Lightweight Event-based Optical Flow Estimation via Iterative Deblurring

Wu, Yilun; Paredes-Vallés, Federico; De Croon, Guido C.H.E.

DOI
10.1109/ICRA57147.2024.10610353

Publication date
2024

Document Version
Final published version

Published in
2024 IEEE International Conference on Robotics and Automation, ICRA 2024

2024 IEEE International Conference on Robotics and Automation (ICRA)
May 13-17, 2024. Yokohama, Japan

Lightweight Event-based Optical Flow Estimation via Iterative Deblurring

Yilun Wu[1], Federico Paredes-Vallés[1] and Guido C. H. E. de Croon[1]

***Abstract*— Inspired by frame-based methods, state-of-the-art event-based optical flow networks rely on the explicit construction of correlation volumes, which are expensive to compute and store, rendering them unsuitable for robotic applications with limited compute and energy budget. Moreover, correlation volumes scale poorly with resolution, prohibiting them from estimating high-resolution flow. We observe that the spatiotemporally continuous traces of events provide a natural search direction for seeking pixel correspondences, obviating the need to rely on gradients of explicit correlation volumes as such search directions. We introduce *IDNet* (Iterative Deblurring Network), a lightweight yet high-performing event-based optical flow network directly estimating flow from event traces without using correlation volumes. We further propose two iterative update schemes: "*ID*" which iterates over the same batch of events, and "*TID*" which iterates over time with streaming events in an online fashion. Our top-performing model (*ID*) sets a new state of the art on DSEC benchmark. Meanwhile, the base model (*TID*) is competitive with prior arts while using 80% fewer parameters, consuming 20x less memory footprint and running 40% faster on the NVidia Jetson Xavier NX. Furthermore, the *TID* scheme is even more efficient offering an additional 5x faster inference speed and 8 ms ultra-low latency at the cost of only a 9% performance drop, making it the only model among current literature capable of real-time operation while maintaining decent performance.**

Code: https://github.com/tudelft/idnet.

I. Introduction

Optical flow estimation, i.e. estimating pixel motion over time on the image plane, is both a central and challenging task in computer vision. As it encodes a primitive form of motion information, optical flow underpins many robotic navigation applications [1–3]. Compared to frame-based cameras, event cameras capture asynchronous brightness changes in continuous time, offering high dynamic range measurements with minimal motion blur at high speeds and low lighting conditions while only consuming milliwatts of power [4]. All these advantages make it the ideal sensor candidate for resource-constrained agile robots such as micro aerial/ground vehicles (MAVs/MGVs) [5, 6].

Learning-based methods have made marked progress in optical flow estimation by incorporating more apt inductive biases [7–9]. Notably, the dense all-pair correlation volumes introduced in [9] effectively estimate high-quality flow for large motions and have been widely adopted in later research [10–12]. Recent methods for event-based optical flow [13–15] emulate these approaches, treating consecutive event frames as discrete images to determine flow.

[1]All authors are with the Micro Air Vehicle Laboratory, Faculty of Aerospace Engineering, Delft University of Technology, Delft, The Netherlands. Correspondences: Y.Wu-9@tudelft.nl

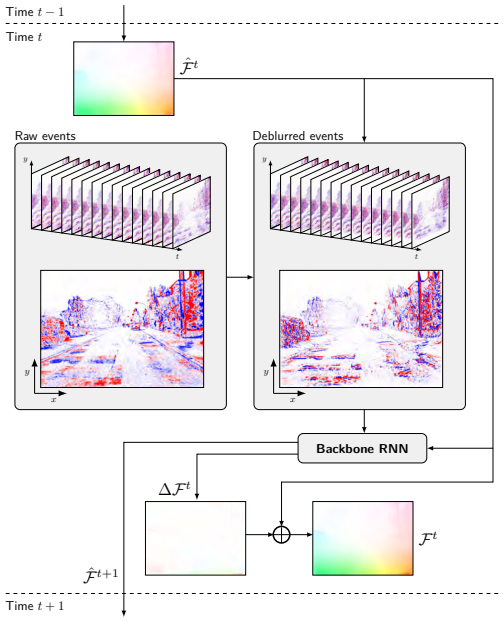

Fig. 1. Illustration of the *IDNet* pipeline for temporal iterative deblurring (i.e. *TID* scheme). Raw events are first deblurred according to the initial coarse optical flow estimate $\hat{\mathcal{F}}^t$ before being processed by the backbone RNN. The RNN extracts the residual motion from the deblurred events and outputs the residual flow $\Delta\mathcal{F}^t$ which is added to the initial estimate $\hat{\mathcal{F}}^t$ to arrive at the final flow estimation $\mathcal{F}^t$. The RNN additionally proposes a coarse estimate $\hat{\mathcal{F}}^{t+1}$ for the next timestep under continuous operation.

However, three major drawbacks of correlation volumes rise when applied to event data. First, constructing them requires accumulation of events which causes high latency. Second, computing and storing the correlation volume is expensive, restricting deployment on memory/compute-limited systems. Finally, as noted in [16, 17], the correlation volume scales poorly to higher input resolutions, limiting the algorithm's ability to process higher resolution events or deliver fine details in optical flow estimates.

We observe that, compared to estimating flow from two images, event-based optical flow benefits from the continuous recording of motion over time and space. This allows tracing the continuous trajectory to estimate flow, whereas frame-based methods have to search for such "trajectory" by estimating gradients in the correlation volume. In other words, while correlation volumes are instrumental for methods operating over frames to propose an update direction, such direction is readily evident in the form of blur (i.e. the continuous spatiotemporal trajectory) in the raw events.

