Time-to-Contact Map by Joint Estimation of Up-to-Scale Inverse Depth and Global Motion using a Single Event Camera

Urbano Miguel Nunes, Laurent U Perrinet, Sio-Hoi Ieng

Time-to-Contact Map by Joint Estimation of Up-to-Scale Inverse Depth and Global Motion using a Single Event Camera

Urbano Miguel Nunes[1], Laurent Udo Perrinet[2] and Sio-Hoi Ieng[1]
[1]Sorbonne University, [2]Aix Marseille Univ, CNRS, INT, Institut de Neurosciences de la Timone
urbano.goncalves-nunes@inserm.fr, laurent.perrinet@univ-amu.fr, sio-hoi.ieng@upmc.fr

Abstract

Event cameras asynchronously report brightness changes with a temporal resolution in the order of microseconds, which makes them inherently suitable to address problems that involve rapid motion perception. In this paper, we address the problem of time-to-contact (TTC) estimation using a single event camera. This problem is typically addressed by estimating a single global TTC measure, which explicitly assumes that the surface/obstacle is planar and fronto-parallel. We relax this assumption by proposing an incremental event-based method to estimate the TTC that jointly estimates the (up-to scale) inverse depth and global motion using a single event camera. The proposed method is reliable and fast while asynchronously maintaining a TTC map (TTCM), which provides per-pixel TTC estimates. As a side product, the proposed method can also estimate per-event optical flow. We achieve state-of-the-art performances on TTC estimation in terms of accuracy and runtime per event while achieving competitive performance on optical flow estimation.

1. Introduction

Event cameras differ from standard frame-based cameras, which capture visual data at a fixed rate and independently of the observing scene. Instead, event cameras respond asynchronously to pixel-wise brightness changes by generating *events* [6, 25]. Event cameras are thus data-driven sensors that offer several advantages, including high temporal resolution in the order of microseconds, low latency, low power consumption, and high dynamic range. These properties place event cameras as suitable candidates to address vision-based problems that involve (high-speed) motion, *e.g.*, optical flow estimation [4, 27, 51], ego-motion estimation [15, 21, 22, 34], motion segmentation [33, 44], and obstacle avoidance [10, 12]. Due to the distinct visual sensing paradigm, however, new methods are necessary to fully exploit the potential of event cameras [13, 23].

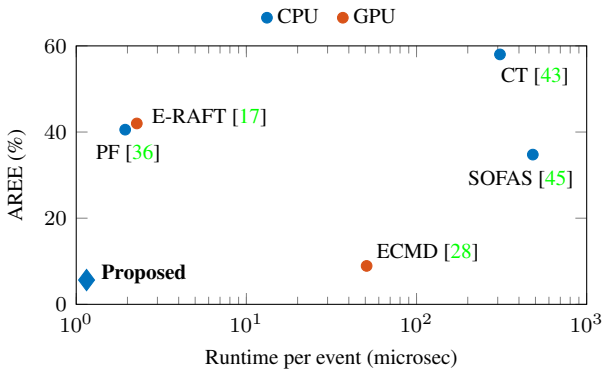

Figure 1. **Runtime *vs*. accuracy comparison** for TTC estimation methods. Average results on the Ventral Landing benchmark [28].

Event cameras have been used to address the problem of fast TTC estimation, which comes up often in the vision-based obstacle avoidance [10, 12, 29, 39] and ventral landing [28, 36] literature. The TTC is the time that would elapse before a camera reaches an obstacle/surface, assuming the current relative motion between them remains constant [10]. Previous methods that use a single event camera have focused on estimating a single global TTC measure, which assumes that the surface is planar and fronto-parallel. To overcome this limitation, other methods use additional sensing, *e.g.*, depth frames, to build a dense TTCM [20, 47].

We instead propose to extend the Dispersion Minimization (DMin) framework [34] to estimate the TTC for each incoming event using a single event camera. The proposed method jointly estimates the relative global motion and per-event (up-to scale) inverse depth. We can then asynchronously maintain a semi-dense TTCM which provides per-pixel TTC estimates or compute a global TTC measure with greater accuracy by averaging over the TTC estimates. Since there is at least one scaling degree of freedom (DOF), we also propose an effective strategy to mitigate event collapse [40, 41]. The proposed method is also computationally fast, reaching $\sim$1 microsecond processing time per event on a standard laptop. Fig. 1 compares the

runtime *vs*. accuracy for TTC estimation methods, whereby the proposed method achieves state-of-the-art performance. We also estimate the per-event optical flow as a side product and achieve competitive performance compared to state-of-the-art optical flow methods that use events.

Main contributions:

1. First event-based method that explicitly estimates the TTC for each event and maintains a semi-dense TTCM using a single event camera.
2. DMin framework [34] extension to jointly handle local and global estimates, *i.e*., inverse depth and global motion, respectively.
3. Effective approach that mitigates event collapse [40, 41] for incremental event-based estimation.

1.1. Time-to-Contact

Consider a freely moving camera with angular and linear velocities $\boldsymbol{\omega}(t) = (\omega_x(t), \omega_y(t), \omega_z(t))^\mathsf{T}$ and $\boldsymbol{\nu}(t) = (\nu_x(t), \nu_y(t), \nu_z(t))^\mathsf{T}$, respectively, that is observing a point $\boldsymbol{\alpha}$, with 3D coordinates $\boldsymbol{\alpha}(t) = (X(t), Y(t), Z(t))^\mathsf{T}$ relative to the camera, as shown in Fig. 2. $Z(t)$ is also referred as the *depth* of point $\boldsymbol{\alpha}$ relative to the camera. The instantaneous TTC between the moving camera and point $\boldsymbol{\alpha}$ is thus given by:

$$\tau(t) \coloneqq -\frac{Z(t)}{\frac{dZ(t)}{dt}} = \frac{Z(t)}{\nu_z(t)}. \tag{1}$$

The minus sign disappears because we define the linear velocity w.r.t. the camera's frame of reference, not w.r.t. the point's frame of reference, *i.e*., $\nu_z(t) = -dZ(t)/dt$. Based on Eq. (1), the exact values of depth and relative approaching motion do not need to be estimated, only the ratio between them.

2. Related Work

We review recent related works on the following topics: TTC, global motion and optical flow estimation. We refer to [13] for a detailed survey.

Time-to-Contact Estimation. The first work on event-based TTC using a single event camera relied solely on the estimation of visual motion flows [10], whereby the motion flows were computed by fitting a local plane to the time surface [4]. Other works that followed were geared towards two main use cases, namely obstacle avoidance [12, 29, 39] and ventral landing [28, 36, 43]. Event-based obstacle avoidance methods are built to be fast reacting and, although they come from either bio-inspired [29, 39] or mathematically grounded principles [12], they typically rely on empirically-validated heuristics to speed-up computations. Existing event-based ventral landing approaches only compute a single TTC estimate, which assumes that the surface is planar and fronto-parallel. To overcome this assumption, other works fuse events with additional sensory information, *e.g*. depth [47]. The proposed approach uses a single event camera while being mathematically grounded and computationally fast.

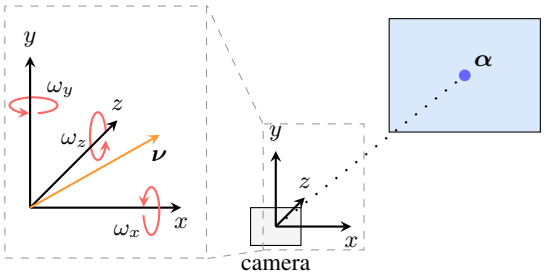

Figure 2. Camera coordinate system.

Global Motion Estimation. Also denoted by ego-motion estimation [13], it refers to estimating the parameters that explain the triggered events according to some global motion model. These methods can be broadly characterized by whether they rely on key-frame registration [8, 9, 14, 21, 22, 38] or perform the estimation without relying on any key-frame [15, 34]. The former methods are reminiscent of the frame-based paradigm and also include methods based on artificial neural networks (ANN) [16, 52], whereby an intermediate frame-based representation is needed for the estimation. The latter methods tend towards a more event-based processing paradigm and include methods based on spiking neural networks (SNN) [18, 37], whereby events are either processed on an event-by-event basis [34] or in batches [15]. Although both approaches have advantages and disadvantages, methods that rely on key-frame registration typically require an intermediate frame-based representation, which is still an open problem in the event-based community. Similarly to [22], the proposed method jointly estimates up-to scale inverse depth and global motion. By building on the DMin framework [34], which also allows processing events on an event-by-event basis, our method does not rely on key-frame registration or background inverse depth regularization to improve convergence.

