Spatio-Temporal Recurrent Networks for Event-Based Optical Flow Estimation

Ziluo Ding, Rui Zhao, Jiyuan Zhang, Tianxiao Gao, Ruiqin Xiong, Zhaofei Yu[*] Tiejun Huang

o Peking University
{ziluo, gtx, rqxiong, yuzf12, tjhuang}@pku.edu.cn, {ruizhao, jyzhang}@stu.pku.edu.cn

Abstract

Event camera has offered promising alternative for visual perception, especially in high speed and high dynamic range scenes. Recently, many deep learning methods have shown great success in providing model-free solutions to many event-based problems, such as optical flow estimation. However, existing deep learning methods did not address the importance of temporal information well from the perspective of architecture design and cannot effectively extract spatio-temporal features. Another line of research that utilizes Spiking Neural Network suffers from training issues for deeper architecture. To address these points, a novel input representation is proposed that captures the events' temporal distribution for signal enhancement. Moreover, we introduce a spatio-temporal recurrent encoding-decoding neural network architecture for event-based optical flow estimation, which utilizes Convolutional Gated Recurrent Units to extract feature maps from a series of event images. Besides, our architecture allows some traditional frame-based core modules, such as correlation layer and iterative residual refine scheme, to be incorporated. The network is end-to-end trained with self-supervised learning on the Multi-Vehicle Stereo Event Camera dataset. We have shown that it outperforms all the existing state-of-the-art methods by a large margin.

Introduction

We have witnessed the prosperous development of neuromorphic cameras, which offer promising alternatives for visual perception. For example, one of the most popular neuromorphic vision sensors, the event camera (Lichtsteiner, Posch, and Delbruck 2008; Brandli et al. 2014), has exhibited better potentials in handling high-speed scenarios and demonstrated superiority in robustness compared with frame-based cameras. Biologically inspired by the retinal periphery, each pixel responds independently to the change of luminance by generating asynchronous events with microsecond accuracy in the event camera. Therefore, event camera greatly reduces the amount of data needed to be processed, and their distinct and novel working principles provide some promising advantages such as extremely low latency, high dynamic range, and low power consumption, making them a good fit for domains such as optical flow, object tracking, or dynamic scene understanding. Optical flow estimation has a wide range of applications, especially in autonomous driving (Aufrère et al. 2003; Capito, Ozguner, and Redmill 2020) and action recognition (Efros et al. 2003). Although optical flow estimation has achieved remarkable success in frame-based vision (Teed and Deng 2020; Wang, Fan, and Liu 2020; Liu et al. 2020), it is unlikely to directly apply traditional frame-based optical flow estimation algorithms to event data since the data formats in the two fields are too different.

Recently, many research studies focus on training an end-to-end neural network to estimate optical flow in a self or unsupervised manner (Zhu et al. 2019; Ye et al. 2018; Lee et al. 2020; Zhu et al. 2018b). In most existing deep learning works, a series of asynchronous events are first transformed into alternative representations–event image that summarizes events into a 2D grid. Then, all the event images or feature maps are concatenated as one entity and fed into Convolutional Neural Network (ConvNet) at one time. Although this practice contains spatial information about scene edges that is familiar to conventional computer vision, it did not address the importance of temporal information well from the perspective of architectural design since ConvNet is originally designed to extract spatial hierarchies of features (Yamashita et al. 2018). In addition, event cameras bear severe noise (Wang et al. 2020). Unfired events can be easily introduced especially in some complex textured and high-speed scenarios, *e.g.* driving on the highway, due to bandwidth limitations (Hu, Liu, and Delbruck 2021). Thus, the problem of event signal enhancement has a strong deviation from its image-based counterpart and requires deliberate modifications when applying frame-based models.

In our work, we address the signal enhancement by proposing a novel input representation that retrieves the missing events in high-speed period. Our input representation accumulates events based on the temporal distribution of the event stream. Event signal is enhanced when the distribution is more concentrated since high-speed movement will lead to more events in a short moment. In addition, we introduce a spatio-temporal recurrent encoding-decoding neural network architecture (STE-FlowNet), an effective Convolutional Gated Recurrent (ConvGRU) (Ballas et al. 2016) based model that can extract spatio-temporal features effec-

[*] Corresponding author.

tively. More specifically, the feature encoder has a novel design, which is utilizing ConvGRU to extract feature maps from a series of event images. Note that event images are fed into STE-FlowNet separately by the temporal order, not as one entity. Also, every timestep the recurrent encoder gets input, the decoder would estimate optical flow between the first event image to the current event image which can be used for next estimation. Event stream has recorded the detailed motion dynamic and provided more intermediate information, intuitively it is more accurate to estimate optical flow based on the estimated optical flow information of previous timestep.

More importantly, recurrent architecture can remove the restriction in module design and provide more convenience in the training procedure. In more detail, as a core in most frame-based algorithms, the correlation layer (Ilg et al. 2017) has been shown to provide important cues for optical flow estimation. But it has been missed in all the previous event-based work since it cannot extract features from one entity input. Unlike previous work (Zhu et al. 2019; Ye et al. 2018; Zhu et al. 2018b), STE-FlowNet processes data frame by frame, allowing the correlation layer to be incorporated to extract extra features. Although event image has an amount of motion blur, it preserves abundant spatial information which is compatible with the correlation layer. Furthermore, inspired by traditional optimization-based approaches, we iteratively estimate the residual flows to refine final results. Besides, multiple intermediate supervised signals are possible to improve the performance of the network since recurrent architecture outputs multiple optical flows corresponding to the different time windows.