From this observation, we propose *IDNet* (Iterative Deblurring Network), an event-based algorithm for iterative optical flow estimation without correlation volumes. At its core,

IDNet processes event bins sequentially via a backbone RNN where flow is estimated from the traces (i.e. blur) of events. Additionally, we adopt an iterative update scheme to improve estimation quality under rapid motion, employing motion compensation (a.k.a. deblurring) [18]. Each iteration deblurs events based on prior flow estimates, then processes them to refine the flow, akin to frame-based warping techniques.

We propose two iterative update schemes: *ID* (iterative deblurring), illustrated in Fig. 2, which iterates over the same batch of events to achieve the best performance, and *TID* (temporal iterative deblurring), shown in Fig. 1, which iterates over events in time for drastically faster processing.

On public benchmarks, our methods achieve comparable results with state-of-the-art methods that use correlation volumes, while using much fewer parameters and memory. Without correlation volumes, our method can estimate optical flow from higher-resolution feature maps, resulting in significant improvements over prior art. Additionally, our *TID* scheme is highly efficient, reaching close-to-state-of-the-art performance while incurring minimal latency.

II. Related Works

A. Event-based Optical Flow

Early event-based optical flow algorithms such as [19] adapt frame-based methods such as KLT [20] to the event-based domain or utilize hand-crafted heuristics to fit event data [21, 22]. The approach in [23] jointly optimizes image brightness and flow by exploiting brightness constancy.

The availability of event simulators [15, 24, 25] and large-scale datasets [26–28] enable learning-based methods to achieve superior performance over model-based ones. EV-Flownet [29] introduces an event representation which split events proportionally to the nearby temporal bins and a U-Net architecture for processing the event representation. ECN [30] jointly estimates optical flow, depth and egomotion. Both methods construct multi-level feature pyramids. E-RAFT [13], an event-based version of RAFT [9] is the first to introduce correlation volumes into the event domain. The method computes the 4D all-pair correlation volume between two neighboring event representations in time, which is then iteratively processed to arrive at the final optical flow estimate. TMA [14] improves upon [13] by constructing multiple correlation volumes at a finer temporal scale.

While methods [13, 14] deliver high performance, the computation and storage of correlation volumes, highlighted in [16, 17], have prohibitive time and storage complexities that worsen with increasing input resolution. For events of resolution H by W, the complexity is $\mathcal{O}((\mathrm{H} \times \mathrm{W})^2)$. Such storage demands limit deployment on constrained memory platforms and prevent scaling to higher resolutions.

B. Iterative Refinement for Optical Flow

The idea of iterative refinement for optical flow can be traced back to the early works of iterative KLT tracking [20] where a Taylor series expansion is applied to linearize the problem and iteratively solve for the residual error. Learning-based methods incorporate this principle as well. Most works [7,31,32] perform coarse-to-fine refinement along the feature pyramid. In addition, other works such as [9, 33–35] stack multiple modules in series to iteratively estimate the residual flow. While some warps images [34], other methods such as [9, 33, 35] resample correlation volumes.

Recently, iterative refinement for optical flow has also been applied to event-based data. Methods such as [36] and [37] use pyramid scale to perform coarse-to-fine estimation similar to PWC-Net [7] but do not perform flow refinement through iterations. The update scheme from [33] is adopted by [38], which processes the correlation volume between warped event slices through time to arrive at a residual flow estimate. E-RAFT [13] instead directly processes the correlation volume through an RNN and extracts the final flow estimate from the final state of the RNN. All aforementioned methods rely on the explicit construction of correlation volumes as a basis for their refinement schemes.

C. Continuous Operation

Since optical flow is temporally highly correlated as the motion in a real-world environment is mostly smooth and continuous, it makes sense to leverage the prior estimate from the past to better predict the present optical flow. This temporal recurrency is introduced in [39] and used in RAFT [9] as a "warm-starting" strategy when applied to video data. RAFT [9], as well as E-RAFT [13], directly transports the previous flow field to the current timestep by the flow itself under the assumption of constant linear motion.

III. Method

A. Event Representation

As in [13, 40], we utilize the discretized event voxel grid as the event representation. Specifically, we create a representation $\mathcal{E} \in \mathrm{R}^{\mathrm{B}\times\mathrm{H}\times\mathrm{W}}$ with B bins out of the event stream $\{e_i = (t_i, x_i, y_i, p_i)\}$ of interest as follows:

$$t_i^* = (B-1)(t_i - t_{\text{begin}})/(t_{\text{end}} - t_{\text{begin}}) \quad (1)$$

$$\mathcal{E}(b, x, y) = \sum_{i|x_i=x, y_i=y} p_i \max(0, 1 - |b - t_i^*|) \quad (2)$$

where b is the integer bin index and H, W correspond to the height and width resolution of the event stream respectively.

B. Motion Compensation

We utilize the principle of motion compensation [18] as a core step in our processing. Specifically, we use the flow estimate $\mathcal{F}$ to deblur the events to a reference time t_{ref} by changing its pixel coordinate $\mathbf{x}_i = (x_i, y_i)$ as follows:

$$\mathbf{x'}_i = \mathbf{x}_i + (t_{\text{ref}} - t_i)\mathcal{F}(x_i) \quad (3)$$

Assuming brightness constancy and linear motion, the ideal flow should neutralize motion in events, resulting in a static representation without any motion. The effect of such motion compensation is depicted in Fig. 1. By projecting all bins of event representation $\mathcal{E}$ onto the same 2D plane, deblurred events show reduced motion, making moving objects appear static with sharp edges. Refer to the supplementary video for a dynamic illustration.

