Spike-FlowNet: Event-Based Optical Flow Estimation with Energy-Efficient Hybrid Neural Networks

Chankyu Lee[1](✉), Adarsh Kumar Kosta[1], Alex Zihao Zhu[2], Kenneth Chaney[2], Kostas Daniilidis[2], and Kaushik Roy[1]

[1] Purdue University, 47907 West Lafayette, IN, USA
{lee2216,akosta,kaushik}@purdue.edu
[2] University of Pennsylvania, 19104 Philadelphia, PA, USA
{alexzhu,chaneyk,kostas}@seas.upenn.edu

Abstract. Event-based cameras display great potential for a variety of tasks such as high-speed motion detection and navigation in low-light environments where conventional frame-based cameras suffer critically. This is attributed to their high temporal resolution, high dynamic range, and low-power consumption. However, conventional computer vision methods as well as deep Analog Neural Networks (ANNs) are not suited to work well with the asynchronous and discrete nature of event camera outputs. Spiking Neural Networks (SNNs) serve as ideal paradigms to handle event camera outputs, but deep SNNs suffer in terms of performance due to the spike vanishing phenomenon. To overcome these issues, we present Spike-FlowNet, a deep hybrid neural network architecture integrating SNNs and ANNs for efficiently estimating optical flow from sparse event camera outputs without sacrificing the performance. The network is end-to-end trained with self-supervised learning on Multi-Vehicle Stereo Event Camera (MVSEC) dataset. Spike-FlowNet outperforms its corresponding ANN-based method in terms of the optical flow prediction capability while providing significant computational efficiency.

Keywords: Event-based vision · Optical flow estimation · Hybrid network · Spiking neural network · Self-supervised learning

1 Introduction

The dynamics of biological species such as winged insects serve as prime sources of inspiration for researchers in the field of neuroscience, machine learning as well as robotics. The ability of winged insects to perform complex, high-speed

The code is publicly available at: https://github.com/chan8972/Spike-FlowNet.

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-58526-6_22) contains supplementary material, which is available to authorized users.

A. Vedaldi et al. (Eds.): ECCV 2020, LNCS 12374, pp. 366–382, 2020.
https://doi.org/10.1007/978-3-030-58526-6_22

maneuvers effortlessly in cluttered environments clearly highlights the efficiency of these resource-constrained biological systems [5]. The estimation of motion patterns corresponding to spatio-temporal variations of structured illumination - commonly referred to as optical flow, provides vital information for estimating ego-motion and perceiving the environment. Modern deep Analog Neural Networks (ANNs) aim to achieve this at the cost of being computationally intensive, placing significant overheads on current hardware platforms. A competent methodology to replicate such energy efficient biological systems would greatly benefit edge-devices with computational and memory constraints (Note, we will be referring to standard deep learning networks as Analog Neural Networks (ANNs) due to their analog nature of inputs and computations. This would help to distinguish them from Spiking Neural Networks (SNNs), which involve discrete spike-based computations).

Over the past years, the majority of optical flow estimation techniques relied on images from traditional frame-based cameras, where the input data is obtained by sampling intensities on the entire frame at fixed time intervals irrespective of the scene dynamics. Although sufficient for certain computer vision applications, frame-based cameras suffer from issues such as motion blur during high speed motion, inability to capture information in low-light conditions, and over- or under-saturation in high dynamic range environments.

Event-based cameras, often referred to as bio-inspired silicon retinas, overcome these challenges by detecting log-scale brightness changes asynchronously and independently on each pixel-array element [20], similar to retinal ganglion cells. Having a high temporal resolution (in the order of microseconds) and a fraction of power consumption compared to frame-based cameras make event cameras suitable for estimating high-speed and low-light visual motion in an energy-efficient manner. However, because of their fundamentally different working principle, conventional computer vision as well as ANN-based methods become no longer effective for event camera outputs. This is mainly because these methods are typically designed for pixel-based images relying on photo-consistency constraints, assuming the color and brightness of object remain the same in all image sequences. Thus, the need for development of handcrafted-algorithms for handling event camera outputs is paramount.

SNNs, inspired by the biological neuron model, have emerged as a promising candidate for this purpose, offering asynchronous computations and exploiting the inherent sparsity of spatio-temporal events (spikes). The Integrate and Fire (IF) neuron is one spiking neuron model [8], which can be characterized by an internal state, known as the membrane potential. The membrane potential accumulates the inputs over time and emits an output spike whenever it exceeds a set threshold. This mechanism naturally encapsulates the event-based asynchronous processing capability across SNN layers, leading to energy-efficient computing on specialized neuromorphic hardware such as IBM's TrueNorth [24] and Intel's Loihi [9]. However, recent works have shown that the number of spikes drastically vanish at deeper layers, leading to performance degradations in deep SNNs [18]. Thus, there is a need for an efficient hybrid architecture, with SNNs in the initial layers, to exploit their compatability with event camera outputs while having ANNs in the deeper layers in order to retain performance.

In regard to this, we propose a deep hybrid neural network architecture, accommodating SNNs and ANNs in different layers, for energy efficient optical flow estimation using sparse event camera data. To the best of our knowledge, this is the first SNN demonstration to report the state-of-art performance on event-based optical flow estimation, outperforming its corresponding fully-fledged ANN counterpart.

The main contributions of this work can be summarized as:

- We present an input representation that efficiently encodes the sequences of sparse outputs from event cameras over time to preserve the spatio-temporal nature of spike events.
- We introduce a deep hybrid architecture for event-based optical flow estimation referred to as Spike-FlowNet, integrating SNNs and ANNs in different layers, to efficiently process the sparse spatio-temporal event inputs.
- We evaluate the optical flow prediction capability and computational efficiency of Spike-FlowNet on the Multi-Vehicle Stereo Event Camera dataset (MVSEC) [33] and provide comparison results with current state-of-the-art approaches.

