur since it contains the fewest spike intervals. Thus, we use features from $D^{(1, 1)}_{SFT}$ as foundation and use features from higher-order DSFT to enhance the information in $D^{(1, 1)}_{SFT}$. This process can be formulated as:

$$R = \Phi \left( \frac{1}{D^{(1, 1)}_{SFT}} \right) + \Upsilon \left( \text{Cat} \left( \{ \psi_1, \cdots, \psi_3 \} \right) \right),$$

where

$$\{\xi_l\}_{l=1}^3 = \{\xi_1, \xi_2, \xi_3\} = \{(1, 2), (2, 1), (2, 2)\},$$

$$\{\chi_l\}_{l=1}^3 = \{\sum_{l=1}^3 (\xi_l - 1)\} = \{2, 2, 3\},$$

where $\Phi$ is the feature extractor, $\Psi^{(n_1, n_2)}$ is for extracting information from $D^{(n_1, n_2)}_{SFT}$, $\xi_l$ is the index of $(n_1, n_2)$, $\Upsilon$ means fusion operation through convolution, and $\text{Cat}$ is channel-wise concatenation. The representations $\{R\}_{i=2}^2$ are then encoded to be features $\{F\}_{i=2}^2$, where the encoder consists of 4 residual blocks.

### 4.3. Multi-granularity Alignment

The utilization of long-term information is key to reconstructing high-quality images from spike streams. We achieve this objective by aligning information from spike sub-streams at different moments: we align the information from features at moments $\{t_l\}_{l=1, 2}$ (reference features) to the $t_0$ moment (key feature) and then fuse all these features.

The MODF module aims to extract stable representations of light intensities from binary spikes. However, multiple factors contribute to the fluctuations of spikes in practice. These factors make corresponding areas in features at different moments have different values, improving the matching error in the alignment process.

To realize robust alignment, we propose a multi-granularity alignment (MGA) module with a pyramid structure. In each pyramidal level, the alignment is from coarse granularity to fine granularity. As shown in the left of Fig. 7, the input features are downsampled by convolutions to construct a pyramid. Suppose $F^\text{key}_{l_i}$ is key feature at $t_0$ moment at the $\ell$-th pyramidal level, and $F^\text{ref}_{l_i}$ is reference feature at $t_i$ moment at the $\ell$-th pyramidal level. $F^\text{key}_{l_i}$ and $F^\text{ref}_{l_i}$ are initial input of MGA. In each pyramidal level, the reference feature is first aligned by coarse-grained alignment (CA) and then aligned by fine-grained alignment (FA). The CA locally aligns features at a patch level, and the FA further aligns features at the pixel level. In CA, we propose a Cross-Attentional Patch-level Alignment (CAPA) with a local search strategy for initial coarse alignment. The design propositions and details of the CAPA are as follows.

The aim of alignment is to align the reference features $F^\text{ref}_{l_i}$ to spatial coordinates of the key feature $F^\text{key}_{l_i}$, i.e., searching for corresponding contents of $F^\text{key}_{l_i}$ in $F^\text{ref}_{l_i}$. Considering the above analysis, we design a cross-attentional operation in CAPA. We use $F^\text{key}_{l_i}$ to construct the Query, and use $F^\text{ref}_{l_i}$ to construct the Key and Value. Patch-level operation is a classic auxiliary strategy in pixel-level tasks [24, 52, 53]. Considering the light intensity information is more stable in a local region than in a pixel, we design CA to be patch-level to provide a foundation for subsequent pixel-level alignment. Thus, the embedding procedure in the CAPA can be formulated as:

$$Q^p_{l_i} = Z[Q_{l_i}] = Z[W_{Q}F^\text{key}_{l_i}],$$

$$K^p_{l_i} = Z[K_{l_i}] = Z[W_{K}F^\text{ref}_{l_i}],$$

$$V^p_{l_i} = Z[V_{l_i}] = Z[W_{V}F^\text{ref}_{l_i}],$$

where $Z$ is the patchification operation with $s_p \times s_p$ size. Suppose after padding for patchification, the spatial resolution of feature $F^\text{key}_{l_i}$ and $F^\text{ref}_{l_i}$ at the $\ell$-th level is $H_{\ell} \times W_{\ell}$, the spatial resolution of $Q^p_{l_i}, K^p_{l_i}$ and $V^p_{l_i}$ is $\tilde{H}_{\ell} \times \tilde{W}_{\ell}$, where $\tilde{H}_{\ell} = H_{\ell}/s_p$ and $\tilde{W}_{\ell} = W_{\ell}/s_p$. In this way, we can real-