We evaluate our method on the Multi Vehicle Stereo Event Camera dataset (MVSEC) (Zhu et al. 2018a) and demonstrate its superior generalization ability in different scenarios. Results show that STE-FlowNet outperforms all the existing state-of-the-art methods (Zhu et al. 2019; Lee et al. 2020). Especially for $dt = 4$ case, we obtain a significant accuracy boost of 23.0 % on average over the baselines. In addition, we validate various design choices of STE-FlowNet through extensive ablation studies.

Related work

Event-based Optical Flow

More recently, deep learning has been applied to event-based optical flow thanks to the introduction of some large-scale event-based benchmarks. Many early works (Moeys et al. 2016; Ghosh et al. 2014) utilize simple ConvNet to estimate optical flow only based on small datasets. EV-FlowNet, presented by Zhu *et al.* (Zhu et al. 2018b), can be regarded as the first deep learning work training on large datasets with an encoder-decoder architecture. The event steam was summarized to compact event image preserving the spatial relationships between events as well as most recent temporal information. As an updated version of EV-FlowNet, an unsupervised framework has been proposed by Zhu *et al.* (Zhu et al. 2019). In more detail, its loss function is designed to measure the motion blur in the event image. The more accurate the optical flow is, the less motion blur the event images possess. ECN (Ye et al. 2018) follows the encoder-decoder architecture. However, it uses an evenly-cascaded structure, instead of standard ConvNet, to facilitate training by providing easily-trainable shortcuts. In summary, it is difficult for ConvNet to find out the temporal correlation between event images from one entity, which is the limitation all the works mentioned above cannot ignore.

SNN, as biologically inspired computational models, can deal with asynchronous computations naturally and exploit spatio-temporal features from events directly. Many researches (Paredes-Vallés, Scheper, and de Croon 2019; Richter, Röhrbein, and Conradt 2014; Orchard et al. 2013; Lee et al. 2020; Giulioni et al. 2016; Haessig et al. 2018) consider it as a perfect fit for event-based vision task. Although they are energy-efficient and hardware-friendly, there still exists a gap in performance between SNN and Analog Neural Network (ANN) since it is hard to retain gradients in deeper layers. Hybrid architecture, Spike-FlowNet, proposed by Lee *et al.* (Lee et al. 2020) seems to be another promising candidate for event-based optical flow. It utilizes SNN as an encoder to extract spatio-temporal features and has ANN as a decoder to enable deeper architecture. Note that this work is a kind of progress for addressing the importance of temporal features compared with ANN methods. However, SNN still limits the capability of the whole architecture as evidenced by that it does not outperform standard ANN methods.

Frame-based Optical Flow

Frame-based optical flow estimation is a classical task in the computer vision area through the years and has been solved well. FlowNet (Dosovitskiy et al. 2015) is the first end-to-end neural network for optical flow estimation, and Fischer *et al.* (Dosovitskiy et al. 2015) propose a large dataset FlyingChairs to train the network via supervised learning. PWC-Net (Sun et al. 2018) and Liteflownet (Hui, Tang, and Loy 2018) introduce the pyramid and cost volume to neural networks for optical flow, warping the features in different levels of the pyramid and learning the flow fields coarse-to-fine. IRR-Net (Hur and Roth 2019) takes the output from a previous iteration as an input for the next iteration using a weights-shared backbone network to iteratively refine the residual of the optical flow, which demonstrates an iterative method can increase the motion analysis performance for an optical flow estimation network. RAFT (Teed and Deng 2020) constructs decoder in the network using ConvGRU, iteratively decoding the correlation and context information in a fixed resolution, showing the promising capability of ConvGRU to extract a spatio-temporal relationship.

Method

Given a series of events from t_{start} to t_{end}, we estimate a dense displacement field $\mathbf{f} = \big(f^1, f^2\big)$, mapping each pixel $\mathbf{x} = (u, v)$ at t_{start} to its corresponding coordinates $\mathbf{x}' = (u', v') = \big(u + f^1(\mathbf{x}), v + f^2(\mathbf{x})\big)$ at t_{end}.

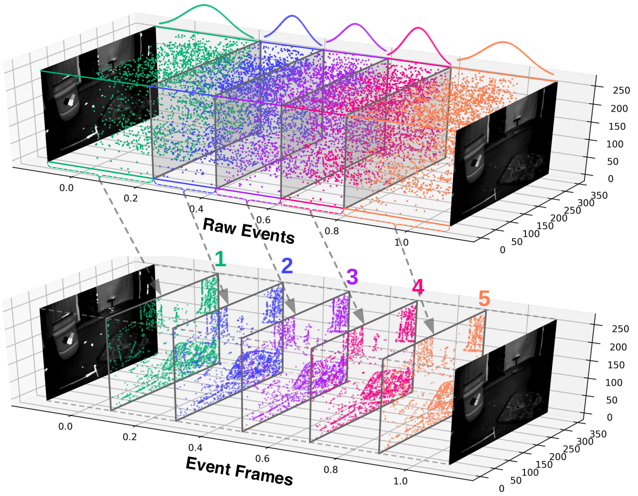

Figure 1: Input events representation. (Top) Raw event steam between two consecutive grayscale images from an event camera. (Bottom) Raw event stream are evenly divided to N splits by number. Each split is converted to a two-channel image, serving as inputs to the network.

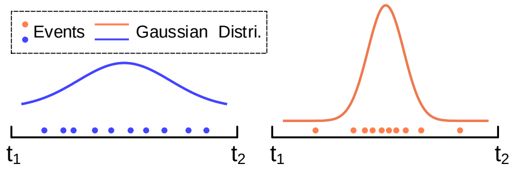

Figure 2: Gaussian model is utilized for fitting the events temporal distribution. Summarizing the number of events at each pixel, as well as the last timestamp, cannot capture the temporal distribution of events.

