3D-FlowNet: Event-based optical flow estimation with 3D representation

Haixin Sun, Minh-Quan Dao, Vincent Frémont

3D-FlowNet: Event-based optical flow estimation with 3D representation

Haixin SUN, Minh-Quan DAO, Vincent FREMONT

***Abstract*— Event-based cameras can overpass frame-based cameras limitations for important tasks such as high-speed motion detection during self-driving cars navigation in low illumination conditions. The event cameras' high temporal resolution and high dynamic range, allow them to work in fast motion and extreme light scenarios. However, conventional computer vision methods, such as Deep Neural Networks, are not well adapted to work with event data as they are asynchronous and discrete. Moreover, the traditional 2D-encoding representation methods for event data, sacrifice the time resolution. In this paper, we first improve the 2D-encoding representation by expanding it into three dimensions to better preserve the temporal distribution of the events. We then propose 3D-FlowNet, a novel network architecture that can process the 3D input representation and output optical flow estimations according to the new encoding methods. A self-supervised training strategy is adopted to compensate the lack of labeled datasets for the event-based camera. Finally, the proposed network is trained and evaluated with the Multi-Vehicle Stereo Event Camera (MVSEC) dataset. The results show that our 3D-FlowNet outperforms state-of-the-art approaches with less training epoch (30 compared to 100 of Spike-FlowNet). The code is released in https://github.com/adosum/3D-FlowNet.**

I. INTRODUCTION

An Autonomous Vehicle (AV) requires an accurate perception of its surrounding environment to reliably and safely operate. The perception system of an AV can transform raw sensory data into semantic information [1], and frame-based monocular cameras are one of the most commonly used sensors for this purpose. They synchronously transmit raw images, frame by frame, at a fixed rate. This feature as the major drawbacks of low temporal resolution, redundant information and low dynamic range. Few years ago, event-based cameras, a bio-inspired technology of silicon retinas, have been proposed to overcome those limitations and to solve both classical and new computer vision tasks [2], [3]. An event-based camera can have a dynamic range of 130 dB and a minimum of 3 μs latency. Those advantages allow the event-based camera to work in extreme scenarios with low light conditions and fast motions. Typically, event-based cameras are used as sensing modalities on Unmanned aerial vehicle (UAV) [4], mobile robots [5] or wearable electronics [6], where operations are under unrealistic lighting conditions and sensitive to the temporal resolution. The main applications for event-based cameras are object stracking [5], surveillance and monitoring [7], and optical flow estimation [8], [9]. Nowadays, more and more researchers focus on using the event-based cameras for autonomous driving. [10] proposed a method that can predict the vehicle's steering angle according to the event data, and [11] proposed a dataset that contains event data along with the vehicle state.

Authors are with Nantes Université, École Centrale de Nantes and CNRS LS2N, 44300 Nantes, France. first-name.last-name@ec-nantes.fr

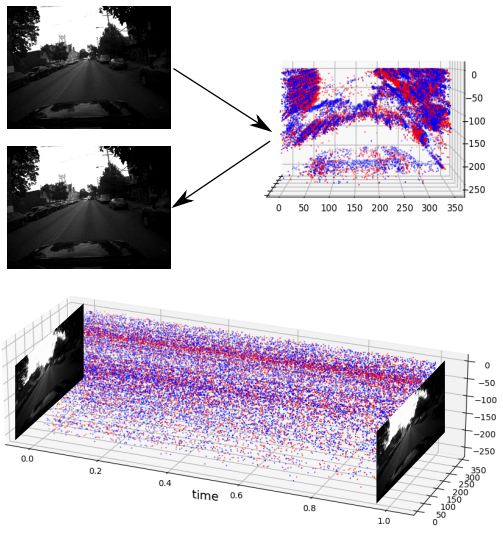


Fig. 1. Visualization of the event data between two gray scale image.

