Optical Flow for Spike Camera with Hierarchical Spatial-Temporal Spike Fusion

Rui Zhao[1,2], Ruiqin Xiong[1,2*], Jian Zhang[3], Xinfeng Zhang[4], Zhaofei Yu[1,2,5], Tiejun Huang[1,2,5]

[1]National Key Laboratory for Multimedia Information Processing, School of Computer Science, Peking University
[2]National Engineering Research Center of Visual Technology, School of Computer Science, Peking University
[3]School of Electronic and Computer Engineering, Peking University
[4]School of Computer Science and Technology, University of Chinese Academy of Sciences
[5]Institute for Artificial Intelligence, Peking University
ruizhao@stu.pku.edu.cn, {rqxiong, zhangjian.sz, yuzf12, tjhuang}@pku.edu.cn, xfzhang@ucas.ac.cn

Abstract

As an emerging neuromorphic camera with an asynchronous working mechanism, spike camera shows good potential for high-speed vision tasks. Each pixel in spike camera accumulates photons persistently and fires a spike whenever the accumulation exceeds a threshold. Such high-frequency fine-granularity photon recording facilitates the analysis and recovery of dynamic scenes with high-speed motion. This paper considers the optical flow estimation problem for spike cameras. Due to the Poisson nature of incoming photons, the occurrence of spikes is random and fluctuating, making conventional image matching inefficient. We propose a Hierarchical Spatial-Temporal (HiST) fusion module for spike representation to pursue reliable feature matching and develop a robust optical flow network, dubbed as HiST-SFlow. The HiST extracts features at multiple moments and hierarchically fuses the spatial-temporal information. We also propose an intra-moment filtering module to further extract the feature and suppress the influence of randomness in spikes. A scene loss is proposed to ensure that this hierarchical representation recovers the essential visual information in the scene. Experimental results demonstrate that the proposed method achieves state-of-the-art performance compared with the existing methods. The source codes are available at https://github.com/ruizhao26/HiST-SFlow.

Introduction

With the development of computer vision, high-speed vision applications attract increasing attention in areas such as autonomous driving and unmanned aerial vehicle. Neuromorphic cameras (NeuCams) are a kind of emerging camera that can handle vision tasks in high-speed scenarios. NeuCams can be roughly divided into event cameras (Lichtsteiner, Posch, and Delbruck 2008; Moeys et al. 2017; Huang, Guo, and Chen 2017) and spike cameras (Dong, Huang, and Tian 2017; Huang et al. 2022a). Both two kinds of cameras work ***asynchronously*** at pixel level and enjoy the advantages of high speed and low latency.

Event cameras are inspired by the retinal periphery and are equipped with a differential sampling model. They detect light intensity change at each pixel in the logarithmic domain, and an event will be fired whenever the change exceeds the threshold. ***Different from event cameras***, spike cameras are inspired by the retinal fovea and work with an integral sampling model. Each pixel of the spike camera continuously accumulates photons, and a spike will be fired whenever the accumulation exceeds the threshold. Compared with event cameras, spike cameras can better recover the scene, especially for regions with fewer textures and motion. Many pixel-level tasks are researched for spike cameras, such as reconstruction (Zhao et al. 2021b; Zheng et al. 2021; Zhao et al. 2021c), optical flow (Hu et al. 2022) and depth estimation (Wang et al. 2022; Zhang et al. 2022).

Optical flow is the pixel-level correspondences among frames (Horn and Schunck 1981), which has been a critical task in the computer vision area. Hu et al. (Hu et al. 2022) propose the first deep learning approach for optical flow estimation for spike cameras, i.e., spike-based optical flow. They propose a lightweight pyramidal network SCFlow with

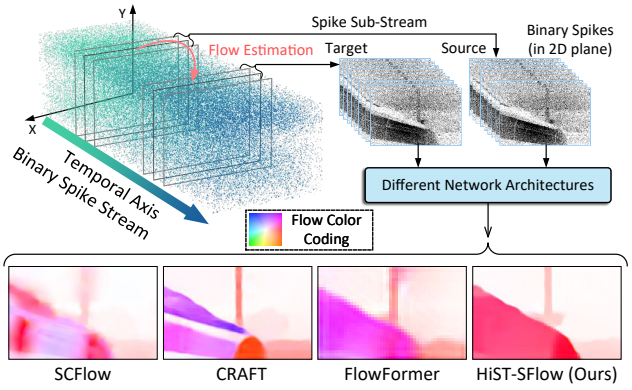

Figure 1: Illustration of spike-based optical flow. The scene content is capturing a high-speed train moving to the right on another train in the opposite direction. On the top-left is a binary spike stream in a spatial-temporal coordinate. On the top-right are spike sub-streams in the spatial plane, where a black point means a spike. The input of the spike-based optical flow is two sub-streams around the source and target time, respectively. As shown at the bottom, our method can better preserve the edges of the motion. All the methods use spikes as inputs and are trained in the same setting.

*Corresponding author.

binary spikes as the input. The differences in the data format introduce challenges to spike-based optical flow. There are multiple kinds of noises in the imaging of the spike camera. First, the arrival of the photons follows the Poisson process. Then, the circuits have thermal noise. Finally, the spike readout is synchronous with quantization noise. Thus, obtaining brightness with high reliability from spikes is difficult.

The factors mentioned above introduce fluctuations in the spike streams. The length of the spike accumulation period may be different even if the light intensity is the same. The randomness of the spike streams causes ambiguities in the correlation between features of different moments, making the feature matching inaccurate. The previous spike-based optical flow methods often have difficulty preserving the edges of the objects and the spatial consistency of the motion, especially in real-captured data.

In this paper, we propose a Hierarchical Spatial-Temporal (HiST) fusion module for spike representation in the spike-based optical flow network HiST-SFlow. The motivation of HiST is to suppress the fluctuations in order to extract stable features from the binary spike streams. In spike-based vision tasks, we usually use a series of spike frames in a period to represent the brightness information of the scene. Previous works fuse all the temporal information with a single operation before embedding them into high-dimensional features. Unlike previous works, we first fuse local temporal information for multiple moments and then extend the scope of the fusion. This hierarchical fusion module can adaptively use the correlated information for spike representation. The contributions of this paper can be summarized as follows.

(1) A HiST-SFlow is proposed for spike-based optical flow. In HiST-SFlow, the spikes are represented by the HiST module and extracted to features for correlation. The optical flow is estimated by a recurrent optimizer.

(2) An inter-moment hierarchical fusion (InterF) module and an intra-moment filtering (IntraF) module are proposed to suppress the randomness in the spikes. A scene loss is proposed to constrain high-fidelity representation to contain the brightness information of the scene.

(3) Experimental results demonstrate that our method gets the state-of-the-art performance on both the PHM dataset (Hu et al. 2022) and real-captured data.

Related Work

Optical Flow Estimation. FlowNet (Dosovitskiy et al. 2015) is the first end-to-end deep neural network for optical flow estimation. Subsequent works introduce the knowledge of traditional methods to the network (Ranjan and Black 2017; Sun et al. 2018; Hui, Tang, and Loy 2018; Hur and Roth 2019; Hui, Tang, and Loy 2020), such as pyramid and warping. RAFT (Teed and Deng 2020) combines the advantages of the above methods, which constructs an all-pairs correlation and recurrently optimizes the optical flow.