Input Data Representation

Event cameras output a stream of events whenever the log intensity changes exceed a set threshold θ. Each event encodes the pixel location of brightness change, timestamp of the event and the polarity (increase or decrease of a brightness), which can be summarized as $e = \{\mathbf{x}, t, p\}$.

However, the output stream might lose some events due to the limitation of pixel bandwidth in some complex textured and high-speed scenarios (Hu, Liu, and Delbruck 2021; Lichtsteiner, Posch, and Delbruck 2008). It is challenging to estimate optical flow in high-speed scenarios, thus we propose a novel input representation for signal enhancement. In more detail, all the events are first evenly divided to N splits by number. As illustrated in Figure 2, the event image by summing the number of events cannot reflect movement details in that period. If most of the events are concentrated in a very short moment, it is more likely that the event camera has experienced a high-speed scene at that moment compared with the smoother distribution. We, hence, incorporated this prior into input representation by accumulating events weighted by the temporal distribution of the events.

Given a set of M input events $(x_i, y_i, t_i, p_i)_{i\in[1:M]}$ in one split, Gaussian model is utilized for fitting the events temporal distribution and events are weighted accordingly. We generate the event image V, see Figure 1, as follows:

$$V^{\pm}(x,y) = \sum_{i=1}^{M} \lceil p_i^{\pm}(x,y)\lambda k_g(t_i) \rceil \tag{1}$$

$$k_g(a) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(a-\mu)^2}{2\sigma^2}\right) \tag{2}$$

Where $k_g(a)$ is a sampling kernel from Gaussian model, μ is the mean and σ is the variance. Both are estimated from all timestamps of M events. λ is normalizing factor and defined as $\frac{M}{\sum_{n=1}^{M} k_g(t_n)}$. The more concentrated the temporal distribution, the greater the weights of events. As a result, the event image is enhanced for the possible high-speed moment. Note that all weighted value are rounded up so that events will not be compressed.

Overall Architecture

The overall architecture of our work, STE-FlowNet, is a variant the encoder-decoder network (Ronneberger, Fischer, and Brox 2015) as illustrated in Figure 3. Each event image $\left(\mathbb{R}^{H\times W\times 2}\right)$ is passed into four encoder modules which doubles output channels and downsamples to 1/2 resolution each time. The output feature maps $\left(\mathbb{R}^{H/8\times W/8\times D}\right)$ from encoder modules then go through two residual blocks. Besides, there is a skip connection that links each encoder to the corresponding decoder module. While passing the decoder module, the input gets upsampled to 2x resolution by transposed convolution. It restores to the original image size after four decoder modules. In addition, STE-FlowNet outputs an intermediate flow at each resolution which is also part of the input for the next decoder. Note that the loss is applied to all the intermediate flows during the training procedure. More details about the parameter settings of architecture can be found in the supplemental material.

Feature Extraction

The key component of feature extraction is ConvGRU. Although ConvGRU plays an important role in some frame-based works (Ren et al. 2019; Teed and Deng 2020), their ConvGRU modules only focus on processing high-level feature maps from ConvNet top-layers. Note that ConvNet discards local information in their top layers and the temporal variation between images tends to vanish. Therefore, it can hardly capture fine motion information. To address this issue, encoder modules utilize ConvGRU to extract spatio-temporal features from event images at different resolutions to preserve local motion patterns.

As illustrated in Figure 3, the input event image goes through two different pyramid-like architectures at the same time. One way is passing ConvNet to generate spatial feature maps of the current input event image at different resolutions. Another way is through spatio-temporal feature encoders at different resolutions and each encoder module consists of one ConvGRU and one strided convolutional layer. The input of ConvGRU at timestep t includes three parts,

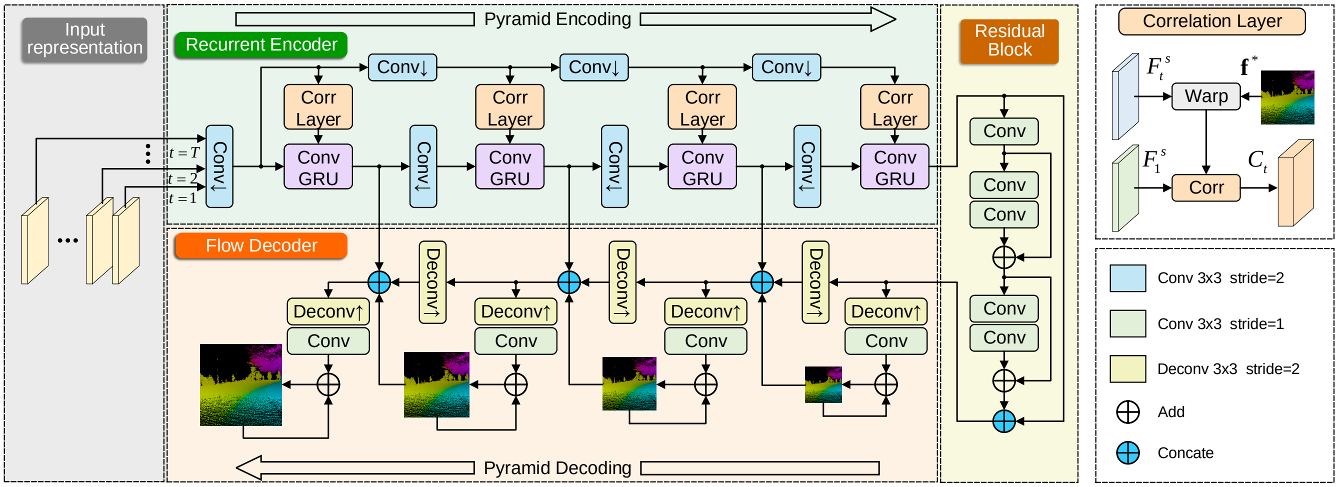

Figure 3: Network architecture of the STE-FlowNet. The event input is downsampled through two different paths. One is to get spatial feature maps of four different resolutions for correlation usage later, the other one is through four ConvGRU based encoder modules with skip connections to the corresponding decoders. After being passed through two residual blocks, the activations are then passed through four decoder modules. In addition, each set of decoder activations is passed through another depthwise convolution layer to generate a flow prediction at its resolution. A loss is applied to this flow prediction, and the prediction is also concatenated to the decoder activations.

