Event Transformer FlowNet for optical flow estimation

Yi Tian
ytian@iri.upc.edu
Juan Andrade-Cetto
cetto@iri.upc.edu

Institut de Robòtica i Informàtica
Industrial, CSIC-UPC
Barcelona, Spain

Abstract

Event cameras are bioinspired sensors that produce asynchronous and sparse streams of events at image locations where intensity change is detected. They can detect fast motion with low latency, high dynamic range, and low power consumption. Over the past decade, efforts have been conducted in developing solutions with event cameras for robotics applications. In this work, we address their use for fast and robust computation of optical flow. We present ET-FlowNet, a hybrid RNN-ViT architecture for optical flow estimation. Visual transformers (ViTs) are ideal candidates for the learning of global context in visual tasks, and we argue that rigid body motion is a prime case for the use of ViTs since long-range dependencies in the image hold during rigid body motion. We perform end-to-end training with self-supervised learning method. Our results show comparable and in some cases exceeding performance with state-of-the-art coarse-to-fine event-based optical flow estimation.

1 Introduction

Event cameras mimic biological retinas in that they generate asynchronous and sparse event signals, providing some advantages (and drawbacks) over conventional frame-based cameras. They can detect fast motion with low latency, have low power consumption, and given their high dynamic range, are robust to changes in illumination conditions, traits that make them ideal candidates for robotic applications. However, in still conditions, they are blind to a stationary scene; and efficient treatment of their unconventional spatio-temporal data still challenges the computer vision community.

Events, reported asynchronously, occur at pixel locations that experience intensity change. Hence, subject to motion, event streams naturally detect edges, making event camera a good sensor for the estimation of optical flow. Some works already exist for the computation of event-based optical flow with artificial neural networks (ANNs) [32, 35, 37]. ANNs cannot deal well with the temporal information of event data and usually require preprocessing chunks of the event stream into time images or voxel representations. On the contrary, spiking neural networks (SNNs) have been suggested as an alternative to address the sparse and asynchronous nature of the event inputs [2, 22]. However, their performance is still uncomparable with that of ANNs as they suffer from training difficulties.

In recent years, transformers have achieved notable success in the solution of both NLP [28] and computer vision problems. In particular, the recent architecture Visual Transformer (ViT) [9, 19] has achieved state-of-the-art performance for various visual tasks, including object detection [5], depth estimation [11] and optical flow [13, 14, 26]. For rigid scenes, optical flow is determined by the camera motion and scene depth structure. Transformer models involve the dot product self-attention mechanism, which can capture such long-range dependencies of the scene; While CNN-only models relying on local convolutional kernels require larger receptive fields or deeper structure. Recent work shows that transformers are also more efficient in solving problems with sparse and spatially-distant patterns [31], which makes them a perfect candidate for event-based data. Given the advantage of transformers already demonstrated in frame-based optical flow tasks [13, 26], we would like to explore their use for event-based optical flow tasks. However, very limited work exists combining transformers with sparse event input data for regression problems [30, 34].

In this paper, we present ET-FlowNet, a hybrid RNN-ViT architecture for event-based optical flow estimation. We incorporate a convolutional gate recurrent unit (ConvGRU) [1] for temporal information extraction with a visual transformer block with token pyramid aggregation (TPA) [30], to learn the global context. We estimate the optical flow in a coarse-to-fine manner and yield comparable results with the state-of-the-art. Our main contributions are: We propose the first pipeline to our knowledge using transformers for event-based optical flow estimation. Our tests demonstrate that our architecture outperforms the CNN-only-based method and yields the best results in most of the MVSEC sequences among self-supervised learning approaches.

2 Related Work

2.1 Learning-based event optical flow estimation

Early research on the optical flow estimation from events focused on model-based approaches, such as gradient-based [4] or spatiotemporal plane fitting [3]. More recently, deep learning techniques have achieved state-of-the-art performance [8, 10, 32, 35, 37]. Of these, Zhu et al. [35] presented EV-FlowNet, which is the first ANN-based optical flow estimation framework using an encoder-decoder architecture. Learning is supervised by photometric loss from the grayscale images. They later improved their work to perform unsupervised learning of optical flow, ego-motion and depth, based on motion-compensation loss [37]. Most of the later work follows this U-Net architecture [8, 12, 16, 17, 32, 37]. Notably, Ye et al. [32] presented the first ANN framework to estimate dense flow, ego-motion, and depth from events only using unsupervised learning. They predicted the optical flow via depth and ego-motion based on the static scene assumption. More recently, Ding et al. [8] proposed STE-FlowNet, adding a correlation layer to the encoder and updating the flow using an Iterative Residual Refine scheme (IRR), also training the network with the aid of grayscale frames. In [21], the authors propose another lightweight neural network architecture, FireFlowNet, trained by a self-supervised method using contrast maximization proxy loss. All these previous works, except for [32], predict sparse flow and evaluate it using a mask due to the limited accuracy where no events are present. Inspired by the frame-based method RAFT [27], Gehrig et al. [10] used cost volumes instead of a U-Net architecture and predicted dense optical flow trained by supervised learning, yielding what we consider to be state-of-the-art performance today on dense optical flow estimation.

