Adaptive Global Decay Process for Event Cameras

Urbano Miguel Nunes, Ryad Benosman and Sio-Hoi Ieng
Sorbonne University, 4 place jussieu, 75005 Paris
urbano.goncalves-nunes@inserm.fr, {ryad.benosman, sio-hoi.ieng}@upmc.fr

Abstract

In virtually all event-based vision problems, there is the need to select the most recent events, which are assumed to carry the most relevant information content. To achieve this, at least one of three main strategies is applied, namely: 1) constant temporal decay or fixed time window, 2) constant number of events, and 3) flow-based lifetime of events. However, these strategies suffer from at least one major limitation each. We instead propose a novel decay process for event cameras that adapts to the global scene dynamics and whose latency is in the order of nanoseconds. The main idea is to construct an adaptive quantity that encodes the global scene dynamics, denoted by event activity. The proposed method is evaluated in several event-based vision problems and datasets, consistently improving the corresponding baseline methods' performance. We thus believe it can have a significant widespread impact on event-based research. Code available: https://github.com/neuromorphic-paris/event_batch.

1. Introduction

Contrary to standard frame-based cameras that capture visual data at a fixed rate and independently of the scene dynamics [31], event cameras respond asynchronously to pixel-wise brightness changes by generating *events*. Event cameras are thus data-driven sensors that inherently react to the scene dynamics [18] while providing additional advantages: high temporal resolution in the order of microseconds, low latency, low power consumption, and high dynamic range. However, due to the different visual sensing paradigm, event cameras pose the challenge of developing new methods that fully unlock their capabilities [7]. By "scene dynamics", we mean the spatio-temporal changes of the visual inputs due to the relative motion of the scene w.r.t. the camera. As in mechanics, the involved quantities are the zero, first, or higher-order derivatives of captured temporal information.

There are two paradigms for event processing, namely: *event-based* [16, 17, 26, 27, 38], whereby each event is processed one-by-one, *e.g.*, by locally updating some state, and *batch-based* [9, 30, 44, 45], whereby events are grouped into a batch for later processing. In both paradigms, there is the intuitive notion that more recent events should be assigned more relevancy. Thus, each event should have a certain lifespan, based on which each one is considered *active* and whose contribution is considered meaningful to the computation. This raises the fundamental question on *what kind of decay strategy should be employed to quantify each event's lifespan*. Currently, there exist three main decay strategies. The first is to consider the events generated within a fixed time window or by applying some constant temporal decay [17, 35]. This strategy does not preserve the data-driven paradigm of event cameras since it imposes an arbitrary finite temporal support in practice, which generally has no relation to the visual information source and does not ac-

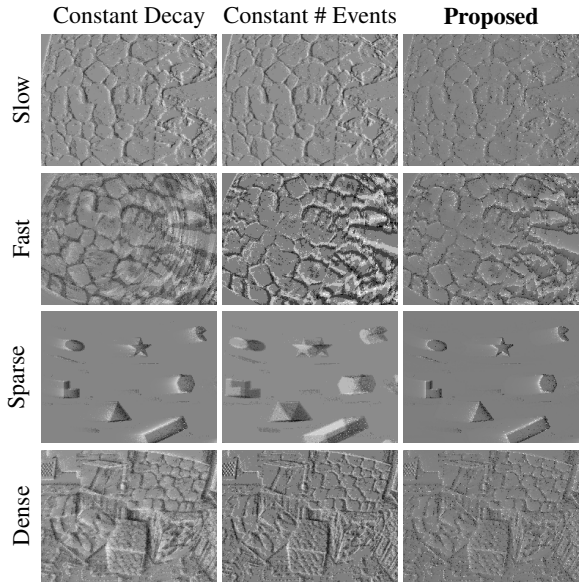


Figure 1. Accumulated events using (left) constant temporal decay, (middle) constant number of events, and (right) the proposed adaptive global decay process. In contrast to the constant processes, the proposed decay process globally adapts to the scene dynamics. Sequences from dataset [24].

count for variations in motion magnitude. A single constant temporal decay can not handle slow and fast motions simultaneously since, *e.g.*, as illustrated in the first two images on the left column in Fig. 1, when the motion is fast, the accumulated events generate a blurry frame. The second strategy considers instead a constant number of events which preserves better the data-driven nature of event cameras as argued in [9]. However, it breaks for variations in the scene texture, whereby, *e.g.*, as illustrated in the last two images on the middle column in Fig. 1, when the texture is sparse, the accumulated events also generate a blurry frame. The third strategy explicitly models the set of active events by estimating each event's lifetime from the corresponding velocity [19, 23]. Although this strategy does not require any explicit decay parameter, it still requires tuning parameters for the flow estimation, which, in turn, may introduce non-negligible latency [23].

