Motion-prior Contrast Maximization for Dense Continuous-Time Motion Estimation

Friedhelm Hamann[1], Ziyun Wang[2], Ioannis Asmanis[2], Kenneth Chaney[2], Guillermo Gallego[1,3], and Kostas Daniilidis[2,4]

[1] TU Berlin and SCIoI Excellence Cluster, Berlin, Germany
[2] University of Pennsylvania, Philadelphia, US
[3] Einstein Center Digital Future and Robotics Institute Germany, Berlin, Germany
[4] Archimedes, Athena RC, Greece

Abstract. Current optical flow and point-tracking methods rely heavily on synthetic datasets. Event cameras are novel vision sensors with advantages in challenging visual conditions, but state-of-the-art frame-based methods cannot be easily adapted to event data due to the limitations of current event simulators. We introduce a novel self-supervised loss combining the Contrast Maximization framework with a non-linear motion prior in the form of pixel-level trajectories and propose an efficient solution to solve the high-dimensional assignment problem between non-linear trajectories and events. Their effectiveness is demonstrated in two scenarios: In dense continuous-time motion estimation, our method improves the zero-shot performance of a synthetically trained model on the real-world dataset EVIMO2 by 29%. In optical flow estimation, our method elevates a simple UNet to achieve state-of-the-art performance among self-supervised methods on the DSEC optical flow benchmark. Our code is available at https://github.com/tub-rip/MotionPriorCMax.

1 Introduction

Determining the motion of arbitrary projected world points on the image plane over long time intervals is a difficult low-level computer vision problem. Researchers have studied it as optical flow and lately as point-tracking, with many practical applications in robotics, computational photography, video compression, and object-level tracking [26]. The best-performing frame-based methods for optical flow estimation and point tracking use large-scale synthetic datasets. Synthetic data is diverse and has highly accurate ground truth (GT), but is unrealistic and methods trained on synthetic data show a sim-to-real gap.

Novel vision sensors called event cameras [16,17,39] have emerged as promising alternatives to take on the problem. Inspired by the transient visual pathway, which is responsible for motion perception, they are particularly fit for capturing scene motion in the form of asynchronous pixel-wise intensity changes. This working principle endows them with advantages, such as high speed, high dynamic range (HDR) and low power consumption.

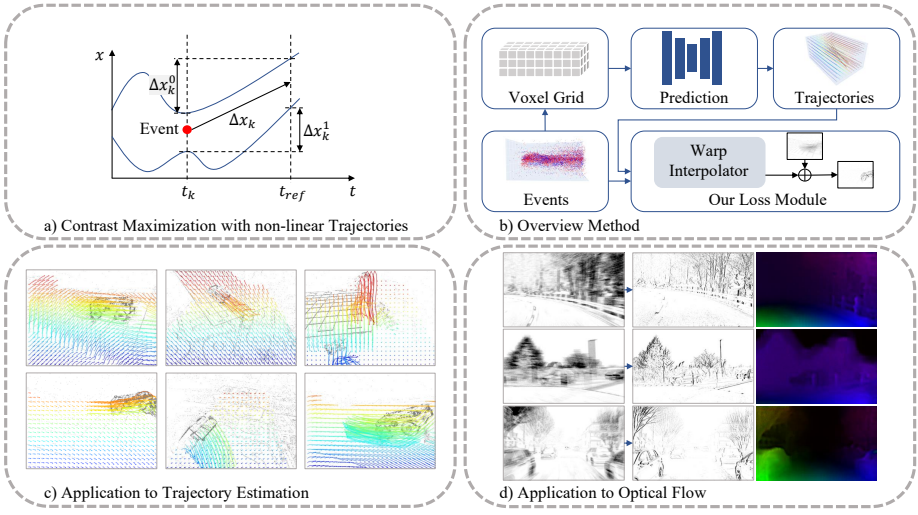


Fig. 1: *Summary.* a) We present an approach to combine Contrast Maximization with dense non-linear trajectories. b) We show how it can be used for self-supervised learning in a pipeline to predict dense point trajectories, and c) evaluate it on the EVIMO2 dataset, for which we generate dense point tracks. d) Additionally, our approach provides state-of-the-art performance on self-supervised optical flow prediction.

Event-based motion estimation methods can be categorized according to the complexity of the motion considered (i.e., number of degrees-of-freedom (DOF)) and to the solution strategy: model-based or learning-based. Low-DOF motions arise in sparse feature tracking and ego-motion estimation, while high-DOF motions describe more complex scenes, e.g., via per-pixel displacement (i.e., densely, over the whole image plane). Focusing on the latter, the event-based optical flow problem has been extensively studied on mobile robotic datasets [23, 75], where GT is calculated as the motion field (from known depth and poses). Most approaches use this GT for direct supervision of dense flow prediction [22, 24, 40, 57, 65]. Alternatively, several works use a contrast loss [18, 19, 77], which allows training in a self-supervised manner. However, the contrast loss is prone to undesired local optima, called event collapse [52, 53], where many events are warped into a few pixels or lines. Evaluation of event-based optical flow has been mostly done on the MVSEC [75] and DSEC [23] datasets, which comprise largely uniform motions and test intervals from 0.022s to 0.1s. Recently, [25] has proposed a supervised method to predict pixel displacement over a larger duration (0.5s), by leveraging synthetic data. It relies on the generation of training event data from images, where current tools [21] are not as mature as frame-based simulation, therefore it suffers from a sim-to-real performance gap.

In short, progress has been made and the research field is moving towards predicting motion over longer time intervals on the whole image plane, i.e., taking on more complex (i.e., nonlinear) motion problems. This is precisely the

problem tackled in this paper (Fig. 1): long-time and dense event-based motion estimation, reducing the domain adaptation gap of previous approaches. It comes with several associated challenges, mainly overcoming the lack of large labeled datasets, and dealing with event noise and data association (events depend on motion, and for large motions, the appearance of "corresponding" events can be wildly different due to changes in motion direction, occlusions, etc.) while leveraging the space-time characteristics of event data.

To this end, we propose tackling the problem in two stages, by leveraging both supervised and self-supervised strategies: first, using supervised learning on synthetic data to provide initial model weights for motion estimation, and secondly, fine-tuning the network on real data via a self-supervised loss to reduce the domain adaptation gap. Our technical contributions involve extending the contrast loss framework to regress continuous-time trajectories over long time intervals via motion priors (parametric functions that provide a good balance between motion generality and regularization [64, 66]). This includes a solution to accurately and efficiently associate events to the trajectories (Fig. 2).

More specifically, finding the association between events and motion trajectories (to warp corresponding events and achieve event alignment) is a high-dimensional problem (e.g., for a time window of 0.3 s one can consider about ten million events, and as many trajectories as pixels), which needs to be implemented in a differentiable and parallelizable manner. We propose two actions to cope with these technical challenges. First, we relax the problem by interpolating over a coarse spatio-temporal displacement field, which serves as a lookup table. Secondly, we use a symbolic matrix framework [15] to calculate the $K = N_{\text{traj}}$ nearest neighbor (KNN) trajectories for each grid point, thereby solving KNN in a memory-efficient and differentiable way on GPU. The warp displacement at the grid point is then set to the average of the neighboring trajectories.

Our approach is versatile, allowing for different types of networks and trajectories. Hence, we evaluate its performance on two applications: dense continuous-time motion estimation and optical flow. The results on EVIMO2 [7] show that fine-tuning with our self-supervised loss improves the zero-shot performance of a model pre-trained on synthetic data by 29%. On DSEC, our model shows state-of-the-art performance among self-supervised methods, on average improving the angular error by 19% and the percentage of inliers by 14%, while having a $5\times$ faster inference time. In summary, our contributions are (Fig. 1):

1. We introduce motion priors (parametric functions with a good balance between generality and regularization) in the event-based contrast maximization framework for continuous-time and dense motion estimation.
2. We combine the self-supervised loss with Bflow [25], the current top-performing supervised model for dense continuous-time event-based motion estimation trained on synthetic data, and show that it can be used to improve over zero-shot performance on unseen real data (EVIMO2).
3. We show that the combination of our loss with a simple U-Net architecture (à la EV-FlowNet [76]) achieves state-of-the-art performance among the self-supervised methods on the DSEC Optical Flow benchmark.