Whereas ANN-based methods need to aggregate event input data into images or voxel representations, losing on the way valuable temporal information, another line of research focuses on the use of asynchronous spiking neural networks (SNNs) [2, 6, 12, 16, 22], or hybrid methods [16, 17]. SNNs naturally match the asynchronous events data format, and they are either trained by unsupervised learning based on STDP [2, 22] –and suffer the common problem of spatio-temporal filter methods for generalization–, or with a surrogate gradient method for backpropagation based on similar self-supervised learning [6, 12]. Hybrid architectures such as Spike-FlowNet [16] and its variant [17] combined an SNN in the encoder with an ANN in the decoder. All the SNN above-mentioned methods demonstrate improvement in energy efficiency, but still underperform their ANN counterparts in terms of accuracy.

With regards to training, some works rely on grayscale images for supervision based on photometric consistency [8, 16, 35], while others use event data based on motion compensation [12, 21, 37]. Recently, Shiba et al. [24] extended the Contrast Maximization (CMax) framework showing state-of-the-art results among model-based methods and also showing applicability to learning-based methods.

2.2 Transformer for event-based vision

Transformer architectures have been introduced in a variety of vision tasks recently making breakthrough achievements. The recent trend shows increasing interest in applying transformer models also to event vision. Most of the existing work using transformer backbones focus on efficient computation with patch-based event representations for classification tasks [18, 23, 29]. Hybrid networks have also been proposed for more complicated tasks, such as single object tracking [34] or event-based video reconstruction [30]. These networks usually extract temporal information relying on RNNs [30] or even SNNs [34] as the backbone network, and a transformer module for global spatial information extraction. We argue that the long-range modeling capability of transformers is an ideal trait for the computation of optical flow tasks on rigid motion cases. To the best of our knowledge, ours is the first work attempting to combine the transformer model for event-based optical flow estimation.

3 Methods

3.1 Event input representations

We use event count images as input for training and testing. Specifically, events are partitioned into sets of event stream $\mathcal{E} = \{e_i\}_{i=1}^{N}$ with a fixed number of events N (1000 in our case). The events in each partition are accumulated in two different channels according to their polarity $p_i \in \{+,-\}$, resulting in an event count image $I \in \mathbb{R}^{2\times H\times W}$, where H and W represent the image height and width, see Fig. 1 frames (a) and (b). For training, however, we also create an image of average timestamps per pixel and per polarity by bilinear interpolation of the event counts at each pixel location (more on this in Sec. 3.3). Whilst there exist more complex input representations that better encode the high temporal resolution of event data, the accumulation representation while preserving polarity, is rather simple, fast to compute, and has proved sufficient to evaluate the convenience of using ViTs. Moreover,

(a) (b) (c)

Figure 1: Input data and optimization: (a) event input data with polarity in red and green and aggregated counts in black, (b) event count image with event counts represented by intensity, (c) deblurred image of warped events (IWE) after motion compensation.

it allows for a fair comparison with other works that do not include time or voxel data, such as those based on SNNs [12].

3.2 Architecture

Our system, ET-FlowNet, is a hybrid RNN-Transformer architecture for event-based optical flow estimation. The pipeline is shown in Fig. 2. The encoder-decoder architecture is well-adopted for event-based optical flow estimation [8, 21, 35, 37]. Optical flow is estimated in a multi-scale coarse-to-fine manner to deal with larger displacement and detect global motion features. However, these multi-scale feature maps usually lack interactions. We follow a similar encoder-decoder architecture at the backbone and incorporate a token pyramid aggregation (TPA) transformer block connecting the encoder and decoder [30]. Instead of connecting only to the single-scale feature pyramid, the TPA module connects to all the feature map outputs from the ConvGRU-based encoders. The transformer encoders model the internal spatial dependency of each feature map while the transformer decoders capture the interactions among all the feature maps via cross-attention. In such a way, we fully model the interactions across both space and scales. Similar structures shows boost performance for tasks such as video reconstruction [30], segmentation [31] and object detection [33]. Our hypothesis is that this capability of modeling global dependency would be also beneficial for the event-based optical flow estimation case. The details for the architecture of our model are described as follows:

ConvGRU-based encoder block: Instead of feeding additional event timestamp pictures to convolutional layers, we use recurrent convolutional layers to extract the temporal information. Specifically, the encoder consists of four convolutional layers followed by ConvGRU blocks. The output spatial dimension of each layer is half of the previous one, while the output channel dimension doubles. Each encoder outputs a feature map $f_l^{en} \in \mathbb{R}^{C_0 \times 2^l \times \frac{H}{2^l} \times \frac{W}{2^l}}$, where C_0 is the channel dimension for the first encoder. We set it to 64, the same as in [35].