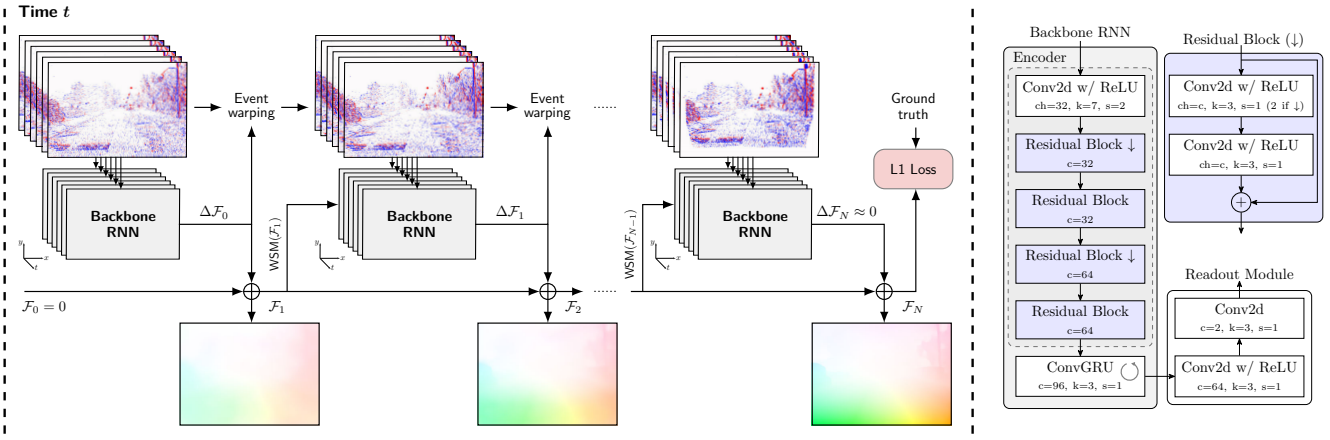

Fig. 2. Overall pipeline of *IDNet* with iterative deblurring scheme (i.e. *ID* scheme). Starting with a zero flow, each iteration deblurs events using prior flow. The deblurred event bins are fed into the RNN sequentially one bin at a time. A residual flow is estimated and used for deblurring in the next iteration. The final flow accumulates all residual flows throughout iterations. An L1 loss is applied between the final flow estimate and ground truth. The detailed network structure is shown on the right. The parameters ch, k, and s of the Conv2d layer refer to the output channel count, kernel size, and stride.

Algorithm 1: Iterative deblurring (*ID*).

input : Event bins $\mathcal{E}$ containing events of duration T
Number of deblurring iterations N
output: Optical Flow Prediction $\mathcal{F}$ for the motion during the time window of events

$\mathcal{F}_0 \leftarrow \mathbf{0}$, $\mathcal{M}_{\text{RNN},0} \leftarrow \mathbf{0}$, $\mathcal{E}_{\text{deblur},0} \leftarrow \mathcal{E}$
for $i \leftarrow 0; i < N; i$++ **do**
  $\Delta\mathcal{F}_i \leftarrow \texttt{RNN}(\texttt{Encoder}(\mathcal{E}_{deblur,i})\,;\, \mathcal{M}_{RNN,i})$
  $\mathcal{F}_{i+1} \leftarrow \mathcal{F}_i + \Delta\mathcal{F}_i$
  $\mathcal{M}_{\text{RNN},i+1} \leftarrow \texttt{WarmStartModule}(\mathcal{F}_{i+1})$
  $\mathcal{E}_{\text{deblur},i+1} \leftarrow \texttt{Deblur}(\mathcal{E}_{deblur,i}; \Delta\mathcal{F}_i)$
$\mathcal{F} \leftarrow \mathcal{F}_N$
return $\mathcal{F}$

C. Iterative Deblurring

The main architecture of *IDNet* under iterative deblurring (*ID* scheme) is shown in Fig. 2. The backbone of our proposed *IDNet* is a recurrent neural network which processes the input event bins sequentially one bin at a time. An optical flow field is read out from the state of the RNN once all bins have been passed through it. In this case, we select a ConvGRU as the choice for the recurrent unit.

Next, we introduce iterative deblurring on top of the sequential processing described above. Our design is inspired by the predictive coding scheme in the visual cortex [41]. In this scheme, our perception of the world is constantly updated, with only the error signal, i.e. the difference between actual visual input and our current prediction, being relayed through the hierarchy of visual cortex for refinement. Analogously, we utilize motion compensation as our prediction model, iterating to refine the flow from the residual signal.

An overview of the algorithm is presented in Algorithm 1. Specifically, as shown in Fig. 2, we start with a zero-initialized flow estimate and backbone RNN memory ($\mathcal{F}_0 = \mathbf{0}$, $\mathcal{M}_{\text{RNN},0} = 0$). After the RNN predicts the flow $\mathcal{F}_1$ from the raw event bins $\mathcal{E}_0$, the event bins are motion-compensated with the optical flow vector $\mathcal{F}_1(x_i)$, resulting in partially deblurred event bins $\mathcal{E}_1$. This starts the iterative refinement process, with diminishing flow refinement over iterations.

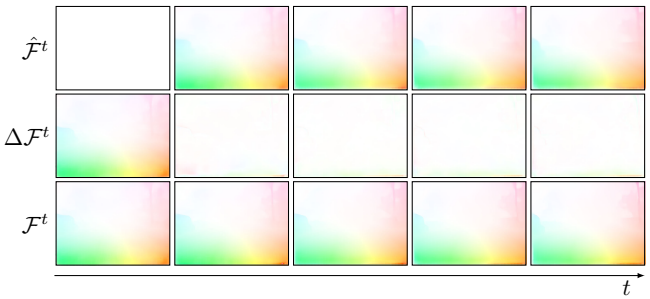

Fig. 3. Initial flow $\hat{\mathcal{F}}^t$, residual flow $\Delta\mathcal{F}^t$ and final flow $\mathcal{F}^t$ as time progresses and new event bins keeps coming. A trend of increasing quality in $\mathcal{F}^t$ and lowering magnitude in $\Delta\mathcal{F}^t$ can be observed, implying that the flow is iteratively refined through time.