We address the fundamental question posed by proposing a novel decay process that inherently *adapts* to the global scene dynamics while introducing negligible latency in the order of nanoseconds, *i.e.*, three orders of magnitude lower than the temporal resolution of event cameras. The main idea is to construct a decay process that depends on a global adaptive quantity that encodes the scene dynamics, which we denote by *event activity*. As shown on the right column in Fig. 1, accumulating events using the proposed decay process always generates sharp edge-like frames, regardless of the event stream dynamics.

2. Event Camera Working Principle

The output of an event camera consists of an asynchronous stream of 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

$$e_i \coloneqq (x_i, y_i, t_i, p_i)\,, \tag{1}$$

where (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 the polarity of the event. An event e_i is generated when the change in log-brightness $\log \mathbf{I}_{x,y}(t) \coloneqq \bar{\mathbf{I}}_{x,y}(t)$ is above a threshold L

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

where Δt_i is the time since the last event at the same pixel. The change in log-brightness can be rewritten in terms of continuous changes in a time interval

$$\Delta\bar{\mathbf{I}}_{x_i,y_i}(t_i) = \int_{t_i-\Delta t_i}^{t_i} \frac{d\bar{\mathbf{I}}_{x_i,y_i}(t)}{dt} dt, \tag{3}$$

from which the brightness signal sensing by a camera can be approximated by a multiplicative process of the global illumination $\bar{\mathbf{L}}$, by the scene reflectance $\bar{\mathbf{R}}$ [28]

$$\Delta\bar{\mathbf{I}}_{x_i,y_i}(t_i) = \int_{t_i-\Delta t_i}^{t_i} \left(\frac{d\bar{\mathbf{L}}(t)}{dt} + \frac{\partial\bar{\mathbf{R}}(t)}{\partial t} + \frac{\partial\bar{\mathbf{R}}(t)}{\partial x}\frac{dx}{dt} + \frac{\partial\bar{\mathbf{R}}(t)}{\partial y}\frac{dy}{dt} \right) dt, \tag{4}$$

where we have omitted the pixel location (x_i, y_i) dependency for readability. The term $d\bar{\mathbf{L}}(t)/dt$ represents the log-illumination change, the term $\partial\bar{\mathbf{R}}(t)/\partial t$ is the log-reflectance temporal change, the terms $\left(\partial\bar{\mathbf{R}}(t)/\partial x, \partial\bar{\mathbf{R}}(t)/\partial y\right)$ are the spatial contrast or the scene texture, and the terms $(dx/dt, dy/dt)$ correspond to the optical flow. The contribution from all these terms is denoted as the *dynamics* of the event stream. We further detail the terminology by noting that only the rate of change of the spatial contrast contribution is w.r.t. space, whereas the rate of change of the remaining contributions is w.r.t. time. We denote the former as the *spatial dynamics* and the latter as the *temporal dynamics*. We also denote the contribution from optical flow as the *motion dynamics*.

Eq. (4) makes explicit what an event encodes. However, each contribution is not easily separable from the event stream alone without further assumptions. In this work, we show that it is still feasible to construct a global decay process that implicitly encodes the event stream dynamics.

3. Related Work

In the literature, besides a few methods [15, 16] that rely on uncertainty priors, there are three main temporal decay strategies applied to process event-based data, which we briefly review next. Tab. 1 highlights the key differences between the decay strategies.

Strategy	Adaptive Spatial	Adaptive Temporal	G/L/H	Latency	# Params
Constant Decay	✓	✗	G	Low	1
Constant # Events	✗	✓	G	Low	1
Event Lifetime [19, 23]	✓	✓	L/H	High	> 1
Proposed	✓	✓	G	Low	≤ 1

Table 1. Comparison of event-based decay processes. G - Global, L - Local, H - Hybrid.

