

Fusing Frame and Event Vision for High-speed Optical Flow for Edge Application

Ashwin Sanjay Lele, Arijit Raychowdhury

School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA
alele9@gatech.edu, arijit.raychowdhury@ece.gatech.edu

Abstract—Optical flow computation with frame-based cameras provides high accuracy but the speed is limited either by the model size of the algorithm or by the frame rate of the camera. This makes it inadequate for high-speed applications. Event cameras provide continuous asynchronous event streams overcoming the frame-rate limitation. However, the algorithms for processing the data either borrow frame like setup limiting the speed or suffer from lower accuracy. We fuse the complementary accuracy and speed advantages of the frame and event-based pipelines to provide high-speed optical flow while maintaining a low error rate. Our bio-mimetic network is validated with the MVSEC dataset showing 19% error degradation at $4\times$ speed up. We then demonstrate the system with a high-speed drone flight scenario where a high-speed event camera computes the flow even before the optical camera sees the drone making it suited for applications like tracking and segmentation. This work shows the fundamental trade-offs in frame-based processing may be overcome by fusing data from other modalities.

Index Terms—Computer Vision, Dynamic Vision Sensors, Drone Tracking, Accuracy-speed trade-off

I. INTRODUCTION

Computation of optical flow (OF) finds applications in many computer vision and robotics tasks ranging from pose estimation [1], video stabilization [2], visual odometry [3], collision avoidance [4] to feature tracking [5] etc. Consistent previous exploration into both model-based algorithms [6] and convolutional neural networks (CNNs) [7] have achieved unsurpassed accuracy levels for this task. Highly accurate CNNs require significant inference latency [8] while smaller models [9] trade off accuracy for speed. Model-based optimization techniques [10] require significant computation time due to a large number of memory accesses and pipelined computations. Even faster methods eventually get limited by the frame rate of the regular optical camera so reducing the camera resolution to accelerate the computation is not enough. Therefore the optical cameras along with conventional computer vision techniques remain inadequate for high-speed edge applications due to discrete data processing modality.¹

Dynamic vision sensors (DVS) or event cameras provide a new mode of visual information with the visual data appearing as a continuous asynchronous stream of binary events instead of discrete intensity frames. An event corresponds to a change in intensity of the pixel and thus the event stream generally corresponds to moving objects in a constant light environment. Event cameras offer low power, very high dynamic range in addition to fine temporal resolution [11]. The asynchronous event generation circumvents the fundamental speed limitation

¹To appear in the proceedings of IEEE International Symposium on Circuits and Systems (ISCAS) 2022

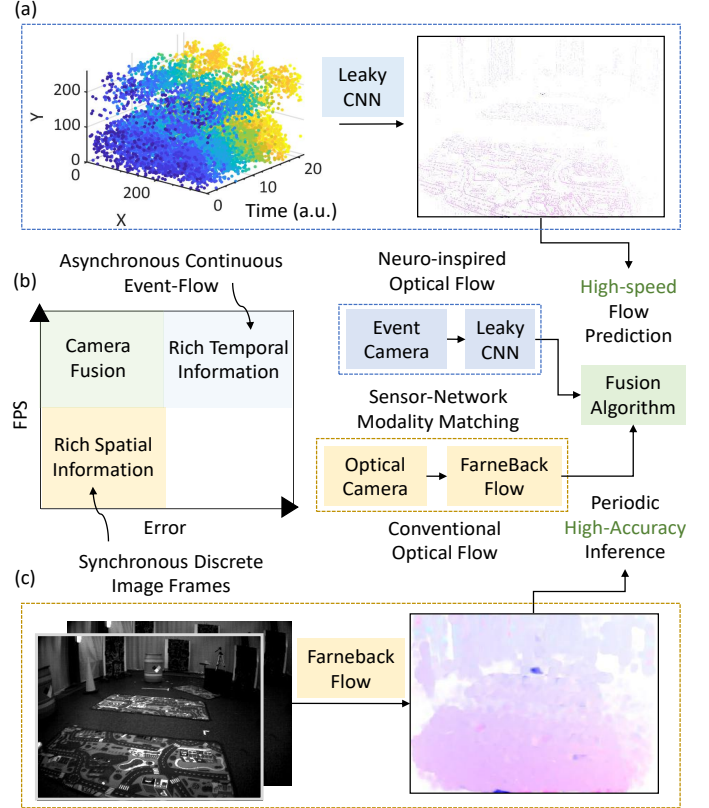


Fig. 1. (a) High speed OF computation using proposed leaky CNN filter for event camera (c) High accuracy OF computation using model-based Farneback algorithm for optical camera (b) Fusing complementary advantages to mitigate accuracy-latency trade-off

put by the frame rate. However, assigning the events to objects and thus matching the features for flow computation is complex because of the unavailability of intensity information.