Many works are proposed based on RAFT. GMA (Jiang et al. 2021a), Separable Flow (Zhang et al. 2021), and KPA-Flow (Luo et al. 2022) focus on the matching of the features to improve accuracy. SCV (Jiang et al. 2021b), Flow1D (Xu et al. 2021b), and DIP (Zheng et al. 2022) reduce the computational complexity with sparse cost volume, orthogonal attention, and inverse patch match, respectively. Recently, transformers are used in optical flow networks (Xu et al. 2022; Huang et al. 2022b; Zhao et al. 2022; Sui et al. 2022).

Spike Camera. Many works around spike cameras are sprung up recently. Image reconstruction is a popular topic among these works. Zhu et al. (Zhu et al. 2019) reconstruct images with the count of spikes and spike intervals, respectively. Subsequent methods reconstruct images from spikes with filtering (Zhao, Xiong, and Huang 2020; Dong et al. 2022), neuron models (Zhu et al. 2020, 2022a; Zheng et al. 2021), optimization (Zhao et al. 2021c), and deep neural networks (Zhao et al. 2021b; Zhu et al. 2021). MGSR (Zhao et al. 2021a), SpikeSRNet (Zhao et al. 2023), and Xiang et al. (Xiang et al. 2021) estimate super-resolved images from spikes based on the fusion of multi frames. Han et al. (Han et al. 2020) and Zhou et al. (Zhou et al. 2020) use spike cameras to realize high dynamic range imaging. Xia et al. (Xia et al. 2023) use spikes to assist video frame interpolation.

Besides getting images, various tasks have been developed. Zhu et al. (Zhu et al. 2022b) and Li et al. (Li et al. 2022) propose object detection methods based on spike streams. SCFlow (Hu et al. 2022) estimates optical flow directly from the binary spike streams based on a pyramidal network. SSDEFormer (Wang et al. 2022) and Spike Transformer (Zhang et al. 2022) estimate binocular and monocular depth for spike cameras with transformers, respectively.

Preliminary of the Spike Camera

The spike camera mimics the retinal fovea in an integrate-and-fire (IF) manner. As shown in Fig. 2. Each pixel of the spike camera has three key components: photon-receptor, integrator, and comparator. The photon-receptor receives photons from the scene and converts them to photoelectrons. The integrator accumulates the photoelectrons, and a spike will be fired whenever the accumulation exceeds the threshold. At the same time, the accumulation in the integrator will be reset. It is noticeable that each pixel implements the IF cycle *independently*, i.e., each pixel of the spike camera works *asynchronously*. The spikes' reading is synchronous with an ultra-high speed of up to 40kHz. Thus, the spike camera generates an $H \times W$ binary spike frame at each reading moment, where the spike density corresponds to the light intensity.

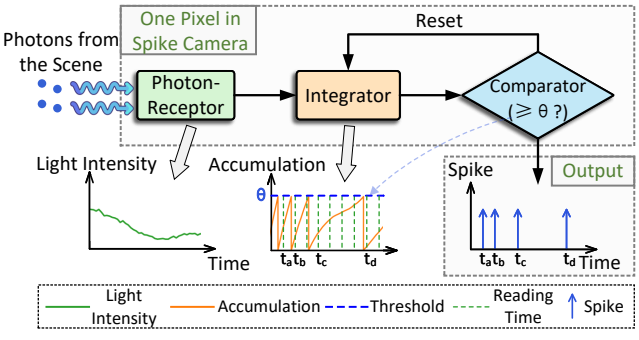

Figure 2: The "integrate-and-fire" working mechanism of a single pixel in the spike camera.

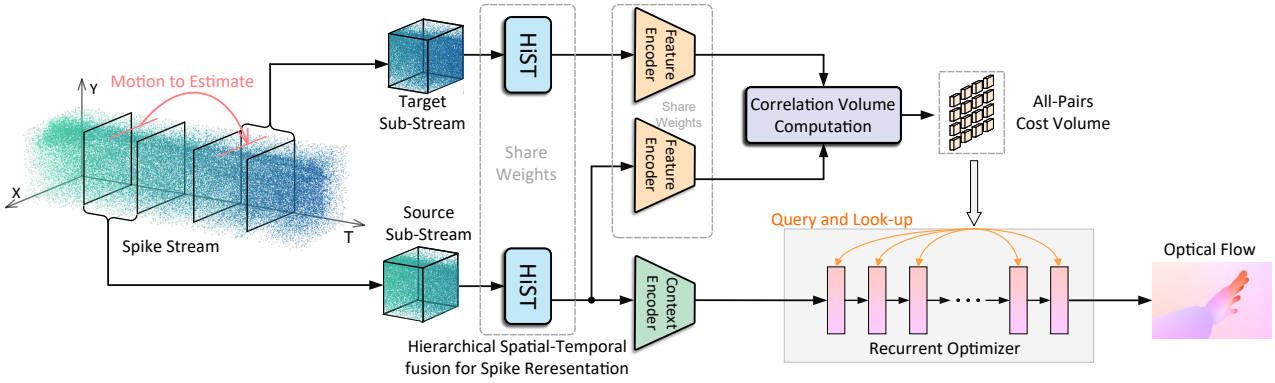

Figure 3: The overall architecture of the HiST-SFlow. Two spike sub-streams represent the scene's brightness at the source and target time, respectively. The hierarchical spatial-temporal (HiST) fusion is used for the two spike sub-streams as representation before being extracted to matching features and context feature. The two matching features construct all-pairs cost volume. The recurrent optimizer estimates the flow based on the context feature and cost volume.

If we denote the incoming light at pixel (x, y) and time t for the spike camera is $\mathbf{I} = \mathbf{I}(x, y, t)$, the accumulation in the integrator $\mathbf{A} = \mathbf{A}(x, y, t)$ can be formulated as:

$$\mathbf{A}(x, y, t) = \int_0^t \mathbf{I}(x, y, t)\, \mathrm{d}t \mod \theta, \tag{1}$$

where θ is the firing threshold adapted to make no more than one spike fired in each reading interval.

Approaches

Overall Architecture

Problem definition. Suppose the spike stream output by the spike camera is $\mathbf{S}(\mathbf{x}, t) \in \mathbb{B}^{H\times W\times T}$, where $\mathbf{x} = (x, y)$ and $\mathbb{B}$ is the binary domain. The spike-based optical flow estimation is to predict the pixel level correspondence $\mathbf{w}(\mathbf{x}; t_i, t_j)$ of the scene captured between moment t_i and t_j based on $\mathbf{S}$. The correspondence can be formulated as:

$$\mathcal{I}(\mathbf{S})\,(\mathbf{x}, t_i) \leftarrow \mathcal{I}(\mathbf{S})\,(\mathbf{x} + \mathbf{w}(\mathbf{x}; t_i, t_j), t_j)\,, \tag{2}$$

where $\mathcal{I}(\mathbf{S})$ is the scene behind the spike stream $\mathbf{S}$, and $\leftarrow$ means pixel-level registration.

Network architecture. As shown in Fig. 3, to estimate the flow field from source time t_i to the target time t_j, we clip two spike sub-streams $\mathbf{H}_i$ and $\mathbf{H}_j$ for representing the scene at time t_i and t_j, respectively. The $\mathbf{H}_t$ is a set of continuous spike frames with t as the central time, which can be formulated as follows:

$$\mathbf{H}_t = \left\{\mathbf{S}(t + t_\mathrm{o}) \mid t_\mathrm{o} \in [-T_\mathrm{s}^\mathrm{half}, T_\mathrm{s}^\mathrm{half}], t_o \in \mathbb{Z}\right\}, \tag{3}$$

where we omit the spatial coordinate $\mathbf{x}$. $\mathbb{Z}$ means the integer domain. $T_\mathrm{s}^\mathrm{half}$ is the half length of the spike sub-strteam.