TPA transformers block: We replace the two residual blocks of the standard encoder-decoder architecture [35] with a TPA module similar to [30] in order to extract the internal and intersected dependency from the multi-scale feature maps. To adapt the TPA module to our network architecture, we split the output of each encoder into the same number of patches $n = HW/P_0^2$, where P_0 is the patch size for the first encoder feature output $f_0^{en} \in \mathbb{R}^{C_0 \times H \times W}$. For all scales, the size of the corresponding patch embedding is $P_l = P_0/2^l$. The patches are flattened into sequences and mapped to dimension D via a trainable linear projection, which is set to be the channel number of the last encoder feature map. After adding the positioning

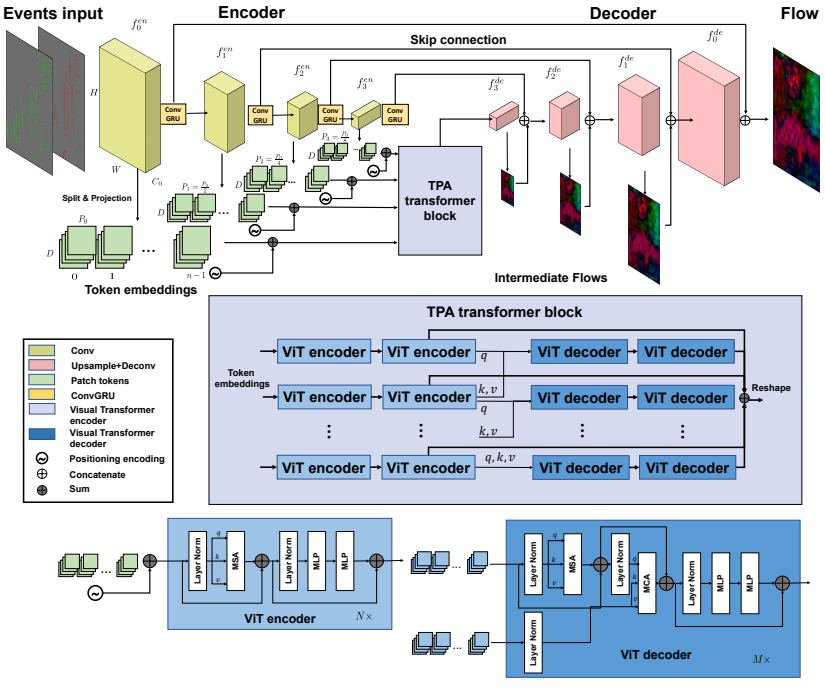


Figure 2: ET-FlowNet architecture.

embedding [28] of the same dimension, we feed the embedding sequences $T_{l,i=0,\ldots n} \in \mathbb{R}^D$ into the transformer encoder to model the internal dependency of each feature map. The output of the transformer encoder is fed as the query vector for the transformer decoder of the same scale, and as the key and value vectors for the decoder of the upper scale. In addition, there is a skip connection for the output of the transformer encoders and decoders at each scale. For the results in Table 1, we use 2 transformer encoders and 2 transformer decoders for all the scales. Finally, all the tokens from the transformer blocks are aggregated and reshaped to $f \in \mathbb{R}^{D \times \frac{H}{P_0} \times \frac{W}{P_0}}$ and fed to the smallest upsample decoder in the U-Net structure.

Upsample decoder block: The decoder consists of four upsample convolutional layers, each increasing the spatial resolution by a factor of two. Similarly, as in the original EVFlowNet [35], there is a skip connection from each encoder concatenated to the predictions from the corresponding decoder sharing the same dimension. The *tanh* activation function is used for flow predictions and loss is applied to the flow prediction generated at each scale and concatenated to the decoders.

3.3 Self-supervised loss

Due to lack of labeled training data, optical flow is often learned in a self-supervised manner. In our work, we train the network using contrast maximization (CMax) loss for self-supervised learning through motion compensation as proposed by Zhu et al. [37]. The method aims to find the optimized parameters that compensate for the motion and provide a deblurred image of warped events (IWE) upon convergence (see Fig. 1 (c)). To this end,

events $(x_i, y_i, t_i, p_i), i = 1, ..., N$, are transformed according to the motion model. In our case, using per pixel optical flow $(u(x_i, y_i), v(x_i, y_i))$ to a single time t':

$$\begin{bmatrix} x_i' \\ y_i' \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \end{bmatrix} + (t' - t_i) \begin{bmatrix} u(x_i, y_i) \\ v(x_i, y_i) \end{bmatrix} . \tag{1}$$

