Institut de Robòtica i Informàtica Industrial

CSIC

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

ICPR 2024 INDIA

Our code is available:

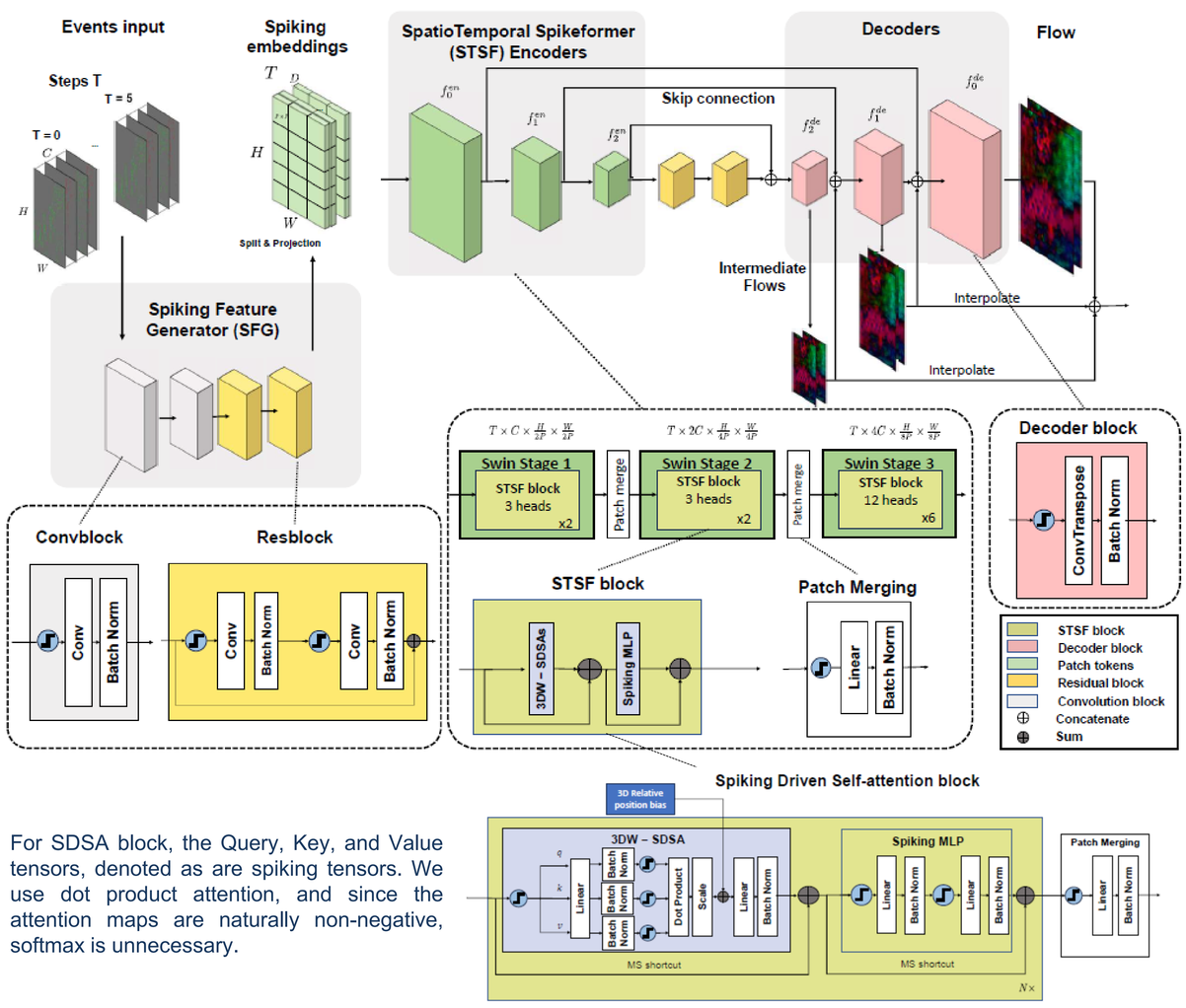
SDformerFlow: Spiking Neural Network Transformer for Event-based Optical Flow