The following contents are structured as follows. In Sect. 2, we elucidate the related works. In Sect. 3, we present the methodology, covering essential backgrounds on the spiking neuron model followed by our proposed input event (spike) representation. This section also discusses the self-supervised loss, Spike-FlowNet architecture, and the approximate backpropagation algorithm used for training. Section 4 covers the experimental results, including training details and evaluation metrics. It also discusses the comparison results with the latest works in terms of performance and computational efficiency.

2 Related Work

In recent years, there have been an increasing number of works on estimating optical flow by exploiting the high temporal resolution of event cameras. In general, these approaches have either been adaptations of conventional computer vision methods or modified versions of deep ANNs to encompass discrete outputs from event cameras.

For computer vision based solutions to estimate optical flow, gradient-based approaches using the Lucas-Kanade algorithm [22] have been highlighted in [4,7]. Further, plane fitting approaches by computing the slope of the plane for estimating optical flow have been presented in [1,3]. In addition, bio-inspired frequency-based approaches have been discussed in [2]. Finally, correlation-based approaches are presented in [12,32] employing convex optimization over events. In addition, [21] interestingly uses an adaptive block matching technique to estimate sparse optical flow.

For deep ANN-based solutions, optical flow estimation from frame-based images has been discussed in Unflow [23], which utilizes a U-Net [28] architecture and computes a bidirectional census loss in an unsupervised manner with an added smoothness term. This strategy is modified for event camera outputs in EV-FlowNet [34] incorporating a self-supervised loss based on gray images as a replacement for ground truth. Other previous works employ various modifications to the training methodology, such as [15], which imposes certain brightness constancy and smoothness constraints to train a network and [17] which adds an adversarial loss over the standard photometric loss. In contrast, [35] presents an unsupervised learning approach using only event camera data to estimate optical flow by accounting for and then learning to rectify the motion blur.

All the above strategies employ ANN architectures to predict the optical flow. However, event cameras produce asynchronous and discrete outputs over time, and SNNs can naturally capture their spatio-temporal dynamics, which are embedded in the precise spike timings. Hence, we posit that SNNs are suitable for handling event camera outputs. Recent SNN-based approaches for event-based optical flow estimation include [13,25,27]. Researchers in [25] presented visual motion estimation using SNNs, which accounts for synaptic delays in generating motion-sensitive receptive fields. In addition, [13] demonstrated real-time model-based optical flow computations on TrueNorth hardware for evaluating patterns including rotating spirals and pipes. Authors of [27] presented a methodology for optical flow estimation using convolutional SNNs based on Spike-Time-Dependent-Plasticity (STDP) learning [11]. The main limitation of these works is that they employ shallow SNN architectures, because deep SNNs suffer in terms of performance. Besides, the presented results are only evaluated on relatively simple tasks. In practice, they do not generally scale well to complex and real-world data, such as that presented in MVSEC dataset [33]. In view of these, a hybrid approach becomes an attractive option for constructing deep network architectures, leveraging the benefits of both SNNs and ANNs.

3 Method

3.1 Spiking Neuron Model

The spiking neurons, inspired by biological models [10], are computational primitives in SNNs. We employ a simple IF neuron model, which transmits the output signals in the form of spike events over time. The behavior of IF neuron at the l^{th} layer is illustrated in Fig. 1. The input spikes are weighted to produce an influx current that integrates into neuronal membrane potential (V^l).

$$V^l[n+1] = V^l[n] + w^l * o^{l-1}[n] \tag{1}$$

where $V^l[n]$ represents the membrane potential at discrete time-step n, w^l represents the synaptic weights and $o^{l-1}[n]$ represents the spike events from the previous layer at discrete time-step n. When the membrane potential overcomes the firing threshold, the neuron emits an output spike and resets the membrane potential to the initial state (zero). Over time, these mechanisms are repeatedly carried out in each IF neuron, enabling event-based computations throughout the SNN layers.

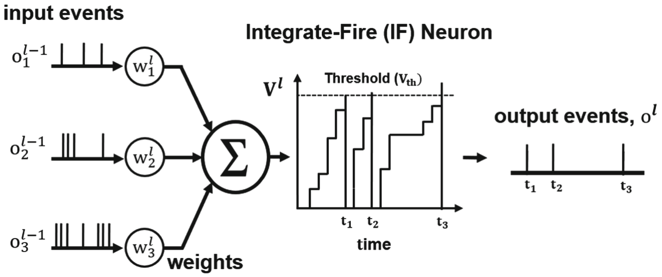

Fig. 1. The dynamics of an Integrate and Fire (IF) neuron. The input events are modulated by the synaptic weight to be integrated as the current influx in the membrane potential. Whenever the membrane potential crosses the threshold, the neuron fires an output spike and resets the membrane potential.

3.2 Spiking Input Event Representation

An event-based camera tracks the changes in log-scale intensity (I) at every element in the pixel-array independently and generates a discrete event whenever the change exceeds a threshold (θ):

$$\|\log(I_{t+1}) - \log(I_t)\| \geq \theta \tag{2}$$

A discrete event contains a 4-tuple $\{x, y, t, p\}$, consisting of the coordinates: x, y; timestamp: t; and polarity (direction) of brightness change: p. This input representation is called Address Event Representation (AER), and is the standard format used by event-based sensors.

