Secrets of Event-based Optical Flow, Depth and Ego-motion Estimation by Contrast Maximization

Shintaro Shiba, Yannick Klose, Yoshimitsu Aoki, Guillermo Gallego

This paper has been accepted for publication at the
IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024. ©IEEE

Secrets of Event-based Optical Flow, Depth and Ego-motion Estimation by Contrast Maximization

Shintaro Shiba, Yannick Klose, Yoshimitsu Aoki, and Guillermo Gallego

Abstract—Event cameras respond to scene dynamics and provide signals naturally suitable for motion estimation with advantages, such as high dynamic range. The emerging field of event-based vision motivates a revisit of fundamental computer vision tasks related to motion, such as optical flow and depth estimation. However, state-of-the-art event-based optical flow methods tend to originate in frame-based deep-learning methods, which require several adaptations (data conversion, loss function, etc.) as they have very different properties. We develop a principled method to extend the Contrast Maximization framework to estimate dense optical flow, depth, and ego-motion from events alone. The proposed method sensibly models the space-time properties of event data and tackles the event alignment problem. It designs the objective function to prevent overfitting, deals better with occlusions, and improves convergence using a multi-scale approach. With these key elements, our method ranks first among unsupervised methods on the MVSEC benchmark and is competitive on the DSEC benchmark. Moreover, it allows us to simultaneously estimate dense depth and ego-motion, exposes the limitations of current flow benchmarks, and produces remarkable results when it is transferred to unsupervised learning settings. Along with various downstream applications shown, we hope the proposed method becomes a cornerstone on event-based motion-related tasks. Code is available at https://github.com/tub-rip/event_based_optical_flow

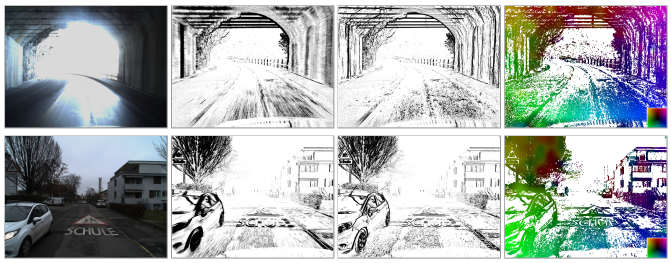
Index Terms—Event camera, Asynchronous sensors, Optical flow, 3D reconstruction, Camera motion estimation, High dynamic range.

1 Introduction

Event cameras are novel bio-inspired vision sensors that naturally respond to motion of edges in image space with high dynamic range (HDR) and minimal blur at high temporal resolution, on the order of µs [1]. These advantages provide a rich signal for accurate motion estimation in difficult real-world scenarios for frame-based cameras. However, such a signal is asynchronous and sparse by nature, hence not compatible with traditional computer vision algorithms. This poses the challenge of rethinking visual processing [2], [3]: motion patterns (i.e., *optical flow*) are no longer obtained by analyzing the intensities of images captured at regular intervals, but by analyzing the stream of per-pixel brightness changes produced by the event camera.

Multiple methods have been proposed for event-based optical flow estimation. They can be broadly categorized in two: (i) model-based methods, which investigate the principles and characteristics of event data that enable optical flow estimation, and (ii) learning-based methods, which exploit correlations in the data and/or apply the above-mentioned principles to compute optical flow. One of the challenges of event-based optical flow is the lack of ground truth flow in real-world datasets (at µs resolution and HDR) [2], which makes it difficult to evaluate and compare the methods properly, and to train supervised-learning ones. Ground truth (GT) in de facto standard datasets [4], [5] is obtained by the *motion field* [6] given additional depth sensors and camera motion. However, such data is limited by the field-of-view (FOV) and resolution (spatial and temporal) of the depth sensor, which do not match those of event cameras. Hence, it is paramount to develop interpretable optical flow methods that exploit the characteristics of event data, and that do not need costly and error-prone ground truth.

Among prior work, Contrast Maximization (CM) [7], [8] is a powerful framework that allows us to tackle multiple motion estimation problems (rotational motion [9]–[12], ho-



(a) Frame (reference) (b) Events (zero flow) (c) Warped events, IWE (d) Estimated flow

Figure 1: DSEC test sequences (interlaken_00b, thun_01a) [5]. Our optical flow estimation method produces sharp images of warped events (IWE) despite the scene complexity, the large pixel displacement and the high dynamic range.

- *S. Shiba and Y. Aoki are with the Department of Electronics and Electrical Engineering, Faculty of Science and Technology, Keio University, Kanagawa, Japan. E-mail: sshiba@keio.jp*
- *S. Shiba, Y. Klose and G. Gallego are with the Department of Electrical Engineering and Computer Science, Technische Universität Berlin, Berlin, Germany.*
- *G. Gallego is with the Science of Intelligence Excellence Cluster and with the Einstein Center Digital Future, Berlin, Germany. E-mail: guillermo.gallego@tu-berlin.de*

- *Funded by the German Academic Exchange Service (DAAD), Research Grant - Bi-nationally Supervised Doctoral Degrees/Cotutelle, 2021/22 (57552338). Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2002/1 "Science of Intelligence" – project number 390523135.*
- *Manuscript received April 28, 2023; accepted April 21, 2024.*

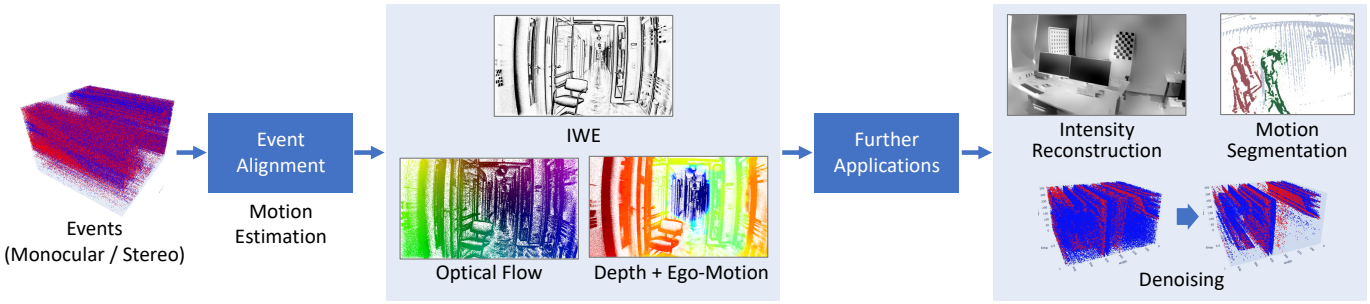


Figure 2: *Overview*. The proposed method solely relies on event data. It not only estimates optical flow, but can also estimate scene depth and ego-motion simultaneously from a monocular or stereo event camera setup. Furthermore, the estimated flow enables various downstream applications such as motion segmentation, intensity reconstruction and event denoising.

mographic motion [7], [13], [14], feature flow estimation [15]–[18], motion segmentation [19]–[22], and also reconstruction [7], [23], [24]). It maximizes an objective function (e.g., contrast) that measures the alignment of events caused by the same scene edge. The intuitive interpretation is to estimate the motion by recovering the sharp (motion-compensated) image of edge patterns that caused the events. Preliminary work on applying CM to estimate optical flow has reported event collapse [25], [26], producing flows at undesired optima that warp events to few pixels or lines [27]. This issue has been tackled by changing the objective function, from contrast to the energy of an average timestamp image [27], [28], but this loss is not straightforward to interpret [8], [29], and is not without its problems [30].

The state-of-the-art performance of CM in low degrees-of-freedom (DOF) motion estimations and its issues in more complex motions (dense flow) suggests that prior work may have rushed to use CM in unsupervised learning of dense flow. There is a gap in understanding how CM can be sensibly extended to estimate dense optical flow accurately. This paper fills this gap and shows a few "secrets" that are also applicable to overcome the issues of previous approaches.

We propose to extend CM for dense optical flow estimation via a tile-based approach covering the image plane (Fig. 1). We present several distinctive contributions:

1) A *multi-reference* focus loss function to improve accuracy and discourage overfitting (Sec. 3.2).
2) A principled *time-aware flow* to better handle occlusions, leveraging the solution of transport problems via differential equations (Sec. 3.3).
3) A *multi-scale* approach to improve convergence to the solution and avoid getting trapped in local optima (Sec. 3.4).

Optical flow is a fundamental visual quantity related to many others, such as camera motion and scene depth. Hence, in this paper we exploit these connections, in monocular and stereo configurations, and show how a dense flow can serve to tackle various related problems in event-based vision, such as depth estimation, motion segmentation, etc. (Fig. 2). This paper is based on our previous work [31], which we substantially extend in the following points:

1) We introduce a new objective function that improves both flow and depth estimation (Sec. 3.2.1).
2) We tackle stationary scenes, estimating monocular depth and ego-motion jointly (Secs. 3.6.1 and 4.4).
3) We also address the stereo setup (Secs. 3.6.2 and 4.5).
4) We discuss current optical flow benchmarks, evaluations and "GT" flow (Sec. 4.2.5).
5) We provide experiments on downstream applications of optical flow: motion segmentation, intensity reconstruction, and denoising (Sec. 4.3).
6) We show experiments on 1Mpixel event cameras, the most recent event camera datasets: TUM-VIE [32] and M3ED [33], both in flow (Sec. 4.2.4) and depth estimation (Sec. 4.4.3).
7) We extend the discussion on computational performance and limitations (Secs. 6 and 7).

The results of our experimental evaluation are surprising: the above design choices are key to our simple, model-based tile-based method achieving the best accuracy among all state-of-the-art methods, including supervised-learning ones, on the de facto benchmark of MVSEC indoor sequences [34]. Since our method is interpretable and produces better event alignment than the ground truth flow, both qualitatively and quantitatively, the experiments also expose the limitations of the current "ground truth". The experiments demonstrate that the above key choices are transferable to unsupervised learning methods, thus guiding future design and understanding of more proficient Artificial Neural Networks (ANNs) for event-based optical flow estimation. Finally, the method allows us to solve many motion-related applications, thus becoming a cornerstone in event-based vision.

Because of the above, we believe that the proposed design choices deserve to be called "secrets" [35]. To the best of our knowledge, they are novel in the context of event-based optical flow, depth and ego-motion estimation, e.g., no prior work considers constant flow along its characteristic lines, designs the multi-reference focus loss to tackle overfitting, or has defined multi-scale (i.e., multi-resolution) contrast maximization on the raw events.

2 Related Work

2.1 Event-based Optical Flow Estimation

Given the identified advantages of event cameras to estimate optical flow, extensive research on this topic has been carried out. Prior work has proposed adaptations of frame-based approaches (block matching [36], Lucas-Kanade [37]), filter-banks [38], [39], spatio-temporal plane-fitting [40], [41],

time surface matching [42], variational optimization on voxelized events [43], and feature-based contrast maximization [7], [15]. For a detailed survey, we refer to [2].

Current state-of-the-art approaches are ANNs [27], [30], [34], [44]–[46], largely inspired by frame-based optical flow architectures [47], [48]. Non-spiking–based approaches need to additionally adapt the input signal, converting the events into a tensor representation (event frames, voxel grids, etc.). These learning-based methods can be classified into supervised, semi-supervised, or unsupervised (see Tab. 1). In terms of architectures, the three most common ones are U-Net [34], [49], FireNet [28], and RAFT [44], [50].

Supervised methods train ANNs in simulation and/or real-data [44], [49]–[54]. This requires accurate GT flow that matches the space-time resolution of event cameras. While this is not a problem in simulation, it incurs a performance gap when trained models are used to predict flow on real data, due to often a large domain gap between training and test data [52], [55]. Besides, real-world datasets have issues in providing accurate GT flow.

Semi-supervised methods use the grayscale images from a colocated camera (e.g., DAVIS [56]) as a supervisory signal: images are warped using the flow predicted by the ANN and their photometric consistency is used as loss function [34], [45], [46]. While such supervisory signal is easier to obtain than real-world GT flow, it may suffer from the limitations of frame-based cameras (e.g., motion blur and low dynamic range), consequently affecting the trained ANNs. EV-FlowNet [34] pioneered these approaches.

Unsupervised methods rely solely on event data. Their loss function consists of an event alignment error using the flow predicted by the ANN [27], [28], [30], [57]–[59]. Zhu et al. [27] extended EV-FlowNet [34] to the unsupervised setting using a motion-compensation loss inspired by the average timestamp images in [19]. This U-Net–like approach has been improved with recurrent blocks in [28], [30]. Paredes-Vallés et al. [28] also proposed FireFlowNet, a lightweight recurrent ANN with no downsampling. More recently, [30] has proposed several variants of EV-FlowNet and FireFlowNet models, and, enabled by the recurrent blocks, has replaced the usual voxel-grid input event representation by sequentially processing short-time event frames. Finally, concurrent work [59] builds upon [30] (sequential processing of event frames), proposing iterative event warping at multiple reference times in a multi-timescale fashion, which allows curved motion trajectories.

2.2 Event-based Depth and Ego-Motion Estimation

Having estimated optical flow, one could try to fit a depth map and camera ego-motion a posteriori consistent with the flow [60]. Instead, it is better to incorporate the assumption of a still scene and a moving camera on the parameterization of the flow using the motion field equation [6]. While this connection exists, the topic of joint ego-motion and dense depth estimation via the motion field is not as explored as optical flow estimation. The problem is difficult, and often one settles for estimating depth alone, with or without knowledge of the camera motion [23], [61], [62].

Closest to our work are [27], [57] because they estimate a depth-parameterized motion field that best fits the event data. They do so by training ANNs in an unsupervised way. The loss functions are based on the energy of an average timestamp image [27] or on the photometric consistency of edge-maps warped by the predicted flow [57].

Similar to the above-mentioned unsupervised-learning works, our method produces dense optical flow and/or depth and does not need ground truth or additional supervisory signals. In contrast to prior work, we adopt a more classical modeling perspective to gain insights into the problem and discover principled solutions that can subsequently be applied to the learning-based setting. Stemming from an accurate and spatially-dependent contrast loss (the gradient magnitude [8]), we model the problem using a tile of patches (in flow or depth parameters) and propose solutions to several problems: overfitting, occlusions, and convergence. To the best of our knowledge, (i) no prior work has proposed to estimate dense optical flow and/or dense depth from a CM model-based perspective, and (ii) no prior unsupervised learning approach based on motion compensation has succeeded in estimating optical flow without the average timestamp image loss. The latter may be due to event collapse [25], but given recent advances on overcoming this issue [31], we show it is possible to succeed.

3 Method

In this section, first we briefly revisit the Contrast Maximization framework (Sec. 3.1). Then, the proposed methods are explained in detail: Section 3.2 proposes the new data fidelity term of the objective function, which discourages event collapse. Section 3.3 proposes a principled model for optical flow that considers the space-time nature of events. We also explain the multi-scale parameterization of the flow (Sec. 3.4), the composite objective function (Sec. 3.5), and the application to the problem of depth and ego-motion estimation in monocular and stereo configurations (Sec. 3.6).

3.1 Event Cameras and Contrast Maximization

Event cameras have independent pixels that operate continuously and generate "events" $e_k \doteq (\mathbf{x}_k, t_k, p_k)$ whenever the logarithmic brightness at the pixel increases or decreases by a predefined amount, called contrast sensitivity. Each event e_k contains the pixel-time coordinates $(\mathbf{x}_k, t_k)$ of the brightness change and its polarity $p_k \in \{+1, -1\}$. Events occur asynchronously and sparsely on the pixel lattice, with a variable rate that depends on the scene dynamics.