To effectively maximize contrast/reduce blur, Zhu et al. [37] adopt a loss function that minimizes the time each event is subjected to the flow vector. This is achieved by minimizing the sum of squares to the average timestamp on a couple of images of the average timestamp at each pixel with bilinear interpolation, one per polarity:

$$T_{p'}(x, y \mid t') = \frac{\sum_{i|p_i=p'} \kappa(x - x_i') \kappa(y - y_i') t_i}{\sum_{i|p_i=p'} \kappa(x - x_i') \kappa(y - y_i') + \varepsilon} \qquad p' \in \{+, -\}, \varepsilon \approx 0 . \tag{2}$$

where $\kappa(a) = \max(0, 1 - |a|)$ is the bilinear sampling kernel and ε is a tiny value to avoid division by zero. The contrast maximization loss is built with the sum of the two average timestamp images squared, scaling with per pixel event count $n(x')$:

$$\mathcal{L}_{CMax}(t') = \frac{\sum_x \sum_y T_{p' \in \{+\}}(x, y \mid t')^2 + T_{p' \in \{-\}}(x, y \mid t')^2}{\sum_x \sum_y [n(x') > 0] + \varepsilon} \qquad \varepsilon \approx 0 . \tag{3}$$

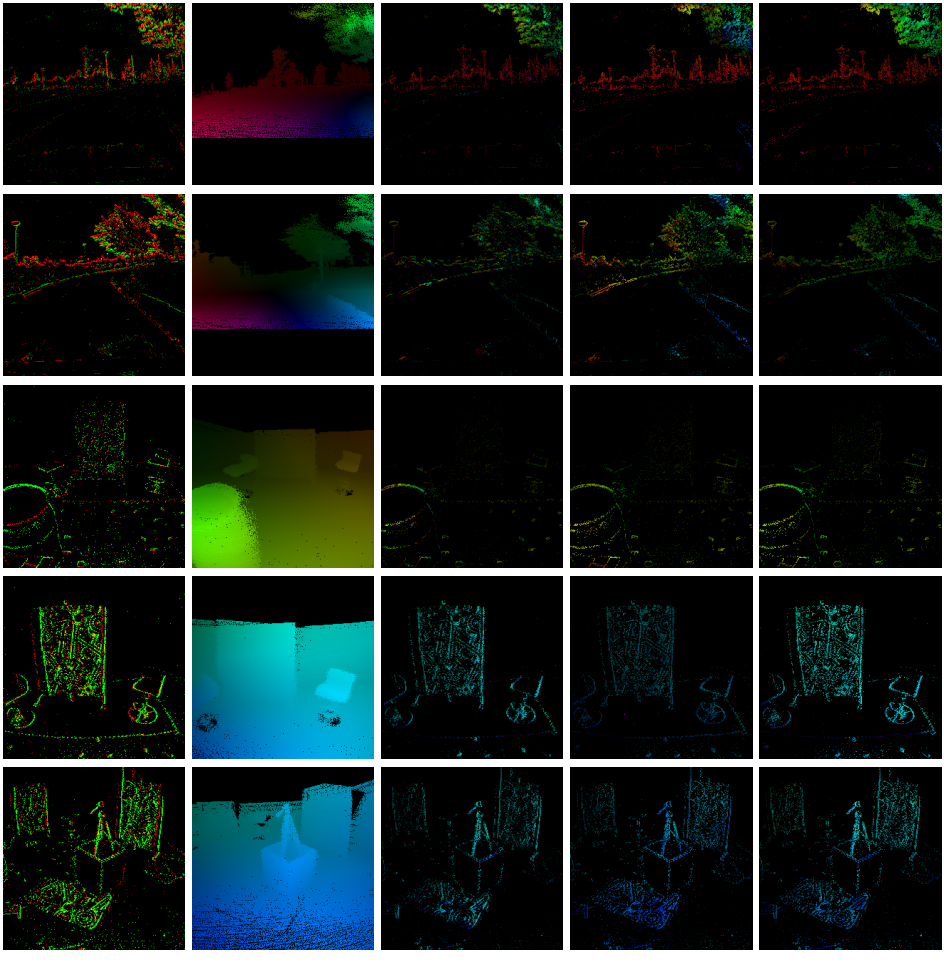
The scaling shows better convexity for the loss function according to [12]. In addition, we follow the forward-backward scheme suggested in [37] to limit the effect of gradient weighting for events further from t', and apply a Charbonnier smoothness loss term in the neighborhood pixels as regularizer. This is a common practice in the optical flow literature to relieve aperture problems and prevent event collapse problems in the CMax framework. The total loss function can be written as follows, where λ is a scaling factor:

$$\mathcal{L}_{flow} = \mathcal{L}_{CMax}^{forward}(t') + \mathcal{L}_{CMax}^{backward}(t') + \lambda \mathcal{L}_{smooth} . \tag{4}$$

4 Experiments

4.1 Datasets and implementation details

We use the Multivehicle Stereo Event Camera Dataset (MVSEC) [36] for evaluation, which provides ground-truth optical flow data and is used as the benchmark for event-based optical flow estimation in many prior works. The common practice in the literature is to train the network on outdoor_day2 sequence of the MVSEC dataset and test it on other sequences. However, this may lead to overfitting the same outdoor scene data. Instead, we follow the training pipeline from [12]: we train our network on the UZH-FPV Drone Racing Dataset [7] and test it on indoor_flying1, indoor_flying2, indoor_flying3 and outdoor_driving1 sequences from MVSEC dataset for quantitative evaluation. We argue that in this way the results show better generalization and it provides a fair comparison with the results from GRU-EVFlowNet and FireNet. We also test our model on the High Quality Frames (HQF) dataset [25] and the Event Camera Dataset (ECD) [20]. These datasets provide sequences covering various scenarios and ranges of motion. However, due to lack of ground-truth data, we only provide qualitative evaluations for them in the supplementary material. We implement our model using Pytorch and train it on an NVIDIA Geforce GTX 2080 GPU with batch size of 8 and epoch of 100. We use Adam optimizer [15] with a learning rate of 0.0002.