2 Related Work

Frame-based Motion Estimation. Progress in deep learning triggered a large series of learned methods for optical flow estimation in classical, frame-based computer vision [2, 14, 32, 33, 60, 62]. The solutions rely on large-scale synthetic datasets [14, 42, 59, 73]. Similarly related is the task of point tracking, which has shown impressive progress on frame-based data [13, 31, 36, 50, 74], equally relying on simulated data [8,12,74]. This approach is unsatisfying, as it is prone to out-of-distribution (OOD) problems that cannot be easily overcome. As an alternative, self-supervised methods have been explored [34, 35, 48, 72] and shown to improve models pre-trained on simulated data for motion estimation tasks [58, 61].

Event-based Optical Flow. Event cameras [39, 46] are a relatively new technology, and have found applications in various computer vision domains, like mobile robotics [27, 45, 68], scene understanding [29, 30], and computational imaging [55, 63]. Their exploration for low-level vision tasks has taken a similar trajectory in a compressed timeline as previously frame-based methods [17]. The first event-based optical flow methods were model-based [3, 4, 6, 41, 43, 54], followed by learning-based approaches [11, 24, 28, 38, 44, 76, 77].

The scarce availability of event data and less mature simulation technology compared to frame-based cameras are major obstacles [21]. Therefore, event-based optical flow has been mostly evaluated on data acquired through ego-motion, such as the robotics dataset MVSEC [75], the driving dataset DSEC [23] or the M3ED dataset [9]. However, GT is calculated using the motion field equation (with data from a synchronized depth sensor) and therefore does not provide an accurate flow at the event rate, nor spatially at occlusions and independently moving objects (IMOs). This GT is used to train supervised learning approaches [22, 24, 40, 57, 65], which inherit limitations from the ground truth.

Self-supervised methods for event-based optical flow lessen the dependency on GT labels by leveraging an event alignment error [28,54,70,77]. Notably, variations of the contrast loss have been proposed [18], however, so far the application has been limited to either low-DOF problems (e.g., feature tracking [10, 19, 51], ego-motion [19,20,37]) or high-DOF but short-time optical flow problem [44, 54] (max. 0.1 s). We expand the frontier to the challenging problem of long-time and high-DOF (complex) motion estimation, by exploiting trajectory priors.

Trajectory Prior. Using trajectories as motion priors has been a widely explored scheme in computer vision. Specifically, [1, 64, 78] propose a linear combination of basis functions for structure from motion. More recently it has been explored for dynamic novel view synthesis [66]. In the context of events, [63] uses cubic motion splines for video frame interpolation, [25] uses B-splines for supervised learning of non-linear optical flow, [67] use learned basis functions for event-based video decompression, and [10,51] show event-based feature tracking (low-DOF) using Bézier or B-spline curves in combination with a contrast loss.

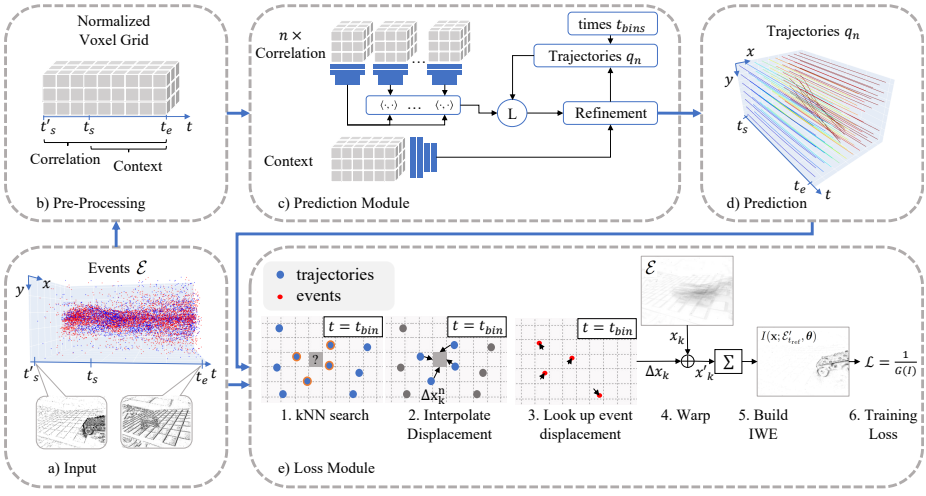


Fig. 2: *Pipeline overview.* (a) Input events in a time interval are (b) voxelized and (c) passed to an artificial neural network that predicts per-pixel coefficients for continuous-time trajectories (d). The raw events and predicted trajectories are fed to the loss module (e). Here, a dense spatio-temporal displacement map is interpolated, and events are warped according to their looked-up displacement. Lastly, an image of warped events (IWE) is built at a random reference time and its gradient magnitude acts as training loss. Note that the prediction method displayed here is specific to the used Bflow backbone [25], with additional events before the prediction start time t_s as input.

3 Methodology

3.1 Contrast Loss

Contrast Maximization (CM) [18,19] is used for self-supervised training. The idea is to find the point trajectories on the image plane best aligning with the events by maximizing the sharpness of warped events. It is an iterative approach, with three steps per iteration: (*i*) Each event $e_k = (\mathbf{x}_k, t_k, p_k)$ contains the pixel coordinates $\mathbf{x}_k = (x_k, y_k)^\top$, timestamp t_k and polarity p_k of a brightness change of predefined size C (contrast sensitivity). Events $\mathcal{E} = \{e_k\}_{k=1}^{N_e}$ are displaced according to a candidate motion hypothesis $\mathbf{x}'_k = \mathbf{W}(\mathbf{x}_k, t_k; \boldsymbol{\theta})$, with parameters $\boldsymbol{\theta}$, to a reference time t_{ref}, producing a set of warped events $\mathcal{E}'_{t_{\text{ref}}} = \{e'_k\}_{k=1}^{N_e}$:

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

Afterwards, (*ii*) the events are summed into an image of warped events (IWE),

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

which essentially counts the number of warped events e'_k per pixel. Lastly (*iii*) the contrast or sharpness of the IWE is computed (e.g., the gradient magnitude of the IWE, $G \doteq \int \|\nabla I(\mathbf{x})\| d\mathbf{x}$), which serves as a proxy for how well the model (motion hypothesis and $\boldsymbol{\theta}$) fit with the events produced by the true motion.

Table 1: Comparison of the contrast loss formulation in the most recent methods.

Steps	Shiba et. al [54]	Paredes et. al [44]	Ours
Warp displacement (3):	$\Delta\mathbf{x}_k = (t_{\text{ref}} - t_k)\mathbf{v}(e_k)$	$\Delta\mathbf{x}_k = \sum_i (\Delta t_i \mathbf{v}_i)(e_k)$	$\Delta\mathbf{x}_k = \frac{1}{N_{\text{traj}}} \sum_n \Delta\mathbf{x}_k^n$
	linear trajectory	flow concatenation	**non-linear trajectory**
IWE	$I(\mathbf{x}) = \sum_k \mathcal{N}(\mathbf{x}; \mathbf{x}'_k, \sigma^2)$	$T_\pm(\mathbf{x}) = \frac{\sum_k t_k \mathcal{N}(\mathbf{x}; \mathbf{x}'_k, \sigma^2)}{\sum_k \mathcal{N}(\mathbf{x}; \mathbf{x}'_k, \sigma^2)}$	$I(\mathbf{x}) = \sum_k \mathcal{N}(\mathbf{x}; \mathbf{x}'_k, \sigma^2)$
Reference time t_{ref}:	$\{t_0, t_{0.5}, t_1\}$	multi-partition p, multi-time-scale s	$\sim \mathcal{U}(0,1)$
Contrast objective	$\frac{G(t_0) + 2G(t_{0.5}) + G(t_1)}{4G(\mathbf{v}=0)}$	$\frac{1}{S}\sum_{s=0}^{S-1} \frac{1}{2^s} \sum_{p=0}^{2^s-1} \mathcal{L}_{\text{CM},p}^{R/2^s}$	$G(t_{\text{ref}})$

3.2 Estimating Continuous-Time and Dense Motion Trajectories

The CM framework has been extended in several works. Table 1 compares recent extensions for optical flow estimation, where events are warped as

$$\mathbf{x}'_k = \mathbf{x}_k + \Delta\mathbf{x}_k. \tag{3}$$

The main differences between these approaches lie in the event displacement (i.e., warp) model, the reference times used, and the loss function (i.e., event-alignment metric), with the design choices having two goals: fitting the event data and regularizing the solution (e.g., avoiding event collapse [52]). Previous methods used mostly a linear model [54,56] or partitioned the inference interval into smaller flow intervals so that its concatenation need not be linear [44]. By contrast, we introduce an explicit continuous-time non-linear trajectory model, a random reference time per iteration, and an easier contrast loss.