Optical Flow Estimation. Several model-based methods have been previously proposed, which can be further divided into: frame-based [2, 7, 26, 27], batch-based [42, 49], and event-based [1, 3, 4]. By selecting the most relevant events, *i.e*., typically the most recent, frame-based methods build frames from which the optical flow is computed using techniques from standard image-based optical flow, *e.g*., Lucas-Kanade [5]. Since each event does not carry much information on its own, batch-based methods aggregate the most recent events by forming batches but perform the computations directly on the events. Event-based methods follow the most event-driven paradigm by performing event-by-event processing, typically being the computationally fastest. However, event-based methods tend to suffer more

from the aperture problem since all the computations are performed locally, and thus frame-based and model-based methods achieve currently better accuracy. In terms of accuracy, learning-based ANN methods [11, 35, 46, 52] generally achieve state-of-the-art performance. Besides the need to convert events into frames for more efficient processing, these methods are known to be very data hungry, sensitive to the training data [48], and consume large amounts of energy [27]. Another line of research in learning-based methods has been to use SNN [19, 24], which combine the event-based processing and learning paradigms and thus do not require an intermediate frame-based representation. However, it is not trivial to train SNN, and the empirical validation is still not on par with ANN methods. Although it is not the primary objective of this work, the proposed method can provide per-event flow estimates while still being competitive in terms of accuracy w.r.t. state-of-the-art methods.

3. Method

In this section, we describe the proposed incremental event-based method for TTCM estimation. We first briefly review the event cameras' working principle, and the DMin framework [34], based on which we develop the proposed method. Refer to the supplementary material for the full mathematical derivations and additional details.

3.1. Event Cameras and Dispersion Minimization

Event cameras output a stream of asynchronous temporal contrast events $\{e_i\}, i \in \mathbb{N}$. Each event e_i represents a spatio-temporal asynchronous brightness change, being defined as a tuple $e_i := (\mathbf{x}_i, t_i, p_i)$, where $\mathbf{x}_i = (x_i, y_i)$ are the pixel coordinates, t_i is the timestamp at which the event was generated, and $p_i \in \{-1, +1\}$ is its polarity. An event e_i is generated when the change in log-brightness $\log \mathbf{I}_{x,y}(t) := \bar{\mathbf{I}}_{x,y}(t)$ is above a threshold L

$$|\Delta\bar{\mathbf{I}}_{x_i,y_i}(t_i)| = |\bar{\mathbf{I}}_{x_i,y_i}(t_i) - \bar{\mathbf{I}}_{x_i,y_i}(t_i - \Delta t_i)| \geq L, \quad (2)$$

where Δt_i is the time since the last event at the same pixel.

The DMin framework [34] estimates the parameters $\boldsymbol{\theta}$ of a transformation model $\mathcal{M}$ from the stream of events $\mathcal{E} = \{e_i\}_{i=1}^{N_e}$ by minimizing a dispersion measure of the transformed events $\mathbf{f}_i = \mathcal{M}(e_i; t_{\text{ref}}, \boldsymbol{\theta})$. We consider the Potential measure with a Gaussian kernel $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$:

$$P(\mathcal{E}; \boldsymbol{\theta}) = -\sum_{i,j}^{N_e} \mathcal{N}(\mathbf{f}_i; \mathbf{f}_j, \mathbf{I}), \quad (3)$$

where $\mathbf{I}$ is the identity matrix. A key distinction of the DMin framework is that it allows to incrementally estimate the model parameters $\boldsymbol{\theta}$ on an event-by-event basis, whereby the model parameters $\boldsymbol{\theta}$ can be iteratively solved by linearizing the transformation model $\mathcal{M}$, such that:

$$\mathbf{f}_i = \mathcal{M}(e_i; t_{\text{ref}}, \boldsymbol{\theta}) = \tilde{\mathbf{f}}_i + \Delta t_{i,\,\text{ref}} \mathbf{B}_i \boldsymbol{\theta}, \; \mathbf{B}_i := \mathbf{B}(e_i), \quad (4)$$

where $\mathbf{B}_i$ is the model-dependent linearization matrix. The parameters $\boldsymbol{\theta}^*$ that minimize Eq. (3) are thus given by:

$$\boldsymbol{\theta}^* = -\left(\sum_{i,j}^{N_e} \mathbf{W}_{i,j} \Delta\mathbf{C}_{i,j}\right)^{-1} \left(\sum_{i,j}^{N_e} \mathbf{W}_{i,j} \Delta\tilde{\mathbf{f}}_{i,j}\right), \quad (5)$$

where $\Delta\mathbf{C}_{i,j} = \mathbf{C}_i - \mathbf{C}_j$, $\Delta\tilde{\mathbf{f}}_{i,j} = \tilde{\mathbf{f}}_i - \tilde{\mathbf{f}}_j$, $\mathbf{W}_{i,j} = w_{i,j} \partial\Delta\mathbf{f}_{i,j}^{\mathsf{T}} / \partial\boldsymbol{\theta}$, $\mathbf{C}_i = \Delta t_{i,\text{ref}} \mathbf{B}_i$ and $w_{i,j} = \mathcal{N}(\mathbf{f}_i; \mathbf{f}_j, \mathbf{I})$.

3.2. Adapted Dispersion Minimization

While the DMin method [34] provides a general framework for global incremental event-based model estimation, it can also be adapted to jointly estimate global and local measures, *i.e.*, global angular and linear velocities and local inverse depth. However, the DMin framework may encounter estimation issues when the global model has at least one scaling DOF, as noted in [34, 40, 41], known as *event collapse*. Event collapse occurs when the events are transformed into a single point, which minimizes the events' dispersion or maximizes the image contrast while the parameters' estimates diverge. So far, to the best of our knowledge, the mitigation discussion in the literature has been on how to constrain the optimization loss to discourage divergent estimates by analyzing the effects of the scaling transformations on the event-based data. Several mitigation strategies have thus been proposed on the events [34, 40] and parameters level [41] by adding terms to regularize the objective measure. While these strategies generally prevent event collapse, they increase the complexity of the optimization framework and introduce additional parameters to tune.

We instead observe that event collapse fundamentally stems from the event transformation to a common time reference t_{ref}, *e.g.*, given by Eq. (4), by identifying two problems: 1) the constant velocity assumption may not hold depending on the time difference $\Delta t_{i,\text{ref}}$, and 2) there is no built-in constraint on the magnitude of the model parameters, *e.g.*, such that the difference between transformed events $\mathbf{f}_i - \mathbf{f}_j$ explicitly penalizes divergent estimates; although, according to Eq. (4), $\mathbf{f}_i$ and $\mathbf{f}_j$ individually diverge if the model parameters also diverge, their difference $\mathbf{f}_i - \mathbf{f}_j$ is not guaranteed to diverge and thus penalize divergent estimates. The first problem is typically addressed by heuristically making the time difference as short as possible, and its effects are of limited significance in practice. The second problem, however, is intrinsically linked to batch-based processing since the alignment of the events in a batch needs to be measured in some common time reference [15, 32].

However, for incremental event-based processing, the event transformation and, consequently, the dispersion measure can be modified without loss of generality such that the event collapse is prevented by implicitly addressing the two problems identified. Fig. 3 depicts the idea whereby we only transform the event neighbors to the current event's

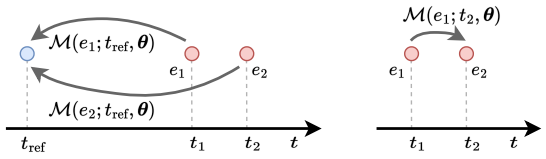

Figure 3. **Incremental event collapse mitigation.** Instead of transforming all the events to some time reference (left), we *locally* transform the events to the current event's timestamp (right).

timestamp instead of transforming all the events to a common time reference, including the current event. Formally, instead of transforming the events according to Eq. (4), only the neighboring events e_j of the current event e_i are transformed according to

$$\mathbf{x}'_j = \mathbf{x}_j + \Delta t_{i,j} \mathbf{B}_j \boldsymbol{\theta} = \mathbf{x}_j + \mathbf{C}_{i,j} \boldsymbol{\theta}, \; e_j \in \text{neigh}(e_i), \quad (6)$$

where 'neigh' is shorthand for neighborhood, and the dependency of $\mathbf{x}'_j$ on $\boldsymbol{\theta}$ was omitted for brevity. From the resultant residual $\mathbf{r}_{i,j} = \mathbf{x}_i - \mathbf{x}'_j = \Delta \mathbf{x}_{i,j} - \mathbf{C}_{i,j} \boldsymbol{\theta}$, we see that the proposed modification to the DMin framework addresses both identified problems. First, the constant velocity assumption is better held since the time difference satisfies $\Delta t_{i,j} \leq \Delta t_{i,\text{ref}}$. Second, if the model parameters $\boldsymbol{\theta}$ diverge, then the residual $\mathbf{r}_{i,j}$ also diverges, which effectively penalizes divergent estimates. We highlight that the proposed adaptation only works for incremental event-based processing: it is not suitable for batch-based processing since the proposed transformation only works locally.