(a) events (b) ground truth (c) FireNet (d) ConvGRU-EVFN (e) ET-FlowNet

Figure 3: Qualitative results for optical flow evaluated on the MVSEC dataset for the $dt = 1$ case. Top two rows are from outdoor_day1 sequence and the last three are from indoor_flying1, indoor_flying2 and indoor_flying3 sequences, respectively. The first column presents the event input and the second column shows the ground truth dense optical flow provided in the MVSEC dataset (Note that for evaluation we use the masked sparse optical flow). Compared to other self-supervised methods trained also on different datasets: (c) FireNet and (d) ConvGRU-EV-FlowNet, our method (e) ET-FlowNet shows qualitatively better results on the outdoor sequences and competitive results in the indoor sequences (best viewed in color).

4.2 Evaluation results

		outdoor_day1		indoor_flying1		indoor_flying2		indoor_flying3		
Training	$dt = 1$ frame	AEE	% Outlier	AEE	% Outlier	AEE	% Outlier	AEE	% Outlier	Learning
MVSEC	EV-FlowNet [35]	0.49	0.20	1.03	2.20	1.72	15.10	1.53	11.90	Semi-SL
	Spike-FlowNet [16]	0.47	-	0.84	-	1.28	-	1.11	-	
	STE-FlowNet [8]	0.42	0.00	0.57	0.1	0.79	1.6	0.72	1.3	
	EV-FlowNet2 [35]	**0.32**	**0.00**	0.58	**0.00**	**1.02**	**4.00**	**0.87**	**3.00**	
	EV-FlowNet2_indoor [24]	0.36	0.09	-	-	-	-	-	-	
FPV	EV-FlowNet2_retrained	0.56	0.17	0.62	**0.26**	**1.10**	**5.97**	**0.90**	**3.54**	Self-SL
	ConvGRU-EV-FlowNet [35]	0.47	0.25	0.60	0.51	1.17	8.06	0.93	5.64	
	FireNet [35]	0.55	0.35	0.89	1.93	1.62	14.65	1.35	10.64	
	ET-FlowNet (ours)	**0.39**	**0.12**	**0.57**	0.53	1.2	8.48	0.95	5.73	
	$dt = 4$ frames	AEE	% Outlier	AEE	% Outlier	AEE	% Outlier	AEE	% Outlier	
MVSEC	EV-FlowNet [35]	1.23	7.30	2.25	24.70	4.05	45.30	3.45	39.70	Semi-SL
	Spike-FlowNet [16]	1.09	-	2.24	-	3.83	-	3.18	-	
	STE-FlowNet [8]	0.99	3.9	1.77	14.7	2.52	26.1	2.23	22.1	
	EV-FlowNet2 [35]	**1.30**	9.70	2.18	24.20	**3.85**	46.80	3.18	47.80	
	EV-FlowNet2_indoor [24]	1.49	11.72	-	-	-	-	-	-	
FPV	EV-FlowNet2_retrained	2.14	20.76	2.35	26.35	3.92	47.84	3.18	37.47	Self-SL
	ConvGRU-EV-FlowNet [35]	1.69	12.50	2.16	21.51	**3.90**	**40.72**	**3.00**	**29.60**	
	FireNet [35]	2.04	20.93	3.35	42.5	5.71	61.03	4.68	53.42	
	ET-FlowNet (ours)	**1.47**	**9.17**	**2.08**	**20.02**	3.99	41.33	3.13	31.70	

Table 1: Quantitative results for optical flow estimation for $dt = 1$ and $dt = 4$, and tested on various sequences of the MVSEC dataset. We sorted the methods according to: a) the training dataset used: MVSEC or UZH-FPV [7]; and b) the learning method: semi-supervised with grayscale images or self-supervised using event data only. Semi-supervised results are only shown for baseline purposes. Our method is compared with other self-supervised methods, with the best performing one shown in black bold. We highlight in blue methods matching the same learning and training conditions as ours (self-supervised and trained on the UZH-FPV dataset). Of these, the best-performing method is highlighted in blue bold unless they are already highlighted in black bold. To relate the success of self-supervised methods to semi-supervised ones, we also underline results for the overall winners in each tested sequence.