Trajectory Representation. Figure 2 shows an overview of the training pipeline. Events in the time interval $[t_s, t_e]$ are fed into a neural network that makes per-pixel predictions of the parameters (coefficients) $\boldsymbol{\theta} \equiv \boldsymbol{\alpha}_n = (\alpha_{n,1}, \ldots, \alpha_{n,N_c})^\top \in \mathbb{R}^{N_c}$ of a continuous-time trajectory $\mathbf{q}_n(t; \boldsymbol{\alpha}_n) \equiv \mathbf{q}_n(t) = (x_n(t), y_n(t))^\top$. There is one trajectory per pixel (i.e., "dense" character), $n = 1, \ldots, N_p$, where $N_p = hw$ is the number of pixels (image height h and width w). Thus, n is the spatial index of the trajectory, identifying it among all trajectories on the image plane.

We model trajectories as weighted combinations of basis functions, $\mathbf{q}_n(t) = \sum_{j=1}^{N_c} g_j(t)\mathbf{p}_{n,j}$, where $g_j(t)$ are temporal basis shared by all trajectories, and $\mathbf{p}_{n,j} = (\alpha_{n,j}^x, \alpha_{n,j}^y)^\top$ are "control points" (we write x and y components explicitly with separate coefficients in $\boldsymbol{\alpha}_n$). We investigate polynomial basis $g_j(t) = t^j$, Bézier curves with basis $g_j(t) = \binom{N_c}{j}(1-t)^{N_c-j}t^j$, as well as a learned basis.

Spatio-Temporal Event Warping. The trajectories can be used to warp events. This would require finding the trajectory that passes through the space-time coordinates of the event and then finding the value of the trajectory at the reference time (i.e., the warped event location). However, the association between events and trajectories is unknown and needs to be estimated simultaneously with the parameters (i.e., shape) of the trajectories. We circumvent the problem by using a soft association between events and trajectories. Each event e_k is associated with its N_{traj} nearest neighboring trajectories $\{\mathbf{q}_n\}_{n=1}^{N_{\text{traj}}}$,

and the event displacement $\Delta\mathbf{x}_k$ in (3) is computed as the average of the respective trajectory displacements $\{\Delta\mathbf{x}_k^n\}_{n=1}^{N_{\text{traj}}}$. The trajectory displacement $\Delta\mathbf{x}_k^n$ is defined as the difference of trajectory locations at the time of the event and the reference time. The event displacement is defined as the mean:

$$\Delta\mathbf{x}_k \doteq \frac{1}{N_{\text{traj}}} \sum_{n=1}^{N_{\text{traj}}} \Delta\mathbf{x}_k^n, \quad \text{with} \quad \Delta\mathbf{x}_k^n \doteq \mathbf{q}_n(t_{\text{ref}}) - \mathbf{q}_n(t_k). \tag{4}$$

Loss Calculation. Once the events are warped using (3), it is straightforward to compute the IWE. We adopt the magnitude of the IWE gradient as loss function [18, 56]. Moreover, we use the contrast loss in a self-supervised learning setting, choosing a different reference time t_{ref}, uniformly sampled in the observation interval, for every batch during training. This simplifies the objective to a single warping operation and loss calculation (e.g., compared to the three warping operations and loss calculations in the optimization-based approach by [54, 56]), while it has added regularization benefits, as mentioned below.

Memory Effective Computation of the Displacement Field. The number of events and trajectories can be very large, and even more their combination. Calculating KNN for every event can quickly become computationally unfeasible, and traditional algorithms cannot be efficiently implemented in deep-learning frameworks. We propose calculating the per-event displacement $\Delta\mathbf{x}_k$ in (3) by first interpolating a dense but coarser spatio-temporal displacement field, and then looking up the per-event displacement in such space-time volume. Moreover, we relax the problem by solving the KNN search in 2D instead of in the volume.

Figure 2 shows an overview of the interpolation process. The displacement map is a tensor of shape $[N_{\text{bins}}, h/4, w/4]$, where N_{bins} is the number of temporal bins, and h and w are the sensor's height and width, respectively. For each temporal bin-center t_{bins} the trajectories $\mathbf{q}_n(t = t_{\text{bins}})$ are calculated and the KNN interpolation is performed between each pixel of a channel $\mathbf{x}[t_{\text{bins}}]$ and $\mathbf{q}_n(t = t_{\text{bins}})$. We implement the KNN approach using KeOps [15], a framework for symbolic matrix computation, which provides a memory-efficient and differentiable solution. We perform a KNN search for every voxel in the table, and events are warped after a lookup operation. Please note that this step is entirely independent of the voxel grid passed as input to the backbone, and all raw events are used to calculate the contrast loss. From the calculated displacement map, we can directly look up per-event displacements $\Delta\mathbf{x}_k$.

Regularization. Objectives based on event alignment are prone to undesired local minima called "event collapse" [52, 53]. Our formulation inherently provides regularization, temporally by the smoothness of the motion prior, and spatially by interpolating the event flow from several trajectories via a soft assignment. Our framework uses two additional regularization sources. Firstly, we penalize the magnitude of the spatial gradient of the interpolated displacement field between consecutive timesteps, $R \doteq \|\nabla(\Delta\mathbf{q}_*(t))\|_{L^1(\Omega)}$. To interpolate $\mathbf{q}_*$ from the sparse trajectories $\mathbf{q}_n$, we use the same KNN indices as for the main volume. This loss encourages spatial smoothness of the trajectories.

Table 2: Results on EVIMO2 [7].

Method	TEPE ↓	TAE ↓	%Out ↓
Paredes et. al [44]	21.69	51.91	0.63
E-RAFT (linear) [24]	19.38	74.52	0.66
BFlow (zero-shot) [25]	8.63	19.94	0.36
BFlow (in-domain)	**3.38**	**11.68**	**0.17**
Ours (self-supervised)	<u>6.14</u>	<u>16.98</u>	<u>0.25</u>

Secondly, we use a multi-reference formulation by randomly sampling a reference time at every training step, while previous work used multiple and fixed timestamps (Tab. 1). This has the advantage that (*i*) IWEs are required to be sharp at truly *any* time and (*ii*) the memory requirement is lowered as it scales linearly with the number of reference times used during a training step.

In summary, the training loss is

$$\mathcal{L} = 1/G + \lambda R, \tag{5}$$

with $\lambda > 0$ as regularization weight. Further details are in the supplementary.

Pre-processing and Prediction-Module Architecture. Events record brightness changes asynchronously, in the form of a sparse spatio-temporal signal. As input to the Prediction Module, events are customarily converted to voxel grids [77] for compatibility with conventional artificial neural networks. An event volume is discretized in the time dimension, and each voxel counts the number of events within it (bi-linearly voted for undistorted, rectified events).

In principle, our self-supervised loss module can be paired with any segmentation or optical flow network architecture. We use a U-Net architecture for the experiments on DSEC (Sec. 4.2), and the architecture in the recent Bflow method [25], inspired by RAFT [62], for the experiments on EVIMO2 (Sec. 4.1).

4 Experiments

We test our method on two applications. First, we test the capabilities for non-linear trajectory estimation on the real-world dataset EVIMO2 [7], with additional results on the synthetic MultiFlow dataset [25], which we use for pre-training (Sec. 4.1). Secondly, we evaluate our method on the DSEC dataset [23, 24] because this has been the previous frontier for event-based optical flow estimation and it allows for direct comparison with prior work (Sec. 4.2). Additional results, including on the MVSEC dataset [75], are given in the supplement.