the feature maps F_t^{st} from the lower-level spatio-temporal feature encoder, the hidden-state of ConvGRU h_{t-1} from previous timestep, and the output of the correlation layer C_t at the same resolution. The ConvGRU is defined by the following equations:

$$z_t = \sigma(\mathrm{Conv}_{3\times3}([h_{t-1}, C_t, F_t^{st}], W_z)) \tag{3}$$

$$r_t = \sigma(\mathrm{Conv}_{3\times3}([h_{t-1}, C_t, F_t^{st}], W_r)) \tag{4}$$

$$\tilde{h}_t = \tanh(\mathrm{Conv}_{3\times3}[r_t \odot h_{t-1}, C_t, F_t^{st}], W_h) \tag{5}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{6}$$

Note that the correlation layer measures the correspondences between the spatial feature maps (output from pyramid-like ConvNet) of the current event image F_t^s and of the first event image F_1^s given a predicted flow $\mathbf{f}^*$ from previous iteration. The correlation layer is defined by the following equation:

$$C_t\left(\mathbf{x}\right) = \mathrm{Corr}\left(F_t^s\left(\mathbf{x} + \mathbf{f}^*\right), F_1^s\left(\mathbf{x}\right)\right) \tag{7}$$

The recurrent architecture is naturally set since we process event images frame by frame and ConvGRU is able to extract information from previous timestep. The reason that we adopt recurrent architecture is that we assume that it is better to estimate optical flow timestep by timestep. It seems more accurate to estimate flow of $t_0 - t_n$ based on the estimated flow information of $t_0 - t_{n-1}$ (hidden layer of ConvGRU from previous timestep).

Iterative Updates

As illustrated in Figure 4, STE-FlowNet adopts the iterative residual refine scheme (IRR) (Hur and Roth 2019) to refine the output, *i.e.* estimated residual flow, iteratively. The final result is the sum of residual flows from all the iteration steps. Besides, we reuse the same network block with shared weights iteratively. It is noted that the concept of iterative refinement has been proved to be an effective way to improve the final result in many frame-based works (Sun et al. 2018; Hui, Tang, and Loy 2018; Teed and Deng 2020; Hur and Roth 2019).

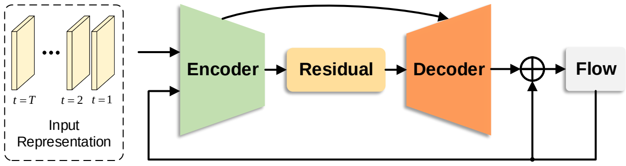

Figure 4: Note that next iteration begins after all the event images are passed through the network frame by frame.

Every iteration after the last input event image is fed into STE-FlowNet, we can get a sequence of output $\{\mathbf{f}_{1,1}^1, \cdots, \mathbf{f}_{T,K}^L\}$, where $\mathbf{f}_{t,k}^l$ represents the predicted flow between the first event image and t^{th} event image at resolution level l after k iterations. The new iteration then begins and the event images, as well as the predicted flows, are passed into STE-FlowNet again to refine the flow. More precisely, the predicted flows from previous iteration will be sent to two places for two different purposes. One is to regard predicted flows as part of the input for STE-FlowNet, which is to warp the current feature maps for correlation use. Another is to perform as a skip connection that adds the predicted flow with the current output of STE-FlowNet to predict the flow at the current iteration. To illustrate, at k iteration, STE-FlowNet outputs the residual flow $\Delta\mathbf{f}_k$ and the current estimation is $\mathbf{f}_k = \mathbf{f}_{k-1} + \Delta\mathbf{f}_k$ with initial starting

$dt = 1$ frame	indoor flying1		indoor flying2		indoor flying3		outdoor day1	
	AEE	% Outlier	AEE	% Outlier	AEE	% Outlier	AEE	% Outlier
EV-FlowNet (Zhu et al. 2018b)	1.03	2.2	1.72	15.1	1.53	11.9	0.49	0.2
Zhu *et al.* (Zhu et al. 2019)	0.58	**0.0**	1.02	4.0	0.87	3.0	**0.32**	**0.0**
Spike-FlowNet (Lee et al. 2020)	0.84	**0.0**	1.28	7.0	1.11	4.6	0.47	**0.0**
Counts+TimeSurface (Zhu et al. 2018b)	0.60	0.1	0.81	2.0	0.73	1.4	0.45	**0.0**
Counts (Lee et al. 2020)	0.62	0.8	0.96	5.5	0.87	3.8	0.42	**0.0**
Ours	**0.57**	0.1	**0.79**	**1.6**	**0.72**	**1.3**	0.42	**0.0**

$dt = 4$ frame	indoor flying1		indoor flying2		indoor flying3		outdoor day1	
	AEE	% Outlier	AEE	% Outlier	AEE	% Outlier	AEE	% Outlier
EV-FlowNet (Zhu et al. 2018b)	2.25	24.7	4.05	45.3	3.45	39.7	1.23	7.3
Zhu *et al.* (Zhu et al. 2019)	2.18	24.2	3.85	46.8	3.18	47.8	1.30	9.7
Spike-FlowNet (Lee et al. 2020)	2.24	23.4	3.83	42.1	3.18	34.8	1.09	5.6
Counts+TimeSurface (Zhu et al. 2018b)	2.02	20.1	2.85	32.1	2.49	26.9	1.14	5.5
Counts (Lee et al. 2020)	2.41	28.2	3.30	39.5	2.86	33.6	1.32	6.1
Ours	**1.77**	**14.7**	**2.52**	**26.1**	**2.23**	**22.1**	**0.99**	**3.9**

Table 1: Quantitative evaluation of our optical flow network compared to EV-FlowNet (Zhu et al. 2018b), Zhu *et al.* work (Zhu et al. 2019), Spike-FlowNet (Lee et al. 2020) and other input representations. For each sequence, Average Endpoint Error (AEE) and % Outlier are computed. $dt = 1$ is computed with a time window between two successive grayscale frames, $dt = 4$ is between four grayscale frames.

point $\mathbf{f}_0 = \mathbf{0}$.