Event-based cameras are asynchronous devices that detect changes in log brightness intensity. When the variation of the brightness of a pixel reaches the threshold, the camera generates an event. The event is usually in the format of a tuple, $e = (x, y, t, p)$, where (x, y) is the pixel's position, t is the precise timestamp of the event which is accurate up to microseconds, and the polarity p of the change that indicates whether the pixel became brighter or darker. Fig. 1 shows the visualization of the event data between two frame-based gray-scale images. The positive events are shown in red, and the negative events are in blue. Between two consecutive images, there is a quasi-continuous stream of events that represents all the brightness change between the two images. The event-based camera's asynchronous nature and tracking in the log image space offer several advantages over traditional frame-based cameras, including extremely low latency for detecting high-speed objects, a very high dynamic range for the poor light conditions, and significantly lower power consumption.

The cameras' unique output, on the other hand, presents new challenges in algorithm developments. Indeed, the events are transmitted asynchronously and lacks the pixel's absolute value and spatial neighborhood. The algorithms for traditional frame images such as optical flow or object detection are no longer valid. As a result, a significant

research effort has been made to develop new algorithms for event-based cameras to solve these traditional vision problems.

Within Deep Learning area, there exist several works that train a neural networks to estimate the optical flow in a self-supervised manner. Zhu et al. [9] accumulate the events into the image-like frames and calculate the optical flow using an encoder-decoder network. Their encoding method loses the temporal information because they summarize the events stream into a four-channel image. Lee et al. [12] try to solve this problem by proposing a deep hybrid neural network architecture called Spike-FlowNet. The use of the Spiking Neural Network allows the approach to process the data asynchronously. So it can best preserve the properties of the event data. However, the training of the Spiking Neural Network is quite slow and unstable. So, although the neural networks avoid the complex problem of modeling and algorithm developments, the encoding representation for the event data and the neural network's design still need to be improved.

The main contribution of this paper is to propose a new encoding method and the corresponding neural network architecture to process an event data stream. We proposed a 3D encoding representation that can better preserve the temporal nature of the event data. We also present the 3D-FlowNet, a novel neural network architecture that can process the 3D input and generate optical flow estimations. Finally, We train and evaluate the proposed 3D-FlowNet using the Multi-Vehicle Stereo Event Camera (MVSEC) dataset [13]. The results show that our approach outperforms current state-of-the-art methods, we achieve 13% improvement compared to the Spike-FlowNet[12], and 32% compared to the EV-FlowNet[9].

The paper is structured as follows: In Section II, we discuss the related work. In section III, we present the methodology, covering the encoding method for the event data and the corresponding neural network architecture. This section also discusses the self-supervised training strategy. In section IV, we present the experimental results, including training details and the evaluation metrics. We also discuss the comparison results with state-of-the-art approaches.

II. RELATED WORKS

Due to the properties of the event-based camera, there has been a lot of interest in developing algorithms that take advantage of them, and optical flow estimation is one of the addressed topics. Benosman et al. [14] fit a plane to the events in spatial-temporal spaces and then estimate the optical flow. Bardow et al. [15] formulate the flow estimation as a convex optimization problem that solves for the image intensity and flow jointly. Almatrafi et al. [8] calculate the spatial and temporal gradients on the frame image and events data, respectively, and then estimate the optical flow by solving the classical optical flow equation.

Besides the traditional optical flow algorithms for the event camera, there are also several model-free methods that use a deep neural networks to predict the optical flow. Zhu et al. [9] accumulate the event into the image like frames and use an encoder-decoder network architecture to estimate the optical flow. The event data are then encoded into a four-channel image representing: Positive events counting, negative events counting, latest timestamp of positive events, and latest timestamp of negative events. This encoding method loses the temporal information because the older timestamp are filtered out. Lee et al. [12] try to solve this problem by proposing a deep hybrid neural network architecture called Spike-FlowNet, a hybrid structure between regular Neural Networks (NN) and Spiking Neural Networks (SNN). Due to the use of the SNN, the events are processed asynchronously to preserve the temporal information of the event data. However, the training of the Spike-FlowNet is relatively slow and unstable. Because the activation function of SNN is not continuous, the backpropagation algorithm can not be directly used to train the SNN.

For the networks' training, several works focuses on self-supervised training for the optical flow prediction because of the lack of labeled event-based datasets. Yu et al. [16] proposed a network that can learn optical flow from brightness constancy and motion smoothness. Based on that, Meister et al. [17] improve the quality of the flow by applying a bidirectional census loss to achieve better performance with less training time. [9], [12] adopt this self-supervised strategy for event-based camera and achieve similar performances.