4.1 Results on Non-Linear Trajectory Estimation

EVIMO2 Continuous Flow Dataset (CF-EVIMO2). MultiFlow [25] was first proposed to evaluate long-term "optical flow" methods. The dataset consists

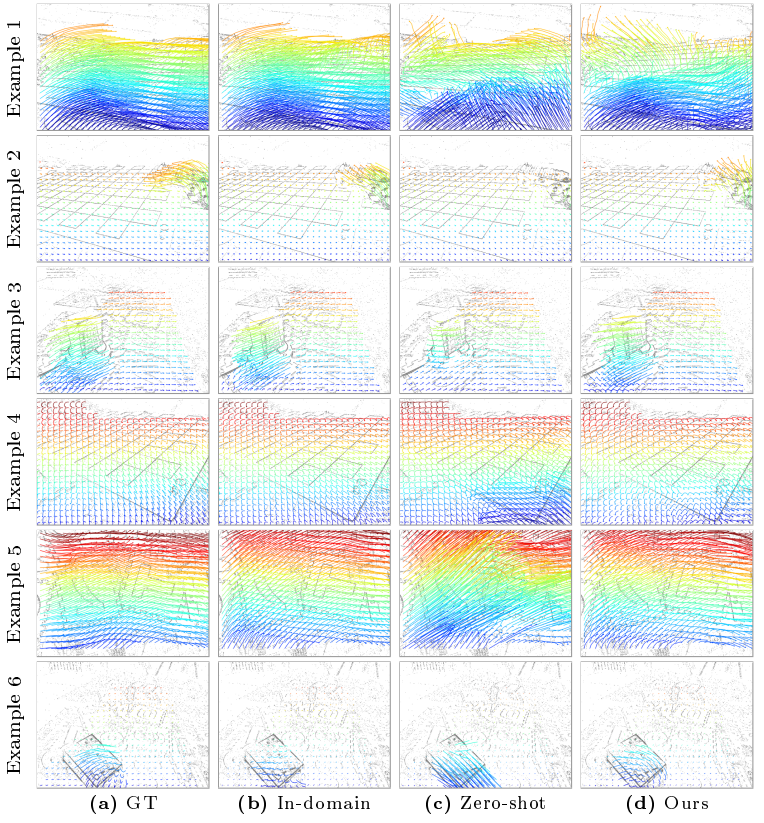


Fig. 3: Visualization of predicted trajectories on EVIMO2 data. *GT*: Ground truth. *In-domain*: fine-tuned on EVIMO2 using GT (supervised). *Zero-shot*: network trained only on synthetic data (out-of-domain prediction). *Ours*: Pre-trained on synthetic data, fine-tuned with self-supervised loss. Note that supervision in-domain is often impossible in practice because dense trajectory labels for real data are difficult to obtain.

of synthetic videos generated with Internet images as foreground and Flickr30K [71] images as background. Events are generated with a simulator [47] by rendering the synthetic scenes at 1000 frames/s. However, due to the sim-to-real gap between real and synthetic events, it is challenging to infer the actual performance of the proposed and baseline methods on real datasets. Furthermore, Multiflow generates trajectories by pasting 2D foreground objects onto background images, which lacks the challenging cases of self-occlusion due to 3D rotation.

To address these data limitations, we present the *EVIMO2 Continuous Flow Dataset* based on the full 3D data and event data provided by the meticulously designed EVIMO2 [7] dataset. The EVIMO2 GT data provides high-quality 3D scans of the objects, camera poses, object poses, and camera intrinsics, which we use to compute optical flow GT as follows. Given a point P_t^i on the object represented in the camera coordinate system at time t, we project it onto a 2D point

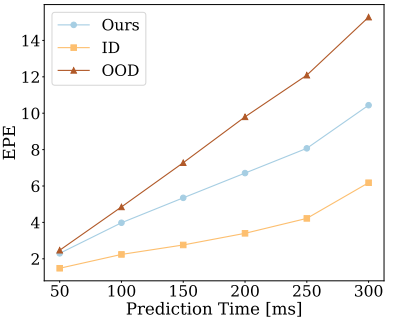


Fig. 4: End-point-error vs. prediction time span for three methods: in-distribution, out-of-distribution and self-supervised (Ours). Using the Bézier curve results from Tab. 3.

Table 3: Sensitivity with respect to motion prior on EVIMO2 [7]. *SL, ID*: supervised, in-distribution, *SL, OOD*: supervised, out of distribution, *SSL*: self-supervised.

	Method	TEPE ↓	TAE ↓	%Out ↓
SL, ID	BFlow (polyn.)	<u>3.51</u>	<u>13.26</u>	<u>0.18</u>
	BFlow (learned)	3.78	13.87	0.19
	BFlow (Bézier)	**3.38**	**11.68**	**0.17**
SL, OOD	BFlow (polyn.)	9.36	20.88	0.36
	BFlow (learned)	8.66	19.64	0.35
	BFlow (Bézier) [25]	8.63	19.94	0.36
SSL	Ours, polyn.	6.78	19.76	0.27
	Ours, learned	7.46	19.78	0.28
	Ours, Bézier	6.14	16.98	0.25

$p_t^i = \pi(P_t^i)$, where the $\pi(\cdot)$ function represents the perspective projection, including the camera intrinsics. The object poses at time t and $t+1$ are provided as $T_o^{c_t}$, $T_o^{c_{t+1}}$ in the GT data. The rigid-body transformation $T_{c_t}^{c_{t+1}} = (T_o^{c_{t+1}})^{-1} T_o^{c_t}$ maps 3D points in the camera frame at time t to the camera frame at time $t+1$ by aligning the shared object coordinate frame. The "flow" (or dense motion, since it may not be linear) is defined as $\Delta p_{t\to t+1}^i \doteq \pi(T_{c_t}^{c_{t+1}} P_t^i) - \pi(P_t^i)$. For each sequence, we generate GT dense motion every 10ms for 300ms. We mask out areas where GT object masks are unavailable. Experiments are carried out on the IMO subset of EVIMO2 using the official train and test splits.

Metrics. Ground truth is provided as dense motion from $t = 0$ to several timesteps in increasing order. We consider $N_s = 6$ timestamps (i.e., subintervals) in a 300ms window and evaluate the quality of predictions with direct extension of the common optical flow metrics end-point-error (EPE), angular error (AE), and percentage of predicted vectors above a specific EPE threshold (%Out). We term the corresponding trajectory metrics TEPE and TAE, where

$$\text{TEPE} = \frac{1}{N_s} \sum_{k=1}^{N_s} \text{EPE}(\Delta \mathbf{x}_{\text{pred}}(t_k), \Delta \mathbf{x}_{\text{gt}}(t_k)). \tag{6}$$

The number of outliers is calculated on TEPE with a threshold of 3px.

Implementation Details. We use the BFlow network architecture and test different motion priors. Specifically, we report results with polynomial, Bézier (with $N_c = 10$) and learned basis $g_j(t)$ (a small dense neural network with three layers and hidden dimension 64 is trained alongside the main network).

The contrast loss module uses one trajectory per 4×4 px, and similarly 4×4 px in the displacement volume. The weight for the spatial smoothness term is $\lambda = 0.003$, and the number of neighbors in the KNN approach is $N_{\text{traj}} = 32$. The loss function G is the L^1 norm of the IWE's gradient magnitude [18,56].

Training Schedule. The model is first pre-trained on MultiFlow ($\approx$ 12000 samples) for 50 epochs, with a batch size of 10 with an L^1 loss on the GT trajectory flow. We include data augmentation by flipping (horizontal and vertical) and cropping and use the training/test split provided in [25]. These weights are what we refer to as *zero-shot* or *out-of-distribution* (OOD) for the main comparison on EVIMO2. Afterwards, the pre-trained model is fine-tuned with two different losses on EVIMO2. Training using our self-supervised loss is carried out for 15 epochs with a batch size of 6. We refer to this method as *Ours*. Lastly, a second version is trained directly on the GT flow of EVIMO for 50 epochs, which we refer to as *in-distribution* (ID). All experiments are performed on Nvidia RTX A6000 GPUs with an AdamW optimizer and a learning rate of 10^{-4}.