Self-Supervised Learning

Many benchmarks provide synchronized events and grayscale images using DAVIS camera (Brandli et al. 2014) so that we can adopt self-supervised learning utilizing grayscale images to guide neural network training. In more detail, the events stream occurring just between two consecutive grayscale images (I_t, I_{t+dt}) is transformed to multiple event images that are passed into the network subsequently. In the meantime, we can apply the predicted per-pixel flow $\mathbf{f} = (f^1, f^2)$ to corresponding grayscale images and generate a self-supervised loss.

The total loss function consists of two parts (Zhu et al. 2018b), photometric reconstruction loss $\mathcal{L}_{\text{photo}}$ and smoothness loss $\mathcal{L}_{\text{smooth}}$, which can be written as,

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{photo}} + \lambda \mathcal{L}_{\text{smooth}} \tag{8}$$

where λ is the weight factor.

The predicted flow is used to warp the second grayscale image using bilinear sampling. The more accurate the predicted flow, the less discrepancy between the first grayscale image and the warped second grayscale image. The photometric loss is computed as follows:

$$\mathcal{L}_{\text{photo}}\left(\mathbf{f}; I_t, I_{t+dt}\right) = \sum_{\mathbf{x}} \rho\left(I_t\left(\mathbf{x}\right) - I_{t+dt}\left(\mathbf{x} + \mathbf{f}\left(\mathbf{x}\right)\right)\right) \tag{9}$$

where ρ is the Charbonnier loss $\rho(x) = (x^2 + \eta^2)^r$, and we set $r = 0.45$ and $\eta = 1\text{e}{-3}$.

Furthermore, we use smoothness loss to regularize the predicted flow. It minimizes the flow difference between neighboring pixels, thus it can enhance the spatial consistency of neighboring flows and mitigate some other issues, such as the aperture problem. It can be written as:

$$\mathcal{L}_{\text{smooth}}\left(f^1, f^2\right) = \sum_{\mathbf{x}} |\nabla f^1(\mathbf{x})| + |\nabla f^2(\mathbf{x})| \tag{10}$$

where ∇ is the difference operator.

Experiments

Datasets and Training Details

The MVSEC dataset (Zhu et al. 2018a) is used for training and evaluating our model since it is designed for the development of visual perception algorithms for event-based cameras. Note that the data is collected in two different scenarios, *e.g.* indoor (recorded in rooms) and outdoor (recorded while driving on public roads). Apart from event data sequences and corresponding ground truth optical flow, it also provides the images from a standard stereo frame-based camera pair for comparison which we can use to generate the self-supervised loss. To provide fair comparisons with prior works (Zhu et al. 2019; Lee et al. 2020; Zhu et al. 2018b), we only use the outdoor day2 sequence to train the models. Indoor flying1, indoor flying2, and indoor flying3, and outdoor day1 sequences are for evaluation only.

We have two models for two different time window lengths. In more detail, one is for 1 grayscale image frame apart ($dt = 1$), and the other is for 4 grayscale images frame apart ($dt = 4$). When it comes to $dt = 1$ case, the model is trained for 40 epochs. The number of event images N_{frame} that summarizes input event sequence is set to 5, and weight factor λ for the smoothness loss is set to 10. The initial learning rate is $4\text{e}{-4}$ and scaled by 0.7 after 5, 10, and 20 epochs. As for $dt = 4$ case, the model is trained for 15 epochs. N_{frame} is set to 20 and λ is set to 10. In addition, the initial learning rate is $4\text{e}{-4}$ with the same scaled strategy. Note that we use Adam optimizer (Kingma and Ba 2014) with mini-batch size of 16 and the number of iteration for IRR N_{irr} is set to 3 in both cases.

As mentioned before, STE-FlowNet would output intermediate flows at different resolutions and the loss computed

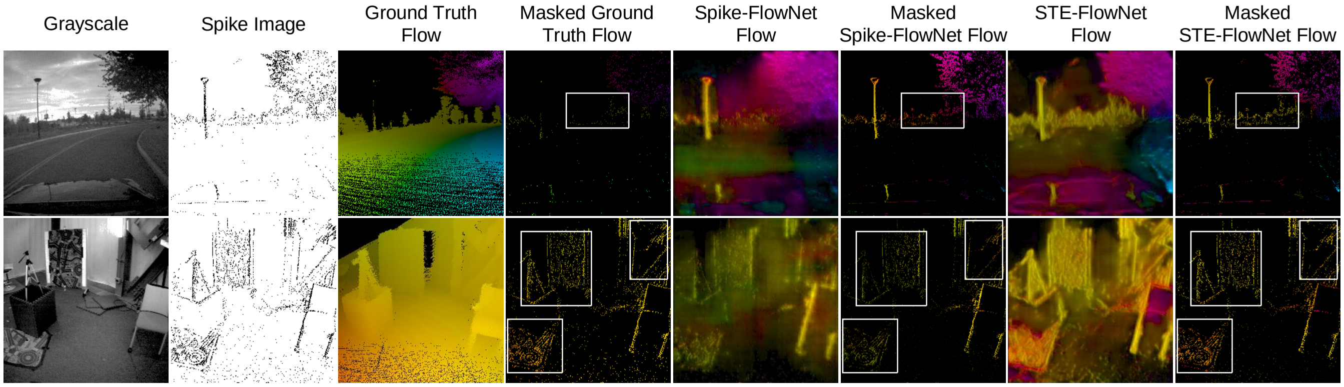

Figure 5: Qualitative results from evaluation for $dt = 1$ case. Examples were collected from (Top) outdoor day1 and (Bottom) indoor flying2. The white boxes are used to highlight the superiority of our method.