Approaches to take advantage of the high throughput data can be briefly categorized into conventional model-based approaches and CNN or more recent spiking neural networks. CNNs [12], [13] and hybrid approaches [14], [15] have CNN backend causing similar latencies as optical frames. Model-based approaches use iterative optimizations [16], [17], [18] causing similar processing modality as that of the optical camera and are speed limited. SNN approaches use Spike-time dependent plasticity [19], delay coding [20], [21] for training with smaller networks achieving high speed. But their accuracy is limited due to the lack of reliable training methods and are typically applied to simple custom made datasets. Thus, although event cameras have high potential speed get

stuck in low throughput with conventional processing or lower accuracy with spiking networks.

This provides us with two modes of visual information namely frame and events with complementary advantages of accuracy and speed respectively (Fig. 1). Temporally detailed event stream promises high speed whereas optical frames offer high accuracy with spatial details. We envision fusing this multi-modal data to extract speed while maintaining accuracy by fusing the inferences from both pipelines. The high speed and lower accuracy event prediction is fused with a low-speed high-accuracy frame inference to induce robustness against noise while boosting the throughput. Computation on events is carried out using the shallow and local computation based leaky CNN filter that imitates correlation-based flow estimation similar to rabbit and insects [22], [23]. The system is validated on the MVSEC dataset [24] to show 19% increase in error while boosting the throughput by 4×. The application to a rapidly moving drone flight shows that the movements that are fast to be captured by the optical camera are successfully captured by the event camera and OF is fused with high accuracy. This work demonstrates the potential of multi-modal fusion systems for overcoming trade-offs in frame-based processing.

II. METHODOLOGY

A. Flow Estimating Leaky CNN Filter

The OF estimation network needs to take the timestamped event stream to predict the flow at each active pixel. The shallow 3 layered leaky CNN is shown in Fig 2(a). Layer 1 acts as a leaky accumulator as shown in the equation. The most recent event adds to the current activation while the leakiness causes the contribution of previous events to diminish over time. This allows the neurons in a neighbourhood to roughly predict the direction of motion of the object in the field of view from smaller activation to higher. The accumulated activations are shown in Fig 2(b).

The second layer computes the difference in the consecutive activation of neurons in both vertical and horizontal directions as this encodes the direction of local flow. The magnitude and polarity of the difference provide the noisy flow at the pixel. This is carried out using differential excitatory and inhibitory synapses connecting from neighbouring neurons of layer 1. Both vertical and horizontal flow are calculated separately using different kernels providing layer2a and Layer2b. Due to the high noise induced by the event camera, layer 2 activations can be seen to possess heavy granularity (Fig. 2(b)). The noise is reduced by the averaging kernel applied to layer 3 where a uniform set of excitatory synapses average the activations to provide a smoothened flow (Layer3a,3b). Each active pixel is assigned a vertical and horizontal flow value which is shown in the flow visualization.

B. Conventional Optical Flow

The noisy flow estimation from the leaky CNN filter is to be corrected using slower but reliable conventional optical flow detectors. Multiple CNN models [9], [25] and optimization