Baselines. This is the first usage of EVIMO2 for dense non-linear flow/motion. For comparison, we provide additional baselines. We use ERAFT [24] with the provided DSEC weights to infer linear flow over the whole prediction time. The timestamps of the intermediate flow are interpolated from the linear prediction. Additionally, we provide results for the prediction of Paredes et. al. [44]. The network is trained self-supervised on DSEC and performs recurrent prediction steps in time in intervals of 10ms, inferring the total flow at every step by accumulation of the shorter flows. We unroll this prediction over the whole interval and take the accumulated flow after each 50ms step as the predicted flow.

Results. Table 2 presents the comparison of the three different training modes. The results of our self-supervised loss can improve the zero-shot performance by nearly 30%. While direct supervision on EVIMO2 GT data delivers the best results, note that this is only possible because we are comparing on a dataset that was recorded with a motion capture setup and exact object models. Under real conditions, pixel-level annotations of motion tracks are very difficult to obtain. Therefore, our loss delivers a promising tool in combination with synthetic pre-training, substantially helping in overcoming the sim-to-real gap.

Figure 3 shows the qualitative comparison of GT tracks and the three training schedules. It delivers insights into the failure cases that our method can overcome. Specifically, the zero-shot model shows errors like overseen object motion (Examples 2 and 3), wrong scale (Ex. 5), and prediction of non-existing motion (Ex. 6), which are improved with self-supervised domain training.

Additionally, Fig. 4 shows the error for different prediction time spans. Intuitively, the error increases for longer intervals. The visualization shows that our loss improves zero-shot performance, especially for the long prediction times.

Table 4: Results on the DSEC optical flow benchmark [24].

		All				interlaken_00_b				interlaken_01_a				thun_01_a			
Method	t_{inf}[ms]	EPE ↓	AE ↓	%Out ↓	FWL ↑	EPE ↓	AE ↓	%Out ↓	FWL ↑	EPE ↓	AE ↓	%Out ↓	FWL ↑	EPE ↓	AE ↓	%Out ↓	FWL ↑
E-RAFT (SL) [24]	46.33	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
IDNet (SL) [69]		**0.72**	**2.72**	**2.04**	–	**1.25**	**2.11**	**4.35**	–	**0.77**	**2.25**	**2.60**	–	**0.57**	**2.66**	**1.47**	–
Paredes et al. (SSL) [44]	40.10	2.33	10.56	17.77	–	3.34	6.22	25.72	–	2.49	6.88	19.15	–	1.73	9.75	10.39	–
RTEF (MB) [5]		4.88	–	41.95	**2.51**	8.59	–	59.84	**2.89**	5.94	–	47.33	**2.92**	3.01	–	29.70	**2.39**
EV-FlowNet (SSL) [77]		3.86	–	31.45	1.30	6.32	–	47.95	1.46	4.91	–	36.07	1.42	2.33	–	20.92	1.32
MultiCM (MB) [54]	$9.9 \cdot 10^3$	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 (poly, $k = 1$)	**7.27**	3.20	8.53	15.21	1.46	3.21	4.89	20.45	1.58	2.38	5.46	17.40	1.70	1.39	6.99	7.36	1.30

	thun_01_b				zurich_city_12_a				zurich_city_14_c				zurich_city_15_a			
Method	EPE ↓	AE ↓	%Out ↓	FWL ↑	EPE ↓	AE ↓	%Out ↓	FWL ↑	EPE ↓	AE ↓	%Out ↓	FWL ↑	EPE ↓	AE ↓	%Out ↓	FWL ↑
E-RAFT (SL) [24]	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
IDNet (SL) [69]	**0.55**	**2.07**	**1.35**	–	**0.60**	**4.56**	1.16	–	0.76	**3.74**	2.74	–	**0.55**	**2.55**	**1.02**	–
Paredes et al. (SSL) [44]	1.66	8.41	9.34	–	2.72	23.16	26.65	–	2.64	10.23	23.01	–	1.69	8.88	9.98	–
RTEF (MB) [5]	3.91	–	34.69	**2.48**	3.14	–	34.08	**1.42**	4.00	–	45.67	**2.67**	3.78	–	37.99	**2.82**
EV-FlowNet (SSL) [77]	3.04	–	25.41	1.33	2.62	–	25.80	1.03	3.36	–	36.34	1.24	2.97	–	25.53	1.33
MultiCM (MB) [54]	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 (poly, $k = 1$)	1.54	6.55	9.69	1.33	8.33	20.16	22.39	1.13	1.78	8.79	12.99	1.56	1.45	6.27	8.34	1.51

Sensitivity to motion prior. Table 3 compares our method for different motion priors. While Bézier curves show slightly improved performance over the two basis function methods, self-supervised domain training can improve performance in all cases, which proves the robustness and generality of our method.

4.2 Results on Optical Flow Estimation

Dataset, Metrics and Details. We compare our method on the DSEC optical flow benchmark [24]. The dataset consists of sequences from a Prophesee Gen3 event camera, with a resolution of 640×480 px, mounted on a driving car. Ground truth is calculated as the motion field from a co-deployed depth sensor.

We compare the common flow metrics EPE, AE, and percentage of outliers (%Out), where an outlier is a flow vector with EPE $>$ 3px. Additionally, we evaluate the Flow Warp Loss (FWL) [57]. FWL > 1 means that the IWE is sharper than pixel-wise event accumulation (i.e., IWE with zero optical flow).

We use a simple U-Net [49] as the backbone in the linear flow experiments (similar architecture like EV-FlowNet [76]) and choose polynomial basis $g_j(t) = t^j$ of degree 1, which effectively leads to a linear motion prior. We use $N_{\text{traj}} = 32$ nearest neighbors and $N_{\text{bins}} = 15$ time bins in the displacement map. The weight for the regularizer is $\lambda = 0.003$. The network trains for 50 epochs with and Adam optimizer, a learning rate of 10^{-4}, and a total batch size of 28 on two RTX A6000.
Results. Table 4 shows the DSEC benchmark results, confirming that our model can generate high-quality optical flow results without having access to any GT. It furthermore achieves the best results among all contrast-maximization–based methods on six out of seven sequences, and only methods directly supervised on GT perform better. Within the self-supervised methods, it improves the AE by 19% and the number of inliers by 14%. The only exception is the night sequence *zurich_city_14*. Contrast maximization is based on the brightness constancy assumption, which is violated here by flickering street lights. Hence, in these areas, the contrast loss does not provide reasonable flow predictions. While our methods still provide a better AE and %Out at night, higher outliers in the

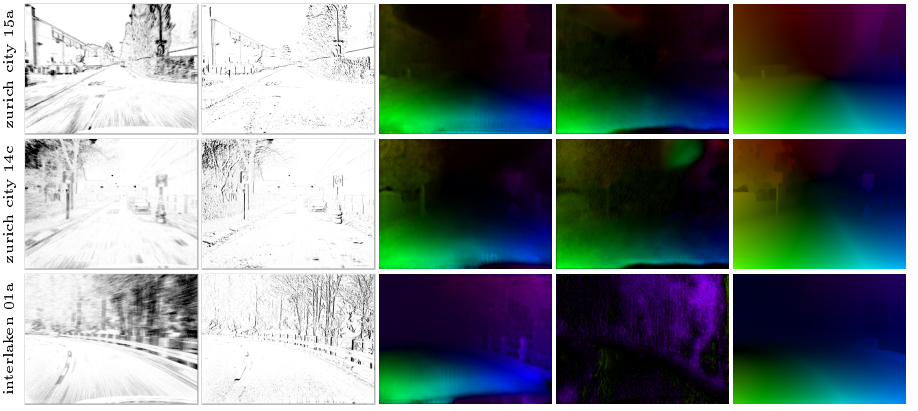


Fig. 5: Results on DSEC. Image of warped events and predicted flow by three methods.

flickering regions prevent the difference in the average EPE from reflecting the improved performance of our method on the other six sequences.