There are prior works that have modified the representations of asynchronous event camera outputs to be compatible with ANN-based methods. To overcome the asynchronous nature, event outputs are typically recorded for a certain time period and transformed into a synchronous image-like representation. In EV-FlowNet [34], the most recent pixel-wise timestamps and the event counts encoded the motion information (within a time window) in an image. However, fast motions and dense events (in local regions of the image) can vastly overlap per-pixel timestamp information, and temporal information can be lost. In addition, [35] proposed a discretized event volume that deals with the time domain as a channel to retain the spatio-temporal event distributions. However, the number of input channels increases significantly as the time dimensions are finely discretized, further aggravating the computation and parameter overheads.

In this work, we propose a discretized input representation (fine-grained in time) that preserves the spatial and temporal information of events for SNNs.

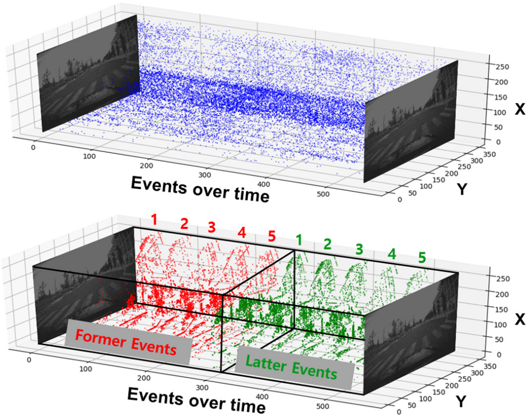

Fig. 2. Input event representation. (*Top*) Continuous raw events between two consecutive grayscale images from an event camera. (*Bottom*) Accumulated event frames between two consecutive grayscale images to form the former and the latter event groups, serving as inputs to the network.

Our proposed input encoding scheme discretizes the time dimension within a time window into two groups (former and latter). Each group contains N number of event frames obtained by accumulating raw events from the timestamp of the previous frame till the current timestamp. Each of these event frames is also composed of two channels for ON/OFF polarity of events. Hence, the input to the network consists of a sequence of N frames with four channels (one frame each from the former and the latter groups having two channels each). The proposed input representation is displayed in Fig. 2 for one channel (assuming the number of event frames in each group equals to five). The main characteristic of our proposed input event representation (compared to ANN-based methods) are as follows:

- Our spatio-temporal input representations encode only the presence of events over time, allowing asynchronous and event-based computations in SNNs. In contrast, ANN-based input representation often requires the timestamp and the event count images in separate channels.
- In Spike-FlowNet, each event frame from the former and the latter groups sequentially passes through the network, thereby preserving and utilizing the spatial and temporal information over time. On the contrary, ANN-based methods feed-forward all input information to the network at once.

3.3 Self-Supervised Loss

The DAVIS camera [6] is a commercially available event-camera, which simultaneously provides synchronous grayscale images and asynchronous event streams. The number of available event-based camera datasets with annotated labels suitable for optical flow estimation is quite small, as compared to frame-based camera datasets. Hence, a self-supervised learning method that uses proxy labels from the recorded grayscale images [15,34] is employed for training our Spike-FlowNet.

The overall loss incorporates a photometric reconstruction loss ($\mathcal{L}_{\text{photo}}$) and a smoothness loss ($\mathcal{L}_{\text{smooth}}$) [15]. To evaluate the photometric loss within each time window, the network is provided with the former and the latter event groups and a pair of grayscale images, taken at the start and the end of the event time window (I_t, I_{t+dt}). The predicted optical flow from the network is used to warp the second grayscale image to the first grayscale image. The photometric loss ($\mathcal{L}_{\text{photo}}$) aims to minimize the discrepancy between the first grayscale image and the inverse warped second grayscale image. This loss uses the photo-consistency assumption that a pixel in the first image remains similar in the second frame mapped by the predicted optical flow. The photometric loss is computed as follows:

$$\mathcal{L}_{\text{photo}}(u, v; I_t, I_{t+dt}) = \sum_{x,y} \rho(I_t(x, y) - I_{t+dt}(x + u(x, y),\ y + v(x, y))) \tag{3}$$

where, I_t, I_{t+dt} indicate the pixel intensity of the first and second grayscale images, u, v are the flow estimates in the horizontal and vertical directions, ρ is the Charbonnier loss $\rho(x) = (x^2 + \eta^2)^r$, which is a generic loss used for outlier rejection in optical flow estimation [30]. For our work, $r = 0.45$ and η =1e-3 show the optimum results for the computation of photometric loss.

Furthermore, a smoothness loss ($\mathcal{L}_{\text{smooth}}$) is applied for enhancing the spatial collinearity of neighboring optical flow. The smoothness loss minimizes the difference in optical flow between neighboring pixels and acts as a regularizer on the predicted flow. It is computed as follows:

$$\begin{aligned}\mathcal{L}_{\text{smooth}}(u, v) = \frac{1}{HD} \sum_{j}^{H} \sum_{i}^{D} (&\|u_{i,j} - u_{i+1,j}\| + \|u_{i,j} - u_{i,j+1}\| \\ &+ \|v_{i,j} - v_{i+1,j}\| + \|v_{i,j} - v_{i,j+1}\|)\end{aligned} \tag{4}$$

where H is the height and D is the width of the predicted flow output. The overall loss is computed as the weighted sum of the photometric and smoothness loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{photo}} + \lambda \mathcal{L}_{\text{smooth}} \tag{5}$$

where λ is the weight factor.