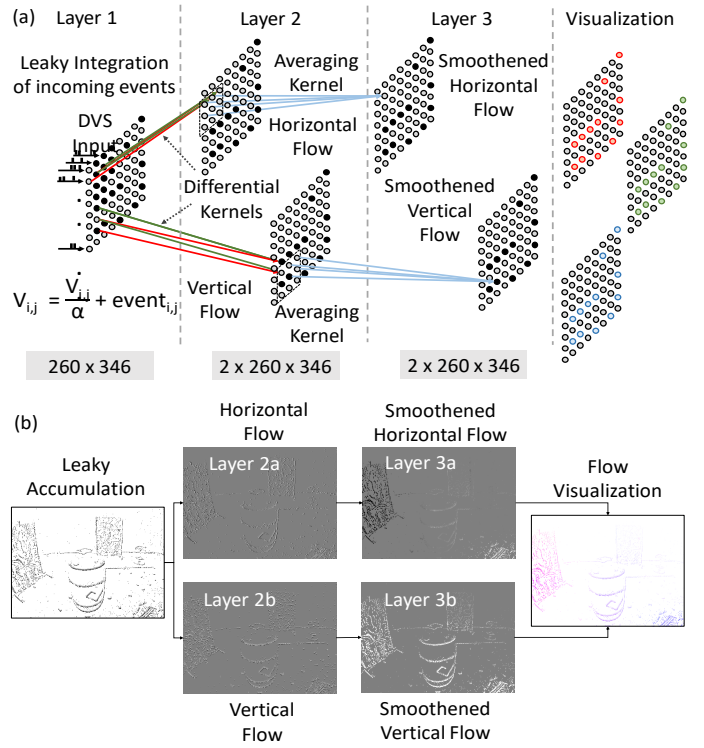


Fig. 2. (a) Leaky CNN Filter - layer 2 computes local flow and layer 3 smoothenes it (b) Activations of each neuron in the layer in determining the flow

models like [26], [27] can be used. However, our dataset under consideration (MVSEC) provides grayscale optical images and DAVIS 346 experiments also provide saturated coloured images incorrectly suited for CNN models. We thus go with the conventional gradient matching Farneback algorithm [26] because of its balance between latency and accuracy [28].

C. Fused Event-Frame system

The fusion between the outputs from two modalities is intended to preserve the accuracy from the Farneback flow while incorporating pixels from leaky CNN that have a high likelihood of correct flow value for faster moving objects. This means that if some object has moved rapidly within the scene, the event pipeline should be able to capture that while the optical camera provides reliable detection for the background scene. This is implemented by Algorithm 1. A confidence map stores the likelihood that prediction from leaky CNN is acceptable. The pixels with high confidence scores are taken from leaky CNN prediction while others are used from previous Farneback flow computation as shown in Fig. 3.

A high confidence score is required for pixels that have seen rapid movement missed by the frame pipeline but captured by the event pipeline. The first condition requires that the leaky CNN flow prediction for the pixel differs from the flow for the same pixel in the previous frame inference (Algorithm 1 - condition 1). The second condition requires the flow to be consistent with the previous leaky CNN estimate to ensure the deviation is not because of noise and corresponds to some moving object (Algorithm 1 - condition 2). Thus

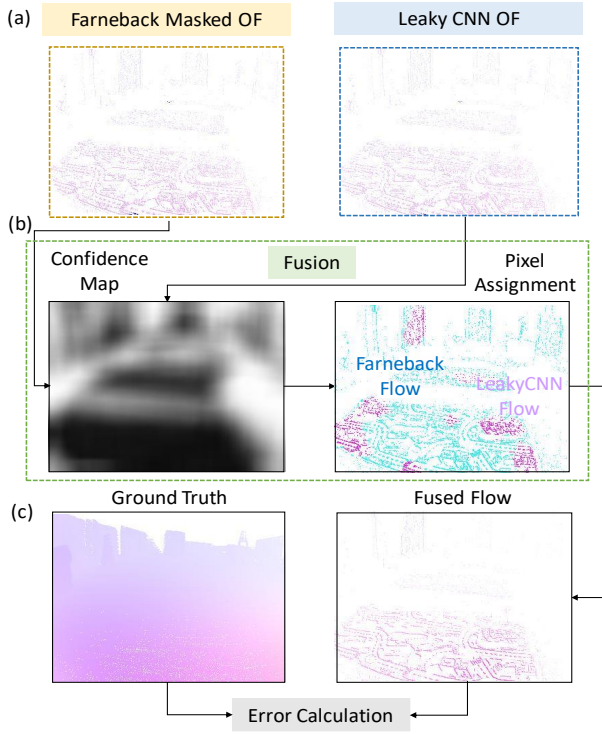


Fig. 3. Flow of fusion algorithm (a) Computed flow maps by leaky filter and Farneback algorithm (b) Confidence map for event-based OF. Pixels with high confidence values are taken from here (c) Fused flow is compared with ground truth for AEE calculation

the confidence score rises for a pixel when its Euclidean distance from the previous frame inference is large but close to the previous leaky CNN flow. The final fused flow map is generated by using pixels from leaky CNN inference wherever the confidence map is higher than a predefined threshold. This is outlined in Algorithm 1. Fig. 3 shows the OF values from both pipelines and the corresponding confidence map. The final fused flow is compared with ground truth to calculate the average endpoint error (AEE).