III. PROPOSED APPROACH

In this section, we explain our approach in details. In III-A, we describe our event encoding method, which encodes a group of event measurements into an 3D temporal-spatial event image. In III-B, we describe the architecture of our network, which uses the 3D convolutions to process the spatial-temporal measurements and output the pixel-wise optical flow. Finally, in III-C, we describe the training strategy and the self-supervised loss is also discussed.

A. Event Data Encoding Method

The event-based camera records the log intensity change of each pixel of the artificial retina, and generates an event whenever the log intensity changes over the threshold θ:

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

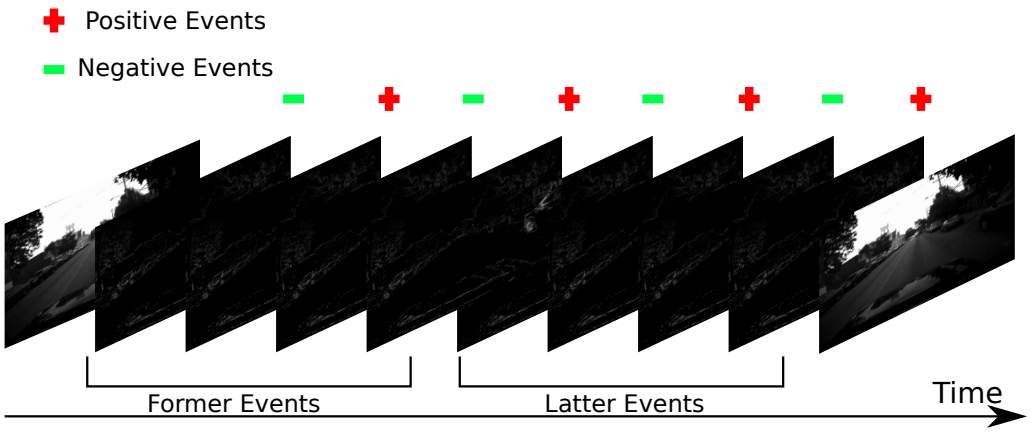
The event measurement is in the format of tuple which consists of location of the pixel, timestamp of the event and polarity of the change:

$$e = (x, y, t, p) \tag{2}$$

Because the events are transmitted asynchronously, they cannot be immediately fed into standard convolutional neural network layers. It is therefore important to keep the necessary information while generate the encoding representation from the event stream.

Several prior works have proposed different methods that transform the event output into a synchronous image-like representation. In EV-FlowNet [9], only the latest pixel-wise timestamps and the event counts are used to encode the event

(a) Example of an event image and a gray scale image



(b) four channels for event data

Fig. 2. Visualization of our event encoding representation. (a) is one slice of the event image $I_{slice} = (1, 1, H, W)$, and the brighter represents the more recent timestamp value. (b) is an example of the event representation where $D = 8$.

representation. However, fast motions and dense scenarios can enormously overlap per-pixel timestamp information. In [18], [19], the time domain is discretized to preserve the temporal distributions. To improve the resolution and the temporal domain beyond the number of bins, the authors insert events into this volume using a linearly weighted accumulation similar to bilinear interpolation. However, the number of input channels increases significantly as the time dimensions are finely discretized, further increasing the computation time for encoding and forward propagation.

Considering all the methods discussed before, we propose in this work, a novel input representation that can better exploit the information in the event data with less computation complexity. Given a set of N input events $E_N = (x_i, y_i, t_i, p_i), i \in [1, N]$, and a time depth D to discretize the time dimension of event data, we accumulate each group of event into images as follows:

$$
\begin{aligned}
t_{norm} &= (t - t_0)/(t_N - t_1) * (D - 1) \\
I(x, y, t, p) &= \sum_i \delta(p - p_i) k_b(x - x_i) k_b(y - y_i) k_b(t - t_{norm}) \\
k_b(a) &= max(0, 1 - |a|)
\end{aligned}
\tag{3}
$$