3.4 Spike-FlowNet Architecture

Spike-FlowNet employs a deep hybrid architecture that accommodates SNNs and ANNs in different layers, enabling the benefits of SNNs for sparse event data processing and ANNs for maintaining the performance. The use of a hybrid architecture is attributed to the fact that spike activities reduce drastically with

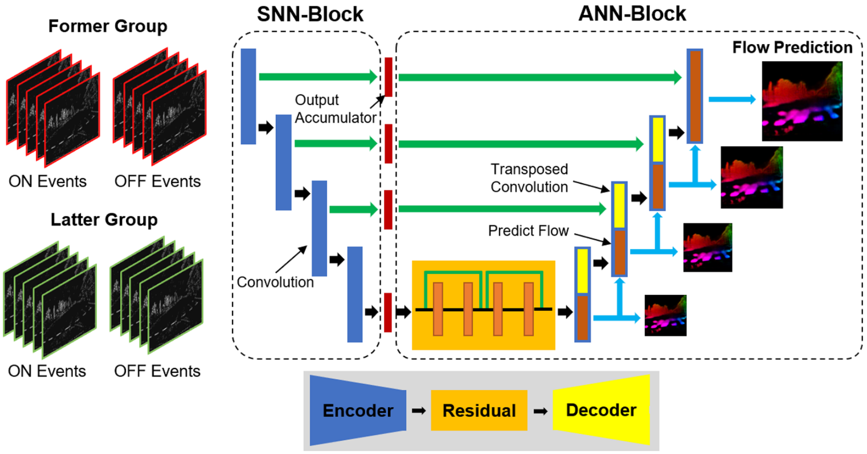

Fig. 3. Spike-FlowNet architecture. The four-channeled input images, comprised of ON/OFF polarity events for former and latter groups, are sequentially passed through the hybrid network. The SNN-block contains the encoder layers followed by output accumulators, while the ANN-block contains the residual and decoder layers. The loss is evaluated after forward propagating all consecutive input event frames (a total of N inputs, sequentially taken in time from the former and the latter event groups) within the time window. The black arrows denote the forward path, green arrows represent residual connections, and blue arrows indicate the flow predictions. (Color figure online)

growing the network depth in the case of full-fledged SNNs. This is commonly referred to as the vanishing spike phenomenon [26], and potentially leads to performance degradation in deep SNNs. Furthermore, high numerical precision is essentially required for estimating the accurate pixel-wise network outputs, namely the regression tasks. Hence, very rare and binary precision spike signals (in input and intermediate layers) pose a crucial issue for predicting the accurate flow displacements. To resolve these issues, only the encoder block is built as an SNN, while the residual and decoder blocks maintain an ANN architecture.

Spike-FlowNet's network topology resembles the U-Net [28] architecture, containing four encoder layers, two residual blocks, and four decoder layers as shown in Fig. 3. The events are represented as the four-channeled input frames as presented in Sect. 3.2, and are sequentially passed through the SNN-based encoder layers over time (while being downsampled at each layer). Convolutions with a stride of two are employed for incorporating the functionality of dimensionality reduction in the encoder layers. The outputs from encoder layers are collected in their corresponding output accumulators until all consecutive event images have passed. Next, the accumulated outputs from final encoder layer are passed through two residual blocks and four decoder layers. The decoder layers upsample the activations using transposed convolution. At each decoder layer, there is a skip connection from the corresponding encoder layer, as well as another convolution layer to produce an intermediate flow prediction, which is concatenated with the activations from the transposed convolutions. The total loss is evaluated after the forward propagation of all consecutive input event frames through the network and is applied to each of the intermediate dense optical flows using the grayscale images.

Algorithm 1. Backpropagation Training in Spike-FlowNet for an Iteration.

```
Input: Event-based inputs ($inputs$), total number of discrete time-steps ($N$), number of SNN/ANN layers ($L_S/L_A$), SNN/ANN outputs ($o/o_A$) membrane potential ($V$), firing threshold ($V_{th}$), ANN nonlinearity ($f$)
Initialize: $V^l[n] = 0,\ \forall l = 1, ..., L_S$
// Forward Phase in SNN-blocks
for $n \leftarrow 1$ to $N$ do
  $o^1[n] = inputs[n]$
  for $l \leftarrow 2$ to $L_S - 1$ do
    $V^l[n] \ = V^l[n-1] + w^l o^{l-1}[n]$ //weighted spike-inputs are integrated to V
    if $V^l[n] > V_{th}$ then
      $o^l[n] = 1,\ V^l[n] = 0$ //if V exceeds $V_{th}$, a neuron emits a spike and reset V
    end if
  end for
  $o_A^{L_S} = V^{L_S}[n] \ = V^{L_S}[n-1] + w^{L_S} o^{L_S - 1}[n]$ //final SNN layer does not fire
end for
// Forward Phase in ANN-blocks
for $l \leftarrow L_S + 1$ to $L_S + L_A$ do
  $o_A^l \ = f(w^l o_A^{l-1})$
end for
// Backward Phase in ANN-blocks
for $l \leftarrow L_S + L_A$ to $L_S$ do
  $\triangle w^l = \frac{\partial \mathcal{L}_{total}}{\partial o_A^l} \frac{\partial o_A^l}{\partial w^l}$
end for
// Backward Phase in SNN-blocks
for $n \leftarrow N$ to 1 do
  for $l \leftarrow L_S - 1$ to 1 do
    //evaluate partial derivatives of loss w.r.t. $w_S$ by unrolling the SNN over time
    $\triangle w^l[n] = \frac{\partial \mathcal{L}_{total}}{\partial o^l[n]} \frac{\partial o^l[n]}{\partial V^l[n]} \frac{\partial V^l[n]}{\partial w^l[n]}$
  end for
end for
```

3.5 Backpropagation Training in Spike-FlowNet

The spike generation function of an IF neuron is a hard threshold function that emits a spike when the membrane potential exceeds a firing threshold. Due to this discontinuous and non-differentiable neuron model, standard backpropagation algorithms cannot be applied to SNNs in their native form. Hence, several approximate methods have been proposed to estimate the surrogate gradient of spike generation function. In this work, we adopt the approximate gradient method proposed in [18,19] for back-propagating errors through SNN layers. The approximate IF gradient is computed as $\frac{1}{V_{th}}$, where the threshold value accounts for the change of the spiking output with respect to the input. Algorithm 1 illustrates the forward and backward pass in ANN-block and SNN-block.