The Potential measure is modified by computing the difference between the current event's coordinates $\mathbf{x}_i$ and the neighboring events' transformed coordinates $\mathbf{x}'_j$:

$$P(\mathcal{E}; \boldsymbol{\theta}) = -\sum_{i,j} \mathcal{N}\left(\mathbf{x}'_j; \mathbf{x}_i, \mathbf{I}\right), \; e_j \in \text{neigh}(e_i). \quad (7)$$

By minimizing Eq. (7) by linearizing the residual according to Taylor's formula $\mathbf{r}_{i,j}(\boldsymbol{\theta} + \Delta\boldsymbol{\theta}) \approx \mathbf{r}_{i,j}(\boldsymbol{\theta}) + \mathbf{J}_{i,j}(\boldsymbol{\theta}) \Delta\boldsymbol{\theta}$, the optimized model parameters $\boldsymbol{\theta}^*$ are iteratively updated:

$$\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}^* + \Delta\boldsymbol{\theta}, \quad (8)$$

$$\Delta\boldsymbol{\theta} = -\underbrace{\left(\sum_j w_{i,j} \mathbf{J}_{i,j}^\mathsf{T} \mathbf{J}_{i,j}\right)}_{\boldsymbol{\Psi}}{}^{-1} \underbrace{\left(\sum_j w_{i,j} \mathbf{J}_{i,j}^\mathsf{T} \mathbf{r}_{i,j}\right)}_{\boldsymbol{\psi}},$$

where $w_{i,j} = \mathcal{N}\left(\mathbf{x}'_j; \mathbf{x}_i, \mathbf{I}\right)$, $\mathbf{J}_{i,j} = \partial \mathbf{r}_{i,j} / \partial \boldsymbol{\theta}$ and $e_j \in \text{neigh}(e_i)$.

3.3. Inverse Depth and Global Motion Model

We consider that a calibrated event camera can freely move and whose global motion is parameterized by the 3D

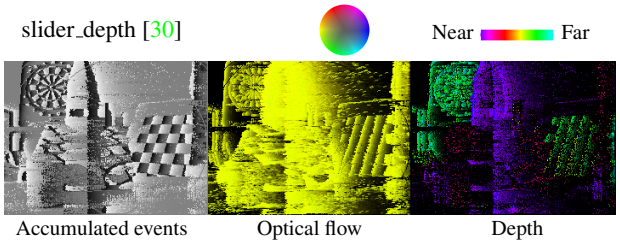

Figure 4. Typical per-event estimation of optical flow and depth.

angular and linear velocities, as defined in Sec. 1.1, $\boldsymbol{\theta} = \left(\boldsymbol{\nu}^\mathsf{T}, \boldsymbol{\omega}^\mathsf{T}\right)^\mathsf{T}$. For each event e_i, we estimate its inverse depth $\rho_i := 1/Z_i$, based on the well-known expression for the apparent velocity on the image plane:

$$\mathbf{v}_i = \begin{pmatrix} \rho_i \mathbf{V}_i & \boldsymbol{\Omega}_i \end{pmatrix} \begin{pmatrix} \boldsymbol{\nu} \\ \boldsymbol{\omega} \end{pmatrix} = \rho_i \mathbf{V}_i \boldsymbol{\nu} + \boldsymbol{\Omega}_i \boldsymbol{\omega} := \mathbf{B}_i \boldsymbol{\theta}, \quad (9)$$

$$\mathbf{V}_i = \begin{pmatrix} -f_x & 0 & x_i - c_x \\ 0 & -f_y & y_i - c_y \end{pmatrix},$$

$$\boldsymbol{\Omega}_i = \begin{pmatrix} \frac{(x_i - c_x)(y_i - c_y)}{f_y} & -f_x - \frac{(x_i - c_x)^2}{f_x} & (y_i - c_y)\frac{f_x}{f_y} \\ f_y + \frac{(y_i - c_y)^2}{f_y} & -\frac{(x_i - c_x)(y_i - c_y)}{f_x} & -(x_i - c_x)\frac{f_y}{f_x} \end{pmatrix}$$

where we make explicit the dependency on the camera intrinsic parameters, namely the horizontal and vertical focal lengths f_x and f_y, respectively, and the horizontal and vertical focal center coordinates c_x and c_y, respectively. In this paper, we assume that the focal center coordinates represent the focus of expansion (FOE). Fig. 4 shows the typical per-event estimation of optical flow and depth using the proposed method.

3.4. Time-to-Contact Map

For each event e_i, we estimate its inverse depth ρ_i and update the global motion parameters $\boldsymbol{\theta} = \left(\boldsymbol{\nu}^\mathsf{T}, \boldsymbol{\omega}^\mathsf{T}\right)^\mathsf{T}$. The model parameters $\boldsymbol{\gamma}_i$ are formed by stacking the motion parameters and inverse depth $\boldsymbol{\gamma}_i = (\boldsymbol{\theta}^\mathsf{T}, \rho_i)^\mathsf{T}$. We impose a smoothness constraint to Eq. (6), so that neighboring inverse depth estimates are assumed to be equal to ρ_i:

$$\mathbf{x}'_j = \mathbf{x}_j + \Delta t_{i,j} \mathbf{B}_{i,j} \boldsymbol{\theta}, \; e_j \in \text{neigh}(e_i), \quad (10)$$

where $\mathbf{B}_{i,j} = \begin{pmatrix} \rho_i \mathbf{V}_j & \boldsymbol{\Omega}_j \end{pmatrix}$. The iterative update $\Delta\boldsymbol{\gamma}_i$ is given by Eq. (8), where $\mathbf{J}_{i,j} = -\Delta t_{i,j} \begin{pmatrix} \mathbf{B}_{i,j} & \mathbf{V}_j \boldsymbol{\nu} \end{pmatrix}$. We maintain the TTCM by computing the TTC for each event e_i based on Eq. (1): $\tau_i = 1/\left(\rho_i \nu_z\right)$. We can also estimate the global TTC by averaging over the values maintained in the TTCM. Fig. 5 shows the typical per-event estimation of optical flow and TTCM using the proposed method.

3.5. Initialization

The motion parameters $\boldsymbol{\theta}$ are global measures which are estimated by aggregating events, while each inverse depth

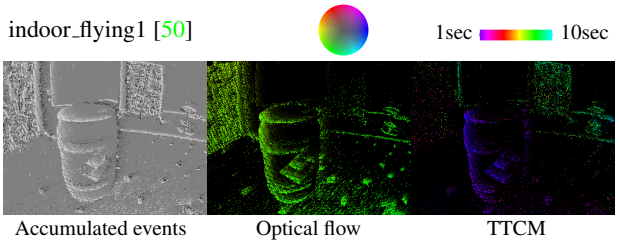

Figure 5. Typical per-event estimation of optical flow and TTCM.

ρ_i value corresponds to the estimate of a single event's inverse depth. The initialization of the motion parameters $\boldsymbol{\theta}$ can thus be almost arbitrarily set, *e.g.*, typically set to 0, and it is only performed once at the beginning. However, a more careful initialization procedure must be considered for the case of inverse depth since it needs to be performed for each event. Based on recent advances in event-based global time decay [31], we perform a weighted average based on the previous neighboring events' inverse depth estimates as the initialization procedure. Each neighboring event's weight $w_i(t)$ is given by:

$$w_i(t) = \frac{1}{1 + a(t)(t - t_i)}, \tag{11}$$

where $a(t)$ is the global *event activity*. If there are no previous neighboring events' inverse depth estimates, which should only occur at the start of the estimation, the initial inverse depth estimate is set to 1.

3.6. Practical Considerations

As mentioned in Sec. 1.1, only the ratio between the depth and relative motion is required to compute the TTC. Since the depth and linear velocities estimates are obtained up-to a scale factor due to the monocular ambiguity, we constrain the linear velocities $\boldsymbol{\nu}$ to have at most unit norm, *i.e.*, $|\boldsymbol{\nu}|_2 \leq 1$, which is useful to improve the method's computational stability by bounding the allowed estimates' values. This is not related to event collapse; rather, it stems from the ratio given by Eq. (1), whereby we can introduce an arbitrary non-zero multiplicative scalar to the numerator and denominator and have the same TTC estimate.

Since the events are generated by 3D points in the field of view (FOV) of the camera and should have positive depth values, we constrain the depth values to be strictly positive by introducing a parameterization variable $\lambda \in \mathbb{R}$ such that $\rho(\lambda) = e^{\lambda} > 0$.