Here, (x, y) denotes the position of the event, p is the polarity of the event, and δ is the Kronecker delta operator. $k_b(\cdot)$ denotes bi-linear sampling kernel. The generated event image I is a $(2, D, H, W)$ matrix, where the number 2 represents the positive and negative polarity, D is the discretized time depth, and (H, W) are respectively the height and width of the image. Then we split the event image into former and latter groups through the time dimension and obtained a new event image with the shape of $(4, \frac{D}{2}, H, W)$. Here the number 4 represents the four channels: Former positive events, former negative events, latter positive events, latter negative events. Fig. 2 shows the proposed input representation. Fig. 2. (a) is the visualization of the event image and the relative grayscale image, left is one slice of the event image, and the brighter represents the more recent timestamp value. Fig. 2. (b) is an example of the event representation where $D = 8$.

B. Proposed Network Architecture

With the input representation $I_{4,D/2,H,W}$ discussed in section III-A, we propose the 3D-FlowNet architecture to predict the optical flow values. The 3D-FlowNet's network adopts an encoder-decoder architecture, containing four encoder layers, two residual blocks, and four decoder layers

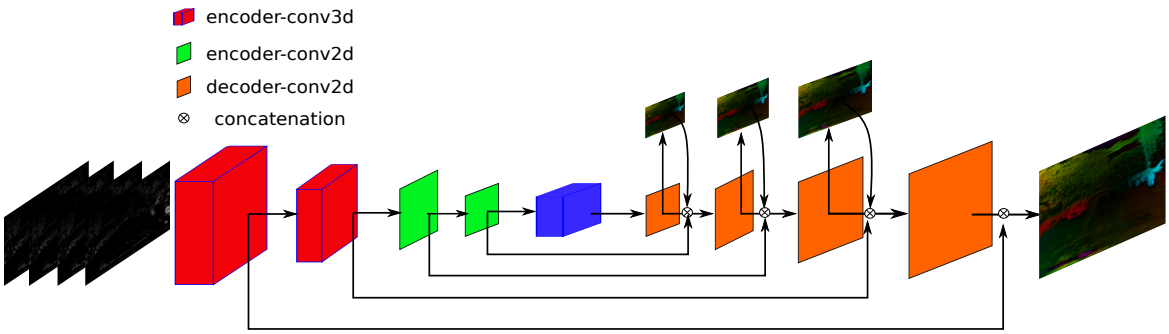


Fig. 3. Network structure of the 3D-FlowNet.

as shown in Fig. 3. First, the input event image is passed through two 3D-decoders. The 3D-decoders down-sample the time dimension $d/2$ to 1, and compress the 3D input into 2D $((4, D/2, H, W) \rightarrow (4, 1, H, W) \rightarrow (4, H, W))$. Then the resulting activation are passed through two 2D-decoders, two residual blocks, and four 2D-decoders. For each decoder, the activation is up-sampled using the 2D transposed convolution and then convolved, to obtain the final optical flow estimation.

There is a skip connection from each encoder to the corresponding decoder. For the skip connection between 2D-encoder and 2D-decoder, the activation of the encoder is directly concatenated with the intermediate optical flow value and the activation of decoder. For the skip connection between 3D-encoder and 2D-decoder, the 3D activation $(C \times D \times W \times H)$ is flattened into 2D tensor $((C * D) \times W \times H)$ first, then it can be concatenated with the activation of the decoder and the intermediate optical flow. The predicted optical flows are then used together with the grayscale image for the loss calculation.

C. Self-Supervised Loss

The event-based camera is a sensor that can produce synchronous grayscale images and asynchronous event data streams simultaneously. Compared to frame-based camera datasets, the number of available event-based camera datasets with annotated labels suitable for optical flow estimation is relatively small. As a result, for training our Spike-FlowNet, we used a self-supervised learning method that uses proxy labels from the recorded grayscale images [16], [17].