Yi Tian, Juan Andrade-Cetto

1. Abstract

Event cameras produce asynchronous and sparse event streams capturing changes in light intensity. Overcoming limitations of conventional frame-based cameras, such as low dynamic range and data rate, event cameras prove advantageous, particularly in scenarios with fast motion or challenging illumination conditions. Leveraging similar asynchronous and sparse characteristics, **Spiking Neural Networks (SNNs)** emerge as natural counterparts for processing event camera data. Recent advancements in Visual Transformer architectures have demonstrated enhanced performance in both Artificial Neural Networks (ANNs) and SNNs across various computer vision tasks. Motivated by the potential of transformers and spikeformers, we propose two solutions for fast and robust optical flow estimation: **STTFlowNet** and **SDformerFlow**. STTFlowNet adopts a U-shaped ANN architecture with spatiotemporal Swin transformer encoders, while SDformerFlow presents its full spike counterpart with **spike-driven Swin transformer** encoders. Notably, our work marks the first utilization of spikeformer for dense optical flow estimation. We conduct end-to-end training for both models using supervised learning on the DSEC-flow Dataset. Our results indicate comparable performance with state-of-the-art SNNs and significant improvement in power consumption compared to the best-performing ANNs for the same task.

2. Network Architecture for SDformerFlow



For SDSA block, the Query, Key, and Value tensors, denoted as are spiking tensors. We use dot product attention, and since the attention maps are naturally non-negative, softmax is unnecessary.

For SDformerFlow, the primary architecture comprises three components: a) **a Spike Feature Generator (SFG) embedding** module, b) **a Spatiotemporal Swin Spikeformer (STSF) encoder**, and c**) spike decoders and flow prediction**. The event stream initially enters the SFG module, which outputs spatiotemporal embeddings for the STSF encoders. The STSF encoders then generate spatiotemporal features hierarchically. Subsequently, the output from each encoder is concatenated to the decoder at the same scale to predict the flow map. Two additional residual blocks exist between the encoder and decoder modules.

3. Training method

$$L = \frac{1}{n}\sum_{i=1}^{n} |\mathbf{u}_i^{pred} - \mathbf{u}_i^{gt}|$$

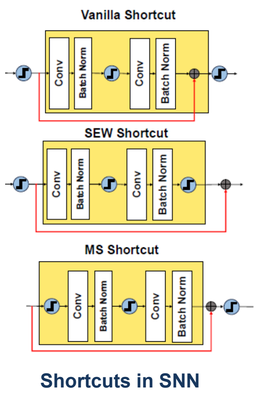
We train our model with **supervised learning** using the **mean absolute error** between the estimated optical flow and the ground-truth flow. For SNN, we employ surrogate gradient (SG) with backpropagation through time (BPTT) to train the network. We use the inverse tangent as the surrogate function

5. Ablation study

Model	AEE		Outlier %		AAE		I	Training res. Param. (M)	
test res: cropped (C) or full (F)	C	F	C	F	C	F			
LIF-EV-FlowNet-en4-s5	3.08	3.47	19.67	23.70	17.90	14.41	voxel10	288,384	14.13
SpikeformerFlowNet-SEW-en3-s8-c4	1.60	3.21	11.90	32.30	12.51	14.77	voxel15*	240,320	19.80
SpikeformerFlowNet-SEW-en3-s4-c8	1.76	3.54	13.43	41.18	14.01	27.81	voxel15*	240,320	19.81
SpikeformerFlowNet-SEW-en3-s5-c4	1.51	2.52	9.85	22.75	10.68	11.10	voxel10	288,384	19.83
SpikeformerFlowNet-MS-en3-s5-c4	1.28	2.01	6.91	15.55	9.01	8.99	voxel10	288,384	19.83
SpikeformerFlowNet-MS-en4-s5-c4	**1.25**	**1.98**	**6.69**	**15.06**	**8.48**	**8.81**	voxel10	288,384	56.48
SpikeCAformerFlow-MS-en4-s5-c4	1.66	2.97	10.65	27.87	12.05	22.55	voxel10	288,384	15.73

*The SEW variant with input voxel size of 15 was trained with a resolution of 240 × 320 due to GPU memory limitations. The rest of the Spikeformer models were trained at 288 × 384 resolution.