The network first embeds the sub-stream $\mathbf{H}_i$ and $\mathbf{H}_j$ into representations $\mathbf{R}_i$ and $\mathbf{R}_j$ with our proposed HiST fusion module. The following network has a similar architecture with RAFT (Teed and Deng 2020). First, matching features $\mathbf{F}_i$ and $\mathbf{F}_j$ are extracted from the spike representations through a feature encoder. The context feature $\mathbf{F}_i^\mathrm{C}$ is extracted from the source representation $\mathbf{R}_i$. We construct the 4D all-pairs correlation volume in a similar way to CRAFT (Sui et al. 2022). The target feature $\mathbf{F}_j$ is first filtered using a semantic smoothing transformer. The all-pairs correlation is based on multiple query and key projections (Li et al. 2021) with K modes. The multi-projected correlations are aggregated using a softmax along the K modes. The recurrent optimizer estimates the residual of the flow based on local cost volume that is looked up according to the current estimated flow.

Hierarchical Spatial-Temporal Fusion Module

As shown in Fig. 4, the structure of the HiST module can be divided into three parts: inter-moment hierarchical fusion (InterF), intra-moment filtering (IntraF), and global temporal aggregation (GTA). In the imaging procedure of spike cameras, multiple kinds of noises are introduced. The photons' arrival follows a Poisson process, introducing Poisson noises. The dark currents in circuits introduce thermal noises, and the reading mechanism of the spikes introduces quantitative noises.

Due to the noises, the binary spikes have fluctuations and randomness. Extracting effective features from binary spikes is a new challenge. For effective pixel-level dense matching based on binary spikes, we propose an InterF module to concentrate the spatial-temporal information in spike streams in a spatial-temporal hierarchical way. In the concentration procedure, we design an IntraF module to reduce the randomness of features at each moment. The InterF and IntraF are implemented alternatively. We also propose a scene loss to constrain the representation with the scene brightness.

Inter-Moment Hierarchical Fusion As shown in the top of Fig. 4, different from previous works that fuse temporal information of spikes in a single operation (Hu et al. 2022; Wang et al. 2022), we retain the time information in the feature extraction procedure. The hierarchical fusion strategy is inspired by video restoration tasks (Maggioni et al. 2021; Isobe et al. 2020; Xu et al. 2021a; Chan et al. 2022; Liu et al.

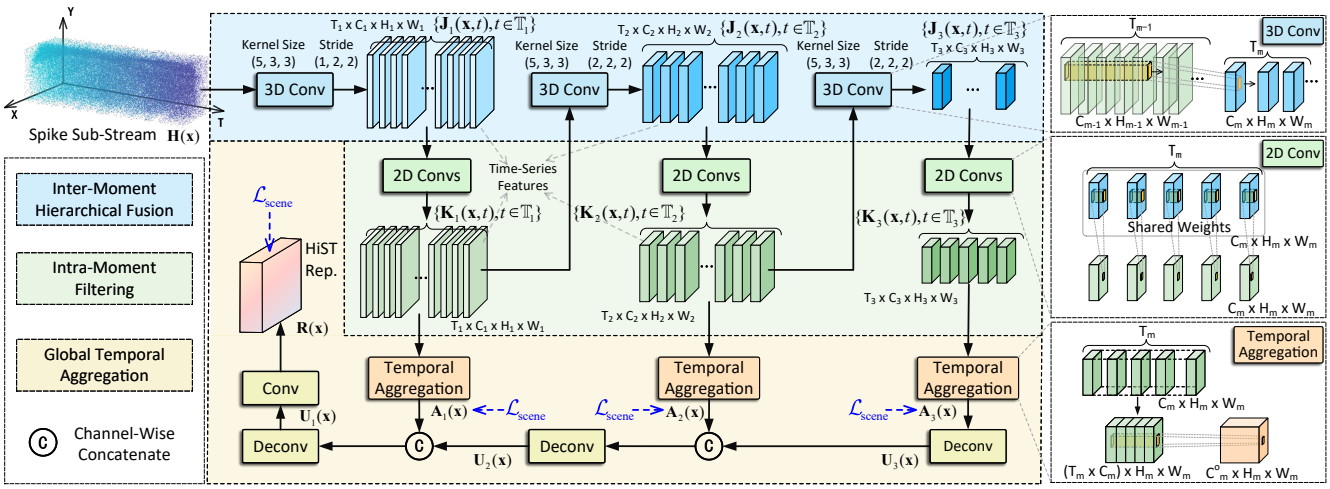

Figure 4: Illustration of the hierarchical spatial-temporal (HiST) fusion for spike representation. A spike sub-stream is extracted as time-series features through passing the inter-moment hierarchical fusion (InterF) and intra-moment filtering (IntraF) module alternatively. The features from all the levels of the IntraF module are aggregated temporally to represent the central time of the input spike sub-stream. The aggregated features are fused to be the final spike representation.

2022) that process a video frame with adjacent frames as references. In the InterF, we construct a pyramid of time-series features using 3D convolutional layers with activation. If we denote the time-series feature output by InterF and IntraF at level m are $\mathbf{J}_m(\mathbf{x}, t)$ and $\mathbf{K}_m(\mathbf{x}, t)$, respectively. The InterF can be formulated as follows:

$$\mathbf{J}_m(\mathbf{x}, t) = \mathscr{J}_m\left[\left\{\mathbf{K}_{m-1}(\mathbf{x}, \tau) \mid \tau \in \mathbb{T}_{m-1}\right\}\right], \quad (4)$$

$$\mathbb{T}_{m-1} = \left\{T_\mathrm{c} - T_{m-1}^\mathrm{half}, \ldots, T_\mathrm{c}, \ldots T_\mathrm{c} + T_{m-1}^\mathrm{half}\right\}, \quad (5)$$

where $\mathscr{J}_m$ is the m-th level InterF's operation. It's a 3D convolution with activation whose kernel size and stride are $(k_m^\mathrm{t}, k_m^\mathrm{h}, k_m^\mathrm{w})$ and $(s_m^\mathrm{t}, s_m^\mathrm{h}, s_m^\mathrm{w})$, respectively. $\mathbb{T}_{m-1}$ is the temporal domain of definition of $\mathbf{J}_{m-1}$ and $\mathbf{K}_{m-1}$. T_c is the central time of spike sub-stream and time-series features. T_{m-1}^half is the half window length of $\mathbf{J}_{m-1}$ and $\mathbf{K}_{m-1}$. We don't pad in the temporal axis since it does not make sense.

We use the raw spike sub-stream as the input of the InterF at the first level, i.e., $\mathbf{K}_0(\mathbf{x}, t) = \mathbf{H}_\tau(\mathbf{x}, t)\big|_{\tau=T_\mathrm{c}}$ and $T_0^\mathrm{half} = T_\mathrm{s}^\mathrm{half}$. In the pyramid, the spatial and temporal information is concentrated through the hierarchical fusion scheme.