The iterative update given by Eq. (8) can be quite noisy since it is only computed in a small neighborhood $\text{neigh}(e_i) = \{e_k : |\mathbf{x}_k - \mathbf{x}_i|_\infty \leq s\}$. We thus introduce two prior parameters for the global motion parameters and the local inverse depth estimates $l_{\boldsymbol{\theta}}$ and l_ρ, respectively. The resultant iterative update of the model parameters is given by $\Delta\boldsymbol{\gamma}_i = (\boldsymbol{\Psi} + \mathbf{L})^{-1}\left(-\boldsymbol{\psi} + \mathbf{L}(\boldsymbol{\gamma}_{\text{prev}} - \boldsymbol{\gamma}_i^*)\right)$, where $\mathbf{L} = \text{diag}(l_{\boldsymbol{\theta}}, \ldots, l_{\boldsymbol{\theta}}, l_\rho)$, and $\boldsymbol{\gamma}_{\text{prev}}$ are the parameters' estimates from the previous event e_{i-1}. We also weigh each event e_j contribution according to the corresponding weight $w_j(t)$, given by Eq. (11), and discard any event whose weight is below a threshold w_{thresh} [31].

Neighborhood size s	w_{thresh}	$l_{\boldsymbol{\theta}}$	l_ρ
3	0.01	$\propto O$(camera resolution)	1

Table 1. Hyper-parameters used across the experiments.

4. Experimental Evaluation

We evaluate the proposed method in TTC estimation and optical flow estimation, given that the apparent velocity on the image plane can be estimated according to Eq. (9) and due to the lack of event datasets dedicated to TTC estimation. The optical flow benchmark also provides a common ground to compare the proposed method with other state-of-the-art methods that estimate optical flow. Tab. 1 presents the hyper-parameters that were used across the experiments. Since the proposed method computes per-event estimates, we only evaluate on the respective pixel locations[1].

4.1. Datasets and Metrics

VL Dataset [28][2]. It consists of 7 real event sequences observing planar prints of landing surfaces and 1 real event sequence observing the 3D print of a landing surface. Each sequence has 15sec of duration, totaling 120sec of dataset duration. The events were recorded by a Prophesee event camera with 1280×720 resolution, and the ground truth (GT) depth measurements were recorded with an Intel RealSense camera. However, only a global GT depth measurement is provided per timestamp.

To comply with the evaluation reported [28], we assess the proposed method at certain timestamps that correspond to event batches of 0.5sec. The comparison metrics are the divergence REE (%) and runtime per event (microsec). The divergence μ is the inverse of the TTC, *i.e.*, $\mu = 1/\tau$, and the REE is given by:

$$\text{REE} = 100\frac{|\hat{\mu} - \mu_{\text{gt}}|}{|\mu_{\text{gt}}|}, \tag{12}$$

where $\hat{\mu}$ is the estimate and μ_{gt} is the GT. The global motion prior $l_{\boldsymbol{\theta}}$ was set to 1000.

MVSEC Dataset [50][3]. It consists of several real indoor and outdoor sequences, providing events, standard grayscale frames, IMU data, camera poses, and scene depth.

[1] All the code will be made available upon acceptance.
[2] https://github.com/s-mcleod/ventral-landing-event-dataset.
[3] https://daniilidis-group.github.io/mvsec.

		Method	2D-1	2D-2	2D-3	2D-4	2D-5	2D-6	2D-7	3D	Avg.
REE		ECMD [28]	13.48	7.41	**6.11**	12.19	10.41	5.12	**3.72**	12.90	8.92
	$s=2$	Proposed Scaling	7.19	8.79	9.13	12.18	4.98	7.79	4.92	11.34	8.29
		Proposed Translation	**5.03**	7.02	6.17	**5.25**	**4.61**	**4.34**	6.18	6.54	**5.64**
		Proposed 6-DOF	6.77	13.73	14.97	7.50	6.91	12.15	12.77	**6.10**	10.11
	$s=3$	Proposed Scaling	14.99	15.19	11.69	21.69	11.19	15.44	11.01	19.43	15.08
		Proposed Translation	11.15	7.01	7.98	13.80	6.37	11.17	4.55	12.46	9.31
		Proposed 6-DOF	6.51	**6.81**	7.25	6.77	6.06	5.59	6.18	6.19	6.42
Runtime		ECMD [28]	63.37	62.51	51.21	41.36	92.69	38.97	25.57	31.53	50.90
	$s=2$	Proposed Scaling	0.82	0.78	0.80	0.81	0.70	0.84	0.78	0.70	0.78
		Proposed Translation	1.10	1.19	1.22	1.08	1.02	1.21	1.22	1.16	1.15
		Proposed 6-DOF	1.40	1.43	1.45	1.44	1.37	1.54	1.60	1.47	1.46
	$s=3$	Proposed Scaling	1.18	1.23	1.25	1.26	1.12	1.29	1.26	1.75	1.29
		Proposed Translation	1.77	1.70	1.80	1.84	1.68	1.71	1.71	1.70	1.74
		Proposed 6-DOF	2.43	2.40	2.36	2.37	2.48	2.34	2.23	2.27	2.36

Table 2. **Global divergence estimation.** Quantitative results on the VL dataset [28], in terms of REE (%) and runtime per event (microsec) averaged over 100 trials. Lower is better.

The events were recorded by a DAVIS [6] with 346×260 resolution. The evaluated sequences span approximately 265sec. The optical flow GT is also provided [51], generated from the scene depth and camera velocity. We generate GT TTCM's by applying Eq. (1) given the GT depth maps and camera velocity.

To assess optical flow accuracy, we use the following metrics: average endpoint error (AEE) (in pixel/frame, as is conventional in the literature [42, 51])

$$\text{AEE} = \frac{1}{N}\sum_{i=1}^{N} |\hat{\mathbf{m}}_i - \mathbf{m}_{\text{gt},i}|_2 \,, \; \mathbf{m} = \Delta t_{\text{frame}} \mathbf{v}, \tag{13}$$

outliers (Out) as the percentage of pixels with AEE greater than 3, average relative endpoint error (AREE) (%) [27]

$$\text{AREE} = \frac{100}{N}\sum_{i=1}^{N} \frac{|\hat{\mathbf{m}}_i - \mathbf{m}_{\text{gt},i}|_2}{|\mathbf{m}_{\text{gt,i}}|_2}, \tag{14}$$

and average angular error (AAE) (°) [27]

$$\text{AAE} = \frac{1}{N}\sum_{i=1}^{N} \arccos \frac{\hat{\mathbf{m}}_i^{\top}\mathbf{m}_{\text{gt},i}}{|\hat{\mathbf{m}}_i|_2 |\mathbf{m}_{\text{gt},i}|_2}. \tag{15}$$

To assess TTCM accuracy, we use the AREE (%) between the divergence estimate $\hat{\mu}_i$ and corresponding GT $\mu_{\text{gt},i}$. The global motion prior $l_{\boldsymbol{\theta}}$ was set to 100.

4.2. Other Global Motion Models

Based on the general 6-DOF global motion model described in Sec. 3.3, we can consider other more constrained motion models depending on the application, as follows.

Translation. This model is parameterized by the 3D linear velocities $\boldsymbol{\nu}$. Thus, $\mathbf{B}_i = \rho_i \mathbf{V}_i$, being $\mathbf{V}_i$ given by Eq. (9).

Model	indoor_flying1	indoor_flying2	outdoor_day1
Driving	153.30	84.80	47.27
Translation	19.82	28.23	45.78
6-DOF	53.65	56.12	56.66

Table 3. **Divergence estimation.** Quantitative results on the MVSEC dataset [50], in terms of AREE (%). Lower is better.

Driving. This model is parameterized by the most significant DOF's when driving a car, namely the angular velocity around the camera y-axis ω_y and the linear velocity in the z-axis ν_z (see Fig. 2). Hence, $\mathbf{B}_i = \begin{pmatrix} \rho_i(x_i - c_x) & -f_x - \frac{(x_i-c_x)^2}{f_x} \\ \rho_i(y_i - c_y) & -\frac{(x_i-c_x)(y_i-c_y)}{f_x} \end{pmatrix}$. When using this motion model, we impose $|\nu_z|_2 = 1$ to ensure that the depth is properly estimated.

Scaling. This model is parameterized by the linear velocity in the z-axis ν_z. Hence, $\mathbf{B}_i = \rho_i \begin{pmatrix} x_i - c_x \\ y_i - c_y \end{pmatrix}$. It is only considered since the global motion model used in [28] is the 1-DOF scaling, which assumes that the scene is planar and it does not estimate the depth.