In the forward phase, neurons in the SNN layers accumulate the weighted sum of the spike inputs in membrane potential. If the membrane potential exceeds a threshold, a neuron emits a spike at its output and resets. The final SNN layer neurons just integrate the weighted sum of spike inputs in the output accumulator, while not producing any spikes at the output. At the last time-step, the integrated outputs of SNN layers propagate to the ANN layers to predict the optical flow. After the forward pass, the final loss ($\mathcal{L}_{total}$) is evaluated, followed by backpropagation of gradients through the ANN layers using standard backpropagation.

Next, the backpropagated errors ($\frac{\partial \mathcal{L}_{total}}{\partial o^{L_S}}$) pass through the SNN layers using the approximate IF gradient method and BackPropagation Through Time (BPTT) [31]. In BPTT, the network is unrolled for all discrete time-steps, and the weight update is computed as the sum of gradients from each time-step. This procedure is displayed in Fig. 4 where the final loss is back-propagated through an ANN-block and a simple SNN-block consisting of a single input IF neuron. The parameter updates of the l^{th} SNN layers are described as follows:

$$\triangle w^l = \sum_n \frac{\partial \mathcal{L}_{total}}{\partial o^l[n]} \frac{\partial o^l[n]}{\partial V^l[n]} \frac{\partial V^l[n]}{\partial w^l}, \text{ where } \frac{\partial o^l[n]}{\partial V^l[n]} = \frac{1}{V_{th}}(o^l[n] > 0) \tag{6}$$

where o^l represents the output of spike generation function. This method enables the end-to-end self-supervised training in the proposed hybrid architecture.

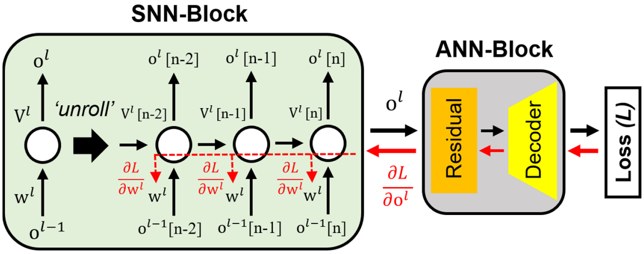

Fig. 4. Error backpropagation in Spike-FlowNet. After the forward pass, the gradients are back-propagated through the ANN block using standard backpropagation whereas the backpropagated errors ($\frac{\partial \mathcal{L}}{\partial o^l}$) pass through the SNN layers using the approximate IF gradient method and BPTT technique.

4 Experimental Results

4.1 Dataset and Training Details

We use the MVSEC dataset [33] for training and evaluating the optical flow predictions. MVSEC contains stereo event-based camera data for a variety of environments (e.g., indoor flying and outdoor driving) and also provides the corresponding ground truth optical flow. In particular, the indoor and outdoor sequences are recorded in dissimilar environments where the indoor sequences (indoor_flying) have been captured in a lab environment and the outdoor sequences (outdoor_day) have been recorded while driving on public roads.

Even though the indoor_flying and outdoor_day scenes are quite different, we only use outdoor_day2 sequence for training Spike-FlowNet. This is done to provide fair comparisons with prior works [34,35] which utilized only outdoor_day2 sequence for training. During training, input images are randomly flipped horizontally and vertically (with 0.5 probability) and randomly cropped to 256 × 256 size. Adam optimizer [16] is used, with the initial learning rate of 5e-5, and scaled by 0.7 every 5 epochs until 10 epoch, and every 10 epochs thereafter. The model is trained on the left event camera data of outdoor_day2 sequence for 100 epochs with a mini-batch size 8. Training is done for two different time windows lengths (i.e, 1 grayscale image frame apart ($dt = 1$) and 4 grayscale image frames apart ($dt = 4$)). The number of event frame (N) and weight factor for the smoothness loss (λ) are set to 5, 10 for a $dt = 1$ case and 20, 1 for a $dt = 4$ case, respectively. The threshold of the IF neurons are set to 0.5 ($dt = 4$) and 0.75 ($dt = 1$) in SNN layers.

4.2 Algorithm Evaluation Metric

The evaluation metric for optical flow prediction is the Average End-point Error (AEE), which represents the mean distance between the predicted flow (y_{pred}) and the ground truth flow (y_{gt}). It is given by:

$$\text{AEE} = \frac{1}{m} \sum_{\text{m}} \|(u, v)_{\text{pred}} - (u, v)_{\text{gt}}\|_2 \tag{7}$$

where m is the number of active pixels in the input images. Because of the highly sparse nature of input events, the optical flows are only estimated at pixels where both the events and ground truth data is present. We compute the AEE for $dt = 1$ and $dt = 4$ cases.