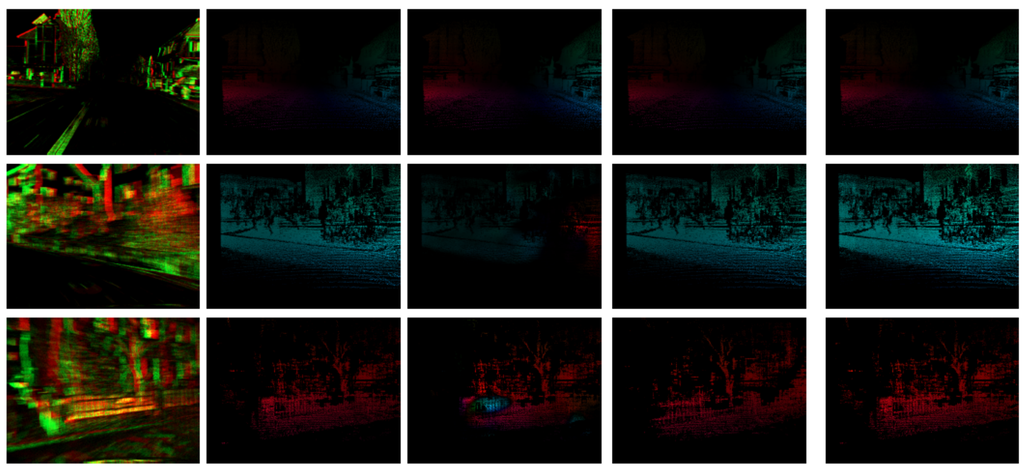
Ablation study for STTFlowNet. Column I stands for the event input type. For the variant of STTFlowNet, en means number of encoders, b stands for number of input blocks, p means spatial patch size, and w stands for swin spatial window size. Best-performing results are highlighted.



Shortcuts in SNN

We studied: a) **the number of time steps/channels**; b) **shortcut variants: SEW or MS shortcuts**; and c) **the number of encoders**. The spikeformer encoders significantly improved performance compared to the baseline model, albeit with reduced robustness when directly tested on scaled-up resolutions. Increasing the number of time steps helped capture temporal information at the expense of increased memory consumption. The MS shortcut variant notably improved results compared to SEW shortcut. One possible explanation is that the MS shortcut provides an information flow path between the states of the neurons before the spike function and is not regulated by their firing status. Increasing the number of encoders from three to four further enhanced performance at the cost of increased parameters. Finally, incorporating convolution-based modules as CAformer in the first two swin encoders yielded a lightweight model but with slightly reduced performance.

4. Qualitative and quantitative results



(a) Events (b) GT (c) EVFlowNet (d) STTFlowNet (e) SDformerNet

Qualitative results for optical flow are evaluated on the DSEC validation dataset. The first column displays the event input, while the second column depicts the ground truth dense optical flow from our split validation dataset. During evaluation, we mask the estimated flow where ground truth flows are available. (Best viewed in color). Notably, when the vehicle moves forward in steady motion, all models achieve accurate flow estimation. However, in scenarios involving sharp turns or large, abrupt motions (third row in the figure), the baseline EVFlowNet struggles to estimate the correct direction. In contrast, both our STTFlowNet and our fully spiking model effectively handle such scenarios,

(a) events (b) EVFlowNet (c) STTFlowNet (d) SDformerFlow

Qualitative results for optical flow are evaluated on the official DSEC test dataset. The first column presents the event input, while the other columns show the corresponding estimated optical flow for the baseline method EVFlow and our methods STTFlowNet and SDformerFlow. (Best viewed in color).

Training		EPE	Outlier %	AAE
A	E-RAFT [10]	0.779	2.684	2.838
	EV-FlowNet_retrained [10]	2.32	18.60	-
	IDNet [35]	0.719	2.036	2.723
	TMA [19]	0.743	2.301	2.684
	E-Flowformer [18]	0.759	2.446	2.676
	TamingCM[26]	2.33	17.771	10.56
	STTFlowNet-en3 (Ours)	0.997	4.588	3.235
S	SNN_3DNet[2]	1.707	10.308	6.338
	SDFormerFlow-en3 (Ours)	2.142	14.021	5.941
M	MultiCM [28]	3.472	30.855	13.983

Quantitative results for optical flow estimation of the DSEC optical flow benchmarks for all the test sequences. The first column shows the methods, A stands for ANN, S stands for SNN, while M stands for model-based method.