III. RESULTS

A. Fusion Mechanism

We begin by exploring the parametric dependence of user-defined parameters on the fusion algorithm. The fraction of pixels coming from event pipeline prediction (shown as event percent in Fig. 4) OF depends crucially upon the error tolerance thresholds (condition 1,2). Low $thresh_{Farneback}$ and high $thresh_{leakyCNN}$ results in a larger fraction of pixels coming from the event pipeline. This is depicted in Fig. 4 (a). The percentage of event OF monotonically rises when these two conditions are met and provides the user with control knobs to tune the algorithm.

A high percentage of pixels coming from the event pipeline is expected to corrupt the accuracy because of the noise it injects. As the FPS increases, more frames from the event pipeline are processed between every inference of the frame pipeline. This causes the percentage contributed from event pipeline prediction to increase as shown in Fig 4(b). The

```

while True do
  Condition 1:
     $Distance_{Farneback} = ||OF_{Farneback},$ 
     $OF_{leakyCNN}||$ 
    if  $Distance_{Farneback} > thresh_{Farneback}$  then
      |  $Error_{Farneback} = 1$ 
    else
      |  $Error_{Farneback} = 0$ 
    end
  Condition 2:
     $Distance_{leakyCNN} = ||OF_{leakyCNN_t},$ 
     $OF_{leakyCNN_{t-1}}||$ 
    if  $Distance_{leakyCNN} < thresh_{leakyCNN}$  then
      |  $Error_{leakyCNN} = 1$ 
    else
      |  $Error_{leakyCNN} = 0$ 
    end
    belief =  $Error_{Farneback} * Error_{leakyCNN}$ 
    belief = belief_prev + belief
    Confidence Map = Confidence Map + belief
    if Confidence Map > thresh then
      |  $Pixels_{leakyCNN} = 1$ 
    else
      |  $Pixels_{Farneback} = 1$ 
    end
     $OF_{fused} = OF_{Farneback} * Pixels_{Farneback}$ 
     $+ OF_{leakyCNN} * pixels_{leakyCNN}$ 
end

```

* is element-wise multiplication

Algorithm 1: Fusion Algorithm

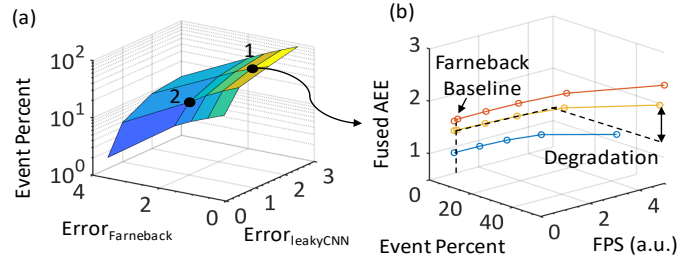


Fig. 4. Fusion parameters sweep. (a) Low $thresh_{Farneback}$ and high $thresh_{leakyCNN}$ allows high percentage of pixels from event camera pipeline in the flow (b) Accuracy gradually degrades with higher percentage of pixels from leaky CNN at a higher FPS rate

error thresholds are from point ‘1’ in Fig. 4(a). The error monotonically rises with FPS and percentage contributed from the event pipeline.