4.3. Results

Time-to-Contact. Tab. 2 reports the results on global divergence estimation on the VL benchmark [28]. The proposed method using the Translation model with neighboring size $s = 2$ achieves the best accuracy, outperforming ECMD [28] by 36.77% on average. The proposed method using the Scaling and full 6-DOF models also outperform ECMD [28] in terms of average accuracy with neighboring size $s = 2$ and $s = 3$, respectively. The results indicate that considering a smaller neighborhood benefits global mod-

		Method	indoor_flying1 AEE	indoor_flying1 Out	indoor_flying2 AEE	indoor_flying2 Out	indoor_flying3 AEE	indoor_flying3 Out	outdoor_day1 AEE	outdoor_day1 Out
LB		EV-FlowNet (MB) [52]	0.58	**0.00**	1.02	4.00	0.87	3.00	0.32	**0.00**
LB		EV-FlowNet (HQF) [46]	0.56	1.00	0.66	1.00	0.59	1.00	0.68	1.00
LB		Ding *et al*. [11]	0.57	0.10	0.79	1.60	0.72	1.30	0.42	**0.00**
LB		ConvGRU-EV-FlowNet [19]	0.60	0.51	1.17	8.06	0.93	5.64	0.47	0.25
MB	FB	Brebion *et al*. [7]	0.52	0.10	0.98	5.50	0.71	2.10	0.53	0.20
MB	BB	Shiba *et al*. [42]	**0.42**	0.10	**0.60**	**0.59**	**0.50**	**0.28**	**0.30**	0.10
MB	EB	Aung *et al*. [3]	-	-	-	-	2.31	-	1.26	-
MB	EB	ARMS [1]	1.52	-	1.59	-	1.89	-	2.75	-
MB	EB	Proposed Driving	1.40	7.50	2.42	26.32	2.14	26.05	0.39	0.12
MB	EB	Proposed Translation	0.63	0.41	0.94	2.83	0.79	1.65	0.55	0.97
MB	EB	Proposed 6-DOF	0.88	1.97	1.60	12.02	1.38	9.99	0.81	2.26

Table 4. **Optical flow estimation.** Quantitative results on the MVSEC dataset [50], in terms of AEE (pixel/frame) and Out (%). Lower is better. Legend: learning-based (LB), model-based (MB), frame-based (FB), batch-based (BB), event-based (EB).

els with fewer parameters; conversely, a larger neighborhood benefits models with more parameters. Even though the motion for all sequences is predominantly dominated by just 1 scaling DOF, the results suggest that considering motion models with additional DOFs, *e.g.*, Translation and 6-DOF, is beneficial. The extra DOFs may explain other small motions, whereas these small motions would just be considered noise for the Scaling model.

In terms of runtime per event, the proposed method outperforms ECMD [28] by between 95.36% and 98.47%, achieving real-time processing for all the sequences in the VL dataset [28], being capable of processing between 420k and 1.28M events per second. The results indicate that the runtime increases with the neighboring size s and the number of parameters of the global motion model.

Tab. 3 reports the results on divergence estimation on the MVSEC benchmark [50]. The Translation model achieves the best performance overall, indicating that the corresponding DOFs are sufficient to explain the perceived motions in the sequences evaluated while minimizing the optimization complexity. As mentioned in Sec. 4.2, the Driving model is tuned to the outdoor_day1 sequence, whose performance is on par with the Translation model. However, it performs poorly on the indoor_flying3 sequence since it can not handle more complex types of motions that are present. Being the most general model, the 6-DOF motion model achieves similar performance for all the sequences. Its performance is worse than the Translation model due to the increased optimization complexity, while the additional DOFs do not contribute to improving the accuracy. The proposed method achieves real-time processing in the MVSEC dataset [50] since, on average, the sequences exhibit a maximum of around 400k events per second.

Optical Flow. Since it is difficult to compare directly the results reported in Tab. 3, we evaluate the proposed method on optical flow estimation, and compare the results with other methods. Tab. 4 reports the results on optical flow estimation on the MVSEC benchmark [50] in terms of AEE and Out, as is commonly found in the literature. The proposed method achieves state-of-the-art performance over the EB methods, on par performance with the FB method, and competitive performance overall.

Method	indoor_flying3 AREE	indoor_flying3 AAE	outdoor_day1 AREE	outdoor_day1 AAE
EV-FlowNet [51]	63.61	32.55	65.54	22.88
Aung *et al*. [3]	95.64	69.35	122.10	69.41
ABMOF [26]	52.69	16.57	90.13	33.76
EDFLOW [27]	**37.52**	12.15	69.40	23.30
Proposed Driving	104.05	60.36	**50.25**	**13.31**
Proposed Translation	44.46	**11.38**	72.67	21.77
Proposed 6-DOF	95.23	21.35	122.04	31.68

Table 5. **Optical flow estimation.** Quantitative results on the MVSEC dataset [50], in terms of AREE (%) and AAE (°). Lower is better.

Tab. 5 reports the results on optical flow estimation on the MVSEC benchmark [50] in terms of AREE and AAE. The proposed method achieves state-of-the-art performance on the outdoor_day1 sequence using the Driving model and on par performance on the indoor_flying3 sequence using the Translation model. The results on Tab. 5 indicate that the proposed method is comparatively more accurate in estimating the direction of the flow, *i.e.*, compared with the other methods, the proposed method achieves lower values of AAE overall. In terms of accuracy, this suggests that more improvements can be achieved by improving the method's per-event inverse depth estimation since inverse depth estimates mainly contribute to the flow magnitude.

Qualitative Results. Fig. 6 presents qualitative results on sequences of the MVSEC dataset [50]. The estimated TTCM and optical flow resemble the GT ones.

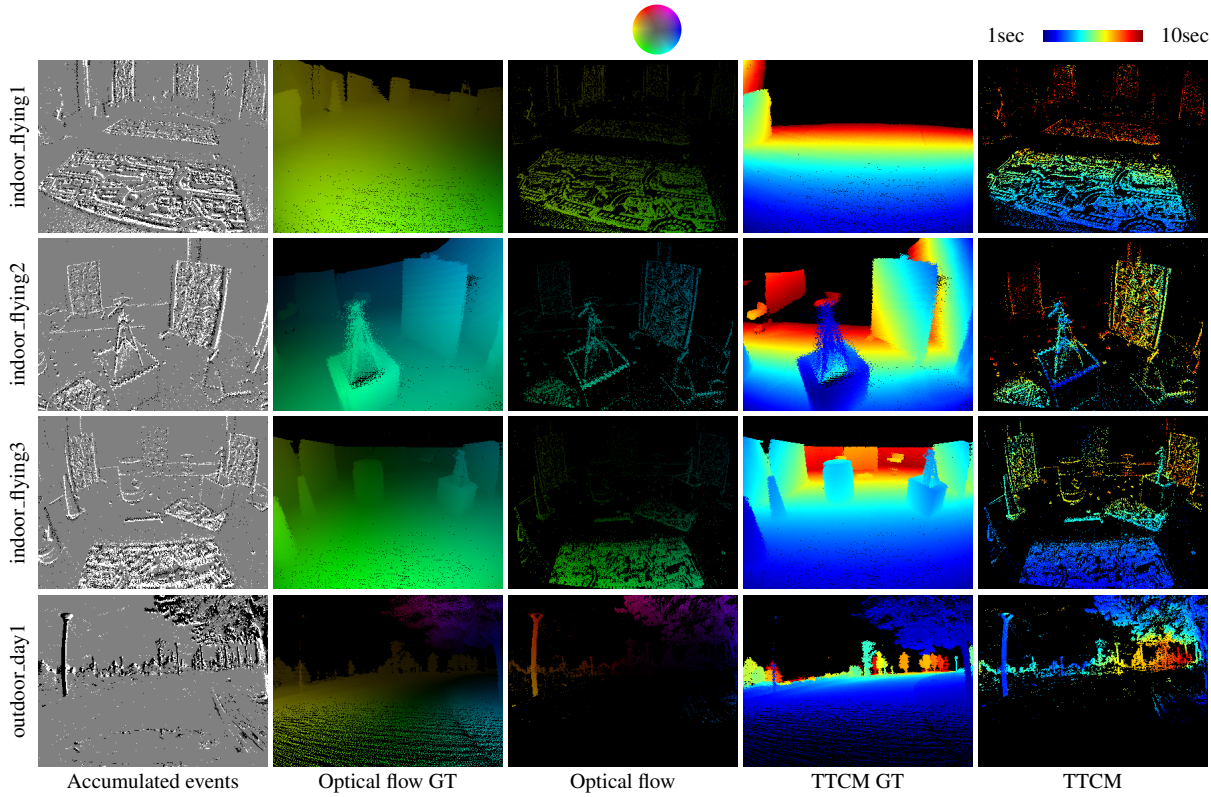

Figure 6. Qualitative results on the MVSEC dataset [50].

4.4. Limitations

Similarly to other event-based methods [15, 22, 35, 52], the proposed method also inherits the brightness assumption from the DMin framework [34]. It can thus provide wrongful estimates for events that are not caused by motion, *e.g.*, due to flickering lights. While this limitation is somewhat mitigated when estimating global quantities, it can struggle to reliably estimate the inverse depth of events that are not caused by motion. Introducing probabilistic uncertainties to the estimates could alleviate this issue while also improving the inverse depth's initialization procedure.