The total loss consists of a smoothness loss (L_{smooth}) and a photometric reconstruction loss (L_{photo}) [16]. The network needs a pair of grayscale images $(I_t, I_{t+\Delta t})$ to calculate the photometric loss, as well as the event data in the time window $(t, t + \Delta t)$. The second grayscale image is warped to the first grayscale image using the network's predicted optical flow. The photometric loss (L_{photo}) is used to minimize the difference between the first grayscale image and the inversely warped second grayscale image. This loss is based on the photometric consistency assumption, which states that a pixel value from the first image will be similar to the second frame warped by the predicted optical flow. The photometric loss can be written as:

$$L_{loss}(u, v, I_t, I_{t+\Delta t}) = \sum_{x,y} \rho(I_t(x, y) - I_{t+dt}(x + u(x, y), y + v(x, y))) \quad (4)$$

Then, the smoothness loss is adopted to improve the spatial consistency of neighboring optical flow. It is calculated as:

$$L_{smooth} = \sum_i \sum_j (||u_{i,j} - u_{i+1,j}|| + ||(u_{i,j} - u_{i,j+1}|| + ||(v_{i,j} - v_{i+1,j}|| + ||(v_{i,j} - v_{i,j+1}||) \quad (5)$$

The total loss for the training is computed as the weighted sum of the photometric and smoothness loss:

$$L_{total} = L_{photo} + \lambda L_{smooth} \quad (6)$$

where λ is the weight factor.

IV. Experiments

A. Dataset and Implementation Details

The MVSEC dataset [13] is used in this paper for training and evaluating the optical flow predictions. The MVSEC dataset contains stereo event-based camera data, including flying, driving, and handheld scenes. Moreover, the dataset provides ground truth poses and depths maps for each event-based camera, and the ground truth optical flow can be generated accordingly. To offer fair comparisons with prior works [12], [9], only the outdoor day2 sequence is used for training.

During the training, the input is centrally cropped to 256×256 size. The ADAM optimizer is used, and the initial learning rate of 1e-4. The model is trained for 30 epochs with a batch size of 16, while [12] takes 100 epochs. This is because the training of the ANN is faster and more stable than the SNN one.

B. Results

Here, the Average End-point Error (AEE) is used to evaluate the optical flow result, and it is defined as:

$$AEE = \frac{1}{n} \sum_n \|(u, v)_{pred} - (u, v)_{gt}\|^2 \quad (7)$$

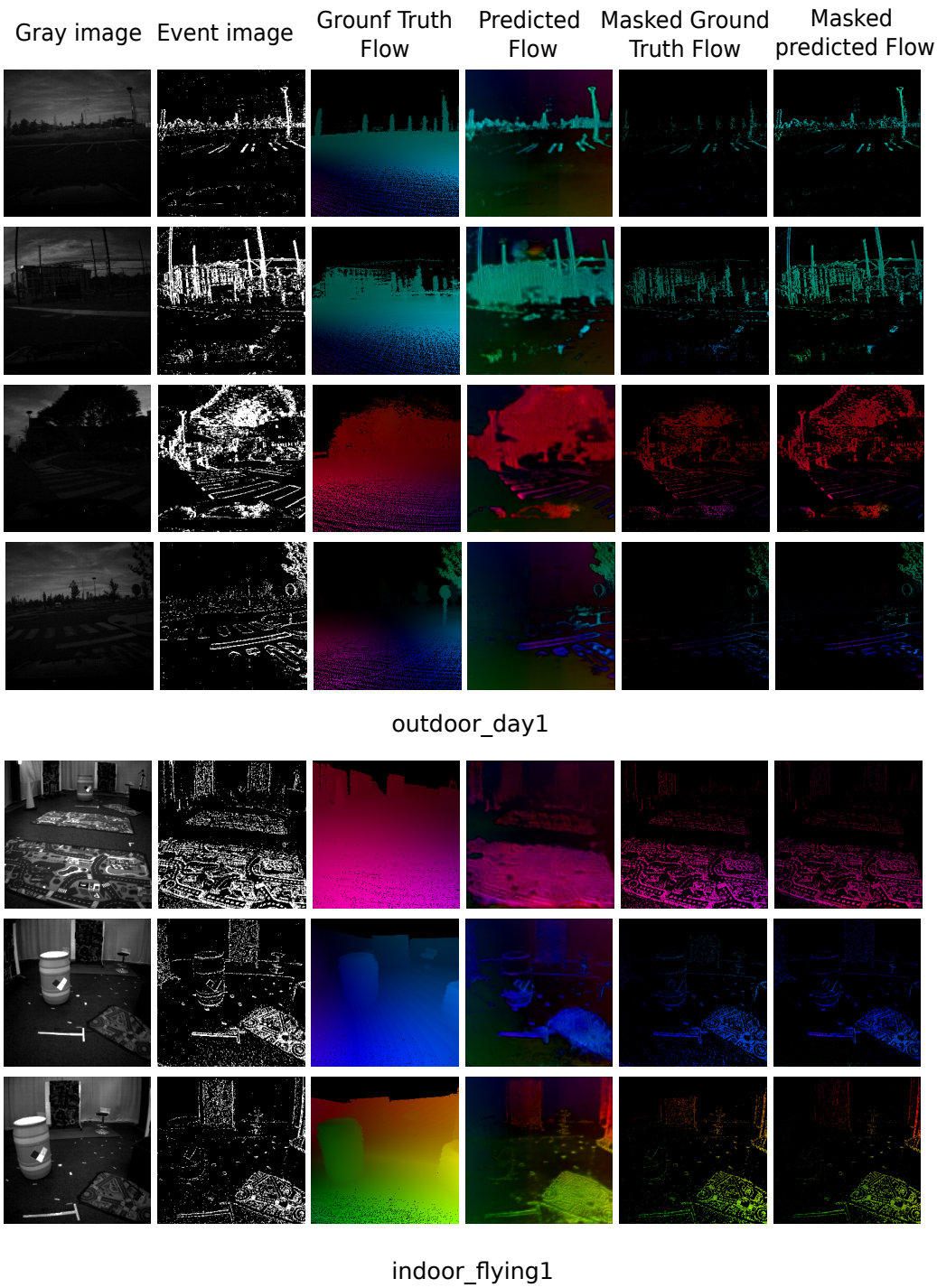


Fig. 4. Visualization of the optical flow estimation.

Where n is the number of the active pixel in the event image, $(u, v)_{pred}$ is the predicted optical flow and $(u, v)_{gt}$ is the groundtruth. We also count the outliers that corresponds to the percentage of points with AEE exceeding three pixels. For each sequence, the AEE is calculated in pixels, and the %Outlier is defined as the percentage of points with $AEE <$ 3 pix. During the testing, the optical flow is also estimated on the centrally cropped 256×256 event images. The sequences of indoor flying 1,2,3 and outdoor day 1 are used. We use all events from the indoor flying sequences and take events within 800 gray scale frames for the outdoor day1 sequence similar to [12].

TABLE I

QUANTITATIVE ASSESSMENT OF OUR APPROACH COMPARED TO EV-FLOWNET AND SPIKE-FLOWNET

	outdoor day1		indoor flying1		indoor flying2		indoor flying3	
	AEE ↓	Outlier ↓	AEE ↓	Outlier ↓	AEE ↓	Outlier ↓	AEE ↓	Outlier ↓
EV-FlowNet [9]	0.49	0.2	1.03	2.2	1.72	15.1	1.53	11.9
Spike-FlowNet [12]	**0.49**	-	0.84	-	1.28	-	1.11	-
Ours	0.51	**0.1**	**0.7**	**0.1**	**1.10**	**0.2**	**0.91**	**0.1**

Table I show the results of the AEE evaluation in comparison to previous event-based camera-based optical flow estimation approaches. Our approach achieves better performances than the others in all the indoor_flying sequences. Our AEE performance is similar to the others in the outdoor_day1 sequence, but we obtain fewer outliers. Fig. 4 shows the qualitative results of our approach. The grayscale, event image, ground truth flow, and corresponding predicted flow images are displayed in this figure. We mask out the optical flow at points where the event data are absent. The masked optical flow is used here because event-based cameras detect the brightness change at pixels. Low texture regions, such as flat surfaces, produce very few events due to fewer brightness changes, resulting in few optical flow predictions in the corresponding areas. Overall, the results show that 3D-FlowNet can predict optical flow accurately in both indoor and outdoor day1 sequences. This proves that the proposed 3D-FlowNet generalizes well to a variety of environments.