The CM framework [7] assumes events $\mathcal{E} \doteq \{e_k\}_{k=1}^{N_e}$ are caused by moving edges (i.e., brightness constancy), and transforms them geometrically according to a motion model $\mathbf{W}$, producing a set of warped events $\mathcal{E}'_{t_{\text{ref}}} \doteq \{e'_k\}_{k=1}^{N_e}$ at a reference time t_{ref}:

$$e_k \doteq (\mathbf{x}_k, t_k, p_k) \;\mapsto\; e'_k \doteq (\mathbf{x}'_k, t_{\text{ref}}, p_k). \tag{1}$$

The warp $\mathbf{x}'_k = \mathbf{W}(\mathbf{x}_k, t_k; \boldsymbol{\theta})$ transports each event from t_k to t_{ref} along the motion curve that passes through it. The vector $\boldsymbol{\theta}$ parameterizes the motion curves. Transformed events are aggregated on an image of warped events (IWE):

$$I(\mathbf{x}; \mathcal{E}'_{t_{\text{ref}}}, \boldsymbol{\theta}) \doteq \sum_{k=1}^{N_e} \delta(\mathbf{x} - \mathbf{x}'_k), \tag{2}$$

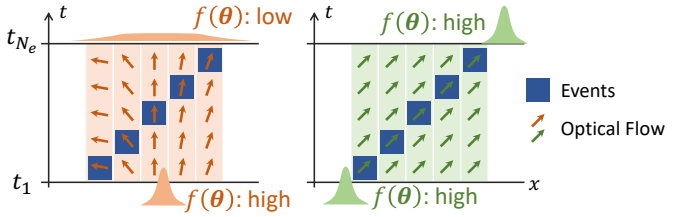


Figure 3: *Multi-reference focus loss*. Assume an edge moves from left to right. Flow estimation with single reference time (t_1) can warp all events into a single pixel, which results in a maximum contrast (at t_1). However, the same flow would produce low contrast (i.e., a blurry image) if events were warped to time t_{N_e}. Instead, we favor flow fields that produce high contrast (i.e., sharp images) at any reference time (here, $t_{\text{ref}} = t_1$ and $t_{\text{ref}} = t_{N_e}$). See also results in Fig. 20.

where each pixel $\mathbf{x}$ sums the number of warped events $\mathbf{x}'_k$ that fall within it. The Dirac delta is approximated by a Gaussian, $\delta(\mathbf{x} - \boldsymbol{\mu}) \approx \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \epsilon^2 \mathtt{Id})$ with $\epsilon = 1$ pixel. Next, an objective function $f(\boldsymbol{\theta})$ is computed, such as the contrast of the IWE (2), given by the variance

$$\text{Var}(I(\mathbf{x};\boldsymbol{\theta})) \doteq \frac{1}{|\Omega|}\int_\Omega \left(I(\mathbf{x};\boldsymbol{\theta}) - \mu_I\right)^2 d\mathbf{x}, \tag{3}$$

with mean $\mu_I \doteq \frac{1}{|\Omega|}\int_\Omega I(\mathbf{x};\boldsymbol{\theta})d\mathbf{x}$. The objective function measures the goodness of fit between the events and the candidate motion curves (warp). Finally, an optimization algorithm iterates the above steps until convergence. The goal is to find the motion parameters that maximize the alignment of events caused by the same scene edge. Event alignment is measured by the strength of the edges of the IWE, which is directly related to image contrast [8].

For *dense optical flow* motion, the warp used is [27], [28]:

$$\mathbf{x}'_k = \mathbf{x}_k + (t_k - t_{\text{ref}})\,\mathbf{v}(\mathbf{x}_k), \tag{4}$$

where $\boldsymbol{\theta} = \{\mathbf{v}(\mathbf{x})\}_{\mathbf{x}\in\Omega}$ is a flow field on the image plane Ω at a set time, e.g., t_{ref}.

3.2 Multi-reference Focus Objective Function

Zhu et al. [27] report that the contrast objective (variance) overfits to the events. This is in part because the warp (4) can describe very complex flow fields, which can push the events to accumulate in few pixels (i.e., event collapse [25], [26]). To mitigate event collapse, we reduce the complexity of the flow field by dividing the image plane into a tile of non-overlapping patches, defining a flow vector at the center of each patch, and interpolating the flow on all other pixels (see Sec. 3.4). Interpolation confers smoothness of the flow field, hence lowering complexity.

However, reducing the complexity of the estimation parameters is not enough. Additionally, we discover that warps that produce sharp IWEs at *any* reference time t_{ref} have a regularizing effect on the flow field, discouraging event collapse. This is illustrated in Fig. 3. In practice we compute the *multi-reference* focus loss using three reference times: t_1 (min), $t_{\text{mid}} \doteq (t_1 + t_{N_e})/2$ (midpoint) and t_{N_e} (max). For each set of events, the flow field is defined only at one reference time and then used to warp to $\{t_1, t_{\text{mid}}, t_{N_e}\}$.

Letting G be the objective function at a single reference time (e.g., (3)), the proposed multi-reference focus objective function is the average of the G functions:

$$f(\boldsymbol{\theta}) \doteq \left(G(\boldsymbol{\theta};t_1) + 2G(\boldsymbol{\theta};t_{\text{mid}}) + G(\boldsymbol{\theta};t_{N_e})\right) / \,4G(\mathbf{0};-), \tag{5}$$

normalized by the value of the G function with zero flow (identity warp): $G(\mathbf{0};-)$. We could choose different convex combinations of normalized G functions and different reference times, but the proposed combination (5) works well in practice. The normalization in (5) provides the same interpretation as the Flow Warp Loss (FWL) [52]: $f < 1$ implies the flow is worse than the zero flow baseline, whereas $f > 1$ means that the flow produces sharper IWEs than the baseline. Such an interpretation is beneficial for model-based and unsupervised-learning methods.

Remark: Warping to two reference times (min and max) was proposed in [27], but with important differences: (i) it was done for the average timestamp loss, hence it did not consider the effect on contrast or focus functions [8], and (ii) it had a completely different motivation: to lessen a back-propagation scaling problem, so that the gradients of the loss would not favor events far from t_{ref}.

3.2.1 Objective Functions based on the IWE Gradient

Among the contrast functions proposed in [7], [8], we use two functions based on the gradient of the IWE:

$$G(\boldsymbol{\theta};t_{\text{ref}}) \doteq \frac{1}{|\Omega|}\int_\Omega \|\nabla I(\mathbf{x};t_{\text{ref}})\|^q\, d\mathbf{x}, \tag{6}$$

with $q = 1$ (the L^1-norm) and $q = 2$ (the squared L^2-norm). Both functions have the following desired properties: (i) they are sensitive to the arrangement (i.e., permutation) of the IWE pixel values, whereas the variance of the IWE (3) is not, (ii) they have top accuracy performance and converge more easily than other objectives we tested, and (iii) they differ from the FWL [52], which is defined using the variance (3) and will be used for evaluation. The two proposed functions have different sensitivities for the number of accumulated events on the IWE, which affects estimation accuracy, especially when the scene has large variations in the number of events per pixel (e.g., scenes with various depth). We find that using L^1 improves the results of L^2 [31] in most cases, as we show in Sec. 4.

3.3 Time-aware Flow

State-of-the-art event-based optical flow approaches are based on frame-based ones, and so they use the warp (4), which defines the flow $\mathbf{v}(\mathbf{x})$ as a function of $\mathbf{x}$ (i.e., a pixel displacement between two given frames). However, this does not take into account the space-time nature of events, which is the basis of CM, because not all events at a pixel $\mathbf{x}_0$ are triggered at the same timestamp t_k. They do not need to be warped with the same velocity $\mathbf{v}(\mathbf{x}_0)$. Figure 4 illustrates this with an occlusion example taken from the slider_depth sequence [63]. Instead of $\mathbf{v}(\mathbf{x})$, the *event-based flow* should be a function of space-time, $\mathbf{v}(\mathbf{x}, t)$, i.e, *time-aware*, and each event e_k should be warped according to the flow value at $(\mathbf{x}_k, t_k)$. Let us propose a more principled warp than (4).

To define a space-time flow $\mathbf{v}(\mathbf{x}, t)$ that is compatible with the propagation of events along motion curves, we are

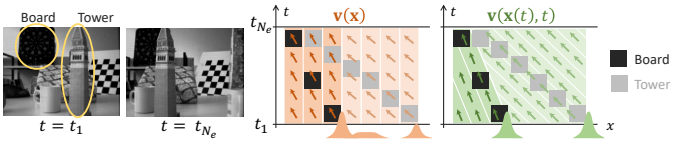


Figure 4: *Time-aware Flow.* Traditional flow (4), inherited from the frame-based one, assumes per-pixel constant flow $\mathbf{v}(\mathbf{x}) = \text{const}$, which cannot handle occlusions properly. The proposed space-time flow assumes constancy along streamlines, $\mathbf{v}(\mathbf{x}(t), t) = \text{const}$, which allows us to handle occlusions more accurately. (See results in Figs. 21 and 24).

inspired by the method of characteristics [64]. Mimicking the mainstream assumption about brightness being constant along the true motion curves in image space, we assume the flow is constant along its streamlines: $\mathbf{v}(\mathbf{x}(t), t) = \text{const}$ (Fig. 4). Differentiating in time and applying the chain rule gives a system of partial differential equations (PDEs):

$$\frac{\partial \mathbf{v}}{\partial \mathbf{x}} \frac{d\mathbf{x}}{dt} + \frac{\partial \mathbf{v}}{\partial t} = \mathbf{0}, \tag{7}$$

where, as usual, $\mathbf{v} = d\mathbf{x}/dt$ is the flow. The boundary condition is given by the flow at say $t = 0$: $\mathbf{v}(\mathbf{x}, 0) = \mathbf{v}^0(\mathbf{x})$. This system of PDEs states how to propagate (i.e., *transport*) a given flow $\mathbf{v}^0(\mathbf{x})$, from the boundary $t = 0$ to the rest of space-time. The PDEs have advection terms and others that resemble those of the inviscid Burgers' equation [64] since the flow is transporting itself. We parameterize the flow at $t = t_{\text{mid}}$ (boundary condition), and propagate it to the volume that encloses the current set of events $\mathcal{E}$. We develop two explicit methods to solve the PDEs, one with upwind differences and one with a conservative scheme adapted to Burgers' terms [65]. Each event e_k is then warped according to a flow $\hat{\mathbf{v}}$ given by the solution of the PDEs at $(\mathbf{x}_k, t_k)$:

$$\mathbf{x}'_k = \mathbf{x}_k + (t_k - t_{\text{ref}})\, \hat{\mathbf{v}}(\mathbf{x}_k, t_k). \tag{8}$$

3.4 Multi-scale Flow parameterization

Inspired by classical estimation methods, we combine our tile-based approach with a multi-scale strategy. The goal is to improve the convergence of the optimizer in terms of speed and robustness (i.e., avoiding local optima).

Some learning-based works [27], [28], [34] also have a multi-scale component, inherited from the use of a U-Net architecture. However, they work on discretized event representations (voxel grid, etc.) to be compatible with DNNs. In contrast, our tile-based approach works directly on raw events, without discarding or quantizing the temporal information in the event stream.

Our multi-scale CM approach is illustrated in Fig. 5. For an event set $\mathcal{E}_i$, we apply the tile-based CM in a coarse-to-fine manner (e.g., $N_\ell = 5$ scales). There are $2^{l-1} \times 2^{l-1}$ tiles at the l-th scale. We use bilinear interpolation to upscale between any two scales. If there is a subsequent set $\mathcal{E}_{i+1}$, the flow estimated from $\mathcal{E}_i$ is used to initialize the flow for $\mathcal{E}_{i+1}$. This is done by downsampling the finest flow to coarser scales. The coarsest scale initializes the flow for $\mathcal{E}_{i+1}$. For finer scales, initialization is computed as the average of the upsampled flow from the coarser scale of $\mathcal{E}_{i+1}$ and the same-scale flow from $\mathcal{E}_i$.

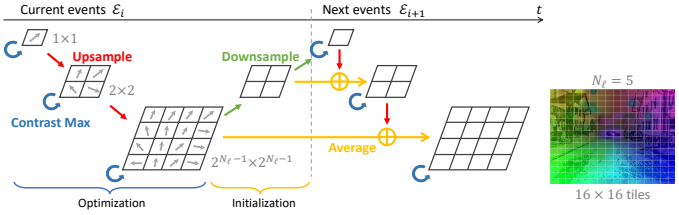


Figure 5: *Multi-scale Approach* using tiles (rectangles) and raw events. (See results in Fig. 22).

3.5 Composite Objective Function

To encourage additional smoothness of the flow, even in regions with few events, we include a flow regularizer $\mathcal{R}(\boldsymbol{\theta})$. The flow is obtained as the solution to the problem:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \left(\frac{1}{f(\boldsymbol{\theta})} + \lambda \mathcal{R}(\boldsymbol{\theta}) \right), \tag{9}$$

where, $\lambda > 0$ is the regularizer weight, and we use the total variation (TV) [66] as regularizer. We use $1/f$ instead of $-f$ because it is convenient for ANN training (Sec. 4.2.3).

3.6 Depth and Ego-Motion Estimation

3.6.1 Monocular

For a still scene but with a moving camera, the motion induced on the image plane has fewer DOFs than the most general case considered so far. In this scenario, it is beneficial to parameterize the optical flow in terms of the scene depth $Z(\mathbf{x})$ and the camera motion (linear velocity $\mathbf{V}$ and angular velocity $\boldsymbol{\omega}$) via the well-known motion field equation [6]:

$$\mathbf{v}(\mathbf{x}) = \frac{1}{Z(\mathbf{x})} A(\mathbf{x}) \mathbf{V} + B(\mathbf{x}) \boldsymbol{\omega}, \tag{10}$$

where the 2×3 matrices $A(\mathbf{x})$ and $B(\mathbf{x})$ solely depend on the pixel coordinate. Substituting (10) in (4) or (8) and using it to warp events yields that the contrast is now maximized with respect to the depth and camera motion parameters while the flow $\mathbf{v}$ acts as an intermediate variable.

Similarly to Sec. 3.4, we parameterize the depth $Z(\mathbf{x})$ using a tile of patches, which results in $6 + N_{\text{patch}}$ DOFs (instead of $2N_{\text{patch}}$ DOFs). By doing this, we not only reduce the complexity of the estimation but also demonstrate the extensibility of the proposed method to the simultaneous estimation of ego-motion and dense depth. Note that parameters $Z(\mathbf{x})$ and $\mathbf{V}$ appear in a product in (10), hence there is a scale ambiguity (typical of monocular setups). Furthermore, we apply an exponential parameterization $\rho \mapsto Z \doteq e^{a\rho+b}$ to avoid negative depth predictions. To mitigate isolated patches with very large depth values we apply median filters [35] and a Charbonnier loss [67] for regularization.

Note that the motion field parameterization (10) is not supposed to handle independently moving objects (IMOs), although it is effective in many event-based optical flow benchmarks (e.g., Secs. 4.2.1 and 4.2.2). We discuss the validity and the limitations of optical flow benchmarking in Sec. 4.2.5, as well as the comprehensive results in Sec. 4.4.