A comparison of this method with previous methods for the MVSEC dataset that captures multiple indoor scenes with an event camera mounted on a drone is carried out in Table 1. Error thresholds are from point ‘2’ in Fig 4(a). The FPS of the Farneback flow is taken as the baseline of 1 while the event stream from the event camera between the consecutive frames is divided into multiple frames and is processed through the event pipeline. Fig. 5(a) shows that as the FPS rises, the noisy contribution of the event pipeline rises. We observe that the error degradation from Farneback to the fused method is small

TABLE I
MVSEC - AVERAGE END-POINT ERROR (AEE) VS FPS

Network	Indoor Flying 1	Indoor Flying 2	Indoor Flying 3	Time ms	FPS
Unflow [29]	0.5	0.7	0.55	-	-
EV Flow [12]	1.03	1.72	1.53	48	21
SpikeFlow [14]	0.84	1.28	1.11	23.11	43
Zhu et. al. [30]	0.58	1.02	0.87	-	-
FusionFlow [15]	0.56	0.95	0.76	-	-
Full ANN [15]	0.68	0.97	0.97	-	-
ECN [13]	0.49	0.43	0.48	4	250
This work	0.95	1.55	1.38	24	> 41

while the frame rate increases significantly (4x). Thus, the fusion algorithm mitigates the trade-off between event and frame pipelines extracting their complementary advantages. The throughput vs. FPS trade-off of various previous methods is shown in Fig. 5(b). The FPS shown for our method is for pipelined execution in FPGA as described in the next section.

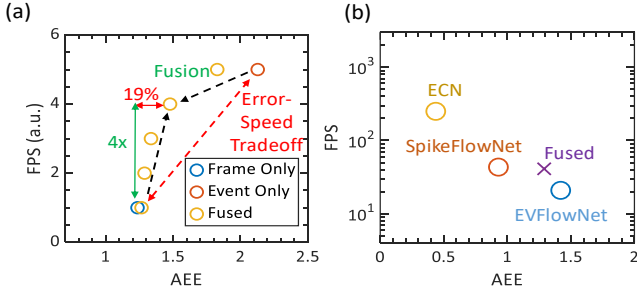


Fig. 5. (a) Overcoming the accuracy latency trade-off (b) Comparison with previous approaches

B. Real World Demonstration

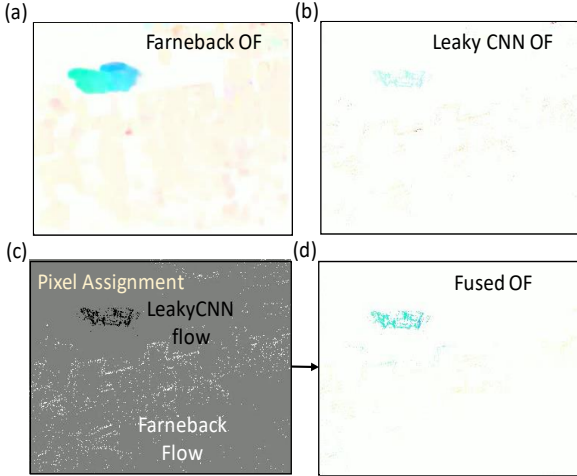


Fig. 6. Screenshots from drone experiment with high-speed flow for drone fused from leakyCNN while the background is taken from Farneback flow

The system is applied to a real-world scenario of a flying drone being captured with a handheld DAVIS346. This provides both color and frame information. The fused optical flow method is applied to this where the rapid movement of the drone that is harder to capture using the only optical camera can be correctly identified using the event-camera pipeline. A screenshot from the processing is shown in Fig. 6. The

drone can be seen to be having a high confidence score while the background information is inserted from the Farneback inference. The link to the video is available at [\(demo-1\)](#)².

C. Throughput Estimation

We estimate the hardware consumption of the algorithm in a pipelined synchronous execution on FPGA using vitis high level synthesis 2021. Leaky CNN and fusion algorithm is executed on the FPGA while Farneback inference is assumed to be fed externally by a conventional processor. Assuming 10000 events are processed in every run, ~ 3 million operations are required per prediction. The synthesis is for Xilinx Virtex UltraScale family FPGA chip xcvu125-CIV-flvd1517-3-e. 8.09×10^6 clock cycles are consumed in generating and fusing one leakyCNN prediction which for a 333 MHz clock promises 41 FPS. 10 DSP(0.83%), 5621 FF(0.39%), 7957 LUT(1.11%) and 1106 BRAM(43.88%) is consumed in the execution highlighting the potential for resource-constrained edge-application. The throughput is currently limited by the number of memory ports and with the incorporation of tiling and event-based hardware design, the latency can be improved further.