In InterF, we simultaneously extract features at different moments $t \in \mathbb{T}_m$ around the central moment T_c. In both spatial and temporal domains, the information in spikes is fused in multiple steps. The fusion procedure in each hierarchy aims to extract the spatial-temporal information structure in a small local neighborhood. With the increasing of hierarchy level, the spatial-temporal information is concentrated. After each hierarchy of fusion, the number of features can be reduced in both spatial and temporal domains

Intra-Moment Filtering We aim to reduce the influence of spikes' fluctuations through pixels with similar distributions. However, due to motion and occlusions, features in other moments cannot always offer effective references through InterF. Thus, we propose IntraF to model the spatial similarity of itself for features at each moment.

In the m-th level time-series features $\mathbf{J}_m$, the feature $\mathbf{J}_m(t_i)$ corresponds to the scene at moment t_i. For $\mathbf{J}_m(t)$ at each moment, we propose to filter them using themselves with weight-shared layers, which can be formulated as:

$$\mathbf{K}_m(\mathbf{x}, t) = \mathscr{K}_m\left[\mathbf{J}_m(\mathbf{x}, t)\right], \ t \in \mathbb{T}_m, \quad (6)$$

where $\mathbf{K}_m$ and $\mathscr{K}_m$ are the output and the operation of the IntraF module at m-th level, respectively. The $\mathcal{K}_m$ is a residual block for filtering features at each moment. Note that the $\mathbf{K}_m(\mathbf{x}, t_i)$ is filtered from $\mathbf{J}_m(\mathbf{x}, t_i)$ rather than the whole $\{\mathbf{J}_m(\mathbf{x}, t) \mid t \in \mathbb{T}_m\}$.

The interF and intraF are alternatively implemented to enhance the time-series features in each hierarchy of the HiST. The collaboration of the InterF and IntraF modules enables us to use the context in diverse scales of scopes for better restoring the scene's brightness. This strategy has been proven to be efficient in video processing tasks such as restoration tasks (Maggioni et al. 2021; Isobe et al. 2020; Xu et al. 2021a; Chan et al. 2022; Liu et al. 2022) and compression (Sullivan et al. 2012; Lainema et al. 2012).

Global Temporal Aggregation In our network, the goal for representation is to describe the scenes' brightness at the source and target time. In the modules mentioned above, we obtain features at moments $t \in \mathbb{T}_m$ at different levels. To represent the scene at central time T_c of the input spike sub-stream, we aggregate the information of features $\{\mathbf{K}_m(t)\} \mid t \in \mathbb{T}_m; m \in \{1, 2, 3\}\}$. In each level m, We concatenate $\{\mathbf{K}_m(t) \mid t \in \mathbb{T}_m\}$ at all the moments in a channel-wise manner and fuse them:

$$\mathbf{A}_m(\mathbf{x}) = \mathscr{A}_m\left[\mathrm{Cat}\left\{\mathbf{K}_m(\mathbf{x}, \tau) \mid \tau \in \mathbb{T}_m\right\}\right], \quad (7)$$

where $\mathbf{A}_m$ and $\mathscr{A}_m$ are the output and operation of the temporal aggregation operation. $\mathscr{A}_m$ means convolutional layers and Cat means the channel-wise concatenation along different moments. As shown in the bottom of Fig. 4, $\mathbf{A}_m$ is

	Architecture	Ball	Cook	Dice	Doll	Fan	Hand	Jump	Poker	Top	**Average**
$\Delta t = 10$	SCFlow	0.51 / 20.3	1.34 / 38.6	1.10 / 30.7	0.22 / 5.6	**0.24** / 10.7	1.30 / 57.3	0.11 / 3.0	0.80 / 41.1	2.14 / 17.7	0.863 / 25.00
	RAFT	0.46 / 12.5	1.32 / 43.7	0.95 / 29.3	0.24 / 6.7	0.28 / 12.7	1.11 / 45.1	0.11 / 3.0	0.67 / 37.1	2.19 / 19.7	0.813 / 23.30
	GMA	0.61 / 21.7	1.84 / 74.7	1.13 / 34.2	0.39 / 9.4	0.36 / 12.1	2.13 / 80.6	0.17 / 2.8	0.88 / 43.5	2.29 / 23.6	1.087 / 33.63
	Flow1D	0.79 / 51.4	1.28 / 50.8	1.15 / 47.9	0.27 / 6.3	0.28 / 11.0	1.86 / 83.1	0.13 / 3.4	0.85 / 50.1	2.19 / 17.7	0.979 / 35.76
	KPA-Flow	0.47 / 14.9	1.41 / 45.9	0.87 / 29.9	0.27 / 7.1	0.29 / 12.7	1.19 / 47.7	0.12 / 3.0	0.65 / 36.6	2.19 / 19.4	0.827 / 24.12
	GMFlow	0.76 / 42.4	1.29 / 61.0	1.54 / 81.7	0.31 / 8.4	0.43 / 14.1	1.83 / 65.0	0.30 / 3.7	0.95 / 54.2	2.29 / 23.3	1.077 / 39.33
	GMFlowNet	0.45 / 12.1	1.22 / 43.8	1.02 / 32.9	0.35 / 7.8	0.25 / 10.7	1.53 / 65.3	0.12 / 3.2	0.65 / 31.5	2.18 / 17.5	0.863 / 24.98
	CRAFT	0.61 / 15.0	1.28 / 43.5	0.93 / 27.6	**0.19** / **5.0**	0.25 / **10.2**	1.67 / 73.3	0.10 / 2.6	0.56 / **23.1**	2.15 / 15.1	0.860 / 23.94
	FlowFormer	0.52 / 13.5	1.48 / 58.7	0.98 / 31.0	0.25 / 6.7	0.29 / 11.5	1.82 / 84.5	0.14 / 3.6	0.94 / 54.9	2.22 / 19.5	0.959 / 31.54
	HiST-SFlow	**0.28 / 7.8**	**0.80 / 27.4**	**0.85 / 23.3**	0.20 / 5.6	0.27 / 12.8	**0.64 / 21.7**	**0.08 / 2.5**	**0.53** / 23.9	**2.11 / 14.8**	**0.640 / 15.54**
$\Delta t = 20$	SCFlow	0.94 / 27.1	3.00 / 50.6	1.72 / 33.2	0.41 / 8.1	**0.46 / 13.6**	3.71 / 71.3	0.19 / 5.9	1.57 / 53.7	4.25 / 18.9	1.804 / 31.37
	RAFT	0.78 / 18.6	2.75 / 54.4	1.57 / 30.1	0.43 / 9.3	0.50 / 14.6	2.81 / 59.9	0.21 / 5.8	1.31 / 46.7	4.30 / 21.2	1.628 / 28.94
	GMA	1.01 / 22.1	4.95 / 96.4	1.52 / 35.9	1.00 / 59.6	1.19 / 98.4	6.66 / 99.5	0.81 / 84.4	1.39 / 45.2	4.64 / 64.9	2.575 / 67.38
	Flow1D	1.19 / 51.6	4.52 / 96.3	1.58 / 50.7	0.78 / 53.3	1.01 / 82.1	6.65 / 99.2	0.72 / 73.1	1.39 / 52.3	4.75 / 79.7	2.510 / 70.90
	KPA-Flow	0.80 / 20.9	2.93 / 55.6	1.48 / 31.4	0.45 / 9.6	0.52 / 14.5	2.86 / 62.5	0.22 / 5.6	1.31 / 48.4	4.28 / 19.7	1.649 / 29.81
	GMFlow	1.49 / 80.3	2.64 / 80.1	2.72 / 91.8	0.54 / 15.3	0.77 / 22.0	3.79 / 81.5	0.55 / 27.8	1.78 / 75.3	4.45 / 32.5	2.080 / 56.28
	GMFlowNet	0.92 / 31.4	2.61 / 70.4	2.17 / 42.7	0.61 / 27.5	0.56 / 13.9	3.30 / 93.2	0.21 / 4.5	1.33 / 53.4	4.33 / 25.3	1.782 / 40.25
	CRAFT	1.16 / 85.5	2.68 / 61.0	1.99 / 46.8	0.39 / 7.8	0.48 / 12.5	3.53 / 87.1	0.20 / **3.6**	**1.23** / 38.9	4.31 / 22.0	1.775 / 40.57
	FlowFormer	0.91 / 13.8	4.41 / 96.3	**1.40** / 32.6	0.80 / 54.8	1.03 / 90.0	6.54 / 99.3	0.74 / 75.8	1.47 / 57.4	4.59 / 61.9	2.432 / 64.67
	HiST-SFlow	**0.55 / 8.8**	**2.04 / 33.6**	1.64 / **26.3**	**0.38 / 7.2**	0.51 / 13.9	**2.00 / 34.7**	**0.17** / 5.0	1.28 / **33.1**	**4.18 / 15.1**	**1.417 / 19.73**