Also related, although we explicitly impose a local smoothness constraint to the inverse depth estimates, given by Eq. (10), the proposed method can still provide inverse depth estimates that differ significantly from the neighboring inverse depth estimates. This typically occurs for events that are generated by noise. Carefully tuning the event camera biases and filtering out outliers could improve the overall inverse depth estimation. Having a back-end inverse depth regularizer [22] could also help to mitigate this issue.

The proposed method can only estimate one global motion. Thus, it can not adequately handle more than one motion simultaneously, *e.g.*, due to cars moving [50]. Considering a multi-scale approach [1, 42] or explicitly modeling more than one possible motion occurring simultaneously [33, 44] are possible avenues for future research.

5. Conclusion

We have proposed a novel method that estimates the TTCM using a single event camera. The proposed method builds on the DMin framework to incrementally estimate local and global quantities, *i.e.*, inverse depth, and global motion, respectively. We have also proposed an approach that effectively prevents event collapse for incremental event-based estimation without introducing regularizers or additional hyper-parameters. The proposed method also achieves state-of-the-art performance in TTC estimation in terms of accuracy and computational runtime while achieving competitive performance in optical flow estimation. Broadly, the proposed work further builds on the increasing amount of evidence that event cameras are especially suited to address visual motion-based problems; in particular, it further shows that incremental event-based processing can provide a flexible and general methodology to consider when using event cameras, which avoids issues introduced when converting events to other representations, *e.g.*, batches and frames.

Acknowledgment. This research received funding from the French National Research Agency (ANR), under grant agreement N° ANR-20-CE23-0021, "AgileNeuRobot".

A. Adapted Dispersion Minimization

In this section, we describe the steps to obtain the optimized model parameters, given by Eq. (8), including the inverse depth parameterization discussed in Sec. 3.6, *i.e.*, $\lambda \in \mathbb{R}$ such that $\rho(\lambda) = e^{\lambda} > 0$. To optimize Eq. (7), we differentiate it w.r.t. the global motion parameters and the parameterized inverse depth $\boldsymbol{\gamma}_i = \left(\boldsymbol{\theta}^{\mathsf{T}}, \lambda_i\right)^{\mathsf{T}}$

$$
\begin{aligned}
\frac{\partial P(\mathcal{E};\boldsymbol{\gamma}_i)}{\partial \boldsymbol{\gamma}_i} &= -\sum_j \frac{\partial \mathcal{N}\left(\mathbf{x}'_j;\mathbf{x}_i,\mathbf{I}\right)}{\partial \boldsymbol{\gamma}_i} \\
&\propto -\sum_j \frac{\partial \exp\left[-\frac{1}{2}\left(\mathbf{x}'_j-\mathbf{x}_i\right)^{\mathsf{T}}\left(\mathbf{x}'_j-\mathbf{x}_i\right)\right]}{\partial \boldsymbol{\gamma}_i} \\
&= -\sum_j \frac{\partial \exp\left[-\frac{1}{2}\mathbf{r}_{i,j}^{\mathsf{T}}\mathbf{r}_{i,j}\right]}{\partial \boldsymbol{\gamma}_i} \\
&\uparrow \\
&\mathbf{r}_{i,j} = \mathbf{x}_i - \mathbf{x}'_j \\
&= \sum_j w_{i,j}\frac{\partial \mathbf{r}_{i,j}^{\mathsf{T}}}{\partial \boldsymbol{\gamma}_i}\mathbf{r}_{i,j} = \sum_j w_{i,j}\mathbf{J}_{i,j}^{\mathsf{T}}\mathbf{r}_{i,j}. \quad (16)
\end{aligned}
$$

By linearizing the residual according to Taylor's formula $\mathbf{r}_{i,j}(\boldsymbol{\gamma}_i + \Delta\boldsymbol{\gamma}_i) \approx \mathbf{r}_{i,j}(\boldsymbol{\gamma}_i) + \mathbf{J}_{i,j}\Delta\boldsymbol{\gamma}_i$ and setting Eq. (16) to 0, we obtain

$$
\begin{aligned}
&\sum_j w_{i,j}\mathbf{J}_{i,j}^{\mathsf{T}}\mathbf{r}_{i,j} \approx \sum_j w_{i,j}\mathbf{J}_{i,j}^{\mathsf{T}}\left(\mathbf{r}_{i,j} + \mathbf{J}_{i,j}\Delta\boldsymbol{\gamma}_i\right) = 0 \\
&\Rightarrow \underbrace{\sum_j w_{i,j}\mathbf{J}_{i,j}^{\mathsf{T}}\mathbf{r}_{i,j}}_{\boldsymbol{\psi}} + \underbrace{\sum_j w_{i,j}\mathbf{J}_{i,j}^{\mathsf{T}}\mathbf{J}_{i,j}}_{\boldsymbol{\Psi}}\Delta\boldsymbol{\gamma}_i = 0 \\
&\Rightarrow \Delta\boldsymbol{\gamma}_i = \boldsymbol{\Psi}^{-1}\boldsymbol{\psi}, \quad (17)
\end{aligned}
$$

thus obtaining the (parameterized) update given by Eq. (8). Lastly, the derivative of the residual w.r.t. the (parameterized) model parameters is given by

$$
\mathbf{J}_{i,j} = -\Delta t_{i,j}\begin{pmatrix}\mathbf{B}_{i,j} & \rho_i \mathbf{V}_j \boldsymbol{\nu}\end{pmatrix}, \quad (18)
$$

where $\mathbf{B}_{i,j}$ and $\mathbf{V}_j$ are given by Eq. (9).

B. Additional Results

We provide additional results regarding the robustness of the proposed method for camera resolution resizing and event sampling. We adopt a simple strategy that resembles an integrate-and-fire model, which depends on a single parameter r that controls both the camera resolution reduction and the threshold that effectively fires an event to be processed. This is a simple strategy that improves the method's runtime by essentially working as an event filter, which can be useful when using cameras with a large resolution and/or in scenarios with limited computational power, *e.g.*, embedded systems.

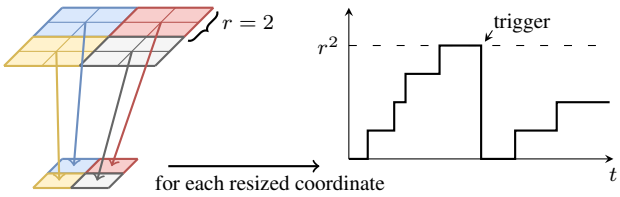

Figure 7. Speed-up strategy resembling an integrate-and-fire model for $r = 2$.

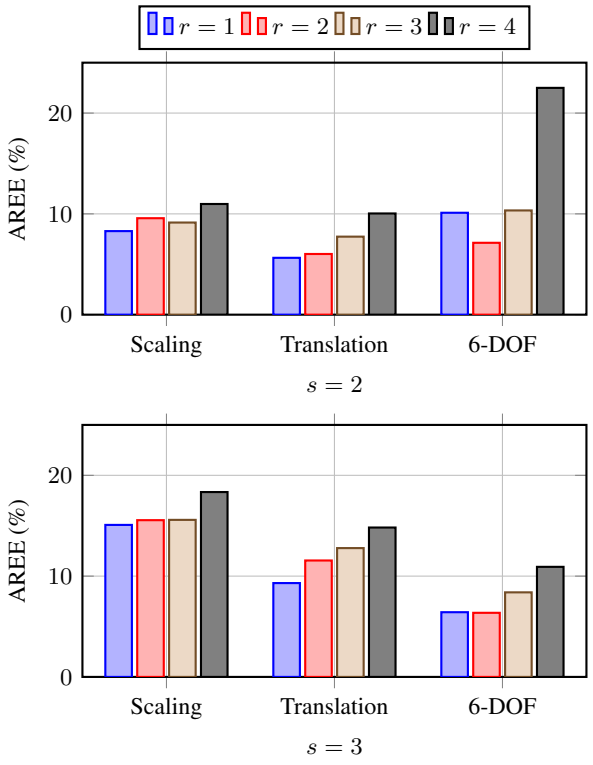

Figure 8. Average global divergence estimation accuracy on the VL dataset [28] in function of the reduction multiplier r.

Fig. 7 illustrates the strategy's main idea. For each event e_i we divide its image coordinates by r, *i.e.*, $\bar{\mathbf{x}}_i = \mathbf{x}/r$, and increment by 1 an integrator variable corresponding to the resized coordinates $\bar{\mathbf{x}}_i$. Once the integrator variable crosses the firing threshold r^2, the corresponding event is processed. This strategy ensures that the original spatial event distribution is largely preserved when the events' coordinates are resized while the number of events processed is reduced by a factor of r^2, which effectively reduces the method's actual runtime by $\approx r^2$. The actual runtime improvement is achieved by skipping and not processing certain events since the runtime per event remains approximately the same. Also, $r = 1$ corresponds to considering the full original resolution.