Regarding the quantitative evaluation tested on DSEC, our ANN model outperforms the baseline model EVFlowNet and other self-supervised trained models. However, it still trails behind correlation-volume-based models. The only SNN model included in the benchmark uses stateless neurons and is trained at full resolution, whereas most other SNN approaches are trained and validated on cropped resolution, with limited representation in the benchmark.

Training	dt = 1 frame	D	outdoor_day1		indoor_flying1		indoor_flying2		indoor_flying3		Avg	
			AEE	% Outlier	AEE	% Outlier	AEE	% Outlier	AEE	% Outlier	AEE	% Outlier
A	EV-FlowNet [44]	M	0.49	0.20	1.03	2.20	1.72	15.10	1.53	11.90	1.19	7.35
	EV-FlowNet2 [46]	M	**0.32**	0.00	0.58	0.00	1.02	4.00	0.87	3.00	0.69	1.75
	GRU-EV-FlowNet [12]	FPV	0.47	0.25	0.60	0.51	1.17	8.06	0.93	5.64	0.79	3.62
	STE-FlowNet [3]	M	0.42	0	0.57	0.1	0.79	1.6	1.72	1.3	0.62	0.75
	ET-FlowNet [32]	FPV	0.39	0.12	0.57	0.53	1.2	8.48	0.95	5.73	0.78	3.72
	ADM-Flow [24]	MDR	0.41	0.00	**0.52**	0.14	**0.68**	1.18	**0.52**	0.04	**0.53**	0.34
	STT-FlowNet (ours)	MDR	0.66	0.29	0.57	0.33	0.88	4.47	0.73	1.58	0.71	1.67
S	Spike-FlowNet [17]	M	0.49		0.84		1.28		1.11		0.93	
	XLIF-EV-FlowNet [12]	FPV	0.45	0.16	0.73	0.92	1.45	12.18	1.17	8.35	0.95	5.40
	Adaptive-SpikeNet [16]	FPV	0.44		0.79		1.37		1.11		0.93	
	SNN3DNet [2]	M	0.85		0.58		0.72		0.67		0.71	
	SDformerFlow (Ours)	MDR	0.69	0.21	0.61	0.60	0.83	3.41	0.76	1.45	0.72	1.42

Qualitative results for optical flow evaluated on the MVSEC dataset for the dt = 1 case. The first row is from outdoor_day1 sequence and the last row is from the indoor_flying sequence. Note that for evaluation we use the masked sparse optical flow.

For the quantitative evaluation tested on MVSEC dataset, both our ANN and SNN models yield competitive results. Our ANN model performs better than another transformer-based U-Net architecture. Our SDformerFlow ranked second for the average AEE for all the sequences among all the SNN methods. However, the best performing model reports their results for the indoor sequences separately trained on the subsets of the same dataset, which may have overfitted to the test dataset.

6. Energy consumption

Model	EPE	Type	Param (M)	FLOPS(G)	Avg. spiking rate	Power(mJ)
EVFlowNet retrained	1.57	ANN	14.14	22.38	-	102.95
LIF-EVFlowNet	3.08	SNN	14.13	22.38	0.29	29.21
STTFlowNet-en3	0.72	ANN	20.30	86.88	-	399.65
SDFlowNet-en3	1.28	SNN	19.83	34.80	0.27	37.64
SDFlowNet-en4	1.25	SNN	56.48	39.10	0.27	41.08

Energy consumption for ANN and SNN models

Energy consumption for ANN: $FLOPS \times E_{MAC}$
Energy consumption for SNN: $FLOPS \times R_s \times T \times E_{AC}$
we estimate energy consumption based on the number of floating-point operations (FLOPS) required, as all operations in ANN layers are multiply-accumulate (MAC) operations. Conversely, SNN models convert multiplication operations into addition operations due to their binary nature. For 32-bit floating-point computation, these energy values are typically EMAC = 4.6pJ and EAC = 0.9pJ, respectively, based on a 45 nm technology Our results demonstrate that the energy consumption of our SNN model is nearly one-tenth that of its ANN counterpart and one-third that of the baseline EVFlowNet model.

Date: December 1-5, 2024

This work received support from projects EBCON (PID2020-119244GB-I00) and RAADICAL (PLEC2021-007817) ; the Consolidated Research Group RAIG (2021 SGR 00510) and by an FI AGAUR PhD grant to Yi Tian.