The quantitative results are shown in Table 1. We use the average endpoint error (AEE) and percentage of outliers as evaluation metrics. The optical flow vectors are scaled to be the displacement between each frame-based image in the $dt = 1$ case, and four frames in the $dt = 4$ case. The flows with endpoint error larger than 3 pixels and 5% of the magnitude of the ground truth flows are treated as outliers, following [35]. The ground truth flow is masked according to the event occurrences to generate corresponding sparse ground truth optical flow for evaluation. To compare with prior work, we show the results from other state-of-the-art networks based on self-supervised or semi-supervised learning. Though some supervised learning methods or model-based methods present better performance [10, 24], we do not include their results here for a fair comparison. The three semi-supervised based models shown in Table 1 (EV-FlowNet [35], Spike-FlowNet [16] and STE-FlowNet [8]) were trained using photometric loss from grayscale images for supervision, while the rest of self-supervised methods all use CMax loss from events only. We also sort all the methods according to their training datasets. Most of the models trained on the MVSEC dataset use the outdoor_day2 sequence [8, 16, 35, 37], except for [24] which trained on indoor sequences and evaluated on outdoor day sequence only. This practice may cause overfitting

on the same dataset. We train our model on a completely different dataset, following the same pipeline as in ConvGRU-EV-FlowNet [12] and FireNet [21]. Thus, these two networks are our main target for evaluation comparison. In addition, they both involve recurrent units and use a similar CMax loss function for self-supervision from pure events data input. We also retrained EV-FlowNet without recurrent units using voxel input representations. They present superior results in some indoor sequences for the $dt = 1$ case but much worse results for the larger time range case ($dt = 4$) without the presence of the recurrent units. In general, our method outperforms FireNet in all sequences and beats ConvGRU-EV-FlowNet, especially with a large margin for the outdoor_day1 sequence, and yields similar results in the indoor sequences. The qualitative results on the sequences shown in Fig. 3 further confirm these numbers with a visual comparison with ground-truth data.

4.3 Ablation studies

	outdoor_day1		indoor_flying1		indoor_flying2		indoor_flying3	
dt = 1 frame	AEE	% Outlier	AEE	% Outlier	AEE	% Outlier	AEE	% Outlier
baseline_2R	0.46	0.16	0.60	**0.40**	1.22	8.61	0.96	**5.58**
baseline_4R	0.82	1.21	0.88	1.50	1.39	9.88	1.16	6.64
ET-FlowNet_2T	0.41	0.14	0.59	0.50	**1.19**	8.81	0.99	6.79
ET-FlowNet_4T	**0.39**	**0.12**	**0.57**	0.53	1.2	**8.48**	**0.95**	5.73
ET-FlowNet_trapezoid	0.51	0.26	0.81	1.96	1.59	14.97	1.31	11.51
dt = 4 frame	AEE	% Outlier	AEE	% Outlier	AEE	% Outlier	AEE	% Outlier
baseline_2R	1.68	11.77	2.26	23.88	4.08	44.60	3.20	33.98
baseline_4R	2.97	39.41	3.51	47.73	4.92	60.24	4.20	53.00
ET-FlowNet_2T	1.55	10.67	2.17	22.61	4.00	42.82	3.31	35.27
ET-FlowNet_4T	**1.47**	**9.17**	**2.08**	**20.02**	**3.99**	**41.33**	**3.13**	**31.70**
ET-FlowNet_trapezoid	1.87	16.15	3.02	35.44	5.51	55.20	4.46	47.30

Table 2: Ablation studies for ET-FlowNet with quantitative evaluation on the MVSEC dataset for $dt = 1$ and $dt = 4$. 2R and 4R stand for two or four residual blocks, while 2T and 4T stand for two or four transformer blocks, respectively.

We perform ablation studies on our ET-FlowNet based on three factors: a) with or without the transformer block b) the number of encoders and decoders included in the transformer block, and c) their stacking structure. Table 2 shows these numbers. The first factor is to show the effectiveness of the transformer block. Since training without a transformer (or residual) block connecting encoders and decoders directly is undesirable, we compare ET-FlowNet with the best-performing self-supervised learning model to date for different training/testing sequences, ConvGRU-EV-FlowNet [12]. We retrain the ConvGRU-EV-FlowNet under our settings as the baseline model. To give an equivalent comparison, we trained the baseline model with 4 residual blocks. The poor results on the outdoor sequence show possible overfitting to the indoor drone racing datasets. This is consistent with the result shown in EV-FlowNet [35]. Our model (ET-FlowNet_4T) beats the baseline models in most of the sequences for $dt = 1$ case and in all the sequences for $dt = 4$ case. In addition, compared to ConvGRU-EV-FlowNet, the qualitative results from our model in Fig. 3 show more consistency in the detail areas of the object thanks to the transformer block. The other two factors

of our ablation study are aimed at identifying the proper design of the TPA transformer block. In our case, the variant with 4 transformer blocks (2 encoders and 2 decoders) outperforms an architecture with only 2 transformer blocks (1 encoder and 1 decoder) by a small margin in most sequences. This result suggests that for the case of ViTs it is better to use a richer block structure in the TPA. Weng et al. [30] carried out extensive ablation studies with regard to the staking fashion inside the TPA. Their conclusion is that a multiple-scale TPA (square stacking structure) performs better than a single-scale variant. Here we compare a square stacking structure (4-4-4-4) with 2 ViT encoders and 2 ViT decoders for each pyramid scale, with a trapezoid stacking one (6-4-4-2) with 3 ViT encoders and 3 ViT decoders in the last scale and 1 for each in the first scale. Since transformers are demonstrated to show effectiveness in the late stages of the network, we include more encoders/decoders in the last pyramid scale. We also ensure that both variants have the same number of transformer encoders/decoders. Our results are compatible with those of [30] where the trapezoid stacking underperforms the square structure. One possible reason is that placing fewer layers in the largest pyramid feature map would overlook short-range dependencies in the transformer.