Fig. 8 plots the average global divergence estimation accuracy on the VL dataset [28] in function of the resolution reduction multiplier r. In absolute terms, on average,

		Method	2D-1	2D-2	2D-3	2D-4	2D-5	2D-6	2D-7	3D	Avg.
		ECMD [28]	13.48	7.41	6.11	12.19	10.41	5.12	3.72	12.90	8.92
$r = 1$	$s = 2$	Proposed Scaling	7.19	8.79	9.13	12.18	4.98	7.79	4.92	11.34	8.29
		Proposed Translation	5.03	7.02	6.17	5.25	4.61	4.34	6.18	6.54	5.64
		Proposed 6-DOF	6.77	13.73	14.97	7.50	6.91	12.15	12.77	6.10	10.11
	$s = 3$	Proposed Scaling	14.99	15.19	11.69	21.69	11.19	15.44	11.01	19.43	15.08
		Proposed Translation	11.15	7.01	7.98	13.80	6.37	11.17	4.55	12.46	9.31
		Proposed 6-DOF	6.51	6.81	7.25	6.77	6.06	5.59	6.18	6.19	6.42
$r = 2$	$s = 2$	Proposed Scaling	10.66	8.61	6.81	13.22	6.94	9.53	6.94	13.82	9.57
		Proposed Translation	7.10	5.04	4.61	9.55	3.61	7.36	2.82	8.08	6.02
		Proposed 6-DOF	3.77	9.93	10.28	7.66	6.61	6.10	7.57	5.10	7.13
	$s = 3$	Proposed Scaling	17.50	14.30	12.91	19.13	12.92	14.77	12.31	20.53	15.55
		Proposed Translation	13.33	10.22	11.55	16.33	6.97	12.96	6.30	14.77	11.55
		Proposed 6-DOF	8.22	4.84	4.22	8.83	6.38	5.95	3.45	8.97	6.36
$r = 3$	$s = 2$	Proposed Scaling	9.74	8.74	5.43	10.45	8.44	8.37	7.12	14.79	9.14
		Proposed Translation	8.62	7.13	6.00	10.13	7.06	8.62	4.06	10.28	7.74
		Proposed 6-DOF	7.85	12.29	10.50	9.76	17.93	9.01	9.62	5.77	10.34
	$s = 3$	Proposed Scaling	17.08	14.40	12.27	17.74	14.38	15.22	12.47	21.06	15.58
		Proposed Translation	13.03	12.23	12.33	17.12	8.54	14.48	7.90	16.61	12.78
		Proposed 6-DOF	7.80	7.04	6.24	10.21	11.13	8.15	5.06	11.45	8.39
$r = 4$	$s = 2$	Proposed Scaling	11.12	10.89	8.08	12.56	9.85	9.42	8.13	17.77	10.98
		Proposed Translation	9.46	10.11	8.48	10.75	11.63	7.60	7.36	14.91	10.04
		Proposed 6-DOF	23.90	18.54	14.22	23.00	38.03	16.50	35.58	10.26	22.50
	$s = 3$	Proposed Scaling	20.24	17.50	14.89	21.50	16.98	17.16	14.01	24.42	18.34
		Proposed Translation	15.90	13.95	14.48	18.01	10.18	14.71	10.12	21.20	14.82
		Proposed 6-DOF	12.36	9.56	8.34	11.89	13.53	8.36	8.74	14.54	10.92

Table 6. **Global divergence estimation.** Quantitative results on the VL dataset [28] in function of the resolution reduction multiplier r, in terms of REE (%). Lower is better.

the performance worsens with the increase of the reduction multiplier r since fewer events are processed and thus less detail is considered. However, the drop in performance only becomes noticeable for $r = 4$. These results suggest that the proposed method can achieve at least a $9\times$ speed-up in actual runtime without a significant drop in accuracy, thus demonstrating its robustness. Tab. 6 presents a detailed breakdown for all the sequences on the VL dataset [28].

References

[1] Himanshu Akolkar, Sio Hoi Ieng, and Ryad Benosman. Real-time High Speed Motion Prediction Using Fast Aperture-Robust Event-Driven Visual Flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2020. 2, 7, 8

[2] Mohammed Almatrafi, Raymond Baldwin, Kiyoharu Aizawa, and Keigo Hirakawa. Distance Surface for Event-Based Optical Flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(7):1547–1556, 2020. 2

[3] Myo Tun Aung, Rodney Teo, and Garrick Orchard. Event-based Plane-fitting Optical Flow for Dynamic Vision Sensors in FPGA. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, 2018. 2, 7

[4] Ryad Benosman, Charles Clercq, Xavier Lagorce, Sio-Hoi Ieng, and Chiara Bartolozzi. Event-Based Visual Flow. *IEEE Transactions on Neural Networks and Learning Systems*, 25(2):407–417, 2014. 1, 2

[5] Ryad Benosman, Sio-Hoi Ieng, Charles Clercq, Chiara Bartolozzi, and Mandyam Srinivasan. Asynchronous Frameless Event-based Optical Flow. *Neural Networks*, 27:32–37, 2012. 2

[6] Christian Brandli, Raphael Berner, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck. A 240 × 180 130 dB 3 µs Latency Global Shutter Spatiotemporal Vision Sensor. *IEEE Journal of Solid-State Circuits*, 49(10):2333–2341, 2014. 1, 6

[7] Vincent Brebion, Julien Moreau, and Franck Davoine. Real-Time Optical Flow for Vehicular Perception With Low- and High-Resolution Event Cameras. *IEEE Transactions on Intelligent Transportation Systems*, 23(9):15066–15078, 2022. 2, 7

[8] William Chamorro, Juan Andrade-Cetto, and Joan Solà. High-Speed Event-Based Camera Tracking. In *British Machine Vision Conference (BMVC)*, 2020. 2

[9] William Chamorro, Joan Solà, and Juan Andrade-Cetto. Event-Based Line SLAM in Real-Time. *IEEE Robotics and Automation Letters*, 7(3):8146–8153, July 2022. 2

[10] Xavier Clady, Charles Clercq, Sio-Hoi Ieng, Fouzhan Houseini, Marco Randazzo, Lorenzo Natale, Chiara Bartolozzi, and Ryad Benosman. Asynchronous Visual Event-based Time-to-Contact. *Frontiers in Neuroscience*, 8, 2014. 1, 2

[11] Ziluo Ding, Rui Zhao, Jiyuan Zhang, Tianxiao Gao, Ruiqin Xiong, Zhaofei Yu, and Tiejun Huang. Spatio-Temporal Recurrent Networks for Event-Based Optical Flow Estimation. In *AAAI Conference on Artificial Intelligence*, volume 36, pages 525–533, 2022. 3, 7

[12] Davide Falanga, Kevin Kleber, and Davide Scaramuzza. Dynamic obstacle avoidance for quadrotors with event cameras. *Science Robotics*, 5(40):eaaz9712, 2020. 1, 2

[13] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J. Davison, Jörg Conradt, Kostas Daniilidis, and Davide Scaramuzza. Event-Based Vision: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):154–180, 2022. 1, 2

[14] Guillermo Gallego, Jon E.A. Lund, Elias Mueggler, Henri Rebecq, Tobi Delbruck, and Davide Scaramuzza. Event-Based, 6-DOF Camera Tracking from Photometric Depth Maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(10):2402–2412, 2018. 2

[15] Guillermo Gallego, Henri Rebecq, and Davide Scaramuzza. A Unifying Contrast Maximization Framework for Event Cameras, with Applications to Motion, Depth, and Optical Flow Estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3867–3876, 2018. 1, 2, 3, 8

[16] Daniel Gehrig, Antonio Loquercio, Konstantinos Derpanis, and Davide Scaramuzza. End-to-End Learning of Representations for Asynchronous Event-Based Data. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5632–5642, 2019. 2

[17] Mathias Gehrig, Mario Millhäusler, Daniel Gehrig, and Davide Scaramuzza. E-RAFT: Dense Optical Flow from Event Cameras. In *International Conference of 3D Vision (3DV)*, pages 197–206, 2021. 1

[18] Germain Haessig, Xavier Berthelon, Sio-Hoi Ieng, and Ryad Benosman. A Spiking Neural Network Model of Depth from Defocus for Event-based Neuromorphic Vision. *Scientific Reports*, 9(1):3744, 2019. 2

[19] Jesse Hagenaars, Federico Paredes-Valles, and Guido de Croon. Self-Supervised Learning of Event-Based Optical Flow with Spiking Neural Networks. In *Advances in Neural Information Processing Systems (NIPS)*, volume 34, pages 7167–7179, 2021. 3, 7