IV. DISCUSSION

Our leaky CNN takes inspiration from the correlation type flow estimation model proposed for animal brains [31]. Spatial delay of the motion response is encoded by the leaky accumulator while the differential synaptic kernel in layer 2 adds direction sensitivity. Many recent works observe [32], [33] and also individually map [23] the direction sensitive activation of visual neurons for flies. Similar behaviour was observed in rabbits [22]. The network may be made more noise-tolerant with additional kernels sensitive to intermediate angles and multi-synaptic kernels instead of the differential kernels presented here.

Previous works have also explored multi-modal systems of optical flow computation. Fusion-flowNet [15] fuses SNN and CNN activations while training them as a single network. [34] used both frame and event data as input to a CNN and train them simultaneously. The key difference lies in the fact that we use a bio-inspired neuronal filter to build a processing pipeline for the event camera and the fusion happens in the final stage. This for the first time to the best of our knowledge uses independent pipelines to take their complementary advantages independently without any composite training.

V. CONCLUSION

We proposed a fusion system for frame and event cameras for high-speed optical flow detection. Our network imitates some characteristics of biological neuronal processing and combines complementary speed and accuracy advantages of the two vision systems. The system is validated on the MVSEC dataset and subsequently applied to high-speed drone motion to demonstrate a real-world application. This shows that the accuracy latency trade-off of frame-based processing can be

²<https://www.youtube.com/watch?v=O587-hzIDwM>

mitigated by using input and processing frameworks from other modalities.