3.6.2 Stereo

The proposed method can be extended to stereo configurations. Parameterizing scene depth and ego-motion on the left camera and using the extrinsic parameters of the stereo setup, we can compute the depth and the motion on the right camera (e.g., by warping the left depth map onto the right camera using the nearest neighbor interpolation). Having depth and ego-motion on each camera, we define the objective function as the sum:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \left(\frac{1}{f_{\mathrm{l}}(\boldsymbol{\theta})} + \lambda \mathcal{R}_{\mathrm{l}}(\boldsymbol{\theta}) + \frac{1}{f_{\mathrm{r}}(\boldsymbol{\theta})} + \lambda \mathcal{R}_{\mathrm{r}}(\boldsymbol{\theta}) \right), \quad (11)$$

where the parameters $\boldsymbol{\theta}$ are only those of the left camera.

In prior works of stereo depth estimation [68], one of the main challenges is how to find correspondences between event streams from multiple cameras. This is a non-trivial problem and is prone to event noise. The proposed method bypasses the event-to-event correspondence problem by parameterizing the depth densely on the whole image plane of one camera and transferring it to the other camera.

Summarizing Remark: All the proposals in Sec. 3 are formulated in the form of an optimization problem, and they are theoretically extensible to learning-based approaches (DNNs), since they are fully differentiable. We will show an example of the learning-based flow estimation in Sec. 4.2.3. Hence, our work provides model-based approaches that can act as baselines for the development of learning-based methods in the context of event-based optical flow, monocular depth, ego-motion, and stereo depth estimation problems.

4 Experiments

We assess the performance of our method on *seven* datasets, which are presented in Sec. 4.1. We provide a comprehensive evaluation of optical flow estimation in Sec. 4.2. Additionally, we demonstrate the learning-based extension (DNN) (Sec. 4.2.3), discuss current optical flow benchmarks (Sec. 4.2.5), and show downstream applications (Sec. 4.3). The results of depth and ego-motion estimation are presented in Sec. 4.4 (monocular) and Sec. 4.5 (stereo).

4.1 Datasets, Metrics and Hyper-parameters

The proposed method works robustly on data comprising different camera motions, scenes, and spatial resolutions. We conduct experiments on the following seven datasets.

Datasets. First, we evaluate our method on sequences from the **MVSEC** dataset [4], [34], which is the de facto standard dataset used by prior works to benchmark optical flow. The dataset contains sequences recorded indoors with a drone, and outdoors with a car. It provides events, grayscale frames, IMU data, camera poses, and scene depth from a LiDAR [4]. The dataset was extended in [34] to provide ground truth (GT) optical flow, computed as the motion field [6] given the camera velocity and the depth of the scene. Notice that the indoor sequences do not have IMOs, and the outdoor sequences do not include scenes with IMOs in the benchmark evaluation. The event camera has 346×260 pixel resolution [56]. In total, we evaluate on 63.5 million events spanning 265 seconds. We quantitatively and qualitatively show results on flow, depth, and ego-motion estimation.

We also evaluate on a recent dataset that provides ground truth flow: **DSEC** [44]. It consists of sequences recorded with Prophesee Gen3 event cameras (stereo), of higher resolution (640×480 pixels), mounted on a car. Optical flow is also computed as the motion field, with the scene depth given by a LiDAR. The flow benchmark contains scenes with IMOs, but performance is assessed only in non-IMO pixels (where the GT from the motion field is valid). In total, we evaluate on 3 billion events spanning the 208 s of the test sequences. We quantitatively/qualitatively show results of flow and stereo depth estimation.

Additionally, we carry out experiments on two HD resolution event camera datasets, **TUM-VIE** [32] and **M3ED** [33], recorded with stereo Prophesee Gen4 event cameras (1280×720 pixels, i.e., 1 Mpixel). The TUM-VIE dataset consists of indoor and outdoor sequences recorded with the sensor rig mounted on a helmet. In the M3ED dataset the sensor rig is mounted on a car (outdoor), a quadruped robot (outdoor), and a drone (indoor and outdoor). We show qualitative results for the flow and depth estimation since the GT data for M3ED is not available at submission time.

The **ECD** dataset [63] is a lower resolution, standard dataset to assess camera ego-motion [9], [16], [25], [69]–[72]. Each sequence provides events, frames, calibration information, and IMU data from a DAVIS240C camera (240×180 pixels [73]), as well as ground truth camera poses from a motion capture system (at 200Hz). We use *slider_depth* and *simulation_3planes* sequences for depth and ego-motion estimation. In the first sequence the event camera moves along a motorized linear slider, recording objects at different depths. The second sequence is synthetic with a circular camera trajectory; since it provides ground truth depth, we report quantitative metrics for depth and ego-motion estimation accuracy. In total, we evaluate on 1.1 million events (3 s) of the slider sequence and on 6.8 million events (2 s) of the simulation sequence.

Finally, we also test sequences from two motion segmentation datasets [20], [21]. The sequences in **EMSMC** [20] are recorded using a hand-held DAVIS240C camera (240×180 pixels). The sequences in **EMSGC** [21] are recorded with a hand-held DAVIS346 camera (346×260 pixels). Both datasets consist of small camera motions and several IMOs in the scene. We demonstrate qualitative results of flow estimation and its application to motion segmentation.

Evaluation Metrics. The metrics used to assess optical flow accuracy are the average endpoint error (AEE), the angular error (AE), and the percentage of pixels with AEE > 3 pixels (denoted by "% Out"), all measured over pixels with valid GT and at least one event in the evaluation interval. We also use the FWL metric (the IWE variance relative to that of the identity warp) to assess event alignment [52].

For depth accuracy evaluation, we use standard metrics following previous work on monocular depth estimation [57], [74]. The depth error metrics are SiLog, Absolute Relative Difference (denoted by "AbsRelDiff"), and the logarithmic RMSE ("logRMSE"). While SiLog is scale-invariant, we substitute the prediction using the mean of the GT for AbsRelDiff and logRMSE. We furthermore report depth accuracy metrics that compute the percentage of pixels whose relative depth with respect to GT is smaller than a threshold. We use three common thresholds: $\delta < \{1.25, 1.25^2, 1.25^3\}$,

denoted by "A1", "A2" and "A3", respectively.

Hyper-parameters. For flow estimation our method uses $N_\ell = 5$ resolution scales, $\lambda = 0.0025$ in (9), and the Newton-CG optimization algorithm with a maximum of 30 iterations per scale. The flow at t_{mid} is transported to each side via the upwind or Burgers' PDE solver (using 5 bins for MVSEC, 40 for DSEC), and used for event warping (8) (see [31]). In the optimization, we use 30k events for MVSEC indoor sequences, 40k events for outdoors, 50k events for ECD, 1.5M events for DSEC, 1.8M events for TUM-VIE and M3ED, and 5k events for the motion segmentation examples.

The number of events was selected guided by the benchmarks and/or experimentally, based on the variables that affect the event generation (camera's spatial resolution, scene texture, motion, etc.) and the CM method (edges should displace enough, e.g., three pixels, see [18]). The estimated flow is scaled and aligned with the benchmark timestamps, if necessary (e.g., MVSEC). There is a trade-off: too few events, then CM does not work (scarce data and there is not enough displacement to have a good objective function landscape); too many events, and the method may not produce a good fit if the constant optical flow assumption does not hold during the time span of the events.

Since the motion-field parameterization reduces the complexity of the problem, we successfully use finer scales $N_\ell = 6$ for MVSEC/DSEC and $N_\ell = 7$ for the 1 Mpixel datasets. By increasing the patch level in static scenes, we expect finer and better flow estimates. While we initialize depth between event packets with the same strategy as that of optical flow, we do not propagate the linear velocity to the subsequent packet in order to avoid errors when abrupt motion changes happen (e.g., during velocity sign changes).

4.2 Optical Flow Estimation

4.2.1 Results on the MVSEC benchmark

We first report the results on the MVSEC benchmark (Table 1). The different methods (rows) are compared on one outdoor and three indoor sequences (columns). This is because many learning-based methods train on the other outdoor sequence, which is therefore not used for testing. Following Zhu et al., outdoor_day1 is tested only on specified 800 frames [34]. The top part of Tab. 1 reports the flow corresponding to a time interval of $dt = 1$ grayscale frame (at $\approx$ 45Hz, i.e., 22.2ms), and the bottom part corresponds to $dt = 4$ frames (89ms).

The table is comprehensive, showing where the proposed methods stand compared to prior work. Our methods provide the best results among all methods in all indoor sequences and are the best among the unsupervised and model-based methods in the outdoor sequence. The errors for $dt = 4$ are about four times larger than those for $dt = 1$, which is sensible given the ratio of time interval durations.

Among different variations of the proposed methods, we observe that (i) the motion field parameterization achieves better accuracy than the direct parameterization of the flow in indoor sequences, (ii) there are no significant differences between the three versions of the flow warp models, and (iii) the L^1 loss improves accuracy over L^2. Elaborating on these three points, (i) the effectiveness of the motion field estimation indoors is due to a good match between the model assumptions and the data (there are no IMOs in the scene), and outdoors depth estimation is generally difficult for driving sequences. (ii) The negligible difference between the flow warp models can be attributed to the fact that the MVSEC dataset does not comprise large pixel displacements or occlusions, which is further discussed in Sec. 5.2. (iii) The L^1 norm grows more slowly than the L^2 norm along the increased number of accumulated events in the IWE. This property makes the L^1 objective function more sensitive to the areas with few events (e.g., pixels of far away objects), resulting in better estimation accuracy.

Qualitative results are shown in Fig. 6, where we compare our method against the state-of-the-art learning-based methods. Our method provides sharper IWEs than the baselines, without overfitting, and the estimated flow resembles the GT. We display flow masked by the events, for consistency with the benchmark. Recall that our method interpolates the flow at pixels with zero events. The USL result [30] is obtained using its official implementation, comprising a recurrent model that sequentially processes sub-partitions of event data. Notice that we use the event mask of the full timestamps ($dt = 4$), which agrees with the quantitative evaluation for a consistent discussion.

Ground truth is not available on the entire image plane (see Fig. 6), such as in pixels not covered by the LiDAR's range, FOV, or spatial sampling. Additionally, there may be interpolation issues in the GT, since the LiDAR works at 20 Hz and the GT flow is given at frame rate (45 Hz). In the outdoor sequences, the GT from the LiDAR and the camera motion cannot provide correct flow for IMOs. These issues of the GT are noticeable in the IWEs: they are not as sharp as expected. In contrast, the IWEs produced by our method are sharp.

4.2.2 Results on the DSEC benchmark

Table 2 gives quantitative results on the DSEC Optical Flow benchmark. No GT flow is available for these test sequences. The proposed methods are compared with an unsupervised-learning method [59] (Sec. 2) and a supervised-learning method E-RAFT [44]. E-RAFT is an ANN that extracts features in event correlation volumes via an iterative update scheme instead of using a U-Net architecture. This version of RAFT [48] was introduced along with the DSEC flow benchmark and showed it can estimate pixel correspondences for large displacements. As expected, E-RAFT is better than ours in terms of flow accuracy because (i) it has additional training information (GT labels), and (ii) it is trained using the same type of GT signal used in the evaluation. Nevertheless, our method provides sensible results and is better in terms of FWL, which exposes similar GT quality issues as those of MVSEC: many pixels have no GT (LiDAR's FOV and IMOs). This is also confirmed in the qualitative results (Fig. 7). Our method provides sharp IWEs, even for IMOs (car) and the road close to the camera. We further discuss the issue of IMOs in the flow benchmarks in Sec. 4.2.5.

Remarkably, the proposed methods achieve competitive results in terms of flow accuracy with the unsupervised-learning method [59]. Among different variations, the "Flow (L^1)" achieves the most competitive results for all sequences except for zurich_city_12a, a night sequence. The night

Table 1: Results on MVSEC dataset [34]. Methods are sorted according to how much data they need: supervised learning (SL) requires ground truth flow; semi-supervised learning (SSL) uses grayscale images for supervision; unsupervised learning (USL) uses only events; and model-based (MB) needs no training data. Bold is the best among all methods; underlined is second best. The results of Nagata et al. [42] are scaled to $dt = 1$.

			indoor_flying1		indoor_flying2		indoor_flying3		outdoor_day1	
			AEE ↓	%Out ↓	AEE ↓	%Out ↓	AEE ↓	%Out ↓	AEE ↓	%Out ↓
Interval duration $dt = 1$	SL	EV-FlowNet-EST [51]	0.97	0.91	1.38	8.20	1.43	6.47	–	–
		EV-FlowNet+ [52]	0.56	1.00	0.66	1.00	0.59	1.00	0.68	0.99
		E-RAFT [44]	–	–	–	–	–	–	0.24	1.70
		E-RAFT [50]	1.10	5.72	1.94	30.79	1.66	25.20	0.24	**0.00**
		DCEIFlow [75]	0.75	1.55	0.90	2.10	0.80	1.77	**0.22**	**0.00**
		TMA [50]	1.06	3.63	1.81	27.29	1.58	23.26	0.25	0.07
		EVA-Flow [76]	–	–	–	–	–	–	0.25	0.00
		Spiking-UNet [49]	0.58	–	0.72	–	0.67	–	1.97	–
		ADM-Flow [54]	0.52	0.14	0.68	1.18	0.52	0.04	0.41	0.00
	SSL	EV-FlowNet (original) [34]	1.03	2.20	1.72	15.10	1.53	11.90	0.49	0.20
		Spike-FlowNet [46]	0.84	–	1.28	–	1.11	–	0.49	–
		Ziluo et al. [45]	0.57	0.10	0.79	1.60	0.72	1.30	0.42	**0.00**
	USL	EV-FlowNet [27]	0.58	**0.00**	1.02	4.00	0.87	3.00	0.32	**0.00**
		EV-FlowNet (retrained) [28]	0.79	1.20	1.40	10.90	1.18	7.40	0.92	5.40
		FireFlowNet [28]	0.97	2.60	1.67	15.30	1.43	11.00	1.06	6.60
		ConvGRU-EV-FlowNet [30]	0.60	0.51	1.17	8.06	0.93	5.64	0.47	0.25
		ET-FlowNet [58]	0.57	0.53	1.20	8.48	0.95	5.73	0.39	0.12
		EV-MGRFlowNet [77]	0.41	0.17	0.70	2.35	0.59	1.29	0.28	0.02
		ConvGRU-EV-FlowNet [59]	0.44	**0.00**	0.88	4.51	0.70	2.41	0.27	0.05
	MB	Akolkar et al. [41]	1.52	–	1.59	–	1.89	–	2.75	–
		Shiba et al. [78]	1.05	2.90	1.68	13.44	1.43	8.97	0.94	3.08
		Nagata et al. [42]	0.62	–	0.93	–	0.84	–	0.77	–
		Brebion et al. [79]	0.52	0.10	0.98	5.50	0.71	2.10	0.53	0.20
		Cuadrado et al. [49]	0.58	–	0.72	–	0.67	–	0.85	–
		Ours (w/o time aware)	0.42	0.09	0.60	0.59	0.50	0.29	0.30	0.11
		Ours (Upwind)	0.42	0.10	0.60	0.59	0.50	0.28	0.30	0.10
		Ours (Burgers')	0.42	0.10	0.60	0.59	0.50	0.28	0.30	0.10
		Ours (L^1)	0.37	0.04	0.53	0.08	0.44	0.02	0.30	0.11
		Ours (Motion field, L^2)	**0.30**	**0.00**	0.50	**0.00**	0.36	**0.00**	0.32	0.19
		Ours (Motion field, L^1)	**0.30**	**0.00**	**0.47**	0.01	**0.34**	**0.00**	0.28	0.21
Interval duration $dt = 4$	SL	E-RAFT [50]	2.81	40.25	5.09	64.19	4.46	57.11	0.72	1.12
		DCEIFlow [75]	2.08	21.47	3.48	42.05	2.51	29.73	0.89	3.19
		TMA [50]	2.43	29.91	4.32	52.74	3.60	42.02	**0.70**	**1.08**
		EVA-Flow [76]	–	–	–	–	–	–	0.82	2.41
		ADM-Flow [54]	1.42	7.78	1.88	16.70	1.61	11.40	1.51	10.20
	SSL	EV-FlowNet (original) [34]	2.25	24.70	4.05	45.30	3.45	39.70	1.23	7.30
		Spike-FlowNet [46]	2.24	–	3.83	–	3.18	–	1.09	–
		Ziluo et al. [45]	1.77	14.70	2.52	26.10	2.23	22.10	0.99	3.90
	USL	EV-FlowNet [27]	2.18	24.20	3.85	46.80	3.18	47.80	1.30	9.70
		ConvGRU-EV-FlowNet [30]	2.16	21.51	3.90	40.72	3.00	29.60	1.69	12.50
		ET-FlowNet [58]	2.08	20.02	3.99	41.33	3.13	31.70	1.47	9.17
		EV-MGRFlowNet [77]	1.50	8.67	2.39	23.70	2.06	18.00	1.10	6.22
	MB	Shiba et al. [78]	4.06	53.88	6.39	71.82	5.36	65.57	3.60	49.04
		Ours (w/o time aware)	1.68	12.79	2.49	26.31	2.06	18.93	1.25	9.19
		Ours (Upwind)	1.69	12.83	2.49	26.37	2.06	19.02	1.25	9.23
		Ours (Burgers')	1.69	12.95	2.49	26.35	2.06	19.03	1.25	9.21
		Ours (L^1)	1.48	8.27	2.10	20.42	1.73	12.81	1.23	9.22
		Ours (Motion field, L^2)	1.22	5.00	2.03	19.17	1.42	8.15	1.35	10.53
		Ours (Motion field, L^1)	**1.18**	**4.77**	**1.87**	**15.51**	**1.38**	**7.26**	1.05	5.68

scenes have many light-induced events that are not due to motion, and naturally the proposed methods tend to fail.