Constant temporal decay & fixed time window. Features extracted using Hierarchy of Time-Surfaces (HOTS) [17] and subsequent works, *e.g.*, Histogram of Averaged Time-Surfaces (HATS) [39], are by definition constructed from an exponential kernel with a fixed time constant. The event-based spatial convolution operator [35] also applies a constant temporal decay, which was derived from designing a high-pass filter to attenuate noise contributions. Fixed time windows have been applied to asynchronous recursive event processing [36], event-based steering prediction [21],

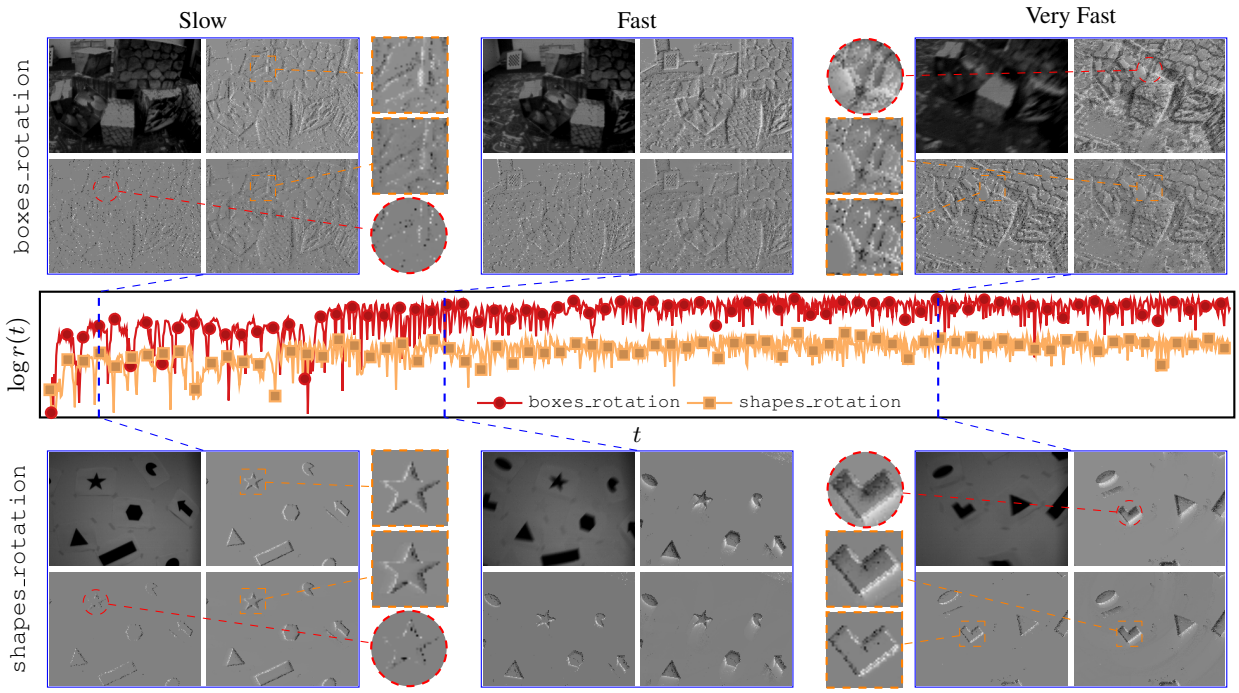


Figure 2. **Comparison of applying constant *vs*. adaptive global decay.** (Middle plot) Event rate $r(t)$ (in logarithmic scale) of two sequences [24] over time, which is used as a proxy for the event stream dynamics. (Top and bottom blocks) Snapshots of the current state of the `boxes_rotation` and `shapes_rotation` sequences. Each block of snapshots consists of (top left) the gray-scale frame, the result of applying the event-based convolution operator [35] using (top right) a large time constant, (bottom left) a small time constant, and (bottom right) an adaptive decay process. (Left) Slow motion, (middle) fast motion, (right) very fast motion. While both sequences exhibit similar motion dynamics, the `boxes_rotation` exhibits higher spatial dynamics (*i.e.*, event density), which explains the higher event rate. The accuracy of the constant decay process significantly depends on the sequence's event rate. (Zoom plots) Smaller time constants can reliably capture higher event rates but can not lower ones; conversely, larger time constants can accurately capture lower event rates but can not higher ones. In contrast, the accuracy of an adaptive decay process should not significantly depend on the event stream dynamics.

optical-flow estimation [1, 3] and feature tracking [2]. Besides depending on the sequence temporal dynamics, fixing the time constant also requires non-trivial tuning.