Table 1: Comparison on average end-point error (AEPE) and percent of outliers (PO%) with comparable methods on PHM datasets in the $\Delta t = 10$ and $\Delta t = 20$ tracks (AEPE / PO%). All the methods use spike stream as input and are retrained in the same setting on SPIFT. The best results for each scene and the best average results are marked in bold.

upsampled through a deconvolutional layer $\mathscr{U}_m$ to be $\mathbf{U}_m$:

$$\mathbf{U}_m = \mathscr{U}_m \left[\mathrm{Cat} \left\{ \mathbf{A}_m, \mathbf{U}_{m-1} \right\} \right], \tag{8}$$

The representation $\mathbf{R}$ is obtained based on $\mathbf{U}_1$.

Scene Loss To ensure the HiST for spike representation contain the scene's brightness information with high fidelity at moment T_c, we propose a scene loss $\mathcal{L}_\mathrm{scene}$. The SPIFT dataset (Hu et al. 2022) offers the brightness ground truth of the scenes based on a graphics simulator. We propose to use a series of simple 3-layer convolutional layers $\{\mathscr{P}_m\}_{m=0}^3$ for the representation $\mathbf{R}_{T_\mathrm{c}}$ and aggregation feature $\mathbf{A}_m$ in each level to predict the brightness $\mathbf{I}_\mathrm{scene}(\mathbf{x}, T_\mathrm{c})$ at moment T_c. The scene loss can be formulated as:

$$\begin{aligned} \mathcal{L}_\mathrm{scene} &= \| \mathbf{I}_\mathrm{scene}(\mathbf{x}, T_\mathrm{c}) - \mathscr{P}_0(\mathbf{R}_{T_\mathrm{c}}(\mathbf{x})) \|_1 \\ &+ \sum_{m=1}^{3} \lambda_m \| \sigma_m(\mathbf{I}_\mathrm{scene}(\mathbf{x}, T_\mathrm{c})) - \mathscr{P}_m(\mathbf{A}_m(\mathbf{x})) \|_1, \end{aligned} \tag{9}$$

where σ_m is the resize operator to interpolate the $\mathbf{I}_\mathrm{scene}$ to the resolution of $\mathbf{A}_m$, and λ_m is the weight of each level. Based on the scene loss, $\mathbf{R}_{T_\mathrm{c}}$ can better focus on the brightness information at the moment T_c. It is noticeable that all the $\mathscr{P}_m$ are used only during training and not for inference.

Loss Function

The loss function for the proposed network is composed of two parts: the flow loss and the scene loss. Suppose the recurrent optimizer of the network has N iterations, and the estimated flow fields of each iteration are $\{\mathbf{w}_1, \ldots, \mathbf{w}_N\}$. The flow loss can be formulated as follows:

$$\mathcal{L}_\mathrm{flow} = \sum_{i=1}^{N} \gamma^{N-i} \| \mathbf{w}_i(\mathbf{x}) - \mathbf{w}_\mathrm{gt}(\mathbf{x}) \|_1, \tag{10}$$

where γ is the decay factor and we set it as 0.8 following RAFT. $\mathbf{w}_\mathrm{gt}$ is the ground truth of optical flow. We construct the scene loss for representations at both the source and target time. Both the $\mathcal{L}_\mathrm{flow}$ and $\mathcal{L}_\mathrm{scene}$ are spatially averaged for training based on eq. (10) and eq. (9), respectively. The total loss can be formulated as follows:

$$\mathcal{L} = \mathcal{L}_\mathrm{flow} + \lambda(\mathcal{L}_\mathrm{scene}^\mathrm{src} + \mathcal{L}_\mathrm{scene}^\mathrm{tgt}), \tag{11}$$

where λ is set as 0.5.

Experiments

Implementation Details

Model details. In the experiments, we set the input spike frame number as 25 following the SCFlow (Hu et al. 2022), i.e., $T_\mathrm{s}^\mathrm{half} = 12$. The temporal kernel size and stride of $\{\mathscr{J}_1, \mathscr{J}_2, \mathscr{J}_3\}$ is $\{5, 5, 5\}$ and $\{1, 2, 2\}$, respectively. Thus, the temporal lengths of $\{\mathbf{J}_1, \mathbf{J}_2, \mathbf{J}_3\}$ are $\{T_1, T_2, T_3\} = \{21, 9, 3\}$. The weights $\{\lambda_1, \lambda_2, \lambda_3\}$ are set as $\{0.5, 0.25, 0.125\}$, respectively. In the correlation volume computation, we set the number of embed modes as 2. The iteration number of the recurrent optimizer is 12.

Datasets. SPIFT (Hu et al. 2022) is a dataset that is designed for the training of spike-based optical flow. The scenes of SPIFT are generated with random contents using a graphics-based simulator. PHM (Hu et al. 2022) is a dataset that is designed for the evaluation of spike-based optical flow. It's also generated through the graphics-based simulator, and the scenes are specially designed with photo-realistic contents and diversified motion. For each of these two datasets, there are two tracks: $\Delta t = 10$ and $\Delta t = 20$. The $\Delta t = 10$ means the distance of the central frame of the

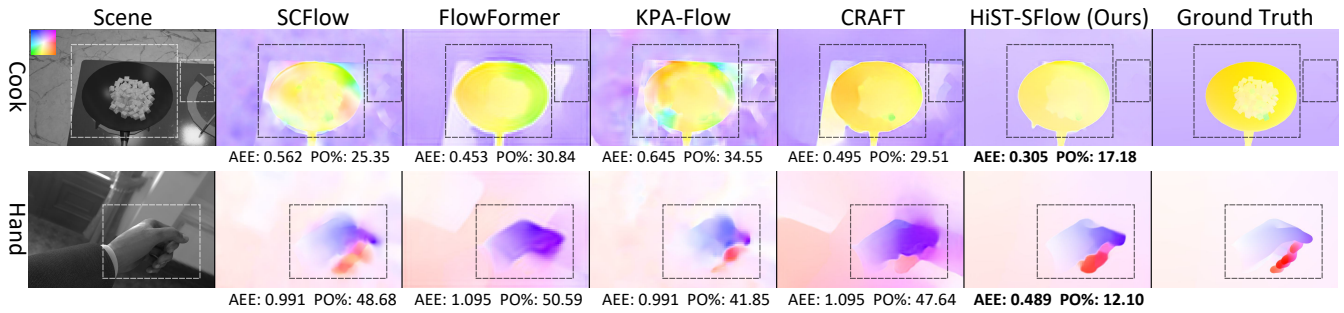

Figure 5: Visual results on PHM dataset in $\Delta t = 10$ track. The meaning of each column is on the top. The performance of each sample is below each color-coded flow. The Scenes are the gray version of the ideal scene in PHM.