Notice that both DNN methods [44], [59] train and evaluate on the DSEC dataset, which is dominantly forward driving motion. As a result, these learning-based methods may overfit to the driving data (i.e., tend to predict forward motion) and fail to produce good results in other motions and datasets [55] (e.g., see E-RAFT rows on the MVSEC indoor seqs. in Tab. 1). On the contrary, the proposed methods rely on the principle of event alignment and generalize to various datasets, producing consistently good results.

Similarly to the MVSEC results, the L^1 loss achieves better accuracy than the L^2 loss. Contrary to MVSEC, the results of the depth parameterization are generally worse than those of the flow parameterization. This can be attributed to the IMOs: although not included in the evaluation pixels, the scenes include IMOs which directly affect the estimated flow. As expected, the motion field estimation fails since it cannot fit the events caused by IMOs.

We observe that the evaluation intervals (100ms) are large for optical flow standards. In the benchmark, 80% of the GT flow has up to 22px displacement, which means that 20% of the GT flow is larger than 22px (on VGA resolution). The apparent motion during such intervals is sufficiently

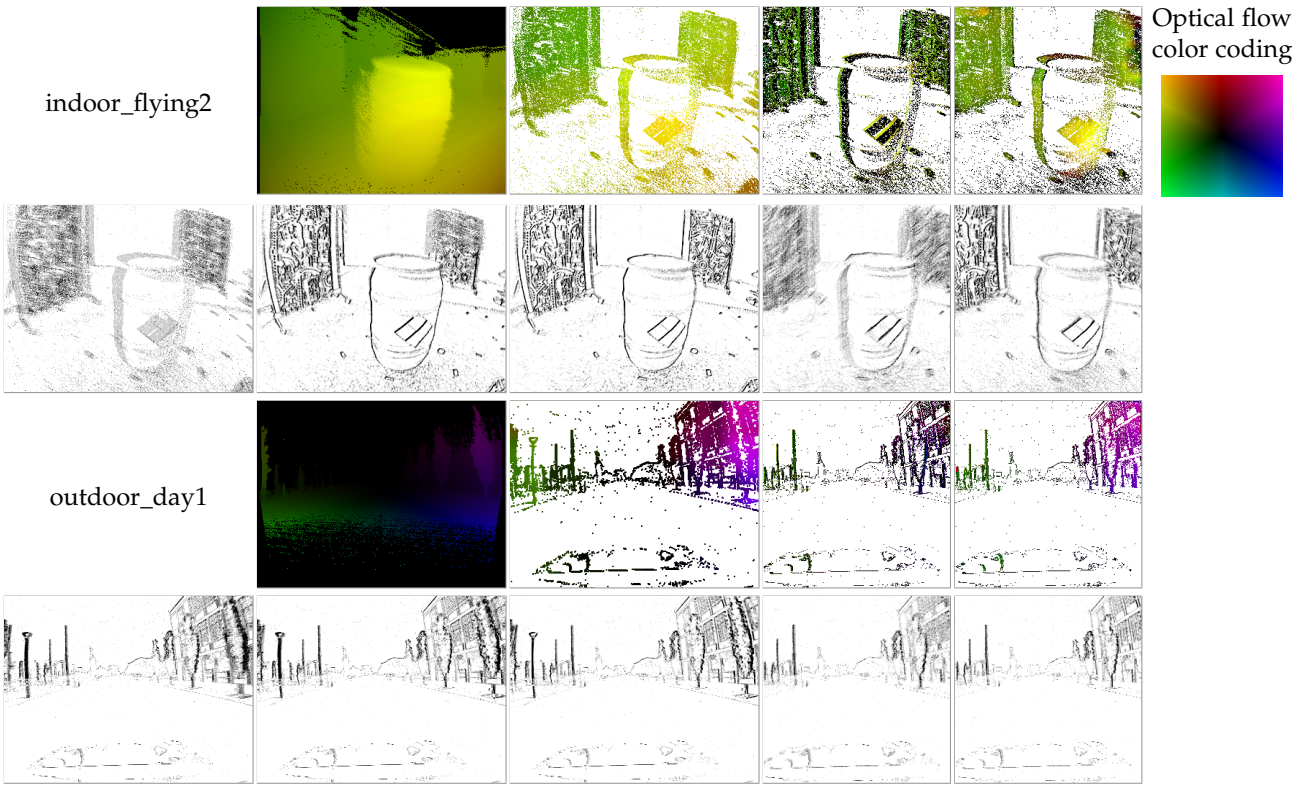


Figure 6: *MVSEC results* ($dt = 4$) of our method and two state-of-the-art baselines: ConvGRU-EV-FlowNet (USL) [30] and EV-FlowNet (SSL) [34]. For each sequence, the upper row shows the flow masked by the input events, and the lower row shows the IWE using the flow. Our method produces the sharpest motion-compensated IWEs. Note that learning-based methods crop the events to the central 256 × 256 pixels, whereas our method does not. Black points in ground truth (GT) flow maps indicate the absence of LiDAR data. Additional plots are given in [31, Fig. 5].

Table 2: Results on the DSEC optical flow benchmark [44].

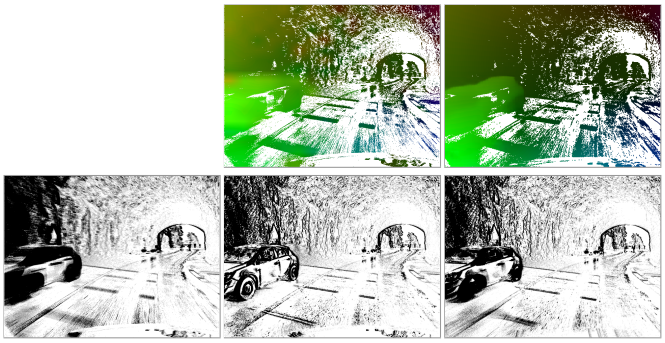
	All				interlaken_00_b				interlaken_01_a				thun_01_a			
	AEE ↓	AE ↓	%Out ↓	FWL ↑	AEE ↓	AE ↓	%Out ↓	FWL ↑	AEE ↓	AE ↓	%Out ↓	FWL ↑	AEE ↓	AE ↓	%Out ↓	FWL ↑
E-RAFT (SL) [44]	0.79	10.56	2.68	1.29	1.39	6.22	6.19	1.32	0.90	6.88	3.91	1.42	0.65	9.75	1.87	1.20
Paredes et al. (USL) [59]	**2.33**	**10.56**	**17.77**	–	**3.34**	6.22	**25.72**	–	**2.49**	**6.88**	**19.15**	–	**1.73**	9.75	**10.39**	–
Ours (USL, L^1)	3.69	12.62	34.62	–	4.37	6.82	36.81	–	3.45	8.54	35.08	–	2.02	**7.51**	17.53	–
Ours (Motion field, L^2)	5.01	14.38	42.60	1.43	6.02	11.75	49.59	1.64	8.52	13.20	58.97	1.41	2.81	12.77	29.89	1.37
Ours (Motion field, L^1)	4.26	12.05	37.05	1.42	3.93	**5.89**	35.14	**1.66**	7.89	11.08	63.98	1.34	1.85	8.20	14.78	**1.37**
Ours (Flow, L^2)	3.47	13.98	30.86	1.37	5.74	9.19	38.93	1.50	3.74	9.77	31.37	1.51	2.12	11.06	17.68	1.24
Ours (Flow, L^1)	3.51	12.31	24.18	**1.47**	5.43	7.76	34.47	1.63	2.99	7.59	23.85	**1.63**	1.84	9.46	13.77	1.35

	thun_01_b				zurich_city_12_a				zurich_city_14_c				zurich_city_15_a			
	AEE ↓	AE ↓	%Out ↓	FWL ↑	AEE ↓	AE ↓	%Out ↓	FWL ↑	AEE ↓	AE ↓	%Out ↓	FWL ↑	AEE ↓	AE ↓	%Out ↓	FWL ↑
E-RAFT (SL) [44]	0.58	8.41	1.52	1.18	0.61	23.16	1.06	1.12	0.71	10.23	1.91	1.47	0.59	8.88	1.30	1.34
Paredes et al. (USL) [59]	**1.66**	8.41	**9.34**	–	**2.72**	**23.16**	**26.65**	–	2.64	10.23	**23.01**	–	**1.69**	8.88	**9.98**	–
Ours (USL, L^1)	3.08	8.16	31.84	–	5.34	32.89	46.89	–	3.00	8.70	32.43	–	2.94	8.72	26.93	–
Ours (Motion field, L^2)	2.90	8.20	28.79	1.37	3.91	29.73	41.63	1.13	3.01	9.95	32.64	**1.57**	3.16	9.75	32.63	1.52
Ours (Motion field, L^1)	2.21	**6.30**	19.39	**1.38**	5.28	46.19	53.25	1.11	2.76	**9.19**	28.54	1.56	2.40	**7.38**	18.64	1.53
Ours (Flow, L^2)	2.48	12.05	23.56	1.24	3.86	28.61	43.96	1.14	2.72	12.62	30.53	1.50	2.35	11.82	20.99	1.41
Ours (Flow, L^1)	1.98	9.63	16.74	1.38	6.36	28.82	35.18	**1.20**	**2.35**	10.53	23.72	1.56	1.98	9.54	15.50	**1.55**

large that it breaks the classical assumption of scene points flowing in linear trajectories (more details in Sec. 4.2.5).

4.2.3 *Application to Deep Neural Networks (DNN)*

The proposed secrets are not only applicable to model-based methods, but also to unsupervised-learning methods. To this end, we train EV-FlowNet [34] in an unsupervised



(a) Events (b) Ours (MB) (c) SL [44]

Figure 7: *DSEC results* on the interlaken_00b sequence (no GT available). Since GT is missing at IMOs and points outside the LiDAR's FOV, the supervised method [44] may provide inaccurate predictions around IMOs and road points close to the camera, whereas our method produces sharp edges. For visualization, we use 1M events.

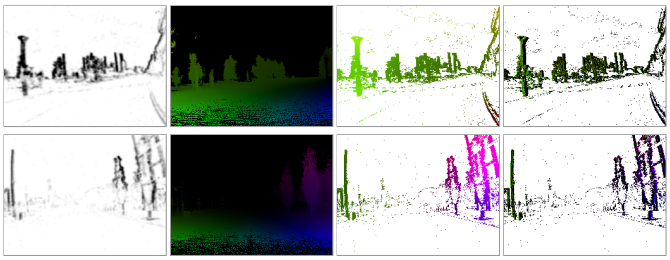
Table 3: Results of unsupervised learning methods on MVSEC's outdoor_day1 sequence.

	$dt = 1$			$dt = 4$		
	AEE ↓	%Out ↓	FWL ↑	AEE ↓	%Out ↓	FWL ↑
EV-FlowNet [27]	**0.32**	**0.00**	–	**1.30**	**9.70**	–
EV-FlowNet (retrained) [28]	0.92	5.40	–	–	–	–
ConvGRU-EV-FlowNet [30]	0.47	0.25	0.94	1.69	12.50	0.94
Our EV-FlowNet (9)	0.36	0.09	**0.96**	1.49	11.72	**1.11**

manner on the MVSEC dataset, using (9) as data-fidelity term and a Charbonnier loss [67] as the regularizer. We convert 40k events into the voxel-grid representation [27] with 5 time bins. The network is trained for 50 epochs with a learning rate of 0.001 and its decay of 0.8 with Adam optimizer [80]. To ensure generalization, we train our network on indoor sequences and test on the outdoor_day1 sequence. Since the time-aware flow does not have a significant influence on the MVSEC benchmark (Tab. 1), we do not port it to the learning-based setting.

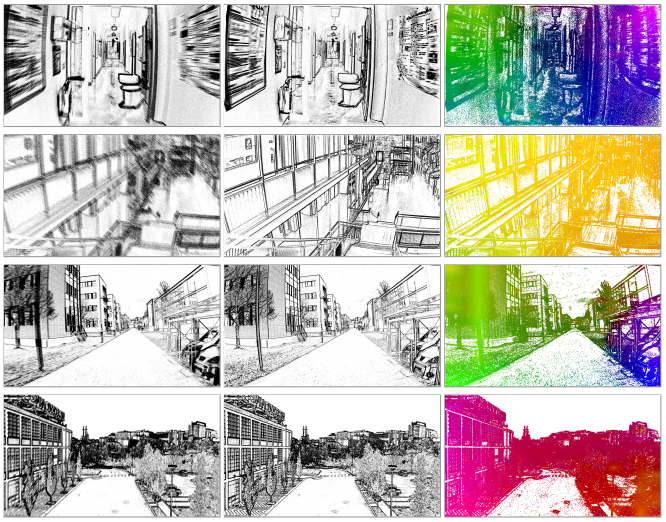
Table 3 shows the quantitative comparison with unsupervised learning methods. Our model achieves the second best accuracy, following [27], and the best sharpness (FWL) among the existing methods. Notice that [27] was trained on the outdoor_day2 sequence, which is a similar driving sequence to the test one, while the other methods were trained on drone data [81]. Hence [27] might be overfitting to the driving data, while ours is not, by the choice of training data. The qualitative results of our unsupervised learning setting are shown in Fig. 8. We compare our method with the state-of-the-art unsupervised learning [30]. Our results resemble the GT flow.