VI. ACKNOWLEDGEMENT

This work was supported by IARPA sponsored Microelectronics for AI program

REFERENCES

- [1] J. Romero, M. Loper, and M. J. Black, "Flowcap: 2d human pose from optical flow," in *German conference on pattern recognition*, pp. 412–423, Springer, 2015.
- [2] A. Lim, B. Ramesh, Y. Yang, C. Xiang, Z. Gao, and F. Lin, "Real-time optical flow-based video stabilization for unmanned aerial vehicles," *Journal of Real-Time Image Processing*, vol. 16, no. 6, pp. 1975–1985, 2019.
- [3] J. Goppert, S. Yantek, and I. Hwang, "Invariant kalman filter application to optical flow based visual odometry for uavs," in *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, pp. 99–104, IEEE, 2017.
- [4] M. B. Milde, O. J. Bertrand, R. Benosman, M. Egelhaaf, and E. Chicca, "Bioinspired event-driven collision avoidance algorithm based on optic flow," in *2015 International Conference on Event-based Control, Communication, and Signal Processing (EBCCSP)*, pp. 1–7, IEEE, 2015.
- [5] T. Senst, V. Eiselein, and T. Sikora, "Robust local optical flow for feature tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 9, pp. 1377–1387, 2012.
- [6] Z. Tu, W. Xie, D. Zhang, R. Poppe, R. C. Veltkamp, B. Li, and J. Yuan, "A survey of variational and cnn-based optical flow techniques," *Signal Processing: Image Communication*, vol. 72, pp. 9–24, 2019.
- [7] T.-W. Hui, X. Tang, and C. C. Loy, "A lightweight optical flow cnn—revisiting data fidelity and regularization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 8, pp. 2555–2569, 2020.
- [8] C. Bailer, B. Taetz, and D. Stricker, "Flow fields: Dense correspondence fields for highly accurate large displacement optical flow estimation," in *Proceedings of the IEEE international conference on computer vision*, pp. 4015–4023, 2015.
- [9] T.-W. Hui, X. Tang, and C. C. Loy, "Liteflownet: A lightweight convolutional neural network for optical flow estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8981–8989, 2018.
- [10] T. Kroeger, R. Timofte, D. Dai, and L. Van Gool, "Fast optical flow using dense inverse search," in *European Conference on Computer Vision*, pp. 471–488, Springer, 2016.
- [11] G. Gallego, T. Delbruck, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis, et al., "Event-based vision: A survey," *arXiv preprint arXiv:1904.08405*, 2019.
- [12] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "Ev-flownet: Self-supervised optical flow estimation for event-based cameras," *arXiv preprint arXiv:1802.06898*, 2018.
- [13] C. Ye, A. Mitrokhin, C. Fermüller, J. A. Yorke, and Y. Aloimonos, "Unsupervised learning of dense optical flow, depth and egomotion from sparse event data," *arXiv preprint arXiv:1809.08625*, 2018.
- [14] C. Lee, A. K. Kosta, A. Z. Zhu, K. Chaney, K. Daniilidis, and K. Roy, "Spike-flownet: event-based optical flow estimation with energy-efficient hybrid neural networks," in *European Conference on Computer Vision*, pp. 366–382, Springer, 2020.
- [15] C. Lee, A. K. Kosta, and K. Roy, "Fusion-flownet: Energy-efficient optical flow estimation using sensor fusion and deep fused spiking-analog network architectures," *arXiv preprint arXiv:2103.10592*, 2021.
- [16] R. Benosman, C. Clercq, X. Lagorce, S.-H. Ieng, and C. Bartolozzi, "Event-based visual flow," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 2, pp. 407–417, 2013.
- [17] M. Almatrafi, R. Baldwin, K. Aizawa, and K. Hirakawa, "Distance surface for event-based optical flow," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 7, pp. 1547–1556, 2020.
- [18] M. Liu and T. Delbruck, "Adaptive time-slice block-matching optical flow algorithm for dynamic vision sensors," 2018.
- [19] F. Paredes-Vallés, K. Y. Scheper, and G. C. de Croon, "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*, vol. 42, no. 8, pp. 2051–2064, 2019.
- [20] G. Haessig, A. Cassidy, R. Alvarez, R. Benosman, and G. Orchard, "Spiking optical flow for event-based sensors using ibm's trueneurosynaptic system," *IEEE transactions on biomedical circuits and systems*, vol. 12, no. 4, pp. 860–870, 2018.
- [21] G. Orchard, R. Benosman, R. Etienne-Cummings, and N. V. Thakor, "A spiking neural network architecture for visual motion estimation," in *2013 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pp. 298–301, IEEE, 2013.
- [22] H. Barlow and W. R. Levick, "The mechanism of directionally selective units in rabbit's retina," *The Journal of physiology*, vol. 178, no. 3, pp. 477–504, 1965.
- [23] B. Schnell, M. Joesch, F. Forstner, S. V. Raghu, H. Otsuna, K. Ito, A. Borst, and D. F. Reiff, "Processing of horizontal optic flow in three visual interneurons of the drosophila brain," *Journal of neurophysiology*, vol. 103, no. 3, pp. 1646–1657, 2010.
- [24] A. Z. Zhu, D. Thakur, T. Özarslan, B. Pfrommer, V. Kumar, and K. Daniilidis, "The multivehicle stereo event camera dataset: An event camera dataset for 3d perception," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2032–2039, 2018.
- [25] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, "FlowNet: Learning optical flow with convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, pp. 2758–2766, 2015.
- [26] G. Farneback, "Two-frame motion estimation based on polynomial expansion," in *Scandinavian conference on Image analysis*, pp. 363–370, Springer, 2003.
- [27] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.
- [28] J. de Boer and M. Kalksma, "Choosing between optical flow algorithms for uav position change measurement," *12th SC@ RUG 2014-2015*, p. 69, 2015.
- [29] S. Meister, J. Hur, and S. Roth, "Unflow: Unsupervised learning of optical flow with a bidirectional census loss," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.
- [30] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "Unsupervised event-based learning of optical flow, depth, and egomotion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 989–997, 2019.
- [31] A. Borst and M. Egelhaaf, "Principles of visual motion detection," *Trends in neurosciences*, vol. 12, no. 8, pp. 297–306, 1989.
- [32] M. Joesch, J. Plett, A. Borst, and D. F. Reiff, "Response properties of motion-sensitive visual interneurons in the lobula plate of drosophila melanogaster," *Current Biology*, vol. 18, no. 5, pp. 368–374, 2008.
- [33] K. D. Longden and H. G. Krapp, "State-dependent performance of optic-flow processing interneurons," *Journal of neurophysiology*, vol. 102, no. 6, pp. 3606–3618, 2009.
- [34] Z. Jiang, P. Xia, K. Huang, W. Stechele, G. Chen, Z. Bing, and A. Knoll, "Mixed frame/event-driven fast pedestrian detection," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8332–8338, IEEE, 2019.