To leverage the RNN's recurrency, we use a warm-starting module (WSM) comprising feed-forward convolutional layers to set the memory of our backbone RNN: $\mathcal{M}_{\text{RNN},1} = \text{WSM}(\mathcal{F}_1)$. We then update the previous belief by adding the residual flow $\Delta\mathcal{F}_2$ estimated from $\mathcal{E}_1$ to arrive at the flow estimate at iteration 2: $\mathcal{F}_2 = \mathcal{F}_1 + \Delta\mathcal{F}_2$, from which a new iteration could be started again.

D. Iterative Deblurring through Time

We also explore another mode of iterative refinement through the temporal scale, as shown in Fig. 1. Instead of deblurring the same events multiple iterations, we give the network access to a continuous stream of events where at each timestep we deblur the event bins $\mathcal{E}^t$ and pass the deblurred bins $\mathcal{E}^t_{\text{deblur}}$ through the backbone RNN only once. In addition to estimating the optical flow $\mathcal{F}^t$ for the current timestep, we predict an initial flow $\hat{\mathcal{F}}^{t+1}$ for deblurring the event bins $\mathcal{E}^{t+1}$ from the next timestep. This recurrent approach refines flow over time, as shown in Fig. 3.

Unlike simply passing the raw event bins through the backbone network once, our network initializes with an informative memory from previous estimates and only processes deblurred event bins with smaller motion range, allowing the RNN to enhance flow accuracy with fewer parameters. A pseudo code of the algorithm is laid out in Algorithm 2.

Algorithm 2: Temporal iterative deblurring (*TID*).

input : Event bins $\mathcal{E}^t$ containing events with $\tau \in [t, t+\mathrm{T}]$
Initial flow estimate at $\hat{\mathcal{F}}^t$ at time t
output: Optical Flow Prediction $\mathcal{F}^t$ at time t for the motion from $\tau \in [t, t+\mathrm{T}]$

$\mathcal{M}^t_{\mathrm{RNN}} \leftarrow \texttt{WarmStartModule}(\hat{\mathcal{F}}^t)$
$\mathcal{E}^t_{\mathrm{deblur}} \leftarrow \texttt{Deblur}(\mathcal{E}^t; \hat{\mathcal{F}}^t)$
$\Delta\mathcal{F}^t, \hat{\mathcal{F}}^{t+1} \leftarrow \mathrm{RNN}(\texttt{Encoder}(\mathcal{E}^t_{deblur})\ ;\ \mathcal{M}^t_{RNN})$
$\mathcal{F}^t \leftarrow \hat{\mathcal{F}}^t + \Delta\mathcal{F}^t$
return $\mathcal{F}^t, \hat{\mathcal{F}}^{t+1}$

E. Network Architecture

We use a single-layer ConvGRU with 96 channels as our backbone network. The `Encoder` network, with 9 convolutional layers (4 having residual connections), produces 64-dim lower-resolution feature maps from each event bin as input to the RNN. The RNN outputs a lower-resolution flow field which is upsampled back to the original input size via convex upsampling, similar to [9,13]. The `WSM` module takes the same architecture as the `Encoder` network.

F. Loss

For iterative deblurring (*ID*) scheme, we apply L1 loss between the final flow and ground truth: $\mathcal{L}_{\mathrm{ID}} = ||\mathcal{F}_{\mathrm{gt}} - \mathcal{F}_N||_1$. For training temporal iterative deblurring (*TID*), we take a prediction sequence of length T and enforce an L1 loss on both the intermediate current flows and future flows and weigh them using a geometric series to penalize error more on later iterations, where $\gamma \in (0, 1)$:

$$\mathcal{L}_{\mathrm{TID}} = \sum_{t=0}^{T} \gamma^{T-t} (||\mathcal{F}^t_{\mathrm{gt}} - \mathcal{F}^t||_1 + ||\mathcal{F}^{t+1}_{\mathrm{gt}} - \hat{\mathcal{F}}^{t+1}||_1) \quad (4)$$

IV. Experiments

Following prior works, we train and evaluate our models on two standard benchmarks: DSEC [26] and MVSEC [27].

A. Training Details

We train our models implemented with PyTorch [42] on a single NVIDIA RTX 4090 GPU. We set the number of deblurring iterations N to be 4 for Algorithm 1 and the sequence length $T = 4$ and $\gamma = 0.8$ in Eq. (4). We use an event representation with $\mathrm{B} = 15$ bins for every 100 ms of events from DSEC and MVSEC under $dt = 4$ and 20Hz setting. B is set to 5 for MVSEC $dt = 1$ case. We use Adam [43] for DSEC and AdamW [44] for MVSEC with the onecycle learning rate scheduler [45] under a maximum learning rate of 1×10^{-4}. We train with a batch size of 3 for 250K steps for DSEC and 40K steps for MVSEC. To account for the longer-duration event trajectories which span greater distances on the image plane, we apply random cropping to a larger size of 384 by 512 for training *TID* models, compared to the 288 by 384 size used for *ID* models. We also apply horizontal flipping and a 10% probability of vertical flipping as data augmentation during training.