[20] Botao He, Haojia Li, Siyuan Wu, Dong Wang, Zhiwei Zhang, Qianli Dong, Chao Xu, and Fei Gao. FAST-Dynamic-Vision: Detection and Tracking Dynamic Objects with Event and Depth Sensing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3071–3078, 2021. 1

[21] Hanme Kim, Ankur Handa, Ryad Benosman, Sio-Hoi Ieng, and Andrew Davison. Simultaneous Mosaicing and Tracking with an Event Camera. In *British Machine Vision Conference (BMVC)*. British Machine Vision Association (BMVA), 2014. 1, 2

[22] Hanme Kim, Stefan Leutenegger, and Andrew J. Davison. Real-Time 3D Reconstruction and 6-DoF Tracking with an Event Camera. In *Computer Vision – ECCV*, Lecture Notes in Computer Science, pages 349–364. Springer International Publishing, 2016. 1, 2, 8

[23] Xavier Lagorce, Garrick Orchard, Francesco Galluppi, Bertram E. Shi, and Ryad B. Benosman. HOTS: A Hierarchy of Event-Based Time-Surfaces for Pattern Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(7):1346–1359, July 2017. 1

[24] Chankyu Lee, Adarsh Kumar Kosta, Alex Zihao Zhu, Kenneth Chaney, Kostas Daniilidis, and Kaushik Roy. Spike-FlowNet: Event-Based Optical Flow Estimation with Energy-Efficient Hybrid Neural Networks. In *Computer Vision – ECCV*, Lecture Notes in Computer Science, pages 366–382. Springer International Publishing, 2020. 3

[25] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128 × 128 120 dB 15 µs Latency Asynchronous Temporal Contrast Vision Sensor. *IEEE Journal of Solid-State Circuits*, 43(2):566–576, 2008. 1

[26] Min Liu and T. Delbruck. Adaptive Time-Slice Block-Matching Optical Flow Algorithm for Dynamic Vision Sensors. In *British Machine Vision Conference (BMVC)*, 2018. 2, 7

[27] Min Liu and Tobi Delbruck. EDFLOW: Event Driven Optical Flow Camera With Keypoint Detection and Adaptive Block Matching. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(9):5776–5789, 2022. 1, 2, 3, 6, 7

[28] Sofia McLeod, Gabriele Meoni, Dario Izzo, Anne Mergy, Daqi Liu, Yasir Latif, Ian Reid, and Tat-Jun Chin. Globally Optimal Event-Based Divergence Estimation for Ventral Landing, 2022. 1, 2, 5, 6, 7, 9, 10

[29] Moritz B. Milde, Olivier J.N. Bertrand, Ryad Benosmanz, Martin Egelhaaf, and Elisabetta Chicca. Bioinspired event-driven collision avoidance algorithm based on optic flow. In *International Conference on Event-based Control, Communication, and Signal Processing (EBCCSP)*, pages 1–7, 2015. 1, 2

[30] Elias Mueggler, Henri Rebecq, Guillermo Gallego, Tobi Delbruck, and Davide Scaramuzza. The Event-Camera Dataset and Simulator: Event-based Data for Pose Estimation, Visual Odometry, and SLAM. *The International Journal of Robotics Research*, 36(2):142–149, 2017. 4

[31] Urbano Miguel Nunes, Ryad Benosman, and Sio-Hoi Ieng. Adaptive Global Decay Process for Event Cameras. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9771–9780, 2023. 5

[32] Urbano Miguel Nunes and Yiannis Demiris. Entropy Minimisation Framework for Event-based Vision Model Estimation. In *Computer Vision – ECCV*, Lecture Notes in Computer Science, pages 161–176. Springer International Publishing, 2020. 3

[33] Urbano Miguel Nunes and Yiannis Demiris. Kinematic Structure Estimation of Arbitrary Articulated Rigid Objects for Event Cameras. In *International Conference on Robotics and Automation (ICRA)*, pages 508–514, 2022. 1, 8

[34] Urbano Miguel Nunes and Yiannis Demiris. Robust Event-Based Vision Model Estimation by Dispersion Minimisation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):9561–9573, 2022. 1, 2, 3, 8

[35] Federico Paredes-Vallés and Guido C. H. E. de Croon. Back to Event Basics: Self-Supervised Learning of Image Reconstruction for Event Cameras via Photometric Constancy. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3445–3454, 2021. 3, 8

[36] Bas J. Pijnacker Hordijk, Kirk Y. W. Scheper, and Guido C. H. E. de Croon. Vertical Landing for Micro Air Vehicles using Event-based Optical Flow. *Journal of Field Robotics*, 35(1):69–90, 2018. 1, 2

[37] Christoph Posch, Teresa Serrano-Gotarredona, Bernabe Linares-Barranco, and Tobi Delbruck. Retinomorphic Event-Based Vision Sensors: Bioinspired Cameras With Spiking Output. *Proceedings of the IEEE*, 102(10):1470–1484, 2014. 2

[38] Henri Rebecq, Timo Horstschaefer, Guillermo Gallego, and Davide Scaramuzza. EVO: A Geometric Approach to Event-Based 6-DOF Parallel Tracking and Mapping in Real Time. *IEEE Robotics and Automation Letters*, 2(2):593–600, 2017. 2

[39] Juan Pablo Rodríguez-Gómez, Raul Tapia, Maria del Mar Guzmán Garcia, Jose Ramiro Martínez-de Dios, and Anibal Ollero. Free as a Bird: Event-Based Dynamic Sense-and-Avoid for Ornithopter Robot Flight. *IEEE Robotics and Automation Letters*, 7(2):5413–5420, 2022. 1, 2

[40] Shintaro Shiba, Yoshimitsu Aoki, and Guillermo Gallego. Event Collapse in Contrast Maximization Frameworks. *Sensors*, 22(14):5190, 2022. 1, 2, 3

[41] Shintaro Shiba, Yoshimitsu Aoki, and Guillermo Gallego. A Fast Geometric Regularizer to Mitigate Event Collapse in the Contrast Maximization Framework. *Advanced Intelligent Systems*, 2022. 1, 2, 3

[42] Shintaro Shiba, Yoshimitsu Aoki, and Guillermo Gallego. Secrets of Event-Based Optical Flow. In *Computer Vision – ECCV*, Lecture Notes in Computer Science, pages 628–645. Springer Nature Switzerland, 2022. 2, 6, 7, 8

[43] Olaf Sikorski, Dario Izzo, and Gabriele Meoni. Event-based Spacecraft Landing using Time-to-Contact. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1941–1950, 2021. ISSN: 2160-7516. 1, 2

[44] Timo Stoffregen, Guillermo Gallego, Tom Drummond, Lindsay Kleeman, and Davide Scaramuzza. Event-Based Motion Segmentation by Motion Compensation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7243–7252, 2019. 1, 8

[45] Timo Stoffregen and Lindsay Kleeman. Simultaneous Optical Flow and Segmentation (SOFAS) using Dynamic Vision Sensor. In *Australasian Conference on Robotics and Automation (ACRA)*, pages 52–61. Australian Robotics and Automation Association (ARAA), 2017. 1

[46] Timo Stoffregen, Cedric Scheerlinck, Davide Scaramuzza, Tom Drummond, Nick Barnes, Lindsay Kleeman, and Robert Mahony. Reducing the Sim-to-Real Gap for Event Cameras. In *Computer Vision – ECCV*, Lecture Notes in Computer Science, pages 534–549. Springer International Publishing, 2020. 3, 7

[47] Celyn Walters and Simon Hadfield. EVReflex: Dense Time-to-Impact Prediction for Event-based Obstacle Avoidance. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1304–1309, 2021. 1, 2

[48] Chengxi Ye, Anton Mitrokhin, Cornelia Fermüller, James A. Yorke, and Yiannis Aloimonos. Unsupervised Learning of Dense Optical Flow, Depth and Egomotion with Event-Based Sensors. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5831–5838, 2020. 3

[49] Alex Zihao Zhu, Nikolay Atanasov, and Kostas Daniilidis. Event-based Feature Tracking with Probabilistic Data Association. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4465–4470, 2017. 2

[50] Alex Zihao Zhu, Dinesh Thakur, Tolga Özaslan, Bernd Pfrommer, Vijay Kumar, and Kostas Daniilidis. The Multivehicle Stereo Event Camera Dataset: An Event Camera Dataset for 3D Perception. *IEEE Robotics and Automation Letters*, 3(3):2032–2039, 2018. 5, 6, 7, 8

[51] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. EV-FlowNet: Self-Supervised Optical Flow Estimation for Event-based Cameras. In *Proceedings of Robotics: Science and Systems*, 2018. 1, 6, 7

[52] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised Event-Based Learning of Optical Flow, Depth, and Egomotion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 989–997, 2019. 2, 3, 7, 8