Figure 5 shows qualitative results of our method. In comparison to another self-supervised optical flow method (d), the visualizations are noticeably sharper, allowing for a more precise representation of the motion in the scene. Moreover, the model performs well in delineating the contours of foreground objects, without the over-smoothing effect often observable in flow methods. The IWE (Fig. 5b) shows sharp results, aligning with the overall high FWL values reported in Tab. 4. While the FWL metric also increases in event collapse (e.g., for RTEF [5]), the IWE visualization reveals that here the model is well-regularized, confirming that the predicted motion aligns the events correctly. The third example in Fig. 5 highlights how our model has improved predictions with a single forward pass, where models relying on temporal recurrence (see (d)) fail at the beginning and need a warm-up stage resulting in higher latency.

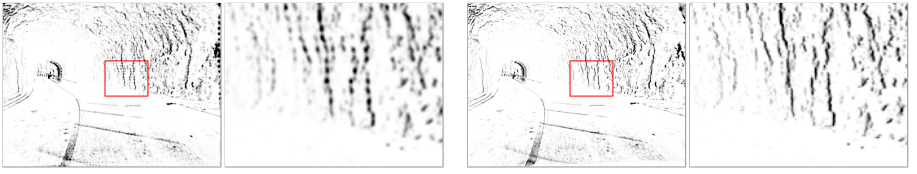
Inference time. Additionally, Tab. 4 provides inference times t_{inf} for several methods. Note that our method is about 5× faster than the competitive baselines. The reason is that we do not rely on any recurrence in this optical flow application, such as recurrence in time or RAFT-inspired refinement steps.

Ablation and Sensitivity. Table 5 lists the performance under different loss settings. It confirms that *the number of neighboring trajectories* N_{traj} in the KNN approach has a regularizing effect and is important for good performance. The decreased performance on the benchmark for a lower N_{traj} aligns with the visual impression in Fig. 6: models with lower N_{traj} show artifacts and a stronger susceptibility to the aperture problem, as is observable on the rock pattern.

Table 5 confirms that the *multi-reference approach* is crucial for regularizing the contrast loss. While using three instead of one t_{ref} improves performance, our approach using a randomized t_{ref} leads to an additional performance boost. Lastly, the non-linear prior showed no performance improvement on DSEC.

Table 5: Sensitivity and Ablation Study for DSEC. Ours corresponds to "Ours" in Tab. 4. Configurations marked with "–" are unchanged from our main result. Δ specifies the change with respect to the original configuration.

	N_{traj}	N_{tref}	N_c	Motion prior	EPE ↓	ΔEPE	AE ↓	ΔAE	%Out ↓	Δ%Out
Ours	32	$\sim\mathcal{U}(0,1)$	1	polynomial	**3.20**		**8.53**		**15.21**	
Number of neighbor trajectories	1	–	–	–	4.58	1.07	13.73	5.20	27.90	12.69
	8	–	–	–	3.51	0.31	13.20	4.67	23.39	8.18
Number of reference times	–	3	–	–	4.46	1.26	14.43	5.90	28.60	13.39
	–	1	–	–	7.26	4.06	18.72	10.19	44.75	29.54
Type and degree of motion prior	–	–	5	learned	3.22	0.02	8.59	0.06	15.34	0.13
	–	–	5	polynomial	3.28	0.06	8.61	0.08	15.68	0.47



$N_{\text{traj}} = 1$ (some event collapse) $N_{\text{traj}} = 32$ (Ours)

Fig. 6: Visual effect of the number of neighbors N_{traj} on the predicted flow.

5 Limitations

Like all CM-based methods, ours is based on the brightness constancy assumption. Therefore, it shows limitations in estimating flow from events that are not caused by motion, e.g., from flickering lights.

Our KNN interpolation shows a good trade-off between granularity and regularization; nevertheless, our method is limited by the aperture problem inherent to optical flow.

Lastly, during training, all raw events are passed to the loss module, increasing training time. Furthermore, gradient calculation is performed through the additional step of event warping. These steps increase memory requirements and training time of our loss compared to supervised methods.

6 Conclusion

We introduced a new loss formulation based on the contrast maximization framework, by combining it with a non-linear trajectory prior. It is a versatile tool that works with a variety of model architectures and trajectory representations. We presented an efficient method to make the high-dimensional assignment between millions of events and thousands of trajectories feasible. Our experiments show clear advantages against supervised methods on unseen data, where under real-world circumstances no GT is available. Additionally, a U-Net trained with our loss shows state-of-the-art performance on the DSEC optical flow benchmark, while being substantially faster than the previously best methods.

Acknowledgments

We thank Dr. Cornelia Fermüller and the NeuroPAC network for fostering collaborations within the event-based community (NSF OISE 2020624). Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2002/1 "Science of Intelligence" – project number 390523135. We furthermore gratefully acknowledge the support by the following grants: NSF FRR 2220868, NSF IIS-RI 2212433, NSF TRIPODS 1934960, ONR N00014-22-1-2677.

References

1. Akhter, I., Sheikh, Y., Khan, S., Kanade, T.: Trajectory space: A dual representation for nonrigid structure from motion. IEEE Trans. Pattern Anal. Mach. Intell. **33**(7), 1442–1456 (jul 2011). https://doi.org/10.1109/TPAMI.2010.201
2. Bailer, C., Varanasi, K., Stricker, D.: CNN-based patch matching for optical flow with thresholded hinge embedding loss. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR). pp. 2710–2719 (2017). https://doi.org/10.1109/CVPR.2017.290
3. Benosman, R., Clercq, C., Lagorce, X., Ieng, S.H., Bartolozzi, C.: Event-based visual flow. IEEE Trans. Neural Netw. Learn. Syst. **25**(2), 407–417 (2014). https://doi.org/10.1109/TNNLS.2013.2273537
4. Benosman, R., Ieng, S.H., Clercq, C., Bartolozzi, C., Srinivasan, M.: Asynchronous frameless event-based optical flow. Neural Netw. **27**, 32–37 (2012). https://doi.org/10.1016/j.neunet.2011.11.001
5. Brebion, V., Moreau, J., Davoine, F.: Real-time optical flow for vehicular perception with low- and high-resolution event cameras. IEEE Trans. Intell. Transport. Syst. pp. 1–13 (2021). https://doi.org/10.1109/TITS.2021.3136358
6. Brosch, T., Tschechne, S., Neumann, H.: On event-based optical flow detection. Front. Neurosci. **9**(137) (Apr 2015). https://doi.org/10.3389/fnins.2015.00137
7. Burner, L., Mitrokhin, A., Fermüller, C., Aloimonos, Y.: EVIMO2: An event camera dataset for motion segmentation, optical flow, structure from motion, and visual inertial odometry in indoor scenes with monocular or stereo algorithms. arXiv e-prints (May 2022). https://doi.org/10.48550/arXiv.2205.03467
8. Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A naturalistic open source movie for optical flow evaluation. In: Eur. Conf. Comput. Vis. (ECCV). pp. 611–625 (2012). https://doi.org/10.1007/978-3-642-33783-3_44
9. Chaney, K., Cladera Ojeda, F., Wang, Z., Bisulco, A., Hsieh, M.A., Korpela, C., Kumar, V., Taylor, C.J., Daniilidis, K.: M3ED: Multi-robot, multi-sensor, multi-environment event dataset. In: IEEE Conf. Comput. Vis. Pattern Recog. Workshops (CVPRW). pp. 4016–4023 (2023). https://doi.org/10.1109/CVPRW59228.2023.00419
10. Chui, J., Klenk, S., Cremers, D.: Event-based feature tracking in continuous time with sliding window optimization. In: arXiv e-prints (2021). https://doi.org/10.48550/arXiv.2107.04536
11. Ding, Z., Zhao, R., Zhang, J., Gao, T., Xiong, R., Yu, Z., Huang, T.: Spatio-temporal recurrent networks for event-based optical flow estimation. In: AAAI Conf. Artificial Intell. vol. 36, pp. 525–533 (2022). https://doi.org/10.1609/aaai.v36i1.19931