**Figure 7.** Illustration of the multi-granularity alignment (MGA) module for aligning $F^\text{ref}_{l_i}$ to $F^\text{key}_{l_i}$. On the left is the architecture of the MGA module. On the right is the architecture of the cross-attentional patch-level alignment (CAPA) module in the MGA.
ize transforming \( \mathbf{F}_i^{ref} \) to approximate \( \mathbf{F}_i^{key} \) according to the relationship of “using \( \mathbf{F}_i^{key} \) to query \( \mathbf{F}_i^{ref} \). In other words, the cross-attention implements alignment from \( \mathbf{F}_i^{ref} \) towards \( \mathbf{F}_i^{key} \). Based on the above discussion, we design a patch-level local search strategy in CAPA. The attention operation can be formulated as follows:

\[
\hat{\mathbf{V}}^p_{i,\ell}(x) = \mathcal{A}(\mathbf{V}^p_{i,\ell}(x)) \sigma \left( \frac{\mathcal{Q}^p_{i,\ell}(x) \mathcal{A}(\mathbf{K}^p_{i,\ell}(x))}{\sqrt{C_k}} \right).
\]

(10)

where \( x \in \mathbb{Q}^{h_i \times w_i} \). \( \mathcal{A}(\cdot) \) is the local sampling operator with \( k_p \times k_p \) size. \( \sigma \) means softmax on the dimension with \( k^2_p \) channels as shown in the right of Fig. 7. \( C_k \) is the channel number of the query, key, and value. The \( \mathcal{A}(\cdot) \) can be formulated as:

\[
\mathcal{A}(\mathbf{V}^p_{i,\ell}(x)) = \{ \mathbf{V}^p_{i,\ell}(x + \delta) \}_{\delta \in \mathcal{N}(x, k_p)}.
\]

(11)

where \( \mathcal{N}(x, k_p) \) is a \( k_p \times k_p \) area centered on \( x \). Through \( \mathcal{A} \), we construct key vectors over a larger range for each query vector, which realizes search operation in \( \mathbf{F}^{ref}_{i,\ell} \) for alignment towards \( \mathbf{F}_i^{key} \). Based on the patch-level operation and the local search strategy, CAPA implements alignment in a coarse granularity with more stable light information than the local search strategy, CAPA implements alignment in a coarse-grained alignment. The illustration of CAPA and the size of each tensor are shown on the right of Fig. 7. The coarse-grained alignment \( \tilde{\mathbf{V}}^p_{i,\ell} \) is obtained through inverse patchification of \( \hat{\mathbf{V}}^p_{i,\ell} \):

\[
\tilde{\mathbf{V}}^p_{i,\ell}(x) = \mathbf{F}^{ref}_{i,\ell}(x) + \lambda Z^{-1} \left[ \mathbf{V}^p_{i,\ell}(x) \right], \quad x \in \mathbb{Q}^{h_i \times w_i},
\]

(12)

where \( Z^{-1} \) means inverse patchification operation and \( \lambda \) is a learnable parameter. Through CAPA, each pixel in \( \mathbf{F}^{ref}_{i,\ell} \) is aligned to \( \mathbf{F}_i^{key} \) based on \( k_p \times k_p \) patches centered at \( \mathbf{F}_i^{key} \).