Additionally, we train the architecture in [59] on DSEC data using the L^1 loss and the Charbonnier loss (with the regularizer weight of 0.15). The accuracy results, reported in Tab. 2 as "Ours (USL, L^1)", are on par with the model-based one. The two experiments in this section (Sec. 4.2.3) confirm the transferability of the techniques in Sec. 3 to learning-based approaches, reaffirming the importance of our contributions.



(a) Events (b) GT (c) Our EV-FlowNet (d) USL [30]

Figure 8: *Results of our DNN on the MVSEC outdoor sequence*. Our DNN (EV-FlowNet architecture) trained with (9) outperforms the unsupervised learning method [30].



(a) Events (b) IWE (c) Estimated flow

Figure 9: *Results on 1Mpixel event camera data*. Sequences are *bike-easy*, *skate-easy* (TUM-VIE [32]), and *falcon* (M3ED [33]).

4.2.4 Results on 1 Mpixel Datasets: TUM-VIE and M3ED

The proposed method generalizes to recent high spatial resolution event cameras. We show qualitative results on the TUM-VIE dataset [32] and the M3ED dataset [33] in Fig. 9. The flow looks realistic and produces sharp IWEs for various motions (forward motion, rotation, translation) and scenes (indoor and outdoor). Also, the flow estimation is stable regardless of the absolute scene intensity, while frames suffer from a limited dynamic range. Hence, we leverage the HDR advantages of event cameras.

4.2.5 Discussion on optical flow benchmarks and "GT" flow

Throughout the quantitative evaluation of the event-based optical flow (Secs. 4.2.1 and 4.2.2), we observe some limitations for the current benchmarks: (i) size of the evaluation interval and (ii) independently moving objects (IMOs).

Evaluation intervals and the linearity of optical flow. The time-aware flow is designed to consider the space-time nature of events. Recently, there have also been other proposals aiming to leverage such nature for per-pixel motion estimation. The main difference between our flow (Sec. 3.3)

and concurrent proposals [53], [59], [82] is the motion hypothesis and its underlying assumptions: (7) assumes that the flow is constant along its streamlines within short time intervals, which produces linear motion trajectories (Fig. 4). The number DOFs of the motion is $2N_p$, and the efficacy of the parameterization for occlusions is shown in Sec. 5.2.

On the other hand, [53], [59] propose non-linear trajectories (e.g., Bézier curves) for the "optical flow". We suspect that the choice of assuming non-linear trajectories stems from the necessity of reporting good figures on the DSEC benchmark (Tab. 2), which has relatively long evaluation intervals. While it is called an "optical flow" benchmark, the ground truth on time intervals of 100 ms at moderate vehicle speeds can result in curved trajectories. The increased complexity of the non-linear trajectory estimation problem has several challenges to be addressed: (i) accuracy is difficult to evaluate with existing benchmarks, which are based on the standard definition of flow, (ii) there is a trade-off between the increased complexity of possible motions and the tendency to overfit, (iii) it is important to assess the efficacy of the curved trajectory in terms of downstream applications. We show various applications of the linear trajectory in Secs. 4.3 to 4.5; for curved trajectories, beyond focusing on beating the current benchmark, it would be interesting to show new applications. Finally, it is worth reconsidering the terminology of the estimation task, such as "instantaneous" (short-baseline) optical flow, vs. "non-instantaneous" (i.e., large-baseline) curved trajectory estimation.

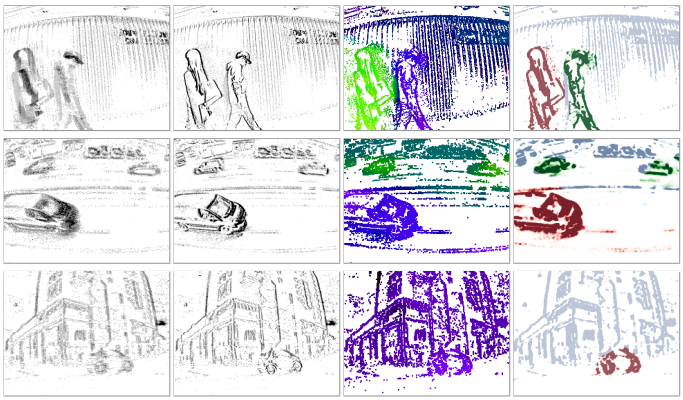
IMOs. The de facto standard flow benchmarks MVSEC and DSEC ignore pixels corresponding to IMOs (because it is difficult to obtain GT labels for IMOs in the real-world). However, optical flow can describe such motions. Indeed, as Tab. 1 shows, the motion-field–parameterized flow achieves better accuracy in still scenes. Training ANNs using only flow from rigid scenes may affect their learning capabilities. To avoid potential pitfalls of optical flow algorithms, it is therefore important that the data used for (training and) evaluation contains IMOs and a variety of ego-motions.

4.3 Applications of Optical Flow

This section demonstrates three exemplary applications of the estimated optical flow: motion segmentation, intensity reconstruction, and denoising.

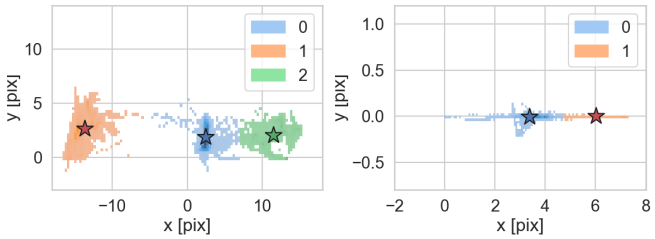
4.3.1 Motion Segmentation

Motion segmentation is the task of splitting a scene into objects moving with different velocities. Thus, it is natural to address it by clustering optical flow [20]. To this end, we show results on three sequences from [20], [21] in Fig. 10 using k-means with 2 to 3 clusters. In the *corridor* scene (first row of Fig. 10) there are 3 clusters: two people are walking in opposite directions while the camera is moving (background). In the second example, the scene includes cars with horizontal motion while the camera tilts. The third example (*car*) has a car moving at a different speed in the same direction as the background, which is the most challenging case among these examples. In all examples, our method successfully provides sensible segmentation masks (last column of Fig. 10) corresponding to the scene objects.



(a) Events (b) IWE (c) Flow (d) Clusters

Figure 10: *Motion Segmentation*. First row: *corridor* sequence from [21]. Second and third rows are sequences from [20].



(a) corridor [21] (b) car [20]

Figure 11: *Visualization of the flow clustering* on the first and third examples in Fig. 10. The stars denote the cluster centroids. Cluster 0 (blue) corresponds to the background, while clusters 1 and 2 are independently moving objects.

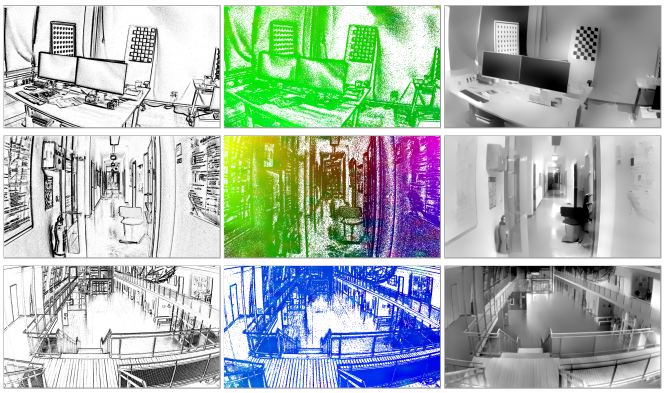
Figure 11 provides detailed analyses of the clustering operation for the *corridor* and *car* examples. Since the proposed method uses a tile-based parameterization of the flow, the interpolation between tiles produces flow vectors that fill in the regions between the distinctive cluster centroids. One could use other clustering algorithms, such as DBSCAN [83], to treat such interpolation effects as outliers.

4.3.2 Image Reconstruction

Events encode the apparent motion of scene edges (e.g., optical flow) as well as their brightness. These two quantities are entangled, and it is possible to use computed optical flow to recover brightness, i.e., reconstruct intensity images [24]. We demonstrate it on a 1 Mpixel dataset in Fig. 12. The estimated flow provides sharp IWEs, which successfully aids reconstruct intensities such as the checkerboard on the wall, the light and its reflection on the corridor, and the complex structure of the stairs. The results are remarkable despite the noise in the corridor scene (see Sec. 4.3.3). Due to the regularizer in [24], the very fine structure (e.g., the poster contents) might not be crisp.

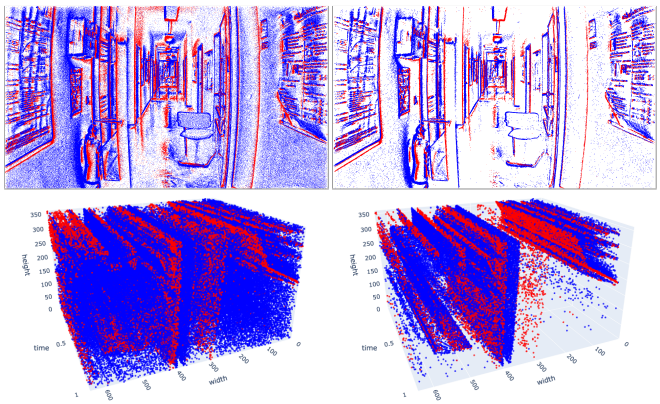
4.3.3 Denoising Event Data

By extending the idea of [84], which classifies events for temporal upsampling into signal or noise based on a predicted 2-DOF motion, we use the estimated optical flow



(a) IWE (b) Flow (c) Reconstruction

Figure 12: *Image reconstruction* after optical flow estimation. Data from the 1Mpixel TUM-VIE dataset [32].



(a) Original Events (b) Denoised Events

Figure 13: *Denoising*. The data is the *skate-easy* sequence from the TUM-VIE dataset. The top row is the image representation of the events, while the bottom row shows them in space-time coordinates (for better visualization, only the bottom-right quarter of the image plane is displayed).

to identify noise events as those where the IWE is smaller than some value (e.g., 3 events). Figure 13 shows qualitative results. The corridor scene has a large amount of noise due to lighting (i.e., flickering events). The denoised event data looks clearer, while it retains the edge structure of the scene.

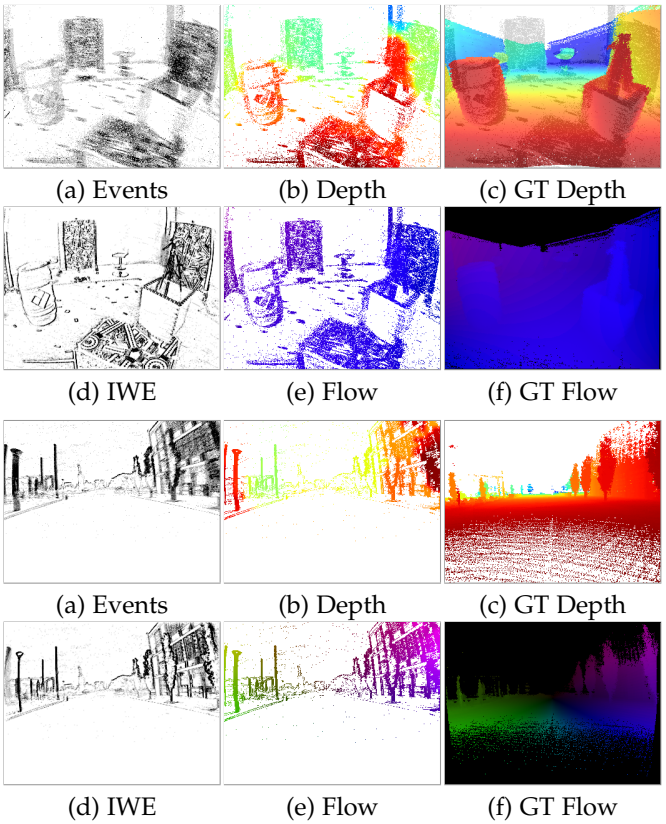
4.4 Monocular Depth and Ego-motion Estimation

4.4.1 Results on MVSEC

Evaluation on Depth. Table 4 summarizes the quantitative results of depth estimation on the MVSEC dataset [34]. Following the convention [57], we report the metrics for indoor as the average of the three indoor sequences. Although prior works use different strategies, such as additional sensor information, different train-test split, and different evaluations, we provide exhaustive comparisons across the existing methods to date: a model-based method where the pose information is given (EMVS) [23], a supervised-learning method [61] trained on real data (outdoor_day2, denoted "SL (R)") or in simulation ("SL (S)"), and two unsupervised-learning methods [27], [57].

Table 4: Depth evaluation on MVSEC (mean of three indoor sequences). The values for EMVS [23] are reported in [62].

		SiLog ↓	AbsRelDiff ↓	logRMSE ↓	A1 ↑	A2 ↑	A3 ↑
indoor	EMVS [23] (MB, w/ pose)	0.04	0.13	0.21	0.85	0.95	0.98
	ECN [57] (USL)	0.11	0.28	0.29	0.98	0.99	1.00
	Ours, L^2 (MB)	0.07	0.17	0.27	0.73	0.89	0.95
	Ours, L^1 (MB)	0.07	0.17	0.26	0.73	0.90	0.96
outdoor	SL (R) [61]	0.25	0.45	0.51	0.47	0.71	0.82
	SL (S) [61]	0.17	0.35	0.42	0.57	0.77	0.88
	EV-FlowNet [27] (USL)	0.16	0.36	0.41	0.46	0.73	0.88
	ECN [57] (USL)	0.14	0.33	0.33	0.97	0.98	0.99
	Ours, L^2 (MB)	0.25	0.39	0.51	0.42	0.70	0.83
	Ours, L^1 (MB)	0.22	0.36	0.48	0.47	0.72	0.85



(a) Events (b) Depth (c) GT Depth

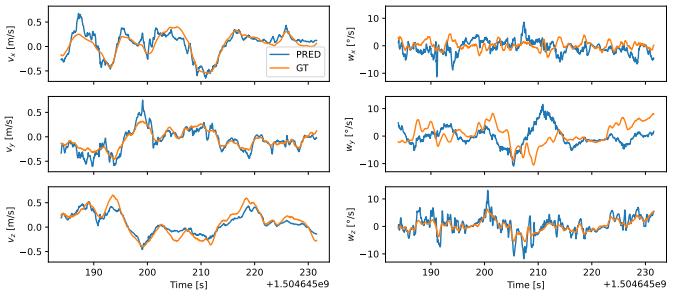
(d) IWE (e) Flow (f) GT Flow

(a) Events (b) Depth (c) GT Depth

(d) IWE (e) Flow (f) GT Flow

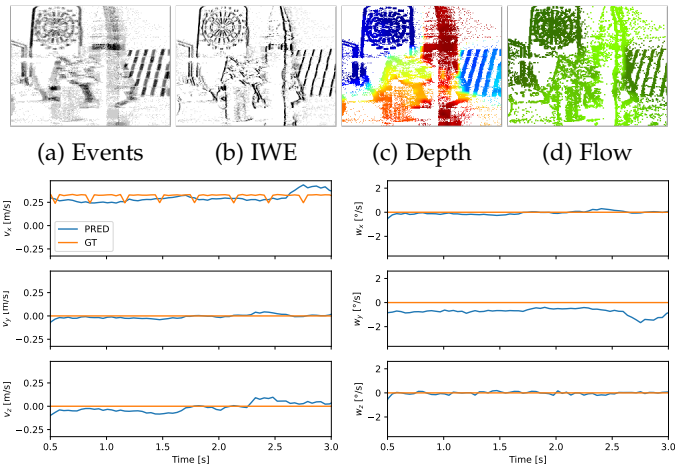
Figure 14: *Depth estimation results* on *indoor_flying3* and *outdoor_day1* sequences of the MVSEC dataset [34]. The 2nd and 3rd columns show the estimation and GT, respectively.