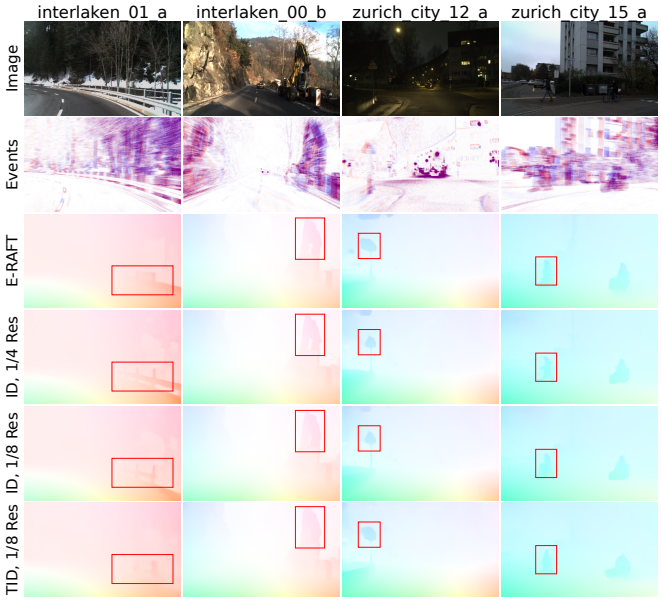

Fig. 4. Qualitative results of optical flow predictions on DSEC-Flow with highlighted regions of interest. Images are for visualization only, as optical flow is event-based. Test samples reveal that our *ID* method at 1/4 resolution produces superior results on fine details and small objects, while the *TID* method yields results that are comparable to those of E-RAFT.

B. DSEC-Flow

Our quantitative evaluations on the DSEC-Flow test set are presented in Table I. We report the following performance metrics regarding flow vector against ground truth: EPE (L2 endpoint error in pixels); AE (angular error in degrees); nPE (percentage of pixels with optical flow magnitude error greater than n pixels). We report the model size in millions of parameters. Additionally, we gather resource usage data for methods when evaluating a single test sample (100ms of events) from DSEC: Memory footprint, the minimum amount of working memory required during inference while executing the computation graph of the model; GMAC, the number of multiply-and-accumulation operations during inference measured in billions of operations. We benchmark the runtime of the algorithms on three different platforms: 8-core AMD Ryzen 7700 CPU, Nvidia RTX 4090 Desktop GPU (D-GPU) and Nvidia Jetson Xavier NX Embedded GPU (E-GPU), commonly used in mobile robots.

Our methods significantly surpass optimization-based MultiCM [46], the self-supervised TamingCM [47], and the feedforward UNet-like EV-FlowNet [29]. Only methods that employ correlation volumes [13–15] yield performance comparable to ours. Notably, our base *ID* model, which processes feature maps of 1/8th the input resolution reaches performance close to, albeit slightly behind, the recent state-of-the-art methods (TMA [14] and E-Flowformer [15]).

On the other hand, our models are much more lightweight, using only a quarter of the parameters and minimal memory. This efficiency enables their deployment on power-efficient edge inference chips with limited on-chip SRAM up to mere MBs, exemplified by [49, 50]. These memory savings arise not merely from a reduced network size but predominantly from eliminating correlation volumes, which are essential for

TABLE I

EVALUATION ON DSEC-FLOW DATASET WITH MODEL STATISTICS. THE BEST METRIC IS IN BOLD, WHILE THE SECOND BEST IS UNDERLINED. VALUES MARKED "-" ARE UNAVAILABLE, WHILE THOSE WITH "∗" ARE INTERPOLATED FROM SIMILAR ARCHITECTURES.

	Performance					Model Size	Memory Footprint	GMAC	Runtime			Real-time
	EPE	AE	1PE	2PE	3PE				CPU	D-GPU	E-GPU	
MultiCM [46] (Optimization-based)	3.47	13.98	76.6	48.4	30.9	-	-	-	-	-	-	-
TamingCM [47] (Self-supervised)	2.33	10.56	68.3	33.5	17.8	-	-	-	-	-	-	-
EV-FlowNet [29]	2.32	7.90	55.4	29.8	18.6	14M	120MB	62	0.20s	**2ms**	**0.02s**	✔
EVA-Flow [48]	0.88	3.31	15.9	-	3.2	-	100MB*	-	-	35ms*	-	✗
E-RAFT [13] (1/8 Resolution)	0.79	2.85	12.7	4.7	2.7	5.3M	132MB	256	0.70s	36ms	0.93s	✗
∟ E-RAFT (1/4 Resolution)	GPU Out of Memory (>40GB)					5.3M	1.9GB	750	1.76s	65ms	1.68s	✗
∟ E-RAFT w/o Correlation Volume	1.20	3.75	27.5	11.5	6.3	4.5M	40MB	208	0.53s	30ms	0.71s	✗
E-Flowformer [15]	0.76	**2.68**	11.2	4.1	2.4	-	-	-	-	-	-	✗
TMA [14]	0.74	**2.68**	10.9	4.0	2.7	6.9M	1.1GB	522	2.30s	17ms	1.06s	✗
Ours (ID, 4 iterations, 1/4 Resolution)	**0.72**	2.72	**10.1**	**3.5**	**2.0**	2.5M	30MB	1200	2.50s	78ms	2.22s	✗
Ours (ID, 4 iterations, 1/8 Resolution)	0.77	3.00	12.1	4.0	2.2	**1.4M**	**20MB**	222	0.68s	46ms	0.63s	✗
Ours (TID, 1 iteration, 1/8 Resolution)	0.84	3.41	14.7	5.0	2.8	1.9M	**20MB**	**55**	**0.12s**	12ms	0.12s	✔

their performance. Removing the correlation volume from E-RAFT results in a smaller memory footprint but significantly deteriorated performance, illustrating its importance.

Compared to [13,14], our base model also has the smallest GMAC and runs fastest on the embedded GPU, showing 40% improvement over TMA [14]. While [13, 14] demonstrate faster desktop GPU runtime, this doesn't genuinely represent their computational demand. This accelerated performance owes largely to the higher throughput of high-end GPUs, stemming from more CUDA cores and higher memory bandwidth, while incurring more power use. This setup naturally favors their more parallel architectures, such as correlation volumes and transformer layers, as used in [13–15]. Conversely, our recurrent networks, inherently sequential, benefit less from such increased throughput. However, complemented by the low power of event cameras, we envision their biggest use cases in power-constrained embedded systems including IoT and robotics. We contend that evaluating event-based algorithms on these systems is more important.