The coarse aligned \( \mathbf{F}^{ref}_{i,\ell} \) is then aligned in fine-granularity by deformable convolutions (DCN) [7, 78]:

\[
\tilde{\mathbf{F}}^{ref}_{i,\ell}(x) = \sum_{\delta \in \mathcal{N}(x, k_d)} K(\delta) \mathbf{F}^{ref}_{i,\ell}(x + \delta + O_i(x, \delta)) \mathcal{M}_i(x, \delta),
\]

(13)

where the subscript \( \ell \) is omitted. \( \mathbf{F}^{ref} \) is the fine-grained aligned feature. \( k_d \) is the kernel size of the DCN. \( O \) and \( \mathcal{M} \) are the offset and mask of the DCN, respectively. They are obtained from the initial aligned \( \mathbf{F}^{ref} \) and \( \mathbf{F}^{key} \), and they are passed and fused between different pyramidal levels by upsampling as shown in the left of Fig. 7. Besides, the aligned \( \tilde{\mathbf{F}}_i \) is also passed and fused between different levels by upsampling. In short, the MGA aligns features at different scales and granularities. The CAPA uses patch-level light information to reduce matching errors caused by spike fluctuations, aiming to provide reliable foundations for fine-grained alignment.

The key feature and aligned reference features are then fused through the reconstruction layer, which is composed of several layers of convolutions and ReLU:

\[
\tilde{I}(t_0) = \text{Recon} \left( \text{Cat} \left( \left\{ \tilde{\mathbf{F}}_{i,\ell} \right\}_{i=2}^{-1}, F_0, \left\{ \tilde{\mathbf{F}}_{i,\ell}^2 \right\}_{i=1} \right) \right).
\]

(14)

5. Experiments

5.1. Data Preparation

To synthesize spike data, we refer to the simulation procedure proposed in literature [64] and extend the simulator. First, we use an advanced video frame interpolation method EMA-VFI [56] to achieve high-fidelity continuous scene generation. Second, based on the temporally interpolated continuous frames, we set a parameter \( \eta \) to simulate different levels of light intensity. Third, we simulate the Poisson process of photons’ arrival.

Since the aperture size and spike firing threshold are adjustable, the settings of parameters in the simulation pipeline are without loss of generality. Based on the continuous scenes, we establish 3 different light intensity factors \( \eta = \{1.00, 0.75, 0.50\} \). \( \gamma \) is the conversion of pixel values to the expected number of arriving photons within a single pixel during one readout interval \( T_r \), and we set \( \gamma \) as 60. We set the quantum conversion factor \( \alpha \) as 0.7. Suppose a pixel value at moment \( t \) is \( I(t) \). During interval \( (t, t + T_r) \), the integral in the accumulator is \( \Delta \mathbf{A}(x) = \alpha \mathcal{P}(\eta I(t, x)) \), where \( \mathcal{P} \) is Poisson sampling. We set the firing threshold as \( \theta = \max(I) \cdot \gamma \). Since \( \alpha < 1 \), the number of fired spikes within \( T_r \) will not exceed one.

We use the REDS dataset [31] at 120 FPS and \( 1280 \times 720 \) resolution to generate the REDS-SCIR dataset. In REDS, there are 240 scenes for training and 30 scenes for evaluation. For each training scene, we crop it to 12 scenes at \( 256 \times 256 \) resolution. For each evaluation scene, we crop it to 4 scenes at \( 384 \times 512 \) resolution. Given that we have 3 light intensity factors \( \eta = \{1.00, 0.75, 0.50\} \), there are \( 240 \times 12 \times 3 = 8640 \) and \( 30 \times 4 \times 3 = 360 \) scenes for training and evaluation, respectively. For each scene, we use 40 frames to generate 400 spike frames. We use the high-quality gray images from the REDS as ground truths.