V. CONCLUSIONS

In this work, we propose 3D-FlowNet, a deep neural network for optical flow estimations using event-based camera data. We improved the encoding methods for the event data and self-training strategy for the network. The results show that our approach can generate more accurate (13%-32%) optical flow estimations (u, v). For future work, we hope to combine frame-based cameras with event-based cameras to achieve better and more robust performance in various scenarios.

REFERENCES

[1] E. Arnold, O. Y. Al-Jarrah, M. Dianati, S. Fallah, D. Oxtoby, and A. Mouzakitis, "A survey on 3d object detection methods for autonomous driving applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3782–3795, 2019.

[2] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128× 128 120 db 15 μs latency asynchronous temporal contrast vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, 2008.

[3] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck, "A 240 × 180 130 db 3 µs latency global shutter spatiotemporal vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 10, pp. 2333–2341, 2014.

[4] N. Sanket, C. Parameshwara, C. Singh, A. V. Kuruttukulam, C. Fermüller, D. Scaramuzza, and Y. Aloimonos, "Evdodgenet: Deep dynamic obstacle dodging with event cameras," in *IEEE International Conference on Robotics and Automation (ICRA)*, 05 2020, pp. 10 651–10 657.

[5] T. Delbruck and M. Lang, "Robotic goalie with 3 ms reaction time at 4% cpu load using event-based dynamic vision sensor," *Frontiers in Neuroscience*, vol. 7, p. 223, 2013. [Online]. Available: https://www.frontiersin.org/article/10.3389/fnins.2013.00223

[6] T. Delbruck, "Neuromorophic vision sensing and processing," in *2016 46th European Solid-State Device Research Conference (ESSDERC)*, 2016, pp. 7–14.

[7] M. Litzenberger, B. Kohn, A. Belbachir, N. Donath, G. Gritsch, H. Garn, C. Posch, and S. Schraml, "Estimation of vehicle speed based on asynchronous data from a silicon retina optical sensor," in *2006 IEEE Intelligent Transportation Systems Conference*, 2006, pp. 653–658.

[8] M. Almatrafi and K. Hirakawa, "Davis camera optical flow," *IEEE Transactions on Computational Imaging*, vol. 6, pp. 396–407, 2020.

[9] A. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "Ev-flownet: Self-supervised optical flow estimation for event-based cameras," in *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, June 2018.

[10] A. Maqueda, A. Loquercio, G. Gallego, N. Garcia, and D. Scaramuzza, "Event-based vision meets deep learning on steering prediction for self-driving cars," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 06 2018, pp. 5419–5427.

[11] Y. Hu, J. Binas, D. Neil, S.-C. Liu, and Delbruck, "Ddd20 end-to-end event camera driving dataset: Fusing frames and events with deep learning for improved steering prediction," in *IEEE International Conference on Intelligent Transportation Systems*, 11 2020.

[12] C. Lee, A. 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*. Springer, 2020, pp. 366–382.

[13] A. Z. Zhu, D. Thakur, T. Özaslan, 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.

[14] R. Benosman, C. Clercq, X. Lagorce, S.-H. Ieng, and C. Bartolozzi, "Event-based visual flow," *IEEE Transactions on Neural Networks*, vol. pp, p. 1, 11 2013.

[15] P. Bardow, A. J. Davison, and S. Leutenegger, "Simultaneous optical flow and intensity estimation from an event camera," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 884–892.

[16] J. J. Yu, A. W. Harley, and K. G. Derpanis, "Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness," in *Computer Vision - ECCV 2016 Workshops, Part 3*, 2016.

[17] S. Meister, J. Hur, and S. Roth, "UnFlow: Unsupervised learning of optical flow with a bidirectional census loss," in *AAAI*, New Orleans, Louisiana, Feb. 2018.

[18] A. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "Unsupervised event-based learning of optical flow, depth, and egomotion," 12 2018.

[19] "How to train your event camera neural network," Mar 2020. [Online]. Available: https://deepai.org/publication/how-to-train-your-event-camera-neural-network