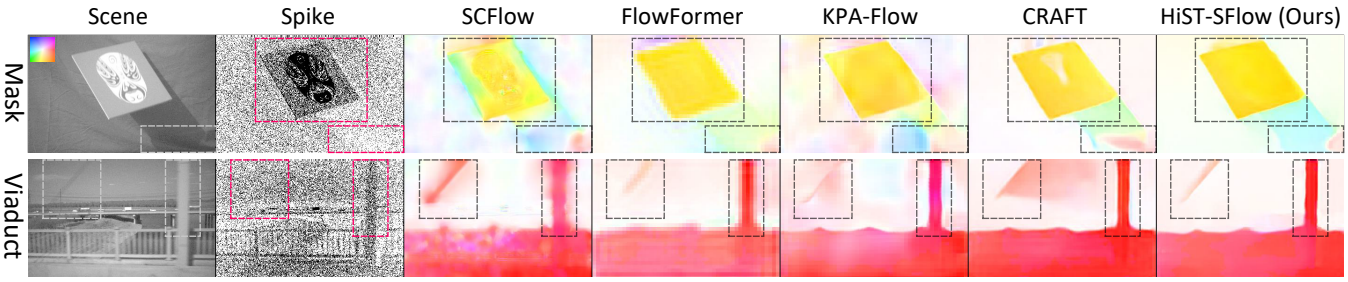

Figure 6: Visual results on real-captured data captured by spike cameras ($\Delta t = 10$). The Scenes are the temporal average of the spikes with gamma transform. The Spikes are the spike frame in the source time of the flow, and a black point means a spike.

two moments for optical flow estimation is 10 spike frames, similarly for $\Delta t = 20$.

Traning details. Similar to SCFlow (Hu et al. 2022), we use SPIFT as the training set and use PHM as the evaluation set. In the training procedure, we randomly crop the spike stream and the flow ground truth to 320×448 in the spatial domain. To balance the motion in the training set, we randomly flip the data horizontally and vertically. Different from previous methods, we mix the "$\Delta t = 10$" and "$\Delta t = 20$" tracks of data during the training. The batch size is set as 6. We use an Adam optimizer (Kingma and Ba 2015) with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The model is trained for 50 epochs. The learning rate is initialized as 1e-4 and scaled by 0.8 every 10 epochs.

Comparable Experiments

We compare the proposed HiST-SFlow with comparable methods on the PHM dataset (Hu et al. 2022) and real-captured data. The comparable methods can be divided into two parts: (a) method designed for spike-based optical flow, and (b) methods straightforwardly adapted from optical flow networks for traditional images. Part (a) includes SCFlow (Hu et al. 2022). Part (b) includes adapted RAFT (Teed and Deng 2020), GMA (Jiang et al. 2021a), Flow1D (Xu et al. 2021b), KPA-Flow (Luo et al. 2022), GMFlow (Xu et al. 2022), GMFlowNet (Zhao et al. 2022), CRAFT (Sui et al. 2022), and FlowFormer (Huang et al. 2022b). The adapted method is inherited from the comparable experiments in SCFlow (Hu et al. 2022), i.e., regarding the binary spike sub-stream as a multi-channel image. All the methods **use spikes as input** and are **retrained** in the same setting as our method. Note that we do not use the event-based methods (Zhu and Yuan 2018; Lee et al. 2020) since it has been shown that these methods are not appropriate to be straightforwardly adapted for spike-based optical flow in Table 2 of literature (Hu et al. 2022). It is noticeable that we **only use the architecture** of the image-based optical flow methods. The straightforwardly adapted methods **are optical flow networks for spike streams rather than images**. Similarly, we don't use image-based datasets since the input of the spike-based optical flow is spike streams.

We use the average end-point error (AEPE) and percent of outliers (PO%) as the metrics for quantitative comparison. The AEPE is the spatially mean value of Euclidean distance between the predicted flow $\mathbf{w}_{\text{pred}}$ and its ground truth $\mathbf{w}_{\text{gt}}$. We define pixels whose end-point error is larger than **0.5** and **5%** of its ground truth simultaneously as outlier pixels. The PO% is the percentage of this kind of outlier pixels.

For the PHM dataset, we evaluate all the methods on both $\Delta t = 10$ and $\Delta t = 20$ tracks, where the Δt means the index difference between the target and source time, i.e., $\Delta t = 10$ corresponds to 0.25 ms when the spike camera works at 40kHz. The quantitative results of these two tracks are shown in Table 1. The average in the last column of the two tables is the arithmetic mean of the metric of all the scenes, which is different from the weighted mean based on the frames of the scenes in SCFlow (Hu et al. 2022). As shown in Table 1, the proposed HiST-SFlow outperforms all the comparable methods in most instances.

The visualization results on the PHM dataset and real-

Index	Settings			$\Delta t = 10$		$\Delta t = 20$	
	InterF	IntraF	$\mathcal{L}_{\text{scene}}$	AEPE	PO%	AEPE	PO%
(A)	✗	✗	✗	0.986	33.17	2.095	56.56
(B)	✓	✗	✗	0.694	18.18	1.449	21.99
(C)	✓	✓	✗	0.676	17.34	1.433	22.79
(D)	✓	✗	✓	0.675	16.63	1.448	21.40
(E)	✓	✓	✓	**0.640**	**15.54**	**1.417**	**19.73**

Table 2: Ablations for the proposed network modules on PHM dataset. All the values are the arithmetic mean of the scenes. The best results are marked in bold.

Representation	$\Delta t = 10$		$\Delta t = 20$	
	AEPE	PO%	AEPE	PO%
Window-Based	0.868	25.72	1.757	34.19
Interval-Based	0.880	29.77	1.824	37.91
Multi-Window	0.799	21.10	1.703	34.58
Flow-Guided Window	0.696	16.99	1.533	23.36
HiST (Ours)	**0.640**	**15.54**	**1.417**	**19.73**

Table 3: Ablations for different representations on PHM dataset. All the values are the arithmetic mean of the scenes. The best results are marked in bold.

captured data are shown in Fig. 5 and Fig. 6, respectively. There are two scenes in the real-captured data. The "Mask" includes a dropping board with mask painting. The "Viaduct" includes a fast-moving view on a viaduct. Note that we use the color-coded scheme in the Middlebury dataset (Baker et al. 2011), which differs from SCFlow (Hu et al. 2022). As shown in these two figures, our HiST-SFlow can better preserve the objects' edges and the motion's consistency compared with other methods.