While *ID* performs better, *TID* is drastically faster by only iterating once. By leveraging the temporal prior, we have managed to retain almost the full quality of prediction while incurring a performance drop of only 9-13% compared to the base model and state-of-the-art method TMA [14], which consumes 10 times more compute and runs 8 times slower. Moreover, [13–15] requires accumulating all events in a time window before processing, resulting in high latency, while *TID* processes event bins sequentially and can thus parallelize the processing of bins as events arrive in real-time, significantly reducing inference latency, as reported in Table II. The recently proposed EVA-Flow [48], employing a recurrent architecture, has also significantly reduced latency through on-the-fly processing. However, it still falls behind our method in both accuracy and latency due to its lack of a temporal prior and a larger network size. Among all published methods, *TID* stands out as the only one nearing a real-time processing rate on the DSEC-Flow dataset (10Hz) while still upholding decent accuracy.

TABLE II

EPE, LATENCY AND PROCESSING MODE OF SELECTED METHODS, MEASURED ON NVIDIA JETSON XAVIER NX GPU.

	EPE	Latency	Processing Mode
E-RAFT [13]	**0.79**	930ms	Batch
EVA-Flow [48]	0.88	93ms*	On-the-fly
Ours (TID)	0.84	**8ms**	On-the-fly

As highlighted in [16,17], all-pair correlation volumes suffer from poor time and space complexity. Thus methods like [13–15] derive correlation volumes from feature maps with 1/8 the resolution of the input size to maintain reasonable memory usage. This limitation impacts performance, as fine details cannot be preserved in low-resolution features. As a concrete example, E-RAFT processing 1/4 resolution would consume 1.9GB of memory per sample, making it not only impractical to deploy on most hardware but also impossible to train on a single GPU due to the even larger memory requirement for training. In contrast, *IDNet* scales well to high-res feature maps. By tweaking the convolution stride in the `Encoder` to produce 1/4 resolution maps, performance leaps by 7% in EPE, outperforming prior state-of-the-art on this benchmark. Qualitative results are presented in Fig. 4. While the upscaled model is no longer lightweight, its top performance may render itself useful in offline non-realtime applications such as computational photography [51].

C. MVSEC

In line with prior works [13, 14, 29], we train on `outdoor_day_2` sequence and evaluate on the `outdoor_day_1` sequence under 45Hz ($dt = 1$), 11.25Hz ($dt = 4$) and original 20Hz rate. The results are presented in Table III. When trained from scratch, our methods lag behind those in [13,14,29], unlike the case on DSEC-Flow dataset. Compared to DSEC, MVSEC holds data of limited variety and poor quality. It features a single sequence, whereas DSEC encompasses eighteen. The events in MVSEC are lower in resolution and sparser than in DSEC. These factors all make generalization on this dataset extremely challenging.

TABLE III

EVALUATION RESULTS ON MVSEC OUTDOOR DAY 1 SEQUENCE.

	$dt = 1$		$dt = 4$		20Hz	
	EPE	3PE	EPE	3PE	EPE	3PE
EV-FlowNet [29]	0.35	0.0	1.09	5.7	0.61	0.45
E-RAFT [13]	0.27	0.0	0.72	**1.1**	**0.47**	**0.24**
└ w/o Correlation Volume	0.32	0.0	1.02	4.1	0.64	0.55
└ Finetune	0.29	0.0	0.67	0.9	0.44	0.22
TMA [14]	**0.25**	0.1	**0.70**	**1.1**	-	-
ID, 1/4 Res, 1 iter	0.31	0.1	1.30	9.1	0.75	0.88
└ Finetune	0.29	0.0	0.75	1.2	0.49	0.23
ID, 1/8 Res, 1 iter	0.33	0.1	1.26	8.5	0.74	0.84
└ Feedforward Init	0.31	0.0	1.11	6.0	0.63	0.86
└ Finetune	0.30	0.0	0.79	1.5	0.46	0.22
ID, 1/8 Res, 4 iter	0.34	0.0	1.48	11.2	0.84	1.06
└ Finetune	0.29	0.0	0.77	1.7	0.46	0.25
TID, 1 iter	0.45	0.2	1.65	13.1	1.03	2.51
└ Finetune	0.35	0.1	0.78	1.5	0.47	0.20

As a result, even state-of-the-art methods can only generalize to an EPE of 0.5 pixels at 20Hz evaluation rate. When adjusted for time window and mean flow magnitude, this translates to a 5-pixel EPE under DSEC, notably poorer than any methods benchmarked there. This is reflected in the qualitative results in Fig. 5 with blurry motion boundaries and poor details, unlike the case in DSEC as shown in Fig. 4.

The deterioriated training signal, owing to the limited and lower-quality data, makes algorithmic generalization more reliant on their inductive biases. We have pinpointed two such biases which lead to this outcome on this dataset: correlation volumes and feedforward processing. Removing correlation volumes from E-RAFT renders its performance comparable to EV-FlowNet. The sparse events also complicate learning useful representations via recurrent connections, which are central to the operations of *IDNet*. By attaching a feedforward network which takes the entire voxel grid as input to initialize the memory of our RNN (denoted by "Feedforward Init" in Table III), we bridge the performance gap with EV-FlowNet. While these biases benefit training on this dataset, they demand substantial computational power while incurring undesired latency and are not essential for high performance when the data quality is improved.

To verify the hypothesis that the reduced performance of *IDNet* is due to the limited and low-quality training set, we finetune models pretrained on DSEC with `outdoor_day_2` sequence before evaluating on `outdoor_day_1` sequence, marked by "Finetune" in Table III. We observe the finetuned performance is close to E-RAFT, surpassing those of EV-FlowNet. Hence, the less good results on this dataset should not be regarded as a limitation of the method, highlighting the need for a more modern and representative benchmark.