4.3 Average End-Point Error (AEE) Results

During testing, optical flow is estimated on the center cropped 256 × 256 left camera images of the indoor_flying 1,2,3 and outdoor_day 1 sequences. We use all events for the indoor_flying sequences, but we take events within 800 grayscale frames for the outdoor_day1 sequence, similar to [34]. Table 1 provides the AEE

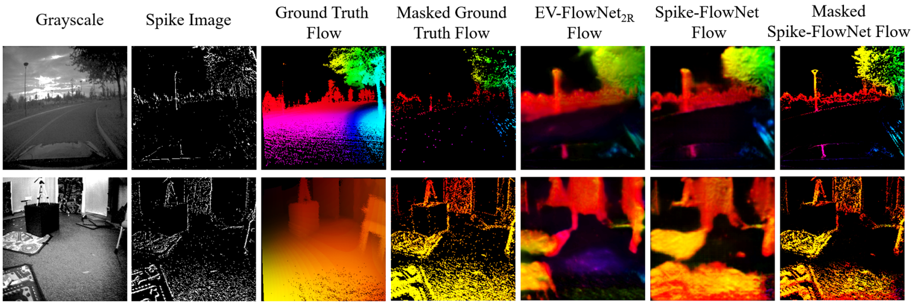

Fig. 5. Optical flow evaluation and comparison with EV-FlowNet. The samples are taken from (*top*) *outdoor_day*1 and (*bottom*) *indoor_day*1. The Masked Spike-FlowNet Flow is basically a sparse optical flow computed at pixels at which events occurred. It is computed by masking the predicted optical flow with the spike image.

evaluation results in comparison with the prior event camera based optical flow estimation works. Overall, our results show that Spike-FlowNet can accurately predict the optical flow in both the indoor_flying and outdoor_day1 sequences. This demonstrates that the proposed Spike-FlowNet can generalize well to distinctly different environments. The grayscale, spike event, ground truth flow and the corresponding predicted flow images are visualized in Fig. 5 where the images are taken from (*top*) *outdoor_day*1 and (*bottom*) *indoor_day*1, respectively. Since event cameras work based on changing light intensity at pixels, the regions having low texture produce very sparse events due to minimal intensity changes, resulting in scarce optical flow predictions in the corresponding areas such as the flat surfaces. Practically, the useful flows are extracted by using flow estimations at points where significant events exist in the input frames.

Moreover, we compare our quantitative results with the recent works [34,35] on event-based optical flow estimation, as listed in Table 1. We observe that Spike-FlowNet outperforms EV-FlowNet [34] in terms of AEE results in both the $dt = 1$ and $dt = 4$ cases. It is worth noting here that EV-FlowNet employs a similar network architecture and self-supervised learning method, providing a

Table 1. Average Endpoint Error (AEE) comparisons with Zhu et al. [35] and EV-FlowNet [34].

	dt = 1 frame				dt = 4 frame			
	indoor1	indoor2	indoor3	outdoor1	indoor1	indoor2	indoor3	outdoor1
Zhu et al. [35]	**0.58**	**1.02**	**0.87**	**0.32**	**2.18**	3.85	**3.18**	1.30
EV-FlowNet [34]	1.03	1.72	1.53	0.49	2.25	4.05	3.45	1.23
This work	0.84	1.28	1.11	0.49	2.24	**3.83**	**3.18**	**1.09**

* EV-FlowNet also uses a self-supervised learning method, providing the fair comparison baseline compared to Spike-FlowNet.

fair comparison baseline for fully ANN architectures. In addition, Spike-FlowNet attains AEE results slightly better or comparable to [35] in the $dt = 4$ case, while underperforming in the $dt = 1$ case. [35] presented an image deblurring based unsupervised learning that employed only the event streams. Hence, it seems to not suffer from the issues related to grayscale images such as motion blur or aperture problems during training. In view of these comparisons, Spike-FlowNet (with presented spatio-temporal event representation) is more suitable for motion detection when the input events have a certain minimum level of spike density. We further provide the ablation studies for exploring the optimal design choices in the supplementary material.

4.4 Computational Efficiency

To further analyze the benefits of Spike-FlowNet, we estimate the gain in computational costs compared to a fully ANN architecture. Typically, the number of synaptic operations is used as a metric for benchmarking the computational energy of neuromorphic hardware [18,24,29]. Also, the required energy consumption per synaptic operation needs to be considered. Now, we describe the procedures for measuring the computational costs in SNN and ANN layers.

In a neuromorphic hardware, SNNs carry out event-based computations only at the arrival of input spikes. Hence, we first measure the mean spike activities at each time-step in the SNN layers. As presented in the first row of Table 2, the mean spiking activities (averaged over indoor1,2,3 and outdoor1 sequences) are

Table 2. Analysis for Spike-FlowNet in terms of the mean spike activity, the total and normalized number of SNN operations in an encoder-block, the encoder-block and overall computational energy benefits.

	indoor1		indoor2		indoor3		outdoor1	
	dt = 1	dt = 4	dt = 1	dt = 4	dt = 1	dt = 4	dt = 1	dt = 4
Encoder spike activity (%)	**0.33**	0.87	0.65	1.27	0.53	1.11	0.41	0.78
Encoder SNN # operation ($\times 10^8$)	**0.16**	1.69	0.32	2.47	0.26	2.15	0.21	1.53
Encoder normalized # operation (%)	**1.68**	17.87	3.49	26.21	2.81	22.78	2.29	16.23
Encoder compute-energy benefit (×)	**305**	28.6	146.5	19.5	182.1	22.44	223.2	31.5
Overall compute-energy reduction (%)	**17.57**	17.01	17.51	16.72	17.53	16.84	17.55	17.07

* For an ANN, the number of synaptic operations is 9.44×10^8 for the encoder-block and 5.35×10^9 for overall network.