Ablations Studies

Ablations for modules. We implement a series of ablations to see the proposed modules' effectiveness. The quantitative results are shown in Table 2. The modules that can be closed optionally include InterF, IntraF, and scene loss. The existence of IntraF depends on InterF since IntraF is used for the output of InterF, i.e., time-series features. Thus, there are five combinations based on the three options. The comparison between experiments {(A), (B)} shows the effectiveness of the InterF module. Experiments {(B), (C)} and {(D), (E)} show the effectiveness of the IntraF module. Experiments {(B), (D)} and {(C), (E)} demonstrate the effectiveness of the scene loss. In summary, Table. 2 shows that all the proposed modules make contributions to the final model.

Ablations for Different Representations. Besides ablations for components. We replace our network's HiST with other spike-based representation schemes. The representations we use are as follows.

(1) Window-based representation. Zhu et al. (Zhu et al. 2019) propose using the average along the temporal axis for spike streams to recover the scene's texture.

(2) Interval-based representation. Zhu et al. (Zhu et al. 2019) propose to use the interval of the spike firing $\Delta t(\mathbf{x})$ as the basement of image reconstruction.

(3) Multi-window representation. SSDEFormer (Wang et al. 2022) uses the multi-window temporal average of the spike stream for representation. The window size varies from 1 to T.

(4) Flow-guided window. SCFlow (Hu et al. 2022) uses an initialized optical flow to guide the direction of convolution for spike streams. In the training procedure, we use the same recurrent strategy with SCFlow.

The quantitative results are shown in Table 3. The HiST outperforms all the comparable schemes on all the metrics. The flow-guided window also performs well, but its computational procedure is complex, especially in training.

Architecture	with HiST	$\Delta t = 10$		$\Delta t = 20$	
		AEPE	PO%	AEPE	PO%
GMA	No	1.087	33.63	2.575	67.38
	Yes	**0.666**	**16.91**	**1.391**	**21.20**
KPA-Flow	No	0.827	24.12	1.649	29.81
	Yes	**0.659**	**16.99**	**1.363**	**22.27**
GMFlowNet	No	0.863	24.98	1.782	40.25
	Yes	**0.730**	**21.22**	**1.452**	**24.93**

Table 4: Comparison between the network with and without HiST on different baselines.

Using HiST for Other Baselines

The main contribution of our HiST-SFlow is a representation module to obtain high-fidelity features. We use a CRAFT-like network as our baseline, and other architectures can also be used as the baseline. We apply our HiST as spike representation for other three advanced optical flow network architectures, i.e., GMA (Jiang et al. 2021a), KPA-Flow (Luo et al. 2022), and GMFlowNet (Zhao et al. 2022). As shown in Table 4, the HiST can improve the performance on the other three baselines. Almost all the metrics in the table have a $20\% \sim 30\%$ error reduction with the HiST.

Conclusion

We propose a hierarchical spatial-temporal fusion module for spike representation and construct a robust network for spike-based optical flow. We propose an inter-moment progressive fusion module and an intra-moment filtering module to suppress the influence caused by the fluctuations in the spikes. We also design a scene loss to constrain the representation containing the brightness information of the scene. Experiments demonstrate that our method achieves state-of-the-art performance on spike-based optical flow and can well preserve the edges and motion consistency of the objects.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grants 62072009, 22127807, 62071449, and in part by the National Key R&D Program of China under Grant 2021YFF0900501.

References

Baker, S.; Scharstein, D.; Lewis, J.; Roth, S.; Black, M. J.; and Szeliski, R. 2011. A database and evaluation methodology for optical flow. *IJCV*, 92(1): 1–31.

Chan, K. C.; Zhou, S.; Xu, X.; and Loy, C. C. 2022. BasicVSR++: Improving video super-resolution with enhanced propagation and alignment. In *CVPR*, 5972–5981.

Dong, S.; Huang, T.; and Tian, Y. 2017. Spike Camera and Its Coding Methods. In *DCC*, 437–437.

Dong, Y.; Zhao, J.; Xiong, R.; and Huang, T. 2022. 3D Residual Interpolation for Spike Camera Demosaicing. In *ICIP*, 1461–1465.

Dosovitskiy, A.; Fischer, P.; Ilg, E.; Hausser, P.; Hazirbas, C.; Golkov, V.; Van Der Smagt, P.; Cremers, D.; and Brox, T. 2015. Flownet: Learning optical flow with convolutional networks. In *ICCV*, 2758–2766.

Han, J.; Zhou, C.; Duan, P.; Tang, Y.; Xu, C.; Xu, C.; Huang, T.; and Shi, B. 2020. Neuromorphic camera guided high dynamic range imaging. In *CVPR*, 1730–1739.

Horn, B. K.; and Schunck, B. G. 1981. Determining optical flow. *AI*, 17(1-3): 185–203.

Hu, L.; Zhao, R.; Ding, Z.; Ma, L.; Shi, B.; Xiong, R.; and Huang, T. 2022. Optical Flow Estimation for Spiking Camera. In *CVPR*.

Huang, J.; Guo, M.; and Chen, S. 2017. A dynamic vision sensor with direct logarithmic output and full-frame picture-on-demand. In *ISCAS*, 1–4.

Huang, T.; Zheng, Y.; Yu, Z.; Chen, R.; Li, Y.; Xiong, R.; Ma, L.; Zhao, J.; Dong, S.; Zhu, L.; et al. 2022a. 1000x Faster Camera and Machine Vision with Ordinary Devices. *Engineering*.

Huang, Z.; Shi, X.; Zhang, C.; Wang, Q.; Cheung, K. C.; Qin, H.; Dai, J.; and Li, H. 2022b. FlowFormer: A Transformer Architecture for Optical Flow. In *ECCV*.

Hui, T.-W.; Tang, X.; and Loy, C. C. 2018. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In *CVPR*, 8981–8989.

Hui, T.-W.; Tang, X.; and Loy, C. C. 2020. A lightweight optical flow CNN—Revisiting data fidelity and regularization. *IEEE TPAMI*, 43(8): 2555–2569.

Hur, J.; and Roth, S. 2019. Iterative residual refinement for joint optical flow and occlusion estimation. In *CVPR*, 5754–5763.

Isobe, T.; Li, S.; Jia, X.; Yuan, S.; Slabaugh, G.; Xu, C.; Li, Y.-L.; Wang, S.; and Tian, Q. 2020. Video super-resolution with temporal group attention. In *CVPR*, 8008–8017.

Jiang, S.; Campbell, D.; Lu, Y.; Li, H.; and Hartley, R. 2021a. Learning to estimate hidden motions with global motion aggregation. In *ICCV*, 9772–9781.

Jiang, S.; Lu, Y.; Li, H.; and Hartley, R. 2021b. Learning optical flow from a few matches. In *CVPR*, 16592–16600.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

Lainema, J.; Bossen, F.; Han, W.-J.; Min, J.; and Ugur, K. 2012. Intra coding of the HEVC standard. *IEEE TCSVT*, 22(12): 1792–1801.

Lee, C.; Kosta, A. K.; Zhu, A. Z.; Chaney, K.; Daniilidis, K.; and Roy, K. 2020. Spike-FlowNet: event-based optical flow estimation with energy-efficient hybrid neural networks. In *ECCV*, 366–382.

Li, J.; Wang, X.; Zhu, L.; Li, J.; Huang, T.; and Tian, Y. 2022. Retinomorphic Object Detection in Asynchronous Visual Streams. In *AAAI*, 1332–1340.