5 Conclusion

In this paper, we proposed ET-FlowNet, the first RNN-ViT framework for event-based optical flow estimation. Our network incorporates a ViT block with TPA to extract the global spatial context and interaction among the multi-scale outputs from the encoder. We perform qualitative and quantitative evaluations on various datasets and compare them to state-of-the-art methods. Our method achieves superior results to other self-supervised methods on some of the sequences when trained and tested on different datasets. In addition, we use the event count representation to simplify the event preprocessing step and to provide a fair comparison with prior work. In future work, we aim to simplify the complexity of the network for less memory consumption and expand the self-attention mechanism as the backbone for the temporal domain.

Acknowledgements

This work was supported by projects EBSLAM DPI2017-89564-P and EBCON PID2020-119244GB-I00 funded by MCIN/AEI/10.13039/501100011033 and by an FI AGAUR PhD grant to Yi Tian.

References

[1] Nicolas Ballas, Li Yao, Chris Pal, and Aaron Courville. Delving deeper into convolutional networks for learning video representations. In *Int. Conf. Learning Representations*, San Juan, PR, 2016.

[2] Thomas Barbier and Jochen Triesch. Spike timing-based unsupervised learning of orientation, disparity, and motion representations in a spiking neural network. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition Workshops*, volume 25, Virtual, 2021.

[3] Ryad Benosman, Charles Clercq, Xavier Lagorce, Sio Hoi Ieng, and Chiara Bartolozzi. Event-based visual flow. *IEEE Trans. Neural Networks and Learning Systems*, 25(2): 407–417, 2014.

[4] Tobias Brosch, Stephan Tschechne, and Heiko Neumann. On event-based optical flow detection. *Frontiers in Neuroscience*, 9:137, 2015.

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conf. Computer Vision*, volume 12346 of *Lecture Notes in Computer Science*, pages 213–229, Virtual, 2020.

[6] Kenneth Chaney, Artemis Panagopoulou, Chankyu Lee, Kaushik Roy, and Kostas Daniilidis. Self-supervised optical flow with spiking neural networks and event based cameras. In *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pages 5892–5899, Prague, 2021.

[7] Jeffrey Delmerico, Titus Cieslewski, Henri Rebecq, Matthias Faessler, and Davide Scaramuzza. Are we ready for autonomous drone racing? the UZH-FPV drone racing dataset. In *IEEE Int. Conf. Robotics and Automation*, pages 6713–6719, Montreal, 2019.

[8] Ziluo Ding, Rui Zhao, Jiyuan Zhang, Tianxiao Gao, Ruiqin Xiong, Zhaofei Yu, and Tiejun Huang. Spatio-temporal recurrent networks for event-based optical flow estimation. In *AAAI Conf. ArtificialIntelligence*, volume 36, pages 525–533, Virtual, 2021.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Int. Conf. Learning Representations*, Virtual, 2021.

[10] Mathias Gehrig, Mario Millhausler, Daniel Gehrig, and Davide Scaramuzza. E-RAFT: Dense optical flow from event cameras. In *Int. Conf. 3D Vision*, pages 197–206, Virtual, 2021.

[11] Vitor Guizilini, Rares Ambrus, Dian Chen, Sergey Zakharov, and Adrien Gaidon. Multi-frame self-supervised depth with transformers. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition*, New Orleans, 2022.

[12] Jesse Hagenaars, Federico Paredes-Vallés, and Guido de Croon. Self-supervised learning of event-based optical flow with spiking neural networks. In *Conf. Neural Information Processing Systems*, volume 34, Virtual, 2021.

[13] Zhaoyang Huang, Xiaoyu Shi, Chao Zhang, Qiang Wang, Ka Chun Cheung, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Flowformer: A transformer architecture for optical flow. In *European Conf. Computer Vision*, Lecture Notes in Computer Science, Tel Aviv, 2022.

[14] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, Olivier Hénaff, Matthew M Botvinick, Andrew Zisserman, Oriol Vinyals, and João

Carreira. Perceiver IO: A general architecture for structured inputs & outputs. In *Int. Conf. Learning Representations*, Virtual, 2021.

[15] Diederik P Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *Int. Conf. Learning Representations*, San Diego, CA, 2015.