0.48% and 1.01% for $dt = 1$ and $dt = 4$ cases, respectively. Note that the neuronal threshold is set to a higher value in $dt = 1$ case; hence the average spiking activity becomes sparser compared to $dt = 4$ case. The extremely rare mean input spiking activities are mainly due to the fact that event camera outputs are highly sparse in nature. This sparse firing rate is essential for exploiting efficient event-based computations in SNN layers. In contrast, ANNs execute dense matrix-vector multiplication operations without considering the sparsity of inputs. In other words, ANNs simply feed-forward the inputs at once, and the total number of operations are fixed. This leads to the high energy requirements (compared to SNNs) by computing both zero and non-zero entities, especially when inputs are very sparse.

Essentially, SNNs need to compute the spatio-temporal spike images over a number of time-steps. Given M is the number of neurons, C is number of synaptic connections and F indicates the mean firing activity, the number of synaptic operations at each time-step in the l^{th} layer is calculated as $M_l \times C_l \times F_l$. The total number of SNN operations is the summation of synaptic operations in SNN layers during the N time-steps. Hence, the total number of SNN and ANN operations become $\sum_l (M_l \times C_l \times F_l) \times N$ and $\sum_l M_l \times C_l$, respectively. Based on these, we estimate and compare the average number of synaptic operations on Spike-FlowNet and a fully ANN architecture. The total and the normalized number of SNN operations compared to ANN operations on the encoder-block are provided in the second and the third row of Table 2, respectively.

Due to the binary nature of spike events, SNNs perform only accumulation (AC) per synaptic operation. On the other hand, ANNs perform the multiply-accumulate (MAC) computations since the inputs consist of analog-valued entities. In general, AC computation is considered to be significantly more energy-efficient than MAC. For example, AC is reported to be 5.1× more energy-efficient than a MAC in the case of 32-bit floating-point numbers (45 nm CMOS process) [14]. Based on this principle, the computational energy benefits of encoder-block and overall Spike-FlowNet are obtained, as provided in the fourth and the fifth rows of Table 2, respectively. These results reveal that the SNN-based encoder-block is 214.2× and 25.51× more computationally efficient compared to ANN-based one (averaged over indoor1,2,3 and outdoor1 sequences) for $dt = 1$ and $dt = 4$ cases, respectively. The number of time-steps (N) is four times less in $dt = 1$ case than in $dt = 4$ case; hence, the computational energy benefit is much higher in $dt = 1$ case.

From our analysis, the proportion of required computations in encoder-block compared to the overall architecture is 17.6%. This reduces the overall energy benefits of Spike-FlowNet. In such a case, an approach of interest would be to perform a distributed edge-cloud implementation where the SNN- and ANN-blocks are administered on the edge device and the cloud, respectively. This would lead to high energy benefits on edge devices, which are limited by resource constraints while not compromising on algorithmic performance.

5 Conclusion

In this work, we propose Spike-FlowNet, a deep hybrid architecture for energy-efficient optical flow estimations using event camera data. To leverage the benefits of both SNNs and ANNs, we integrate them in different layers for resolving the spike vanishing issue in deep SNNs. Moreover, we present a novel input encoding strategy for handling outputs from event cameras, preserving the spatial and temporal information over time. Spike-FlowNet is trained with a self-supervised learning method, bypassing expensive labeling. The experimental results show that the proposed architecture accurately predicts the optical flow from discrete and asynchronous event streams along with substantial benefits in terms of computational efficiency compared to the corresponding ANN architecture.

Acknowledgment. This work was supported in part by C-BRIC, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA, the National Science Foundation, Sandia National Laboratory, and the DoD Vannevar Bush Fellowship.

References

1. Aung, M.T., Teo, R., Orchard, G.: Event-based plane-fitting optical flow for dynamic vision sensors in FPGA. In: 2018 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1–5, May 2018. https://doi.org/10.1109/ISCAS.2018.8351588
2. Barranco, F., Fermuller, C., Aloimonos, Y.: Bio-inspired motion estimation with event-driven sensors. In: Rojas, I., Joya, G., Catala, A. (eds.) IWANN 2015. LNCS, vol. 9094, pp. 309–321. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19258-1_27
3. Benosman, R., Clercq, C., Lagorce, X., Ieng, S., Bartolozzi, C.: Event-based visual flow. IEEE Transa. Neural Networks Learn. Syst. **25**(2), 407–417 (2014). https://doi.org/10.1109/TNNLS.2013.2273537
4. Benosman, R., Ieng, S.H., Clercq, C., Bartolozzi, C., Srinivasan, M.: Asynchronous frameless event-based optical flow. Neural Networks **27**, 32–37 (2012). https://doi.org/10.1016/j.neunet.2011.11.001. http://www.sciencedirect.com/science/article/pii/S0893608011002930
5. Borst, A., Haag, J., Reiff, D.F.: Fly motion vision. Ann. Rev. Neurosci. **33**(1), 49–70 (2010). https://doi.org/10.1146/annurev-neuro-060909-153155. https://doi.org/10.1146/annurev-neuro-060909-153155, pMID: 20225934
6. Brandli, C., Berner, R., Yang, M., Liu, S., Delbruck, T.: A 240 × 180 130 db 3 µs latency global shutter spatiotemporal vision sensor. IEEE J. Solid-State Circuits **49**(10), 2333–2341 (2014). https://doi.org/10.1109/JSSC.2014.2342715
7. Brosch, T., Tschechne, S., Neumann, H.: On event-based optical flow detection. Front. Neurosci. **9**, 137 (2015). https://doi.org/10.3389/fnins.2015.00137
8. Burkitt, A.N.: A review of the integrate-and-fire neuron model: I. homogeneous synaptic input. Biol. Cybern. **95**(1), 1–19 (2006)
9. Davies, M., et al.: Loihi: a neuromorphic manycore processor with on-chip learning. IEEE Micro **38**(1), 82–99 (2018). https://doi.org/10.1109/MM.2018.112130359