5.2. Implementation Details

In the experiments, we set the patchification size \( s_p \) in CAPA as 3. Since CAPA is used for initial alignment, we set kernel size \( k_p \) of local sampling operation as 3 for simplicity. During training, we randomly crop the spikes to \( 96 \times 96 \) spatially, and we use random horizontal and vertical flips as well as random rotation for data augmentation. The network is trained for 60 epochs with a batch size of 8. We use Adam optimizer [19] with \( \beta_1 = 0.9 \) and \( \beta_2 = 0.999 \). The learning rate is initially set as \( 1e-4 \) and scaled by 0.5 every 10 epoch. The network is trained based on \( \ell_1 \) loss between the normalized estimated \( \tilde{I}(t_0) \) and its ground truth \( I_{gt}(t_0) \):

\[
\mathcal{L} = \| \tilde{I}(t_0) / (\alpha \cdot \eta) - I_{gt}(t_0) \|_1.
\]

(15)

5.3. Comparison with Existing Methods

We divide existing methods for comparison into 4 parts: (A) traditional training-free methods, (B) event-based image re-
Table 1. Quantitative results on the evaluation set on REDS-SCIR with full-reference metrics. ♣ means the network is retrained on REDS-SCIR with the same settings as ours. Best in red bold and second best in blue. † and ‡ means larger and smaller is better, respectively.

<table>
<thead>
<tr>
<th>Method</th>
<th>PSNR †</th>
<th>SSIM †</th>
<th>LPIPS ‡</th>
<th>PSNR †</th>
<th>SSIM †</th>
<th>LPIPS ‡</th>
<th>PSNR †</th>
<th>SSIM †</th>
<th>LPIPS ‡</th>
<th>Param (M)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Part (A)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TFP [73]</td>
<td>27.27</td>
<td>0.711</td>
<td>0.265</td>
<td>26.73</td>
<td>0.669</td>
<td>0.300</td>
<td>25.62</td>
<td>0.581</td>
<td>0.370</td>
<td>—</td>
</tr>
<tr>
<td>TFI [73]</td>
<td>23.55</td>
<td>0.634</td>
<td>0.239</td>
<td>24.77</td>
<td>0.673</td>
<td>0.293</td>
<td>26.77</td>
<td>0.713</td>
<td>0.249</td>
<td>—</td>
</tr>
<tr>
<td>TFPST [69]</td>
<td>20.35</td>
<td>0.678</td>
<td>0.270</td>
<td>19.62</td>
<td>0.685</td>
<td>0.252</td>
<td>21.10</td>
<td>0.707</td>
<td>0.247</td>
<td>—</td>
</tr>
<tr>
<td>MAHTF [65]</td>
<td>29.57</td>
<td>0.879</td>
<td>0.112</td>
<td>30.07</td>
<td>0.884</td>
<td>0.113</td>
<td>29.65</td>
<td>0.869</td>
<td>0.136</td>
<td>—</td>
</tr>
<tr>
<td><strong>Part (B)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FireNet [39]♣</td>
<td>34.38</td>
<td>0.922</td>
<td>0.077</td>
<td>33.87</td>
<td>0.911</td>
<td>0.084</td>
<td>32.62</td>
<td>0.884</td>
<td>0.105</td>
<td>0.038</td>
</tr>
<tr>
<td>ETNet [46]♣</td>
<td>33.24</td>
<td>0.918</td>
<td>0.082</td>
<td>32.85</td>
<td>0.909</td>
<td>0.089</td>
<td>31.96</td>
<td>0.889</td>
<td>0.109</td>
<td>22.179</td>
</tr>
<tr>
<td><strong>Part (C)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSSL [5]♣</td>
<td>32.60</td>
<td>0.920</td>
<td>0.088</td>
<td>32.09</td>
<td>0.907</td>
<td>0.097</td>
<td>31.00</td>
<td>0.879</td>
<td>0.122</td>
<td>2.385</td>
</tr>
<tr>
<td>SSSL [5]♣</td>
<td>33.94</td>
<td>0.923</td>
<td>0.075</td>
<td>33.27</td>
<td>0.909</td>
<td>0.088</td>
<td>32.01</td>
<td>0.883</td>
<td>0.116</td>
<td>2.385</td>
</tr>
<tr>
<td>Spk2ImgNet [64]</td>
<td>35.21</td>
<td>0.953</td>
<td>0.036</td>
<td>34.70</td>
<td>0.945</td>
<td>0.044</td>
<td>33.75</td>
<td>0.926</td>
<td>0.064</td>
<td>3.904</td>
</tr>
<tr>
<td>Spk2ImgNet [64]</td>
<td>39.16</td>
<td>0.966</td>
<td>0.024</td>
<td>38.27</td>
<td>0.958</td>
<td>0.032</td>
<td>36.59</td>
<td>0.940</td>
<td>0.051</td>
<td>3.904</td>
</tr>
<tr>
<td>WGSE [58]♣</td>
<td>35.21</td>
<td>0.950</td>
<td>0.039</td>
<td>34.98</td>
<td>0.947</td>
<td>0.042</td>
<td>34.11</td>
<td>0.931</td>
<td>0.057</td>
<td>3.806</td>
</tr>
<tr>
<td>WGSE [58]♣</td>
<td>38.97</td>
<td>0.964</td>
<td>0.027</td>
<td>38.23</td>
<td>0.957</td>
<td>0.034</td>
<td>36.75</td>
<td>0.940</td>
<td>0.049</td>
<td>3.806</td>
</tr>
<tr>
<td>BSF (Ours)</td>
<td>39.76</td>
<td>0.970</td>
<td>0.021</td>
<td>39.09</td>
<td>0.964</td>
<td>0.027</td>
<td>37.76</td>
<td>0.951</td>
<td>0.040</td>
<td>2.477</td>
</tr>
</tbody>
</table>