[16] Chankyu Lee, Adarsh Kumar Kosta, Alex Zihao Zhu, Kenneth Chaney, Kostas Daniilidis, and Kaushik Roy. Spike-FlowNet: Event-based optical flow estimation with energy-efficient hybrid neural networks. In *European Conf. Computer Vision*, volume 12374 of *Lecture Notes in Computer Science*, pages 366–382, Glasgow, 2020.

[17] Chankyu Lee, Adarsh Kumar Kosta, and Kaushik Roy. Fusion-FlowNet: Energy-efficient optical flow estimation using sensor fusion and deep fused spiking-analog network architectures. In *IEEE Int. Conf. Robotics and Automation*, pages 6504–6510, Xian, 2021.

[18] Zhihao Li, M. Salman Asif, and Zhan Ma. Event transformer, 2022. CoRR arXiv.2204.05172.

[19] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *IEEE/CVF Int. Conf. Computer Vision*, pages 9992–10002, Virtual, 2021.

[20] Elias Mueggler, Henri Rebecq, Guillermo Gallego, Tobi Delbruck, and Davide Scaramuzza. The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM. *Int. J. Robotics Research*, 36(2):142–149, 2017.

[21] Federico Paredes-Valles and Guido C H E de Croon. Back to event basics: Self-supervised learning of image reconstruction for event cameras via photometric constancy. *IEEE/CVF Conf. Computer Vision and Pattern Recognition*, 2021.

[22] Federico Paredes-Valles, Kirk Yannick Willehm Scheper, and Guido C H E Cornelis Henricus Eugene De Croon. Unsupervised learning of a hierarchical spiking neural network for optical flow estimation: From events to global motion perception. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 42(8):2051–2064, 2020.

[23] Alberto Sabater, Luis Montesano, and Ana C. Murillo. Event transformer. a sparse-aware solution for efficient event data processing. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition Workshops*, New Orleans, 2022.

[24] Shintaro Shiba, Yoshimitsu Aoki, and Guillermo Gallego. Secrets of event-based optical flow. In *European Conf. Computer Vision*, Lecture Notes in Computer Science, Tel Aviv, 2022.

[25] Timo Stoffregen, Cedric Scheerlinck, Davide Scaramuzza, Tom Drummond, Nick Barnes, Lindsay Kleeman, and Robert Mahony. Reducing the sim-to-real gap for event cameras. In *European Conf. Computer Vision*, volume 12372 of *Lecture Notes in Computer Science*, pages 534–549, Virtual, 2020.

[26] Xiuchao Sui, Shaohua Li, Xue Geng, Yan Wu, Xinxing Xu, Yong Liu, Rick Goh, and Hongyuan Zhu. CRAFT: Cross-attentional flow transformer for robust optical flow. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition*, New Orleans, 2022.

[27] Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *European Conf. Computer Vision*, volume 12347 of *Lecture Notes in Computer Science*, pages 402–419, 2020.

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Conf. Neural Information Processing Systems*, pages 5999–6009, 2017.

[29] Zuowen Wang, Yuhuang Hu, and Shih-Chii Liu. Exploiting spatial sparsity for event cameras with visual transformers, 2022. CoRR arXiv.2202.05054.

[30] Wenming Weng, Yueyi Zhang, and Zhiwei Xiong. Event-based video reconstruction using transformer. In *IEEE/CVF Int. Conf. Computer Vision*, Virtual, 2021.

[31] Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Zhicheng Yan, Masayoshi Tomizuka, Joseph Gonzalez, Kurt Keutzer, and Peter Vajda. Visual transformers: Where do transformers really belong in vision models? In *IEEE/CVF Int. Conf. Computer Vision*, pages 579–589, Virtual, 2021.

[32] Chengxi Ye, Anton Mitrokhin, Cornelia Fermuller, James A Yorke, and Yiannis Aloimonos. Unsupervised learning of dense optical flow, depth and egomotion with event-based sensors. In *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pages 5831–5838, Virtual, 2020.

[33] Dong Zhang, Hanwang Zhang, Jinhui Tang, Meng Wang, Xiansheng Hua, and Qianru Sun. Feature pyramid transformer. In *European Conf. Computer Vision*, Lecture Notes in Computer Science, pages 323–339, Virtual, 2020.

[34] Jiqing Zhang, Bo Dong, Haiwei Zhang, Jianchuan Ding, Felix Heide, Baocai Yin, and Xin Yang. Spiking transformers for event-based single object tracking. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition*, New Orleans, 2022.

[35] Alex Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. EV-FlowNet: Self-supervised optical flow estimation for event-based cameras. In *Robotics: Science and Systems*, Pittsburgh, 2018.

[36] Alex Zihao Zhu, Dinesh Thakur, Tolga Özaslan, Bernd Pfrommer, Vijay Kumar, and Kostas Daniilidis. The multivehicle stereo event camera dataset: An event camera dataset for 3D perception. *IEEE Robotics and Automation Letters*, 3(3):2032–2039, 2018.

[37] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised event-based learning of optical flow, depth, and egomotion. *IEEE/CVF Conf. Computer Vision and Pattern Recognition*, pages 989–997, 2019.