D. Ablation Studies

We study the impact of introducing iterative deblurring on a slightly scaled-down model to save time and resources. The results are reported in Table IV. For this experiment, we first set the number of deblurring iterations down to 1. Without either iterative processing or deblurring, the performance dropped by 47% on EPE. Introducing recurrency with 4 iterations but no deblurring, where the network would take raw event bins instead of the deblurred ones as in Algorithm 1, results in a 41% decrease in EPE, demonstrating the effectiveness of deblurring as an inductive bias. Lastly, we remove the warm starting module (WSM) so that the memory of the backbone RNN would be initialized to zero. This time, EPE and AE went up by 16%, revealing the recurrency is better complemented with an informative prior.

TABLE IV

ABLATION STUDY ON DSEC-FLOW W.R.T THE EFFECTIVENESS OF ITERATIVE DEBLURRING. THE STUDY USES A SLIGHTLY SCALED-DOWN MODEL THAN THE ONES PRESENTED IN TABLE I.

	EPE	AE	1PE	3PE
ID 4 iters	**0.88**	**3.21**	**15.6**	**3.2**
ID 2 iters	0.94	3.43	18.1	3.7
ID 1 iter	1.30	4.82	33.7	6.7
ID 4 iters w/o deblurring	1.24	4.49	31.1	6.6
ID 4 iters w/o WSM	1.02	4.07	20.8	4.0

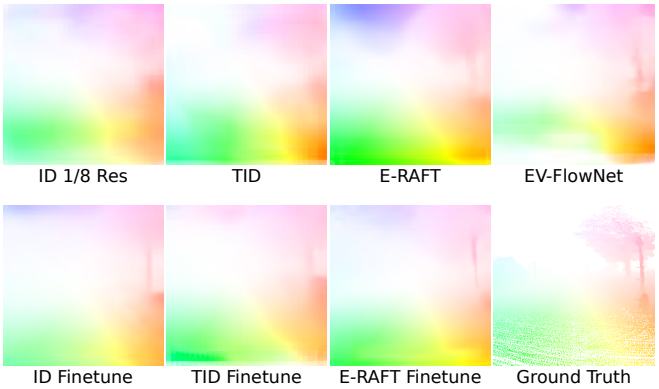

Fig. 5. Qualitative results on MVSEC `outdoor_day_1` sequence.

We also study the impact of the number of deblurring iterations on the performance by comparing *ID* models with $N = 1, 2, 4$ iterations. We observe that iterating twice already lowers EPE by 27%, while two extra deblurring iterations bring a diminishing further improvement of 5%.

Note the improvement brought by multiple iterations is diminishing for scenes with very small motion, such as in MVSEC (see Table III) where the mean flow magnitude is only around 3 pixels at 20Hz evaluation rate.

V. CONCLUSION

In this work, we introduce IDNet, a lightweight yet high-performing event-based optical flow algorithm. We demonstrate through experiments our network are competitive in performance while being much more efficient than state-of-the-art methods. We believe our work provides valuable insights into efficiently solving event-based optical flow problems and hope to encourage the community to adopt our methods for potential applications and explore the principle of iterative deblurring for other architectures and vision tasks.

ACKNOWLEDGEMENTS

This work was sponsored by the Office of Naval Research (ONR) Global under grant number N629092112014. The views and conclusions contained herein are those of the authors only and should not be interpreted as representing those of the U.S. Government.

REFERENCES

[1] H. Chao, Y. Gu, and M. Napolitano, "A survey of optical flow techniques for robotics navigation applications," *Journal of Intelligent & Robotic Systems*, vol. 73, no. 1, pp. 361–372, 2014. 1

[2] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018. 1

[3] R. J. Bouwmeester, F. Paredes-Vallés, and G. C. H. E. de Croon, "NanoFlowNet: Real-time dense optical flow on a nano quadcopter," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 1996–2003. 1

[4] G. Gallego, T. Delbruck, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, "Event-based vision: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 01, pp. 154–180, 1 2022. 1

[5] D. Falanga, S. Kim, and D. Scaramuzza, "How fast is too fast? the role of perception latency in high-speed sense and avoid," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1884–1891, 2019. 1

[6] B. Forrai, T. Miki, D. Gehrig, M. Hutter, and D. Scaramuzza, "Event-based agile object catching with a quadrupedal robot," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 12 177–12 183. 1

[7] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8934–8943. 1, 2

[8] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. v. d. Smagt, D. Cremers, and T. Brox, "FlowNet: Learning optical flow with convolutional networks," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 2758–2766. 1

[9] Z. Teed and J. Deng, "RAFT: Recurrent all-pairs field transforms for optical flow," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 402–419. 1, 2, 4

[10] S. Jiang, D. Campbell, Y. Lu, H. Li, and R. Hartley, "Learning to estimate hidden motions with global motion aggregation," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 9752–9761. 1

[11] Z. Huang, X. Shi, C. Zhang, Q. Wang, K. C. Cheung, H. Qin, J. Dai, and H. Li, "FlowFormer: A transformer architecture for optical flow," in *Computer Vision – ECCV 2022*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds. Cham: Springer Nature Switzerland, 2022, pp. 668–685. 1

[12] H. Xu, J. Zhang, J. Cai, H. Rezatofighi, and D. Tao, "GMFlow: Learning optical flow via global matching," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8121–8130. 1

[13] M. Gehrig, M. Millhausler, D. Gehrig, and D. Scaramuzza, "E-RAFT: Dense optical flow from event cameras," in *2021 International Conference on 3D Vision (3DV)*. Los Alamitos, CA, USA: IEEE Computer Society, Dec 2021, pp. 197–206. 1, 2, 4, 5, 6

[14] H. Liu, G. Chen, S. Qu, Y. Zhang, Z. Li, A. Knoll, and C. Jiang, "TMA: Temporal motion aggregation for event-based optical flow," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, Oct 2023, pp. 9651–9660. 1, 2, 4, 5, 6