12. Doersch, C., Gupta, A., Markeeva, L., Recasens, A., Smaira, L., Aytar, Y., Carreira, J., Zisserman, A., Yang, Y.: Tap-vid: A benchmark for tracking any point in a video. Adv. Neural Inf. Process. Syst. (NeurIPS) **35**, 13610–13626 (2022)
13. Doersch, C., Yang, Y., Vecerik, M., Gokay, D., Gupta, A., Aytar, Y., Carreira, J., Zisserman, A.: Tapir: Tracking any point with per-frame initialization and temporal refinement. arXiv preprint arXiv:2306.08637 (2023)
14. Dosovitskiy, A., Fischer, P., Ilg, E., Häusser, P., Hazırbaş, C., Golkov, V., van der Smagt, P., Cremers, D., Brox, T.: FlowNet: Learning optical flow with convolutional networks. In: Int. Conf. Comput. Vis. (ICCV). pp. 2758–2766 (2015). https://doi.org/10.1109/ICCV.2015.316
15. Feydy, J., Glaunès, A., Charlier, B., Bronstein, M.: Fast geometric learning with symbolic matrices. Adv. Neural Inf. Process. Syst. (NeurIPS) **33**, 14448–14462 (2020)
16. Finateu, T., Niwa, A., Matolin, D., Tsuchimoto, K., Mascheroni, A., Reynaud, E., Mostafalu, P., Brady, F., Chotard, L., LeGoff, F., Takahashi, H., Wakabayashi, H., Oike, Y., Posch, C.: A 1280x720 back-illuminated stacked temporal contrast event-based vision sensor with 4.86μm pixels, 1.066Geps readout, programmable event-rate controller and compressive data-formatting pipeline. In: IEEE Int. Solid-State Circuits Conf. (ISSCC). pp. 112–114 (2020). https://doi.org/10.1109/ISSCC19947.2020.9063149
17. Gallego, G., Delbruck, T., Orchard, G., Bartolozzi, C., Taba, B., Censi, A., Leutenegger, S., Davison, A., Conradt, J., Daniilidis, K., Scaramuzza, D.: Event-based vision: A survey. IEEE Trans. Pattern Anal. Mach. Intell. **44**(1), 154–180 (2022). https://doi.org/10.1109/TPAMI.2020.3008413
18. Gallego, G., Gehrig, M., Scaramuzza, D.: Focus is all you need: Loss functions for event-based vision. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR). pp. 12272–12281 (2019). https://doi.org/10.1109/CVPR.2019.01256
19. Gallego, G., Rebecq, H., Scaramuzza, D.: 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). pp. 3867–3876 (2018). https://doi.org/10.1109/CVPR.2018.00407
20. Gallego, G., Scaramuzza, D.: Accurate angular velocity estimation with an event camera. IEEE Robot. Autom. Lett. **2**(2), 632–639 (2017). https://doi.org/10.1109/LRA.2016.2647639
21. Gehrig, D., Gehrig, M., Hidalgo-Carrió, J., Scaramuzza, D.: Video to Events: Recycling video datasets for event cameras. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR). pp. 3583–3592 (2020). https://doi.org/10.1109/CVPR42600.2020.00364
22. Gehrig, D., Loquercio, A., Derpanis, K.G., Scaramuzza, D.: End-to-end learning of representations for asynchronous event-based data. In: Int. Conf. Comput. Vis. (ICCV). pp. 5632–5642 (2019). https://doi.org/10.1109/ICCV.2019.00573
23. Gehrig, M., Aarents, W., Gehrig, D., Scaramuzza, D.: DSEC: A stereo event camera dataset for driving scenarios. IEEE Robot. Autom. Lett. **6**(3), 4947–4954 (2021). https://doi.org/10.1109/LRA.2021.3068942
24. Gehrig, M., Millhäusler, M., Gehrig, D., Scaramuzza, D.: E-RAFT: Dense optical flow from event cameras. In: Int. Conf. 3D Vision (3DV). pp. 197–206 (2021). https://doi.org/10.1109/3DV53792.2021.00030
25. Gehrig, M., Muglikar, M., Scaramuzza, D.: Dense continuous-time optical flow from event cameras. IEEE Trans. Pattern Anal. Mach. Intell. pp. 1–12 (2024). https://doi.org/10.1109/TPAMI.2024.3361671

26. Guney, F., Sevilla-Lara, L., Sun, D., Wulff, J.: "what is optical flow for?": Workshop results and summary. In: Eur. Conf. Comput. Vis. Workshops (ECCVW). pp. 0–0 (2018)
27. Guo, S., Gallego, G.: CMax-SLAM: Event-based rotational-motion bundle adjustment and SLAM system using contrast maximization. IEEE Trans. Robot. **40**, 2442–2461 (2024). https://doi.org/10.1109/TRO.2024.3378443
28. Hagenaars, J., Paredes-Vallés, F., De Croon, G.: Self-supervised learning of event-based optical flow with spiking neural networks. Adv. Neural Inf. Process. Syst. (NeurIPS) **34**, 7167–7179 (2021)
29. Hamann, F., Gallego, G.: Stereo co-capture system for recording and tracking fish with frame- and event cameras. In: 26th Int. Conf. on Pattern Recognition (ICPR), Visual observation and analysis of Vertebrate And Insect Behavior (VAIB) Workshop (2022). https://doi.org/10.48550/ARXIV.2207.07332
30. Hamann, F., Ghosh, S., Juárez-Martínez, I., Hart, T., Kacelnik, A., Gallego, G.: Low-power, continuous remote behavioral localization with event cameras. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR) (2024)
31. Harley, A.W., Fang, Z., Fragkiadaki, K.: Particle video revisited: Tracking through occlusions using point trajectories. In: Eur. Conf. Comput. Vis. (ECCV). pp. 59–75 (2022)
32. Huang, Z., Shi, X., Zhang, C., Wang, Q., Cheung, K.C., Qin, H., Dai, J., Li, H.: Flowformer: A transformer architecture for optical flow. In: Eur. Conf. Comput. Vis. (ECCV). pp. 668–685 (2022)
33. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: FlowNet 2.0: Evolution of optical flow estimation with deep networks. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR). pp. 1647–1655 (2017). https://doi.org/10.1109/cvpr.2017.179
34. Janai, J., Guney, F., Ranjan, A., Black, M., Geiger, A.: Unsupervised learning of multi-frame optical flow with occlusions. In: Eur. Conf. Comput. Vis. (ECCV). pp. 690–706 (2018)
35. Jonschkowski, R., Stone, A., Barron, J.T., Gordon, A., Konolige, K., Angelova, A.: What matters in unsupervised optical flow. In: Eur. Conf. Comput. Vis. (ECCV). pp. 557–572 (2020)
36. Karaev, N., Rocco, I., Graham, B., Neverova, N., Vedaldi, A., Rupprecht, C.: Cotracker: It is better to track together. arXiv preprint arXiv:2307.07635 (2023)
37. Kim, H., Kim, H.J.: Real-time rotational motion estimation with contrast maximization over globally aligned events. IEEE Robot. Autom. Lett. **6**(3), 6016–6023 (2021). https://doi.org/10.1109/LRA.2021.3088793
38. Lee, C., Kosta, A., Zhu, A.Z., Chaney, K., Daniilidis, K., Roy, K.: Spike-flownet: Event-based optical flow estimation with energy-efficient hybrid neural networks. In: Eur. Conf. Comput. Vis. (ECCV). pp. 366–382 (2020)
39. Lichtsteiner, P., Posch, C., Delbruck, T.: A 128×128 120 dB 15 μs latency asynchronous temporal contrast vision sensor. IEEE J. Solid-State Circuits **43**(2), 566–576 (2008). https://doi.org/10.1109/JSSC.2007.914337
40. Liu, H., Chen, G., Qu, S., Zhang, Y., Li, Z., Knoll, A., Jiang, C.: TMA: Temporal motion aggregation for event-based optical flow. In: Int. Conf. Comput. Vis. (ICCV). pp. 9651–9660 (Oct 2023). https://doi.org/10.1109/ICCV51070.2023.00888
41. Liu, M., Delbruck, T.: Adaptive time-slice block-matching optical flow algorithm for dynamic vision sensors. In: British Mach. Vis. Conf. (BMVC). pp. 1–12 (2018)

42. Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T.: A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR). pp. 4040–4048 (2016)
43. Orchard, G., Benosman, R., Etienne-Cummings, R., Thakor, N.V.: A spiking neural network architecture for visual motion estimation. In: IEEE Biomed. Circuits Syst. Conf. (BioCAS). pp. 298–301 (2013). https://doi.org/10.1109/biocas.2013.6679698
44. Paredes-Vallés, F., Scheper, K.Y., De Wagter, C., de Croon, G.C.: Taming contrast maximization for learning sequential, low-latency, event-based optical flow. In: Int. Conf. Comput. Vis. (ICCV). pp. 9661–9671 (Oct 2023). https://doi.org/10.1109/ICCV51070.2023.00889
45. Paredes-Vallés, F., Hagenaars, J.J., Dupeyroux, J., Stroobants, S., Xu, Y., de Croon, G.C.H.E.: Fully neuromorphic vision and control for autonomous drone flight. Science Robotics **9**(90), eadi0591 (2024). https://doi.org/10.1126/scirobotics.adi0591
46. Posch, C., Serrano-Gotarredona, T., Linares-Barranco, B., Delbruck, T.: Retinomorphic event-based vision sensors: Bioinspired cameras with spiking output. Proc. IEEE **102**(10), 1470–1484 (Oct 2014). https://doi.org/10.1109/jproc.2014.2346153
47. Rebecq, H., Gehrig, D., Scaramuzza, D.: ESIM: an open event camera simulator. In: Conf. on Robotics Learning (CoRL). Proc. Machine Learning Research, vol. 87, pp. 969–982. PMLR (2018)
48. Ren, Z., Yan, J., Ni, B., Liu, B., Yang, X., Zha, H.: Unsupervised deep learning for optical flow estimation. In: AAAI Conf. Artificial Intell. vol. 31 (2017)
49. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. In: Int. Conf. Medical Image Computing and Computer-Assisted Intervention (MICCAI). pp. 234–241 (2015)
50. Sand, P., Teller, S.: Particle video: Long-range motion estimation using point trajectories. Int. J. Comput. Vis. **80**, 72–91 (2008)
51. Seok, H., Lim, J.: Robust feature tracking in dvs event stream using Bezier mapping. In: IEEE Winter Conf. Appl. Comput. Vis. (WACV). pp. 1647–1656 (2020). https://doi.org/10.1109/WACV45572.2020.9093607
52. Shiba, S., Aoki, Y., Gallego, G.: Event collapse in contrast maximization frameworks. Sensors **22**(14), 1–20 (2022). https://doi.org/10.3390/s22145190
53. Shiba, S., Aoki, Y., Gallego, G.: A fast geometric regularizer to mitigate event collapse in the contrast maximization framework. Adv. Intell. Syst. p. 2200251 (2022). https://doi.org/10.1002/aisy.202200251
54. Shiba, S., Aoki, Y., Gallego, G.: Secrets of event-based optical flow. In: Eur. Conf. Comput. Vis. (ECCV). pp. 628–645 (2022). https://doi.org/10.1007/978-3-031-19797-0_36
55. Shiba, S., Hamann, F., Aoki, Y., Gallego, G.: Event-based background oriented schlieren. IEEE Trans. Pattern Anal. Mach. Intell. **46**(4), 2011–2026 (2024). https://doi.org/10.1109/TPAMI.2023.3328188
56. Shiba, S., Klose, Y., Aoki, Y., Gallego, G.: Secrets of event-based optical flow, depth, and ego-motion by contrast maximization. IEEE Trans. Pattern Anal. Mach. Intell. pp. 1–18 (2024). https://doi.org/10.1109/TPAMI.2024.3396116
57. Stoffregen, T., Scheerlinck, C., Scaramuzza, D., Drummond, T., Barnes, N., Kleeman, L., Mahony, R.: Reducing the sim-to-real gap for event cameras. In: Eur. Conf. Comput. Vis. (ECCV). pp. 534–549 (2020). https://doi.org/https://doi.org/10.1007/978-3-030-58583-9_32

58. Stone, A., Maurer, D., Ayvaci, A., Angelova, A., Jonschkowski, R.: Smurf: Self-teaching multi-frame unsupervised raft with full-image warping. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR). pp. 3887–3896 (2021)
59. Sun, D., Vlasic, D., Herrmann, C., Jampani, V., Krainin, M., Chang, H., Zabih, R., Freeman, W.T., Liu, C.: Autoflow: Learning a better training set for optical flow. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR). pp. 10093–10102 (2021)
60. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR) (2018)
61. Sun, X., Harley, A.W., Guibas, L.J.: Refining pre-trained motion models. arXiv preprint arXiv:2401.00850 (2024)
62. Teed, Z., Deng, J.: RAFT: Recurrent all pairs field transforms for optical flow. In: Eur. Conf. Comput. Vis. (ECCV). pp. 402–419 (2020). https://doi.org/10.1007/978-3-030-58536-5_24
63. Tulyakov, S., Bochicchio, A., Gehrig, D., Georgoulis, S., Li, Y., Scaramuzza, D.: Time lens++: Event-based frame interpolation with parametric non-linear flow and multi-scale fusion. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR). pp. 17755–17764 (Jun 2022)
64. Valmadre, J., Lucey, S.: General trajectory prior for non-rigid reconstruction. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR). pp. 1394–1401 (2012)
65. Wan, Z., Dai, Y., Mao, Y.: Learning dense and continuous optical flow from an event camera. IEEE Trans. Image Process. **31**, 7237–7251 (2022)
66. Wang, C., Eckart, B., Lucey, S., Gallo, O.: Neural trajectory fields for dynamic novel view synthesis. arXiv preprint arXiv:2105.05994 (2021)
67. Wang, Z., Hamann, F., Chaney, K., Jiang, W., Gallego, G., Daniilidis, K.: Event-based continuous color video decompression from single frames. arXiv preprint arXiv:2312.00113 (2023)
68. Weikersdorfer, D., Hoffmann, R., Conradt, J.: Simultaneous localization and mapping for event-based vision systems. In: Int. Conf. Comput. Vis. Syst. (ICVS). pp. 133–142 (2013). https://doi.org/10.1007/978-3-642-39402-7_14
69. Wu, Y., Paredes-Vallés, F., de Croon, G.C.: Lightweight event-based optical flow estimation via iterative deblurring. arXiv preprint arXiv:2211.13726 (2022)
70. Ye, C., Mitrokhin, A., Parameshwara, C., Fermüller, C., Yorke, J.A., Aloimonos, Y.: Unsupervised learning of dense optical flow, depth and egomotion with event-based sensors. In: IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS). pp. 5831–5838 (2020). https://doi.org/10.1109/IROS45743.2020.9341224
71. Young, P., Lai, A., Hodosh, M., Hockenmaier, J.: From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. Trans. Assoc. Computational Linguistics **2**, 67–78 (2014)
72. Yu, J.J., Harley, A.W., Derpanis, K.G.: Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. In: Eur. Conf. Comput. Vis. Workshops (ECCVW). pp. 3–10 (2016)
73. Zen, G., Ricci, E., Sebe, N.: Exploiting sparse representations for robust analysis of noisy complex video scenes. In: Eur. Conf. Comput. Vis. (ECCV). pp. 199–213 (2012)
74. Zheng, Y., Harley, A.W., Shen, B., Wetzstein, G., Guibas, L.J.: PointOdyssey: A large-scale synthetic dataset for long-term point tracking. In: Int. Conf. Comput. Vis. (ICCV). pp. 19855–19865 (2023)
75. Zhu, A.Z., Thakur, D., Ozaslan, T., Pfrommer, B., Kumar, V., Daniilidis, K.: The multivehicle stereo event camera dataset: An event camera dataset for 3D

perception. IEEE Robot. Autom. Lett. **3**(3), 2032–2039 (Jul 2018). https://doi.org/10.1109/lra.2018.2800793
76. Zhu, A.Z., Yuan, L., Chaney, K., Daniilidis, K.: EV-FlowNet: Self-supervised optical flow estimation for event-based cameras. In: Robotics: Science and Systems (RSS). pp. 1–9 (2018). https://doi.org/10.15607/RSS.2018.XIV.062
77. Zhu, A.Z., Yuan, L., Chaney, K., Daniilidis, K.: Unsupervised event-based learning of optical flow, depth, and egomotion. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR). pp. 989–997 (2019). https://doi.org/10.1109/CVPR.2019.00108
78. Zhu, Y., Lucey, S.: Convolutional sparse coding for trajectory reconstruction. IEEE Trans. Pattern Anal. Mach. Intell. **37**(3), 529–540 (2013)