Figure 8. The impact of the window length on TFP [73].

Figure 9. The real-captured scene used for evaluation. Each scene is shown through a frame reconstructed by our method.

construction networks, (C) self-supervised SCIR methods, and (D) supervised SCIR methods. We do not compare SNM [77] on REDS-SCIR since the official program will be unresponsive with spikes. Given that spikes are also streaming data, we select them for comparison. We use three full-reference metrics: PSNR, SSIM [45], and LPIPS [59] (Alex version). All these three metrics measure the distance between predicted images and their corresponding ground truths. The evaluation on REDS-SCIR is based on predicted images normalized by $\alpha$ and $\eta$ in the way like Eq. (15): $l_{norm} = l_{pred} / (\alpha \cdot \eta)$. Methods in part (A) are training-free. TFP [73] is the temporal mean of a segment of spikes, which has a hyper-parameter of window length $w_T$. As shown in Fig. 8, we test the $w_T$ in a range of $\{2n + 1\}_{n=4}^{10}$ on the REDS-SCIR. We select a $w_T = 41$ that performs well on different $\eta$. For part (B), we clip the 60 frames centered at the moment to be reconstructed into 10 segments for recurrent inputting. For parts (C) and (D), they are originally trained on spike data synthesized from REDS [64], thus, we preserve both the original and retrained version. Besides, when retraining SSSL [5], we use its original self-supervised loss. As shown in Table 1, our method achieves the best performance across all the $\eta$ on the three metrics. Note that part (A) has no parameters since they are not deep learning methods.

Besides synthetic data, we also compare the above methods using data captured by spike cameras in the real world. We use spikes of 12 scenes, which are shown in Fig. 9. For quantitative comparison, since there are no ground truths, we employ three blind image quality assessment metrics, namely BRISQUE [28], PIQE [42], and HOSA [51]. BRISQUE uses statistics of locally normalized luminance to quantify possible losses of naturalness. PIQE estimates quality only from perceptually significant spatial regions with local features. HOSA uses a small codebook based on high-order statistics aggregation to build the global quality-
Figure 10. Visual comparison on real-captured data. In the visualization of spikes, an orange point means a spike. Gamma transformation with parameter 2.2 is used for visualization.