[15] Y. Li, Z. Huang, S. Chen, X. Shi, H. Li, H. Bao, Z. Cui, and G. Zhang, "BlinkFlow: A dataset to push the limits of event-based optical flow estimation," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 3881–3888. 1, 2, 4, 5

[16] S. Jiang, Y. Lu, H. Li, and R. Hartley, "Learning optical flow from a few matches," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 16 587–16 595. 1, 2, 5

[17] H. Xu, J. Yang, J. Cai, J. Zhang, and X. Tong, "High-resolution optical flow from 1d attention and correlation," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 10 478–10 487. 1, 2, 5

[18] G. Gallego, H. Rebecq, and D. Scaramuzza, "A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3867–3876. 2

[19] R. Benosman, S.-H. Ieng, C. Clercq, C. Bartolozzi, and M. Srinivasan, "Asynchronous frameless event-based optical flow," *Neural Networks*, vol. 27, pp. 32–37, 2012. 2

[20] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, ser. IJCAI'81. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1981, p. 674–679. 2

[21] 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, 2014. 2

[22] E. Mueggler, C. Forster, N. Baumli, G. Gallego, and D. Scaramuzza, "Lifetime estimation of events from dynamic vision sensors," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 4874–4881. 2

[23] 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. 2

[24] D. Gehrig, M. Gehrig, J. Hidalgo-Carrió, and D. Scaramuzza, "Video to events: Recycling video datasets for event cameras," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 3583–3592. 2

[25] Y. Hu, S.-C. Liu, and T. Delbruck, "v2e: From video frames to realistic dvs events," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2021, pp. 1312–1321. 2

[26] M. Gehrig, W. Aarents, D. Gehrig, and D. Scaramuzza, "DSEC: A stereo event camera dataset for driving scenarios," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4947–4954, 2021. 2, 4

[27] 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. 2, 4

[28] K. Chaney, F. Cladera, Z. Wang, A. Bisulco, M. A. Hsieh, C. Korpela, V. Kumar, C. J. Taylor, and K. Daniilidis, "M3ED: Multi-robot, multi-sensor, multi-environment event dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2023, pp. 4015–4022. 2

[29] A. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "EV-FlowNet: Self-supervised optical flow estimation for event-based cameras," *Robotics: Science and Systems XIV*, Jun 2018. 2, 4, 5, 6

[30] C. Ye, A. Mitrokhin, C. Fermüller, J. A. Yorke, and Y. Aloimonos, "Unsupervised learning of dense optical flow, depth and egomotion with event-based sensors," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 5831–5838. 2

[31] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2720–2729. 2

[32] T.-W. Hui, X. Tang, and C. C. Loy, "LiteFlowNet: A lightweight convolutional neural network for optical flow estimation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8981–8989. 2

[33] J. Hur and S. Roth, "Iterative residual refinement for joint optical flow and occlusion estimation," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 5747–5756. 2

[34] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1647–1655. 2

[35] Y. Lu, J. Valmadre, H. Wang, J. Kannala, M. Harandi, and P. H. S. Torr, "Devon: Deformable volume network for learning optical flow," in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020, pp. 2694–2702. 2

[36] L. Hu, R. Zhao, Z. Ding, L. Ma, B. Shi, R. Xiong, and T. Huang, "Optical flow estimation for spiking camera," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 17 823–17 832. 2

[37] Z. Li, J. Shen, and R. Liu, "A lightweight network to learn optical flow from event data," in *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021, pp. 1–7. 2

[38] Z. Ding, R. Zhao, J. Zhang, T. Gao, R. Xiong, Z. Yu, and T. Huang, "Spatio-temporal recurrent networks for event-based optical flow estimation," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, vol. 36, 2022, pp. 525–533. 2

[39] M. Neoral, J. Šochman, and J. Matas, "Continual occlusion and optical flow estimation," in *Computer Vision – ACCV 2018*, C. Jawahar,

H. Li, G. Mori, and K. Schindler, Eds. Cham: Springer International Publishing, 2019, pp. 159–174. 2

[40] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "Unsupervised event-based learning of optical flow, depth, and egomotion," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 989–997. 2

[41] R. P. N. Rao and D. H. Ballard, "Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects," *Nature Neuroscience*, vol. 2, no. 1, pp. 79–87, 1999. 3

[42] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2019. 4

[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. 4

[44] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2019. 4

[45] L. N. Smith and N. Topin, "Super-convergence: very fast training of neural networks using large learning rates," *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, May 2019. 4

[46] S. Shiba, Y. Aoki, and G. Gallego, "Secrets of event-based optical flow," in *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVIII*. Berlin, Heidelberg: Springer-Verlag, 2022, p. 628–645. 4, 5

[47] F. Paredes-Vallés, K. Y. W. Scheper, C. De Wagter, and G. C. H. E. De Croon, "Taming contrast maximization for learning sequential, low-latency, event-based optical flow," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 9661–9671. 4, 5

[48] Y. Ye, H. Shi, K. Yang, Z. Wang, X. Yin, Y. Wang, and K. Wang, "Towards anytime optical flow estimation with event cameras," 2023, arXiv:2307.05033 [cs.CV]. 5

[49] G. Technologies, "GAP9 processor," https://greenwaves-technologies.com/gap9_processor/, Jan 2023, accessed: 2024-03-01. 4

[50] Intel, "Intel® Movidius™ Myriad™ X VPU Product Brief," https://www.intel.com/content/www/us/en/products/docs/processors/movidius-vpu/myriad-x-product-brief.html, 2018, accessed: 2024-03-01. 4

[51] S. Tulyakov, D. Gehrig, S. Georgoulis, J. Erbach, M. Gehrig, Y. Li, and D. Scaramuzza, "Time lens: Event-based video frame interpolation," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 16 150–16 159. 5