by each intermediate flow is weighted equally in the final loss. In addition, in the $dt = 4$ cases, there are some more grayscale images available between $\mathrm{t}_{\mathrm{start}}$ and $\mathrm{t}_{\mathrm{end}}$, which means more intermediate self-supervised loss can be generated since recurrent architecture allows STE-FlowNet to predict flows for different time window lengths. The intermediate loss can be used to address vanishing gradients issues and improve the performance of the network (Newell, Yang, and Deng 2016). The multiple intermediate losses (MIL) from different time window lengths then combine to form the final loss with equal weights.

Quantitative and Qualitative Results

Average End-point Error (AEE) is used as evaluation metric. It measures the mean distance between the predicted flow $\mathbf{f}_{\mathrm{pred}}$ and the ground truth $\mathbf{f}_{\mathrm{gt}}$ provided by the MVSEC dataset. Note that only active pixels are reported and they are defined as places where both the ground truth data and the events are present (at least one event can be observed). Besides, we also report the percentage of points with AEE greater than 3 pixels and 5% of the magnitude of the flow vector, denoted as % Outlier. During the evaluation, we estimate the optical flow on the center cropped 256×256 of input event images. As for evaluation sequences, we use all the data from the indoor flying sequences. However, only 800 grayscale images from the outdoor day1 sequence are chosen, following the settings in (Zhu et al. 2019).

Comparison for Networks We compare STE-FlowNet with three existing methods on event-based optical flow estimation: EV-FlowNet (Zhu et al. 2018b), Spike-FlowNet (Lee et al. 2020) , and the unsupervised framework of Zhu *et al.* (Zhu et al. 2019). Table 1 provides the evaluation results in comparison with all the baselines mentioned above.

For $dt = 1$ case, we can find out that STE-FlowNet has achieved better results compared with all other baselines on three indoor sequences. Specifically, STE-FlowNet achieves AEE of 0.57, 0.79 and 0.72 on the indoor flying1, indoor flying2 and indoor flying3 sequences respectively, 2%, 23% and 17% error reduction from the best prior deep network. Note that STE-FlowNet achieves the lowest % Outlier in most of evaluation sequences. Although our model doesn't get the best result on outdoor day1, we have demonstrated the better generalization ability of the model on indoor sequences than baselines.

As for $dt = 4$ case, our model has made a remarkable achievement. We outperform all existing approaches in all the test sequences by a large margin. In more detail, we get AEE of 1.77, 2.52, and 2.23 on the indoor flying1, indoor flying2, and indoor flying3 sequences respectively. The error reduction from the best prior work on indoor sequences is 18%, 34%, and 30%. Even on the outdoor day1 sequence, we still have a satisfying result. Our model achieves AEE of 0.99 and gets 9% error reduction. Furthermore, STE-FlowNet achieves the lowest % Outlier in all the evaluation sequences. The results show that recurrent architecture can better handle more input data since it processes data frame by frame, especially in the scenario where the time window length is long and the number of input event images is large. On the contrary, standard ConvNet architecture might be overwhelmed by massive data at one time.

The grayscale, spike event, ground truth flow, and the corresponding predicted flow images are visualized in Figure 5 where the images are taken from outdoor day1, indoor flying2, and indoor flying3 in $dt = 1$. Since the event data is quite sparse, STE-FlowNet doesn't predict flows in most of the regions. In summary, the results show that STE-FlowNet can accurately estimate the optical flow in both the indoor and outdoor scenes. In addition, STE-FlowNet can estimate flows in more edge regions where Spike-FlowNet has no output. Also, in some regions with rich texture, the directions of predicted flows (viewed in color) of STE-FlowNet are closer to the ground truth than Spike-FlowNet.

Comparison for Input Representations We compare our input representation with that of other optical flow algorithms. In more detail, Spike-FlowNet (Lee et al. 2020) generates an event image by summing the number of events at each pixel, denoted as Counts. Ev-FlowNet (Zhu et al. 2018b) proposes input representation that summarizes the number of events at each pixel as well as the last timestamp, denoted as Counts+TimeSurface. For fair comparison, the

		MIL	Corr	IRR	CGRU	indoor flying1		indoor flying2		indoor flying3		outdoor day1	
						AEE	% Outlier	AEE	% Outlier	AEE	% Outlier	AEE	% Outlier
dt=1	STE-G	–	✗	✗	✗	1.37	9.4	2.16	22.7	2.03	20.8	0.59	0.1
	STE-I	–	✗	✗	✓	0.63	0.2	0.87	2.8	0.79	2.4	0.42	**0.0**
	STE-C	–	✗	✓	✓	0.59	0.2	0.87	2.8	0.79	2.4	**0.41**	**0.0**
	STE	–	✓	✓	✓	**0.57**	**0.1**	**0.79**	**1.6**	**0.72**	**1.3**	0.42	**0.0**
dt=4	STE-G	✓	✗	✗	✗	4.77	65.6	7.36	75.8	6.55	74.8	3.10	44.5
	STE-I	✓	✗	✗	✓	1.91	17.7	2.82	31.2	2.41	24.5	1.10	4.6
	STE-C	✓	✗	✓	✓	2.11	21.9	2.73	33.7	2.46	27.4	1.06	4.1
	STE-L	✗	✓	✓	✓	1.96	18.6	2.70	30.6	2.40	25.7	1.01	4.5
	STE	✓	✓	✓	✓	**1.77**	**14.7**	**2.52**	**26.1**	**2.23**	**22.1**	**0.99**	**3.9**

Table 2: Ablation studies of our design choices for $dt = 1$ and $dt = 4$ case. STE-FlowNet without IRR, and STE-FlowNet without ConvGRU and IRR. The ablation baselines are denoted as STE-C, STE-I, and STE-G respectively. For $dt = 4$, STE-I means removing multiple intermediate losses.

backbones of Counts and Counts+TimeSurface are the same as that of STE-FlowNet.