The proposed methods achieve overall better accuracy on the indoor sequences and competitive results on the outdoor sequence compared with ECN [57], the closest work to ours. However, ECN uses the 80/20 train-test split within each sequence (i.e., the training data consists of the same sequences as the test data), hence it might suffer from data leakage. For the outdoor sequence, our methods provide better results than the real-world supervised-learning method ("SL (R)"), and competitive results with the other learning-based approaches. We find that outdoor sequences are in general more challenging for the proposed approach. This can be attributed to the facts that (i) the MVSEC outdoor data has considerably sparse events, which affects the convergence of the method, and (ii) events in a scene



(a) Linear Velocity (b) Angular Velocity

Figure 15: *Ego-motion estimation results* on the *indoor_flying1* sequence from the MVSEC dataset [34].



(a) Events (b) IWE (c) Depth (d) Flow

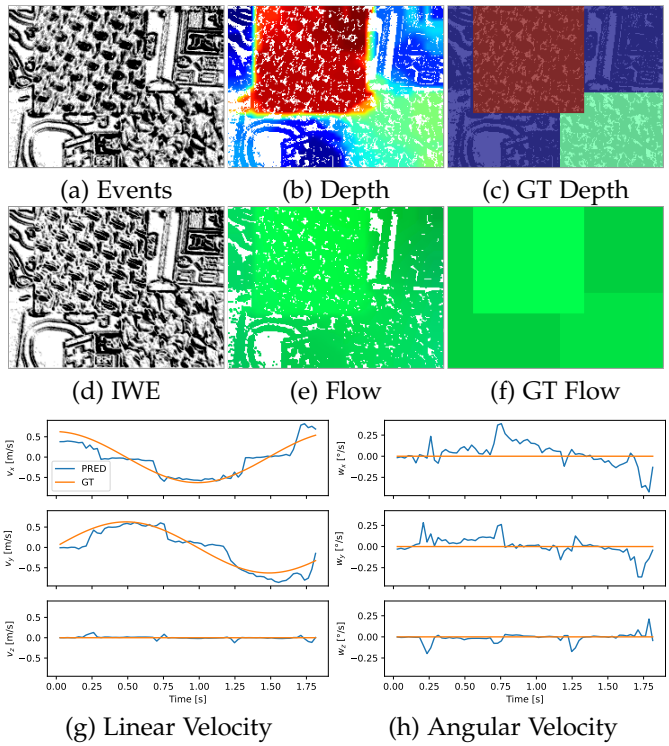
(e) Linear Velocity (f) Angular Velocity

Figure 16: *Depth and Ego-motion estimation* for the *slider_depth* sequence (real data) from the ECD dataset [63]. RMS errors: 0.11 m/s (in $\mathbf{V}$) and 0.94 °/s (in $\boldsymbol{\omega}$).

comprise various displacements with uneven distribution on the image plane. Indeed, the L^1 gradient magnitude loss achieves better results than the L^2 loss.

Qualitative results are shown in Fig. 14. For completeness, we show the flow (i.e., motion field) computed from the estimated depth and ego-motion. The estimated depth resembles the GT for both sequences, resulting in sharp IWEs. Moreover, similarly to the flow estimation (Sec. 4.2.1), the proposed depth covers the pixels where the GT does not exist, such as the middle board in the indoor scene and poles in the outdoor scene. Also, the estimated depth looks reasonable where LiDAR may fail to produce reliable depth maps due to the differences in the sampling frequency (e.g., the left-most board in the indoor results). Overall, the results illustrate that the proposed method is effective in estimating depth for these standard, real-world sequences.

Ego-Motion Estimation. Figure 15 shows ego-motion estimation results on the *indoor_flying1* sequence. The estimated linear velocity is scaled using the GT (IMU). The linear velocities resemble the GT, indicating that our method successfully estimates the camera motion of the freely-moving (6-DOF) drone. The pitch/yaw angular velocities are challenging to estimate since the motion field due to the pitch/yaw rotations is similar to that of a translation.



(a) Events (b) Depth (c) GT Depth

(d) IWE (e) Flow (f) GT Flow

(g) Linear Velocity (h) Angular Velocity

Figure 17: *Depth and Ego-motion estimation* for the *simulation_3planes* sequence from the ECD dataset [63]. GT flow is generated using GT poses and GT depth. RMS errors: 0.30 m/s (in $\mathbf{V}$) and 0.20 °/s (in $\boldsymbol{\omega}$).

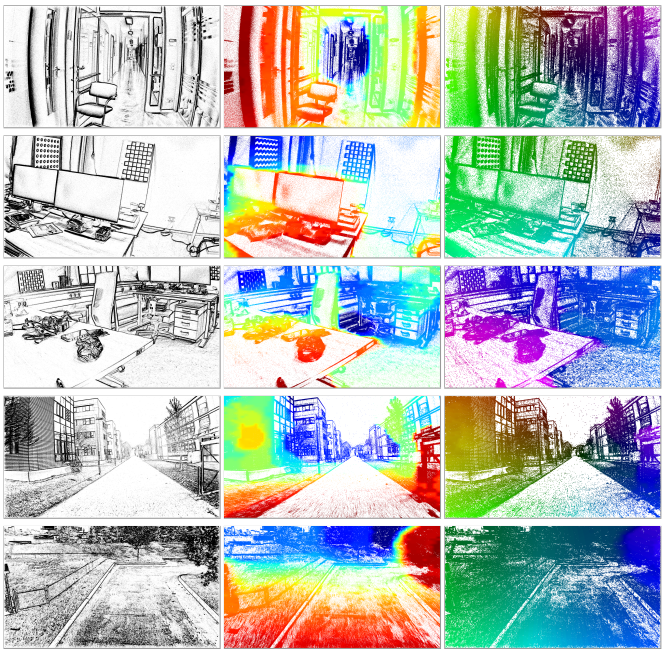
Table 5: Pose evaluation on MVSEC [34]. RMS errors in linear velocity $\mathbf{V}$ (m/s) and angular velocity $\boldsymbol{\omega}$ (°/s).

	indoor_flying1		indoor_flying2		indoor_flying3		outdoor_day1	
	$\mathbf{V}\downarrow$	$\boldsymbol{\omega}\downarrow$	$\mathbf{V}\downarrow$	$\boldsymbol{\omega}\downarrow$	$\mathbf{V}\downarrow$	$\boldsymbol{\omega}\downarrow$	$\mathbf{V}\downarrow$	$\boldsymbol{\omega}\downarrow$
ECN [57]	–	–	–	–	–	–	0.70	–
Ours	0.24	7.72	0.27	11.50	0.31	9.53	5.90	6.85

Quantitative results are reported in Tab. 5. Linear velocity errors are sensible: 20–30 cm/s for indoor (drone) sequences and 5.9 m/s for the outdoor (car) sequence. Forward-moving motion is more challenging for depth estimation, as the scene contains less parallax than lateral translational motions, which is also confirmed by our results. Angular velocity errors are small in all sequences, as they do not contain rotational-dominant motions. Few prior works report numerical values for comparison. As discussed in (Sec. 4.4.1 and Tab. 4), ECN [57] might have overfit to this outdoor sequence that reports a very small error (0.7m/s). On the other hand, our results provide constantly reasonable/similar metrics for all sequences. We hope Tab. 5 will encourage more works to benchmark monocular ego-motion estimation on these datasets.

4.4.2 *Results on ECD*

Depth and ego-motion estimation results on the *slider_depth* sequence from the ECD dataset [63] are shown on Fig. 16. Our method produces a sharp IWE as well as reasonable depth map, flow and poses, handling complex objects with



(a) IWE (b) Depth (c) Flow

Figure 18: *Depth estimation on 1Mpixel event datasets* [32], [33].

occlusion and at different distances. The camera pose RMS errors are: 0.11 m/s (in $\mathbf{V}$) and 0.94 °/s (in $\boldsymbol{\omega}$). We observe that the predicted linear velocity stays relatively constant, as expected. Also, the angular velocity error stays small, as the dominant motion of the sequence is translational. This is favorable for future extension of the proposed method to global adjustment (e.g., SLAM).

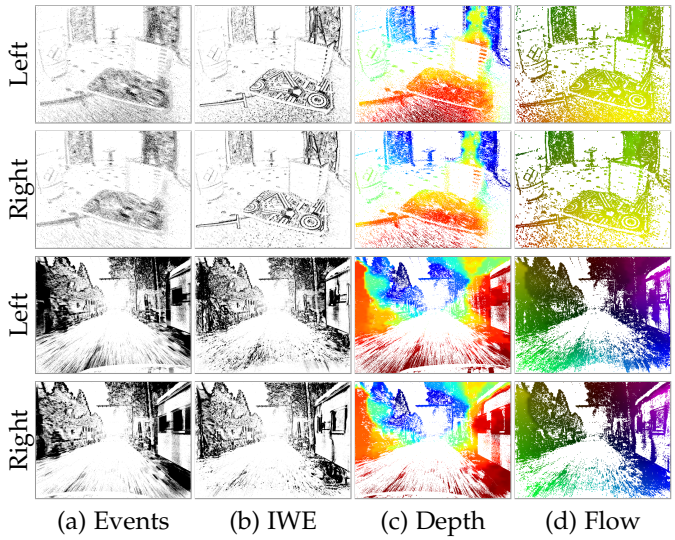
Figure 17 shows the results on a synthetic sequence from [63]. Since it has ground truth poses and depth, we also report these evaluation metrics, as SiLog[x100]: 1.16, AbsRelDiff: 0.09, logRMSE: 0.11, A1: 0.98, A2: 1.0 and A3: 1.0 for depth, and RMS: 0.30 m/s (in $\mathbf{V}$) and 0.20 °/s (in $\boldsymbol{\omega}$) for velocities. The estimated depth, flow, and ego-motion resemble those of GT, producing a sharp IWE.

4.4.3 Results on 1 Mpixel Datasets: TUM-VIE and M3ED

Figure 18 shows the qualitative depth estimation results on the TUM-VIE and M3ED datasets [32], [33]. The estimated depth is realistic, even for the challenging corridor sequence, which contains a large amount of noise and large variations of contour displacement in the scene due to the forward motion. The resulting flows are reasonable and the IWEs are sharp. Since the datasets do not have GT depth, we cannot conduct the quantitative evaluation.

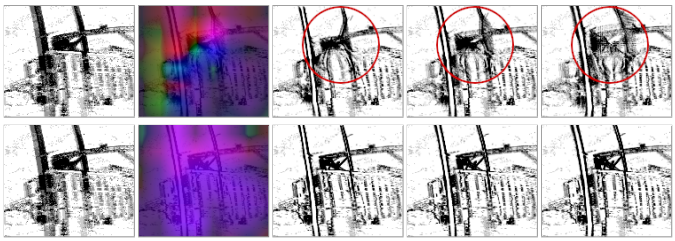
4.5 Stereo Depth Estimation

As explained in Sec. 3.6.2, our method can also tackle the event-based stereo scenario. Figure 19 shows stereo depth estimation results on the DSEC and MVSEC datasets. By parameterizing the depth and ego-motion on one camera only, the proposed model-based method successfully converges and provides sharp IWEs for both event cameras. We observe that, while IMOs are not explicitly modeled, depth estimation becomes more robust against them in the stereo setting. We leave a detailed analysis, evaluation, and benchmarks for future work.



(a) Events (b) IWE (c) Depth (d) Flow

Figure 19: *Stereo depth estimation results on MVSEC (indoor2) and DSEC (zurich_05b) datasets.*



(a) Events (b) Flow (c) Warp t_1 (d) t_{mid} (e) t_{N_e}

Figure 20: *Effect of the multi-reference focus loss*. Top row: single reference (t_1). Bottom row: proposed multi-reference.

5 Ablation and Sensitivity Analysis

5.1 Effect of the Multi-reference Focus Loss

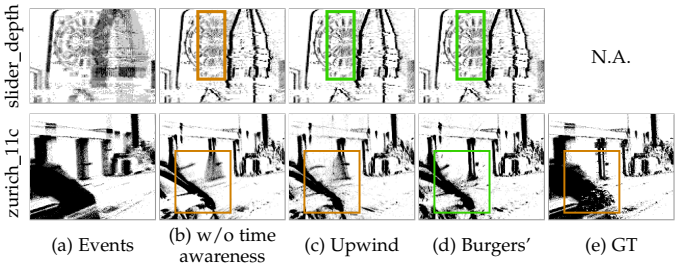
The effect of the proposed multi-reference focus loss is shown in Fig. 20. The single-reference focus loss function can easily overfit to the only reference time, pushing all events into a small region of the image at t_1 while producing blurry IWEs at other times (t_{mid} and t_{N_e}). Instead, our proposed multi-reference focus loss discourages such overfitting, as the loss favors flow fields which produce sharp IWEs at *any* reference time. The difference is also noticeable in the flow: the flow from the single-reference loss is irregular, with a lot of spatial variability in terms of directions (many colors, often in opposite directions of the color wheel). In contrast, the flow from the multi-reference loss is considerably more regular.

5.2 Effect of the Time-Aware Flow

To assess the effect of the proposed time-aware warp (8), we conducted experiments on MVSEC, DSEC and ECD [63]

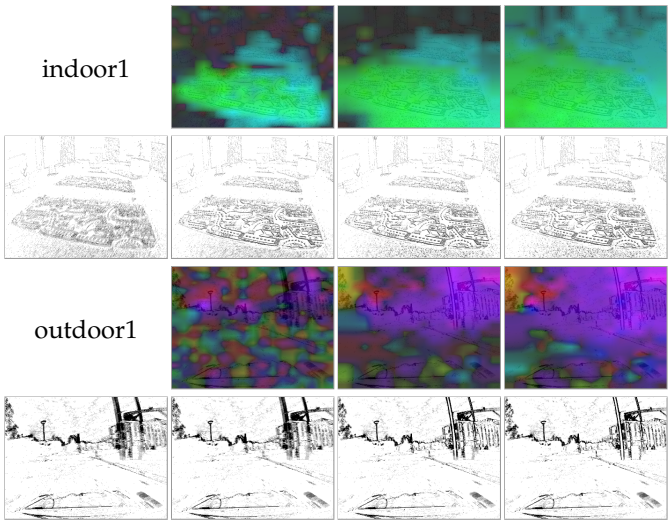
Table 6: FWL (IWE sharpness) results on MVSEC, DSEC, and ECD. Higher is better.

	MVSEC ($dt = 4$)				ECD	DSEC	
	indoor1	indoor2	indoor3	outdoor1	slider_depth	thun_00a	zurich_07a
Ground truth	1.09	1.20	1.12	1.07	–	1.01	1.04
Ours: w/o time aware	**1.17**	**1.30**	**1.23**	**1.11**	1.88	1.39	1.57
Ours: Upwind	**1.17**	**1.30**	**1.23**	**1.11**	1.92	1.40	1.60
Ours: Burgers'	**1.17**	**1.30**	**1.23**	**1.11**	**1.93**	**1.42**	**1.63**



(a) Events (b) w/o time awareness (c) Upwind (d) Burgers' (e) GT

Figure 21: *Effect of the time-aware flow*. Comparison between three flow models: Burgers', upwind, and no time-aware (4). At occlusions (dartboard in slider_depth [63] and garage door in DSEC [5]), upwind and Burgers' produce sharper IWEs. Due to the smoothness of the flow conferred by the tile-based approach, some small regions are still blurry.