<table>
<thead>
<tr>
<th>Case</th>
<th>MODF</th>
<th>MGA</th>
<th>PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Spike</td>
<td>38.44</td>
<td>37.69</td>
</tr>
<tr>
<td>2</td>
<td>D (1,1)</td>
<td>38.78</td>
<td>38.10</td>
</tr>
<tr>
<td>3</td>
<td>✓</td>
<td>38.99</td>
<td>38.32</td>
</tr>
<tr>
<td>4</td>
<td>✓ ✓</td>
<td>38.91</td>
<td>38.26</td>
</tr>
<tr>
<td>5</td>
<td>✓ ✓ ✓</td>
<td>39.06</td>
<td>38.39</td>
</tr>
<tr>
<td>6</td>
<td>✓ ✓ ✓</td>
<td>39.02</td>
<td>38.39</td>
</tr>
<tr>
<td>7</td>
<td>✓ ✓ ✓</td>
<td>39.39</td>
<td>38.73</td>
</tr>
<tr>
<td>8</td>
<td>✓ ✓ ✓</td>
<td>39.38</td>
<td>38.77</td>
</tr>
<tr>
<td>9</td>
<td>✓ ✓ ✓</td>
<td>39.76</td>
<td>39.09</td>
</tr>
</tbody>
</table>

Table 3. Ablation studies of the proposed network. Best in bold.

We also study the impact of the number of input frames on the network. As shown in Table 4, we select {1,3,5,7} as the number of sub-streams, i.e., the number of input frames.

<table>
<thead>
<tr>
<th>NIF</th>
<th>η = 1.00</th>
<th>η = 0.75</th>
<th>η = 0.50</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>38.14 0.959 0.032</td>
<td>37.44 0.952 0.039</td>
<td>36.08 0.935 0.055</td>
</tr>
<tr>
<td>41</td>
<td>39.35 0.969 0.021</td>
<td>38.67 0.963 0.027</td>
<td>37.34 0.949 0.041</td>
</tr>
<tr>
<td>61</td>
<td>39.76 0.970 0.021</td>
<td>39.09 0.964 0.027</td>
<td>37.76 0.951 0.040</td>
</tr>
<tr>
<td>81</td>
<td>39.70 0.969 0.022</td>
<td>39.03 0.963 0.028</td>
<td>37.69 0.950 0.042</td>
</tr>
</tbody>
</table>

Table 4. Ablation studies on the number of input frames NIF.

NIF is {21,41,61,81}. Since the space is limited, we use the initial letters to represent the 3 reference metrics. When NIF is small, the performance grows with NIF grows, but this growth tends to converge when NIF is large. We set 61 as the NIF since the performance is converged.

6. Conclusions

We propose a method for reconstructing clear images from spike streams with boosted approaches for dealing with spike fluctuations. We reveal the unbiased estimation attribute of the reciprocal of DSFT and design a multi-order DSFT fusion (MODF) module. We also propose a pyramidal multi-granularity alignment (MGA) module. The MGA uses a cross-attentional patch-level operation with a local search strategy for initialization and uses deformable convolution for pixel-level alignment. Experimental results show that the proposed method achieves state-of-the-art performance on both synthetic and real-captured data.
References

[16] Jin Han, Yixin Yang, Peiqi Duan, Chu Zhou, Lei Ma, Chao Xu, Tiejun Huang, Imari Sato, and Boxin Shi. Hybrid high dynamic range imaging fusing neuromorphic and conventional images. IEEE TPAMI, 2023. 2


[48] Lujie Xia, Ziluo Ding, Rui Zhao, Jiyuan Zhang, Lei Ma, Zhaofei Yu, Tiejun Huang, and Ruiqin Xiong. Unsupervised optical flow estimation with dynamic timing representation for spike camera. In NeurIPS, 2023. 2


[60] Yiyang Zhang, Ruiqin Xiong, and Tiejun Huang. Spike signal reconstruction based on inter-spike similarity. In VCIP, pages 1–5, 2022. 2


[77] Lin Zhu, Siwei Dong, Jianing Li, Tiejun Huang, and Yonghong Tian. Ultra-high temporal resolution visual reconstruction from a fovea-like spike camera via spiking neuron model. *IEEE TPAMI*, 2022. 1, 2, 7