10. Dayan, P., Abbott, L.F.: Theoretical Neurosci., vol. 806. MIT Press, Cambridge (2001)
11. Diehl, P.U., Cook, M.: Unsupervised learning of digit recognition using spike-timing-dependent plasticity. Front. Comput. Neurosci. **9**, 99 (2015)
12. Gallego, G., Rebecq, H., Scaramuzza, D.: A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation. CoRR abs/1804.01306 (2018). http://arxiv.org/abs/1804.01306
13. Haessig, G., Cassidy, A., Alvarez, R., Benosman, R., Orchard, G.: Spiking optical flow for event-based sensors using ibm's truenorth neurosynaptic system. IEEE Trans. Biomed. Circuits Syst. **12**(4), 860–870 (2018)
14. Horowitz, M.: 1.1 computing's energy problem (and what we can do about it). In: 2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC), pp. 10–14. IEEE (2014)
15. Yu, J.J., Harley, A.W., Derpanis, K.G.: Back to basics: unsupervised learning of optical flow via brightness constancy and motion smoothness. In: Hua, G., Jégou, H. (eds.) ECCV 2016. LNCS, vol. 9915, pp. 3–10. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49409-8_1
16. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
17. Lai, W.S., Huang, J.B., Yang, M.H.: Semi-supervised learning for optical flow with generative adversarial networks. In: Guyon, I., et al. (eds.) Advances in Neural Information Processing Systems 30, pp. 354–364. Curran Associates, Inc. (2017). http://papers.nips.cc/paper/6639-semi-supervised-learning-for-optical-flow-with-generative-adversarial-networks.pdf
18. Lee, C., Sarwar, S.S., Panda, P., Srinivasan, G., Roy, K.: Enabling spike-based backpropagation for training deep neural network architectures. Front. Neurosci. **14**, 119 (2020)
19. Lee, J.H., Delbruck, T., Pfeiffer, M.: Training deep spiking neural networks using backpropagation. Front. Neurosci. **10**, 508 (2016)
20. Lichtsteiner, P., Posch, C., Delbruck, T.: A 128× 128 120 db 15 μs latency asynchronous temporal contrast vision sensor. IEEE J. Solid-State Circuits **43**(2), 566–576 (2008). https://doi.org/10.1109/JSSC.2007.914337
21. Liu, M., Delbrück, T.: ABMOF: A novel optical flow algorithm for dynamic vision sensors. CoRR abs/1805.03988 (2018). http://arxiv.org/abs/1805.03988
22. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2, IJCAI 1981, pp. 674–679. Morgan Kaufmann Publishers Inc., San Francisco (1981). http://dl.acm.org/citation.cfm?id=1623264.1623280
23. Meister, S., Hur, J., Roth, S.: Unflow: unsupervised learning of optical flow with a bidirectional census loss. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
24. Merolla, P.A., et al.: A million spiking-neuron integrated circuit with a scalable communication network and interface. Science **345**(6197), 668–673 (2014)
25. Orchard, G., Benosman, R.B., Etienne-Cummings, R., Thakor, N.V.: A spiking neural network architecture for visual motion estimation. In: 2013 IEEE Biomedical Circuits and Systems Conference (BioCAS), pp. 298–301 (2013)
26. Panda, P., Aketi, S.A., Roy, K.: Toward scalable, efficient, and accurate deep spiking neural networks with backward residual connections, stochastic softmax, and hybridization. Front. Neurosci. **14**, 653 (2020)

27. Paredes-Vallés, F., Scheper, K.Y.W., De Croon, G.C.H.E.: Unsupervised learning of a hierarchical spiking neural network for optical flow estimation: from events to global motion perception. IEEE Trans. Pattern Anal. Mach. Intell. **42**, 2051–2064 (2019). https://ieeexplore.ieee.org/abstract/document/8660483
28. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. CoRR abs/1505.04597 (2015). http://arxiv.org/abs/1505.04597
29. Rueckauer, B., Lungu, I.A., Hu, Y., Pfeiffer, M., Liu, S.C.: Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. Front. Neurosci. **11**, 682 (2017)
30. Sun, D., Roth, S., Black, M.J.: A quantitative analysis of current practices in optical flow estimation and the principles behind them. Int. J. Comput. Vision **106**(2), 115–137 (2014). https://doi.org/10.1007/s11263-013-0644-x. http://dx.doi.org/10.1007/s11263-013-0644-x
31. Werbos, P.J.: Backpropagation through time: what it does and how to do it. Proc. IEEE **78**(10), 1550–1560 (1990)
32. Zhu, A.Z., Atanasov, N., Daniilidis, K.: Event-based feature tracking with probabilistic data association. In: 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 4465–4470, May 2017. https://doi.org/10.1109/ICRA.2017.7989517
33. Zhu, A.Z., Thakur, D., Özaslan, T., Pfrommer, B., Kumar, V., Daniilidis, K.: The multivehicle stereo event camera dataset: an event camera dataset for 3D perception. IEEE Robot. Autom. Lett. **3**(3), 2032–2039 (2018)
34. Zhu, A.Z., Yuan, L., Chaney, K., Daniilidis, K.: Ev-flownet: self-supervised optical flow estimation for event-based cameras. arXiv preprint arXiv:1802.06898 (2018)
35. Zhu, A.Z., Yuan, L., Chaney, K., Daniilidis, K.: Unsupervised event-based learning of optical flow, depth, and egomotion. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 989–997 (2019)