(a) Events (b) Single (fine) scale (c) Three scales (d) $N_\ell = 5$ scales

Figure 22: *Effect of the multi-scale approach*. For each sequence, the top row shows the estimated flow and the bottom row shows the IWEs.

Table 7: Sensitivity analysis on the choice of loss function.

MVSEC	indoor1		indoor2		indoor3		outdoor1	
($dt = 4$)	AEE ↓	%Out ↓	AEE ↓	%Out ↓	AEE ↓	%Out ↓	AEE ↓	%Out ↓
Gradient L^2	1.68	12.79	2.49	26.31	2.06	18.93	1.25	9.19
Gradient L^1	**1.48**	**8.27**	**2.10**	**20.42**	**1.73**	**12.81**	**1.23**	9.22
Image variance [9]	1.70	11.25	2.18	21.91	1.93	15.84	1.82	15.89
Avg. timestamp [27]	>99	>99	>99	>99	>99	>99	>99	>99
Norm. avg. timestamp [30]	>99	>99	>99	>99	>99	>99	>99	>99

datasets. Accuracy results are already reported in Tabs. 1 and 2. We now report values of the FWL metric in Tab. 6. For MVSEC, $dt = 1$ is a very short time interval, with small motion and therefore few events, hence the sharpness of the IWE with or without motion compensation are about the same (FWL ≈ 1). Instead, $dt = 4$ provides more events, and larger FWL values (1.1–1.3), which means that the contrast of the motion-compensated IWE is larger than that of the zero flow baseline. All three methods provide sharper IWEs than ground truth. The advantages of the time-aware warp (8) over (4) to produce better IWEs (higher FWL) are most noticeable on sequences like slider_depth [63] and DSEC (see Fig. 21) because of the occlusions and larger motions. Notice that FWL differences below 0.1 are significant as seen in [52, Fig. 1] (cf. last two columns) and [52, Fig. 3], demonstrating the efficacy of time-awareness.

5.3 Effect of the Multi-scale Approach

The effect of the proposed multi-scale approach (Fig. 5) is shown in Fig. 22. This experiment compares the results of using multi-scale approaches (in a coarse-to-fine fashion) vs. using a single (finest) scale. With a single scale, the optimizer gets stuck in a local extremal, yielding an irregular flow field (see the optical flow rows), which may produce a blurry IWE (e.g., outdoor_day1 scene). With three scales (finest tile and two downsampled ones), the flow becomes less irregular than with one single scale, but there may be regions with few events where the flow is difficult to estimate. With five scales the flow becomes smoother, more coherent over the whole image domain, while still being able to produce sharp IWEs.

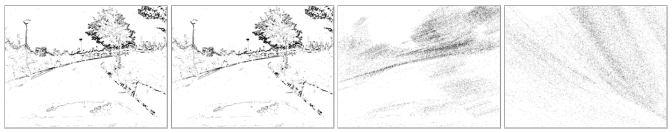
5.4 The choice of loss function

Table 7 shows the results on the MVSEC benchmark for different loss functions. We compare the gradient-based functions (L^1 and L^2), image variance [7], average timestamp [27], and normalized average timestamp [30]. The contrast functions (L^1, L^2, and variance) yield consistently better accuracy than the two average timestamp losses. Although the variance gives competitive results, we use the functions based on the IWE gradient for the reasons described in Sec. 3.2.1. Both average timestamp losses are trapped in the global optima which pushes most events out of the image plane (see Fig. 23), hence, they provide very large errors (marked as "> 99" in Tab. 7). Despite this, they have been successfully used in several learning-based methods.

Remark: Maximization of (5) does not suffer from the problem mentioned in [30] that affects the average timestamp loss function, namely that the optimal flow warps all events outside the image so as to minimize the loss (undesired global optima shown in Fig. 23c-d). If most events were warped outside of the image, then (5) would be smaller than the identity warp, which contradicts maximization.

5.5 The regularizer weight

Table 8 shows the sensitivity analysis on the regularizer weight λ in (9). $\lambda = 0.0025$ provides the best accuracy in the outdoor sequence, while $\lambda = 0.025$ provides slightly better accuracy in the indoor sequences. Comparing their accuracy, we use the former because it has a higher gain.



(a) (b) (c) (d)

Figure 23: *IWEs for different loss functions*: (a) Gradient Magnitude (L^2); (b) Variance; (c) Avg. timestamp [27]; (d) Normalized avg. timestamp [30].

Table 8: Sensitivity analysis on the regularizer weight.

MVSEC	indoor1		indoor2		indoor3		outdoor1	
($dt = 4$)	AEE ↓	%Out ↓	AEE ↓	%Out ↓	AEE ↓	%Out ↓	AEE ↓	%Out ↓
$\lambda = 0.0025$	1.68	12.79	2.49	26.31	2.06	18.93	1.25	9.19
$\lambda = 0.025$	1.52	9.07	2.39	26.26	1.94	18.44	1.86	17.11
$\lambda = 0.25$	1.89	16.54	3.19	36.95	2.91	30.85	2.57	27.86

6 Computational Performance

Each scale of our method has the same computational complexity as CM [7], $O(N_e + N_p)$ because the multi-reference warps yield a constant scaling factor. Our unoptimized implementation using PyTorch (v1.9) running on a GPU (NVIDIA Quadro RTX 8000) without time awareness takes about 9.9 s per batch to converge on the MVSEC experiments ($3\times$ more in case of Burgers' scheme) (Sec. 4.2.1). However, if we apply the proposed method to a DNN (EV-FlowNet), training takes about 10 h, preprocessing (crop center image and voxelization) takes 74 ms, and inference takes about 3 ms (Sec. 4.2.3). This inference time is on par with other DNN-based methods.

7 Limitations

Like previous unsupervised works [27], [30], our method is based on the brightness constancy assumption. Hence, it struggles to estimate flow from events that are not due to motion, such as those caused by flickering lights. SL and SSL methods may forego this assumption, but they require high quality supervisory signal, which is challenging due to the HDR and high speed of event cameras.

Like other optical flow methods, our approach may suffer from the aperture problem. The flow could still cause event collapse if tiles become too small (higher DOFs), or if the regularization is too small compared with the texture density that drives the data-fidelity term. This effect can be observed in Fig. 1, where the flow becomes irregular for the tree leaves (in the example on row 2). Optical flow is also difficult to estimate in regions with few events, such as homogeneous brightness regions and regions with small apparent motion. Regularization fills in the homogeneous regions, whereas recurrent connections could help with small apparent motion.

The monocular depth and ego-motion estimation approach considers each event packet (i.e., time interval) independently, hence it only recovers camera velocities. Absolute poses could be estimated if the camera velocities were simultaneously recovered over multiple event packets while sharing a common depth map. The stereo approach enables the recovery of the absolute scale.

While the computational effort of the proposed approach is high in our current (unoptimized) implementation, it allowed us to focus on modeling the problem and uncovering the "secrets" of event-based optical flow, i.e., identifying the successful ingredients for accurate motion estimation. Then, we showed how such knowledge could be transferred to learning-based settings, with the same computational cost and speed as prior work (ms inference time on GPUs).

8 Conclusion

We have extended the CM framework to estimate dense optical flow, depth and ego-motion from events alone. The proposed principled method overcomes problems of overfitting, occlusions, and convergence by sensibly modeling the space-time nature of event data. The comprehensive experiments show that our method achieves the best flow accuracy among all methods in the MVSEC indoor benchmark, and among the unsupervised and model-based methods in the outdoor sequence. It also provides competitive results in the DSEC optical flow benchmark and generalizes to various datasets, including the latest 1 Mpixel ones, delivering the sharpest IWEs. The method exposes the limitations of the current flow benchmarks and produces remarkable results when it is transferred to unsupervised learning settings. We show downstream applications of the estimated flow, such as motion segmentation, intensity reconstruction and event denoising. Finally, the method achieves competitive results in depth and ego-motion estimation in both monocular and stereo settings. As demonstrated, the proposed framework is able to handle a broad set of motion-related tasks across multiple datasets and event camera resolutions, hence we believe it is a cornerstone in event-based vision. We hope our work inspires future model-based and learning-based approaches in these motion-related problems.

Appendix

Time-Awareness: PDE solutions

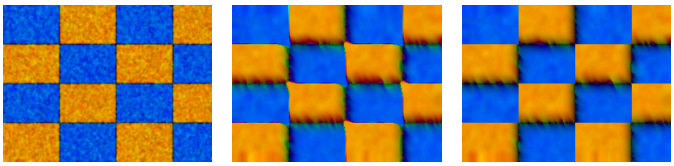
The proposed *time-aware flow* is given as the solution to (7). Letting the flow be $\mathbf{v} = (v_x, v_y)^\top$, the system of PDEs can be written as:

$$\begin{aligned} v_x \frac{\partial v_x}{\partial x} + v_y \frac{\partial v_x}{\partial y} + \frac{\partial v_x}{\partial t} = 0, \\ v_x \frac{\partial v_y}{\partial x} + v_y \frac{\partial v_y}{\partial y} + \frac{\partial v_y}{\partial t} = 0. \end{aligned} \tag{12}$$

Upwind and Burgers' schemes can be used to discretize and numerically solve the system of PDEs [64], [65].

Discretization. Let $\mathbf{v}^n(x, y)$ be the flow vector at discretized space- (e.g., pixel) and time-indices (x, y, n), with discretization steps $\Delta x, \Delta y$, and Δt, respectively, and let the forward $(+)$ and backward $(-)$ differences of a scalar field w (e.g., v_x^n or v_y^n) be defined as

$$\begin{aligned} D_x^+ w \equiv \frac{\partial w}{\partial x}^+ &= \frac{1}{\Delta x}\big(w(x + \Delta x, y) - w(x, y)\big), \\ D_y^+ w \equiv \frac{\partial w}{\partial y}^+ &= \frac{1}{\Delta y}\big(w(x, y + \Delta y) - w(x, y)\big), \end{aligned} \tag{13}$$



(a) Original flow (b) Upwind (c) Burgers'

Figure 24: *Comparison of the two flow propagation schemes.* Original flow (a) has large shock and fan waves (the color changes between orange and blue) to highlight the difference. The propagated flows with both schemes are shown in (b) (c). Same color notation as Figs. 1 and 6.

and

$$\begin{aligned} D_x^- w \equiv \frac{\partial w}{\partial x}^- &= \frac{1}{\Delta x}\big(w(x,y) - w(x-\Delta x, y)\big), \\ D_y^- w \equiv \frac{\partial w}{\partial y}^- &= \frac{1}{\Delta y}\big(w(x,y) - w(x, y-\Delta y)\big). \end{aligned} \tag{14}$$

We discretize in time using forward differences, $\frac{\partial w}{\partial t} \approx (w(t+\Delta t) - w(t))/\Delta t$, to yield explicit update schemes: $w(t+\Delta t) \approx w(t) + \Delta t \frac{\partial w}{\partial t}$.

Upwind scheme. The first-order upwind scheme is an explicit scheme that updates the flow as follows, based on the sign of the variables: it uses $D_x^+ v_x^n$ and $D_x^+ v_y^n$ for $v_x^n > 0$ ($D_x^- v_x^n$ and $D_x^- v_y^n$ otherwise), and $D_y^+ v_x^n$ and $D_y^+ v_y^n$ for $v_y^n > 0$ ($D_y^- v_x^n$ and $D_y^- v_y^n$ otherwise). The scheme is stable if the flow satisfies $\Delta t \max\{|v_x|/\Delta x + |v_y|/\Delta y\} < 1$ (CFL stability condition [85]). For example, in case that $v_x^n > 0$ and $v_y^n > 0$ at the current discretization time n:

$$\begin{aligned} v_x^{n+1} &= v_x^n - \Delta t \left(v_x^n D_x^+ v_x^n + v_y^n D_y^+ v_x^n\right), \\ v_y^{n+1} &= v_y^n - \Delta t \left(v_y^n D_y^+ v_y^n + v_x^n D_x^+ v_y^n\right). \end{aligned} \tag{15}$$

Burgers' scheme. The study of the inviscid Burgers' equation provides a more conservative scheme solution, especially at "shock" and "fan wave" cases [65]. In this explicit scheme, the product terms in the same variable (which convey that the flow is transporting itself), $v_x^n D_x^+ v_x^n$ and $v_y^n D_y^+ v_y^n$ in (15), are replaced with U_x and U_y respectively, which are given by:

$$\begin{aligned} U_x &= \frac{1}{2}\left(\operatorname{sgn}\big(v_x^n(x,y)\big)\big(v_x^n(x,y)\big)^2 + F_x - B_x\right), \\ F_x &= \begin{cases} \big(v_x^n(x+\Delta x, y)\big)^2, & \text{if } v_x^n(x+\Delta x, y) < 0 \\ 0, & \text{otherwise} \end{cases} \\ B_x &= \begin{cases} \big(v_x^n(x-\Delta x, y)\big)^2, & \text{if } v_x^n(x-\Delta x, y) > 0 \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{16}$$

and

$$\begin{aligned} U_y &= \frac{1}{2}\left(\operatorname{sgn}\big(v_y^n(x,y)\big)\big(v_y^n(x,y)\big)^2 + F_y - B_y\right), \\ F_y &= \begin{cases} \big(v_y^n(x, y+\Delta y)\big)^2, & \text{if } v_y^n(x, y+\Delta y) < 0 \\ 0, & \text{otherwise} \end{cases} \\ B_y &= \begin{cases} \big(v_y^n(x, y-\Delta y)\big)^2, & \text{if } v_y^n(x, y-\Delta y) > 0 \\ 0. & \text{otherwise} \end{cases} \end{aligned} \tag{17}$$

Comparison of schemes. Figure 24 shows the comparison between the two schemes, especially for the "shock" and "fan wave" cases. After some propagation iterations, the upwind scheme starts to produce artifacts around the shock and fan flows (the color boundary of orange and blue), while the Burgers' scheme provides a more stable flow.

REFERENCES

[1] C. Posch, T. Serrano-Gotarredona, B. Linares-Barranco, and T. Delbruck, "Retinomorphic event-based vision sensors: Bioinspired cameras with spiking output," *Proc. IEEE*, vol. 102, no. 10, pp. 1470–1484, Oct. 2014.

[2] G. Gallego, T. Delbruck, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, "Event-based vision: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 154–180, 2022.

[3] X. Lagorce, G. Orchard, F. Gallupi, B. E. Shi, and R. Benosman, "HOTS: A hierarchy of event-based time-surfaces for pattern recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 7, pp. 1346–1359, Jul. 2017.

[4] A. Z. Zhu, D. Thakur, T. Ozaslan, B. Pfrommer, V. Kumar, and K. Daniilidis, "The multivehicle stereo event camera dataset: An event camera dataset for 3D perception," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 2032–2039, Jul. 2018.

[5] M. Gehrig, W. Aarents, D. Gehrig, and D. Scaramuzza, "DSEC: A stereo event camera dataset for driving scenarios," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 4947–4954, 2021.

[6] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1998.

[7] 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 *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018, pp. 3867–3876.