From Table 1, our representation has demonstrated the superiority over other methods. Counting events (Lee et al. 2020) discards the rich temporal information in the events, and is susceptible to motion blur. Time Surface method used in (Zhu et al. 2018b) is only able to capture some temporal information around some specific moment. Different with these representations, our method accumulates the events based on the temporal distribution of the events stream. We aim to enhance the signal and highlight the period when the event camera encounters a high-speed scene.

Qualitative Results The grayscale, spike event, ground truth flow, and the corresponding predicted flow images are visualized in Figure 5 where the images are taken from outdoor day1, indoor flying2, and indoor flying3 in $dt = 1$. Since the event data is quite sparse, STE-FlowNet doesn't predict flows in most of the regions. In summary, the results show that STE-FlowNet can estimate flows in more edge regions where Spike-FlowNet has no output. Also, in some regions with rich texture, the directions of predicted flows (viewed in color) of STE-FlowNet are closer to the ground truth than Spike-FlowNet.

Moreover, Figure 6 shows the intermediate outputs from parts timestep of the network. We can find out the optical flow is gradually enhanced and the network indeed can get optical flow information from previous timesteps.

Ablation Studies

There are three components needing to be assessed, *i.e.* ConvGRU, correlation layer, and IRR scheme. Therefore, we have three baselines for ablation studies, namely, STE-FlowNet without correlation layer, STE-FlowNet without IRR, and STE-FlowNet without ConvGRU and IRR. The ablation baselines are denoted as STE-C, STE-I, and STE-G respectively. Note that the correlation layer is meaningless when the IRR scheme is absent since the flows used for correlation come from the previous iteration. Besides, we concatenate the predicted flow with the input event image directly, serving as input for the next iteration in STE-C. As for $dt = 4$ case, we have additional ablation for removing multiple intermediate losses (MIL) for different time windows, denoted as STE-L.

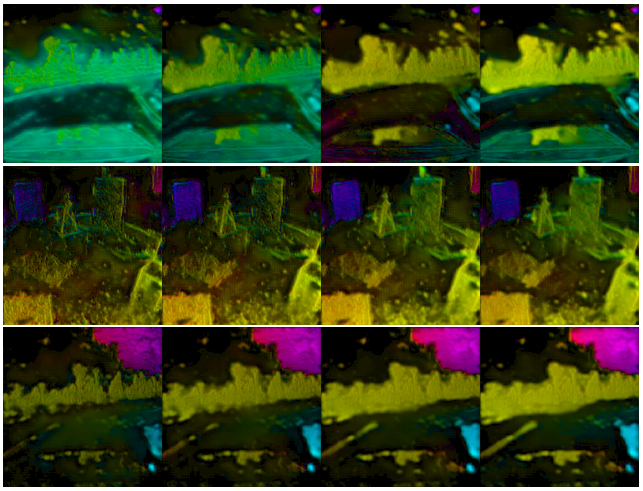

Figure 6: The intermediate outputs from parts timesteps. Timestep increases from left to right.

Table 2 shows the results of ablation studies in terms of both AEE and % Outliers for $dt = 1$ case. STE-G performs the worst compared with others in all evaluation sequences. This is because that it only depends on one input event image to predict flows without the ConvGRU to provide extra spatio-temporal information from other event images. STE-FlowNet and STE-C perform better than STE-I in almost all evaluation sequences, which demonstrates the effectiveness of the IRR scheme. In addition, STE-FlowNet outperforms STE-C. It shows correlation layer is able to provide more valuable features than directly sending flows. Also, it proves that the correlation layer can be applied in dealing with the event data. Note that STE-I can still achieve promising results compared with some prior works. It again demonstrates the superiority of our architecture. The same conclusions can be obtained in $dt = 4$ case. Moreover, we find out that STE-FlowNet performs better than STE-L. Therefore, we believe that the multiple intermediate losses from different time windows indeed help to improve the performance of the model. Note that the prior works (Zhu et al. 2019, 2018b; Ye et al. 2018; Lee et al. 2020) are unable to utilize additional grayscale images.

Conclusion

We propose a ConvGRU-based encoding-decoding network with a novel input representation to effectively extract the spatio-temporal information from event input. Moreover, the correlation layer is used to provide more valuable clues for the IRR scheme to further refine the predicted flow. Empirically, results show that STE-FlowNet outperforms all existing methods in a variety of indoor and outdoor scenes.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (62176003, 62088102, 61961130392), Beijing Major Science and Technology Project (Z191100010618003), PKU-Baidu Fund (Project 2019BD001).

References

Aufrère, R.; Gowdy, J.; Mertz, C.; Thorpe, C.; Wang, C.-C.; and Yata, T. 2003. Perception for collision avoidance and autonomous driving. *Mechatronics*, 13(10): 1149–1161.

Ballas, N.; Yao, L.; Pal, C.; and Courville, A. C. 2016. Delving Deeper into Convolutional Networks for Learning Video Representations. In *International Conference on Learning Representations (ICLR)*.