Li, S.; Sui, X.; Luo, X.; Xu, X.; Liu, Y.; and Goh, R. 2021. Medical image segmentation using squeeze-and-expansion transformers. In *IJCAI*.

Lichtsteiner, P.; Posch, C.; and Delbruck, T. 2008. A 128×128 120 dB 15μ s latency asynchronous temporal contrast vision sensor. *IEEE JSSC*, 43(2): 566–576.

Liu, C.; Yang, H.; Fu, J.; and Qian, X. 2022. Learning Trajectory-Aware Transformer for Video Super-Resolution. In *CVPR*, 5687–5696.

Luo, A.; Yang, F.; Li, X.; and Liu, S. 2022. Learning Optical Flow With Kernel Patch Attention. In *CVPR*, 8906–8915.

Maggioni, M.; Huang, Y.; Li, C.; Xiao, S.; Fu, Z.; and Song, F. 2021. Efficient multi-stage video denoising with recurrent spatio-temporal fusion. In *CVPR*, 3466–3475.

Moeys, D. P.; Corradi, F.; Li, C.; Bamford, S. A.; Longinotti, L.; Voigt, F. F.; Berry, S.; Taverni, G.; Helmchen, F.; and Delbruck, T. 2017. A sensitive dynamic and active pixel vision sensor for color or neural imaging applications. *IEEE TBCS*, 12(1): 123–136.

Ranjan, A.; and Black, M. J. 2017. Optical flow estimation using a spatial pyramid network. In *CVPR*, 4161–4170.

Sui, X.; Li, S.; Geng, X.; Wu, Y.; Xu, X.; Liu, Y.; Goh, R.; and Zhu, H. 2022. CRAFT: Cross-Attentional Flow Transformer for Robust Optical Flow. In *CVPR*, 17602–17611.

Sullivan, G. J.; Ohm, J.-R.; Han, W.-J.; and Wiegand, T. 2012. Overview of the high efficiency video coding (HEVC) standard. *IEEE TCSVT*, 22(12): 1649–1668.

Sun, D.; Yang, X.; Liu, M.-Y.; and Kautz, J. 2018. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *CVPR*, 8934–8943.

Teed, Z.; and Deng, J. 2020. RAFT: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 402–419.

Wang, Y.; Li, J.; Zhu, L.; Xiang, X.; Huang, T.; and Tian, Y. 2022. Learning stereo depth estimation with bio-inspired spike cameras. In *ICME*, 1–6.

Xia, L.; Zhao, J.; Xiong, R.; and Huang, T. 2023. SVFI: spiking-based video frame interpolation for high-speed motion. In *AAAI*, 2910–2918.

Xiang, X.; Zhu, L.; Li, J.; Wang, Y.; Huang, T.; and Tian, Y. 2021. Learning Super-Resolution Reconstruction for High Temporal Resolution Spike Stream. *IEEE TCSVT*.

Xu, G.; Xu, J.; Li, Z.; Wang, L.; Sun, X.; and Cheng, M.-M. 2021a. Temporal modulation network for controllable space-time video super-resolution. In *CVPR*, 6388–6397.

Xu, H.; Yang, J.; Cai, J.; Zhang, J.; and Tong, X. 2021b. High-Resolution Optical Flow from 1D Attention and Correlation. In *ICCV*, 10498–10507.

Xu, H.; Zhang, J.; Cai, J.; Rezatofighi, H.; and Tao, D. 2022. GMFlow: Learning Optical Flow via Global Matching. In *CVPR*, 8121–8130.

Zhang, F.; Woodford, O. J.; Prisacariu, V. A.; and Torr, P. H. 2021. Separable Flow: Learning Motion Cost Volumes for Optical Flow Estimation. In *ICCV*, 10807–10817.

Zhang, J.; Tang, L.; Yu, Z.; Lu, J.; and Huang, T. 2022. Spike Transformer: Monocular Depth Estimation for Spiking Camera. In *ECCV*.

Zhao, J.; Xie, J.; Xiong, R.; Zhang, J.; Yu, Z.; and Huang, T. 2021a. Super Resolve Dynamic Scene from Continuous Spike Streams. In *ICCV*, 2533–2542.

Zhao, J.; Xiong, R.; and Huang, T. 2020. High-speed motion scene reconstruction for spike camera via motion aligned filtering. In *ISCAS*, 1–5.

Zhao, J.; Xiong, R.; Liu, H.; Zhang, J.; and Huang, T. 2021b. Spk2ImgNet: Learning to Reconstruct Dynamic Scene from Continuous Spike Stream. In *CVPR*, 11996–12005.

Zhao, J.; Xiong, R.; Xie, J.; Shi, B.; Yu, Z.; Gao, W.; and Huang, T. 2021c. Reconstructing Clear Image for High-Speed Motion Scene With a Retina-Inspired Spike Camera. *IEEE TCI*, 8: 12–27.

Zhao, J.; Xiong, R.; Zhang, J.; Zhao, R.; Liu, H.; and Huang, T. 2023. Learning to super-resolve dynamic scenes for neuromorphic spike camera. In *AAAI*, 3579–3587.

Zhao, S.; Zhao, L.; Zhang, Z.; Zhou, E.; and Metaxas, D. 2022. Global Matching with Overlapping Attention for Optical Flow Estimation. In *CVPR*, 17592–17601.

Zheng, Y.; Zheng, L.; Yu, Z.; Shi, B.; Tian, Y.; and Huang, T. 2021. High-speed Image Reconstruction through Short-term Plasticity for Spiking Cameras. In *CVPR*, 6358–6367.

Zheng, Z.; Nie, N.; Ling, Z.; Xiong, P.; Liu, J.; Wang, H.; and Li, J. 2022. DIP: Deep Inverse Patchmatch for High-Resolution Optical Flow. In *CVPR*, 8925–8934.

Zhou, C.; Zhao, H.; Han, J.; Xu, C.; Xu, C.; Huang, T.; and Shi, B. 2020. Unmodnet: Learning to unwrap a modulo image for high dynamic range imaging. In *NeurIPS*, 1559–1570.

Zhu, A. Z.; and Yuan, L. 2018. EV-FlowNet: Self-Supervised Optical Flow Estimation for Event-based Cameras. In *RSS*.

Zhu, L.; Dong, S.; Huang, T.; and Tian, Y. 2019. A retina-inspired sampling method for visual texture reconstruction. In *ICME*, 1432–1437.

Zhu, L.; Dong, S.; Li, J.; Huang, T.; and Tian, Y. 2020. Retina-like visual image reconstruction via spiking neural model. In *CVPR*, 1438–1446.

Zhu, L.; Dong, S.; Li, J.; Huang, T.; and Tian, Y. 2022a. Ultra-high Temporal Resolution Visual Reconstruction from a Fovea-like Spike Camera via Spiking Neuron Model. *IEEE TPAMI*.

Zhu, L.; Li, J.; Wang, X.; Huang, T.; and Tian, Y. 2021. NeuSpike-Net: High Speed Video Reconstruction via Bio-Inspired Neuromorphic Cameras. In *ICCV*, 2400–2409.

Zhu, Y.; Zhang, Y.; Xie, X.; and Huang, T. 2022b. An FPGA Accelerator for High-Speed Moving Objects Detection and Tracking With a Spike Camera. *Neural Computation*, 34(8): 1812–1839.