[8] G. Gallego, M. Gehrig, and D. Scaramuzza, "Focus is all you need: Loss functions for event-based vision," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019, pp. 12 272–12 281.

[9] G. Gallego and D. Scaramuzza, "Accurate angular velocity estimation with an event camera," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 632–639, 2017.

[10] H. Kim and H. J. Kim, "Real-time rotational motion estimation with contrast maximization over globally aligned events," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 6016–6023, 2021.

[11] C. Gu, E. Learned-Miller, D. Sheldon, G. Gallego, and P. Bideau, "The spatio-temporal Poisson point process: A simple model for the alignment of event camera data," in *Int. Conf. Comput. Vis. (ICCV)*, 2021, pp. 13 495–13 504.

[12] S. Guo and G. Gallego, "CMax-SLAM: Event-based rotational-motion bundle adjustment and SLAM system using contrast maximization," *IEEE Trans. Robot.*, pp. 1–20, 2024.

[13] U. M. Nunes and Y. Demiris, "Robust event-based vision model estimation by dispersion minimisation," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2021.

[14] X. Peng, L. Gao, Y. Wang, and L. Kneip, "Globally-optimal contrast maximisation for event cameras," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3479–3495, 2022.

[15] A. Z. Zhu, N. Atanasov, and K. Daniilidis, "Event-based feature tracking with probabilistic data association," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2017, pp. 4465–4470.

[16] ——, "Event-based visual inertial odometry," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017, pp. 5816–5824.

[17] H. Seok and J. Lim, "Robust feature tracking in dvs event stream using Bezier mapping," in *IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, 2020, pp. 1647–1656.

[18] T. Stoffregen and L. Kleeman, "Event cameras, contrast maximization and reward functions: an analysis," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019, pp. 12 292–12 300.

[19] A. Mitrokhin, C. Fermuller, C. Parameshwara, and Y. Aloimonos, "Event-based moving object detection and tracking," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2018, pp. 1–9.

[20] T. Stoffregen, G. Gallego, T. Drummond, L. Kleeman, and D. Scaramuzza, "Event-based motion segmentation by motion compensation," in *Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 7243–7252.

[21] Y. Zhou, G. Gallego, X. Lu, S. Liu, and S. Shen, "Event-based motion segmentation with spatio-temporal graph cuts," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–13, 2021.

[22] C. M. Parameshwara, N. J. Sanket, C. D. Singh, C. Fermüller, and Y. Aloimonos, "0-MMS: Zero-shot multi-motion segmentation with a monocular event camera," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2021, pp. 9594–9600.

[23] H. Rebecq, G. Gallego, E. Mueggler, and D. Scaramuzza, "EMVS: Event-based multi-view stereo—3D reconstruction with an event camera in real-time," *Int. J. Comput. Vis.*, vol. 126, no. 12, pp. 1394–1414, Dec. 2018.

[24] Z. Zhang, A. Yezzi, and G. Gallego, "Formulating event-based image reconstruction as a linear inverse problem with deep regularization using optical flow," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022.

[25] S. Shiba, Y. Aoki, and G. Gallego, "Event collapse in contrast maximization frameworks," *Sensors*, vol. 22, no. 14, pp. 1–20, 2022.

[26] ——, "A fast geometric regularizer to mitigate event collapse in the contrast maximization framework," *Adv. Intell. Syst.*, p. 2200251, 2022.

[27] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "Unsupervised event-based learning of optical flow, depth, and egomotion," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019, pp. 989–997.

[28] F. Paredes-Valles and G. C. H. E. de Croon, "Back to event basics: Self-supervised learning of image reconstruction for event cameras via photometric constancy," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2021, pp. 3445–3454.

[29] A. Z. Zhu, "Event-based algorithms for geometric computer vision," Ph.D. dissertation, Univ. Pennsylvania, 2019.

[30] J. J. Hagenaars, F. Paredes-Valles, and G. C. H. E. de Croon, "Self-supervised learning of event-based optical flow with spiking neural networks," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 34, 2021, pp. 7167–7179.

[31] S. Shiba, Y. Aoki, and G. Gallego, "Secrets of event-based optical flow," in *Eur. Conf. Comput. Vis. (ECCV)*, 2022, pp. 628–645.

[32] S. Klenk, J. Chui, N. Demmel, and D. Cremers, "TUM-VIE: The TUM stereo visual-inertial event dataset," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2021, pp. 8601–8608.

[33] K. Chaney, F. Cladera Ojeda, 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 *IEEE Conf. Comput. Vis. Pattern Recog. Workshops (CVPRW)*, 2023, pp. 4016–4023.

[34] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "EV-FlowNet: Self-supervised optical flow estimation for event-based cameras," in *Robotics: Science and Systems (RSS)*, 2018, pp. 1–9.

[35] D. Sun, S. Roth, and M. J. Black, "A quantitative analysis of current practices in optical flow estimation and the principles behind them," *Int. J. Comput. Vis.*, vol. 106, no. 2, pp. 115–137, Sep. 2013.

[36] M. Liu and T. Delbruck, "Adaptive time-slice block-matching optical flow algorithm for dynamic vision sensors," in *British Mach. Vis. Conf. (BMVC)*, 2018, pp. 1–12.

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

[38] G. Orchard, R. Benosman, R. Etienne-Cummings, and N. V. Thakor, "A spiking neural network architecture for visual motion estimation," in *IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, 2013, pp. 298–301.

[39] T. Brosch, S. Tschechne, and H. Neumann, "On event-based optical flow detection," *Front. Neurosci.*, vol. 9, no. 137, Apr. 2015.

[40] R. Benosman, C. Clercq, X. Lagorce, S.-H. Ieng, and C. Bartolozzi, "Event-based visual flow," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 2, pp. 407–417, 2014.

[41] H. Akolkar, S.-H. Ieng, and R. Benosman, "Real-time high speed motion prediction using fast aperture-robust event-driven visual flow," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 361–372, 2022.

[42] J. Nagata, Y. Sekikawa, and Y. Aoki, "Optical flow estimation by matching time surface with event-based cameras," *Sensors*, vol. 21, no. 4, 2021.

[43] P. Bardow, A. J. Davison, and S. Leutenegger, "Simultaneous optical flow and intensity estimation from an event camera," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2016, pp. 884–892.

[44] M. Gehrig, M. Millhäusler, D. Gehrig, and D. Scaramuzza, "E-RAFT: Dense optical flow from event cameras," in *Int. Conf. 3D Vision (3DV)*, 2021, pp. 197–206.

[45] 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," *AAAI Conf. Artificial Intell.*, vol. 36, no. 1, pp. 525–533, 2022.

[46] 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 *Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 366–382.

[47] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Int. Conf. Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015, pp. 234–241.

[48] Z. Teed and J. Deng, "RAFT: Recurrent all pairs field transforms for optical flow," in *Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 402–419.

[49] J. Cuadrado, U. Rançon, B. R. Cottereau, F. Barranco, and T. Masquelier, "Optical flow estimation from event-based cameras and spiking neural networks," *Front. Neurosci.*, vol. 17, p. 1160034, 2023.

[50] 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 *Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 9685–9694.

[51] D. Gehrig, A. Loquercio, K. G. Derpanis, and D. Scaramuzza, "End-to-end learning of representations for asynchronous event-based data," in *Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 5632–5642.

[52] T. Stoffregen, C. Scheerlinck, D. Scaramuzza, T. Drummond, N. Barnes, L. Kleeman, and R. Mahony, "Reducing the sim-to-real gap for event cameras," in *Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 534–549.

[53] M. Gehrig, M. Muglikar, and D. Scaramuzza, "Dense continuous-time optical flow from event cameras," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–12, 2024.

[54] X. Luo, K. Luo, A. Luo, Z. Wang, P. Tan, and S. Liu, "Learning optical flow from event camera with rendered dataset," *Int. Conf. Comput. Vis. (ICCV)*, 2023.

[55] 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," *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2023.

[56] G. Taverni, D. P. Moeys, C. Li, C. Cavaco, V. Motsnyi, D. S. S. Bello, and T. Delbruck, "Front and back illuminated Dynamic and Active Pixel Vision Sensors comparison," *IEEE Trans. Circuits Syst. II (TCSII)*, vol. 65, no. 5, pp. 677–681, 2018.

[57] C. Ye, A. Mitrokhin, C. Parameshwara, C. Fermüller, J. A. Yorke, and Y. Aloimonos, "Unsupervised learning of dense optical flow, depth and egomotion with event-based sensors," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2020, pp. 5831–5838.

[58] Y. Tian and J. Andrade-Cetto, "Event transformer FlowNet for optical flow estimation," in *British Mach. Vis. Conf. (BMVC)*, 2022.

[59] F. Paredes-Vallés, K. Y. Scheper, C. De Wagter, and G. C. de Croon, "Taming contrast maximization for learning sequential, low-latency, event-based optical flow," in *Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 9661–9671.

[60] A. Jaegle, S. Phillips, and K. Daniilidis, "Fast, robust, continuous monocular egomotion computation," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2016, pp. 773–780.

[61] D. G. Javier Hidalgo-Carrio and D. Scaramuzza, "Learning monocular dense depth from events," in *Int. Conf. 3D Vision (3DV)*, Nov. 2020, pp. 534–542.

[62] S. Ghosh and G. Gallego, "Multi-event-camera depth estimation and outlier rejection by refocused events fusion," *Adv. Intell. Syst.*, vol. 4, no. 12, p. 2200221, 2022.

[63] E. Mueggler, H. Rebecq, G. Gallego, T. Delbruck, and D. Scaramuzza, "The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM," *Int. J. Robot. Research*, vol. 36, no. 2, pp. 142–149, 2017.

[64] L. C. Evans, *Partial Differential Equations*, ser. Graduate Studies in Mathematics. American Mathematical Society, 2010.

[65] J. Sethian, *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, ser. Cambridge Monographs on Applied and Computational Mathematics. Cambridge Univ. Press, 1999.

[66] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1–4, pp. 259–268, Nov. 1992.

[67] P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barlaud, "Deterministic edge-preserving regularization in computed imaging," *IEEE Trans. Image Process.*, vol. 6, no. 2, pp. 298–311, 1997.

[68] S.-H. Ieng, J. Carneiro, M. Osswald, and R. Benosman, "Neuromorphic event-based generalized time-based stereovision," *Front. Neurosci.*, vol. 12, p. 442, 2018.

[69] A. Rosinol Vidal, H. Rebecq, T. Horstschaefer, and D. Scaramuzza, "Ultimate SLAM? combining events, images, and IMU for robust visual SLAM in HDR and high speed scenarios," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 994–1001, Apr. 2018.

[70] H. Rebecq, T. Horstschäfer, G. Gallego, and D. Scaramuzza, "EVO: A geometric approach to event-based 6-DOF parallel tracking and mapping in real-time," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 593–600, 2017.

[71] E. Mueggler, G. Gallego, H. Rebecq, and D. Scaramuzza, "Continuous-time visual-inertial odometry for event cameras," *IEEE Trans. Robot.*, vol. 34, no. 6, pp. 1425–1440, Dec. 2018.

[72] Y. Zhou, G. Gallego, and S. Shen, "Event-based stereo visual odometry," *IEEE Trans. Robot.*, vol. 37, no. 5, pp. 1433–1450, 2021.

[73] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck, "A 240x180 130dB 3μs latency global shutter spatiotemporal vision sensor," *IEEE J. Solid-State Circuits*, vol. 49, no. 10, pp. 2333–2341, 2014.

[74] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017, pp. 6612–6619.

[75] Z. Wan, Y. Dai, and Y. Mao, "Learning dense and continuous optical flow from an event camera," *IEEE Trans. Image Process.*, vol. 31, pp. 7237–7251, 2022.

[76] Y. Ye, H. Shi, K. Yang, Z. Wang, X. Yin, Y. Wang, and K. Wang, "Towards anytime optical flow estimation with event cameras," *arXiv e-prints*, 2023.

[77] H. Zhuang, X. Huang, K. Hou, D. Kong, C. Hu, and Z. Fang, "EV-MGRFlowNet: Motion-guided recurrent network for unsupervised event-based optical flow with hybrid motion-compensation loss," *arXiv e-prints*, 2023.

[78] S. Shiba, Y. Aoki, and G. Gallego, "Fast event-based optical flow estimation by triplet matching," *IEEE Signal Process. Lett.*, vol. 29, pp. 2712–2716, 2022.

[79] V. Brebion, J. Moreau, and F. Davoine, "Real-time optical flow for vehicular perception with low- and high-resolution event cameras," *IEEE Trans. Intell. Transport. Syst.*, pp. 1–13, 2021.

[80] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *Int. Conf. Learn. Representations (ICLR)*, 2015.

[81] J. Delmerico, T. Cieslewski, H. Rebecq, M. Faessler, and D. Scaramuzza, "Are we ready for autonomous drone racing? the UZH-FPV drone racing dataset," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2019, pp. 6713–6719.

[82] Y. Wu, F. Paredes-Vallés, and G. C. H. E. de Croon, "Rethinking event-based optical flow: Iterative deblurring as an alternative to correlation volumes," *arXiv e-prints*, 2023.

[83] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Int. Conf. Knowledge Discovery and Data Mining (KDD)*, vol. 96, no. 34, 1996, pp. 226–231.

[84] X. Xiang, L. Zhu, J. Li, Y. Tian, and T. Huang, "Temporal upsampling for asynchronous events," in *IEEE Int. Conf. Multimedia and Expo (ICME)*, 2022, pp. 01–06.

[85] C. Hirsch, "The resolution of numerical schemes," in *Numerical Computation of Internal and External Flows*, 2nd ed., C. Hirsch, Ed. Oxford: Butterworth-Heinemann, 2007, pp. 411–412.

Shintaro Shiba received the Ph.D. degree in Engineering from Keio University, Japan, in 2023. From 2021 to 2023 he was a visiting researcher at the Robotics Interactive Perception Laboratory, Technische Universität Berlin, supported by a German Academic Exchange Service (DAAD) bi-national doctoral research fellowship. In 2024, he received the 14th Ikushi Prize by Japan Society of the Promotion of Science. He received his Master's in Cognitive Neuroscience from the University of Tokyo in 2017. His research interests include computer vision, machine learning, robotics, and neuroscience.

Yannick Klose received the Master's degree in Electrical Engineering from Technische Universität Berlin, in 2023. As part of his Master's thesis, he worked at the Robotic Interactive Perception Laboratory on event-based vision. His research interests include computer vision, signal processing and FPGAs. He is currently working as an Embedded Software Engineer in communication technologies.

Yoshimitsu Aoki received the Ph.D. degree in engineering from Waseda University in 2001. From 2002 to 2008, he was an Associate Professor with the Department of Information Engineering, Shibaura Institute of Technology. He is currently a Professor with the Department of Electronics and Electrical Engineering, Keio University. He performs research in the areas of computer vision, pattern recognition, and media understanding.

Guillermo Gallego (SM'19) is Associate Professor at Technische Universität Berlin and at the Einstein Center Digital Future, Berlin, where he leads the Robotic Interactive Perception Laboratory. He is also a Principal Investigator at the Science of Intelligence Excellence Cluster and co-director of the HEIBRiDS research school, Berlin, Germany. He received the PhD degree in Electrical and Computer Engineering from the Georgia Institute of Technology, USA, in 2011, supported by a Fulbright Scholarship. He serves as Associate Editor for IEEE T-PAMI, IEEE RA-L and IJRR.