Brandli, C.; Berner, R.; Yang, M.; Liu, S.; and Delbruck, T. 2014. A 240 × 180 130 dB 3 µs Latency Global Shutter Spatiotemporal Vision Sensor. *IEEE Journal of Solid-State Circuits*, 49(10): 2333–2341.

Capito, L.; Ozguner, U.; and Redmill, K. 2020. Optical flow based visual potential field for autonomous driving. In *IEEE Intelligent Vehicles Symposium (IV)*, 885–891.

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 *IEEE International Conference on Computer Vision (ICCV)*, 2758–2766.

Efros, A. A.; Berg, A. C.; Mori, G.; and Malik, J. 2003. Recognizing action at a distance. In *IEEE International Conference on Computer Vision (ICCV)*, volume 3, 726–726.

Ghosh, R.; Mishra, A.; Orchard, G.; and Thakor, N. V. 2014. Real-time object recognition and orientation estimation using an event-based camera and CNN. In *2014 IEEE Biomedical Circuits and Systems Conference (BioCAS) Proceedings*, 544–547.

Giulioni, M.; Lagorce, X.; Galluppi, F.; and Benosman, R. B. 2016. Event-based computation of motion flow on a neuromorphic analog neural platform. *Frontiers in Neuroscience*, 10: 35.

Haessig, G.; Cassidy, A.; Alvarez, R.; Benosman, R.; and Orchard, G. 2018. Spiking optical flow for event-based sensors using ibm's truenorth neurosynaptic system. *IEEE Transactions on Biomedical Circuits and Systems*, 12(4): 860–870.

Hu, Y.; Liu, S.-C.; and Delbruck, T. 2021. V2e: From video frames to realistic DVS events. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1312–1321.

Hui, T.-W.; Tang, X.; and Loy, C. C. 2018. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 8981–8989.

Hur, J.; and Roth, S. 2019. Iterative residual refinement for joint optical flow and occlusion estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5754–5763.

Ilg, E.; Mayer, N.; Saikia, T.; Keuper, M.; Dosovitskiy, A.; and Brox, T. 2017. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2462–2470.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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 *European Conference on Computer Vision (ECCV)*, 366–382. Springer.

Lichtsteiner, P.; Posch, C.; and Delbruck, T. 2008. A 128×128 120dB 15μs latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-state Circuits*, 43(2): 566–576.

Liu, L.; Zhang, J.; He, R.; Liu, Y.; Wang, Y.; Tai, Y.; Luo, D.; Wang, C.; Li, J.; and Huang, F. 2020. Learning by analogy: Reliable supervision from transformations for unsupervised optical flow estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6489–6498.

Moeys, D. P.; Corradi, F.; Kerr, E.; Vance, P.; Das, G.; Neil, D.; Kerr, D.; and Delbruck, T. 2016. Steering a Predator Robot using a Mixed Frame/Event-Driven Convolutional Neural Network. arXiv:1606.09433.

Newell, A.; Yang, K.; and Deng, J. 2016. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision (ECCV)*, 483–499. Springer.

Orchard, G.; Benosman, R.; Etienne-Cummings, R.; and Thakor, N. V. 2013. A spiking neural network architecture for visual motion estimation. In *IEEE Biomedical Circuits and Systems Conference (BioCAS)*, 298–301.

Paredes-Vallés, F.; Scheper, K. Y.; and de Croon, G. C. 2019. Unsupervised learning of a hierarchical spiking neural network for optical flow estimation: From events to global motion perception. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(8): 2051–2064.

Ren, Z.; Gallo, O.; Sun, D.; Yang, M.-H.; Sudderth, E. B.; and Kautz, J. 2019. A fusion approach for multi-frame optical flow estimation. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2077–2086.

Richter, C.; Röhrbein, F.; and Conradt, J. 2014. Bio-inspired optic flow detection using neuromorphic hardware. In *BCCN conference, Göttingen*.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 234–241. Springer.

Sun, D.; Yang, X.; Liu, M.-Y.; and Kautz, J. 2018. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 8934–8943.

Teed, Z.; and Deng, J. 2020. Raft: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision (ECCV)*, 402–419. Springer.

Wang, H.; Fan, R.; and Liu, M. 2020. Cot-amflow: Adaptive modulation network with co-teaching strategy for unsupervised optical flow estimation. *arXiv preprint arXiv:2011.02156*.

Wang, Z. W.; Duan, P.; Cossairt, O.; Katsaggelos, A.; Huang, T.; and Shi, B. 2020. Joint filtering of intensity images and neuromorphic events for high-resolution noise-robust imaging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1609–1619.

Yamashita, R.; Nishio, M.; Do, R. K. G.; and Togashi, K. 2018. Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*, 9(4): 611–629.

Ye, C.; Mitrokhin, A.; Fermüller, C.; Yorke, J. A.; and Aloimonos, Y. 2018. Unsupervised learning of dense optical flow, depth and egomotion from sparse event data. *arXiv preprint arXiv:1809.08625*.

Zhu, A. Z.; Thakur, D.; Özaslan, T.; Pfrommer, B.; Kumar, V.; and Daniilidis, K. 2018a. The multivehicle stereo event camera dataset: An event camera dataset for 3D perception. *IEEE Robotics and Automation Letters*, 3(3): 2032–2039.

Zhu, A. Z.; Yuan, L.; Chaney, K.; and Daniilidis, K. 2018b. EV-FlowNet: Self-supervised optical flow estimation for event-based cameras. *arXiv preprint arXiv:1802.06898*.

Zhu, A. Z.; Yuan, L.; Chaney, K.; and Daniilidis, K. 2019. Unsupervised event-based learning of optical flow, depth, and egomotion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 989–997.