Constant number of events. This is the predominant decay strategy in the literature, and can be found in several applications, such as global motion estimation [8, 9, 13, 25, 27], feature tracking [10, 43], depth estimation [32, 44], optical-flow estimation [19, 45], and image intensity reconstruction [30, 33]. Besides depending on the sequence spatial dynamics, fixing the number of events also requires prior knowledge of the event stream. Adjusting the number of events based on the sequence spatial dynamics, *i.e.*, scene texture, was proposed in [10]. However, the method requires standard frames, whose availability and suitability are not always guaranteed. An ensemble method that employs both fixed time window and constant number of events strategies was proposed in the context of visual place recognition [6]. Ensembles generally increase the accuracy since each component's errors are averaged out, while increasing computational costs, and thus a trade-off must be reached regarding the ensemble size. The proposed method adapts to the event stream dynamics, which can be seen as dynamically selecting the most suitable component(s) without incurring the cost of running the entire ensemble.

Flow-based lifetime of events. The lifetime of an event quantifies how long an event is considered active [23], being defined as the duration until a new event is generated in the event's neighborhood. To compute each event's lifetime, the corresponding velocity, *i.e.*, optical-flow, must be estimated, which is achieved by robust plane-fitting [3, 20, 23] or by converting events into frames and applying the classic Lucas-Kanade estimation method [19]. By design, these methods do not rely on explicit decay parameters. However, they require additional thresholds for inlier/outlier separation, whose dependency on the event stream dynamics is not clear and can thus be cumbersome to properly set. Also, the latency introduced may be non-negligible [23], which further hinders their widespread adoption. Although the proposed method models the decay process globally, it has applicability to several event-based vision problems, consistently improving the corresponding baseline methods' performance while introducing negligible latency.

4. Method

In this section, we describe how an adaptive global decay process from an event stream can be built without having to explicitly model each contribution in Eq. (4). We first review the constant decay process, based on which the adaptive decay process is constructed. Refer to the supplementary material for the full mathematical derivations.

4.1. Constant Decay

The constant decay strategy consists in applying an exponential decay kernel with time constant τ

$$\beta_\tau(t, t_p) = e^{-\frac{t-t_p}{\tau}}, \tag{5}$$

where t_p is some previous timestamp. This decay strategy applies a decrease at a rate proportional to its current value which can be clearly seen by differentiating Eq. (5) w.r.t. t:

$$\frac{d\beta_\tau(t, t_p)}{dt} = -\frac{1}{\tau} e^{-\frac{t-t_p}{\tau}} = -\lambda \beta_\tau(t, t_p). \tag{6}$$

The proportionality constant $\lambda = 1/\tau$, *i.e.*, *decay rate*, directly influences the behavior of the decay process and must be empirically set for each sequence since the event stream dynamics depend on different contributions according to Eq. (4). This issue is depicted in Fig. 2, where we show the result of applying asynchronous convolution operations [35] using large and small time constants. Larger time constants can reliably capture lower event rates, whereas smaller time constants can accurately capture higher event rates. Based on Fig. 2, we thus intuitively assert that the decay rate λ should scale with the dynamics of the event stream for an adequate adaptive decay process.

4.2. Event Activity

Defining the number of events as a counting process

$$n(t) \coloneqq \sum_{i=0}^{\infty} J(t_i \leq t), \tag{7}$$

where $J(C)$ is the indicator function, being 1 if the condition C is true and 0 otherwise, and t_i is the timestamp of the i-th event in the event stream, event activity is defined as the number of events in a given time interval $(s, t]$

$$a(t, s) \coloneqq n(t) - n(s) = \sum_{i=0}^{\infty} J(s < t_i \leq t). \tag{8}$$

This definition resembles applying a variable temporal sliding window strategy to estimate the event activity, whose temporal length is given by $t - s$. By abusing notation, we can consider a more useful event activity definition in practice which resembles applying a temporal decay strategy

$$a(t) \coloneqq \beta(t, s)a(s) + n(t) - n(s), \quad a(0) = a_0, \tag{9}$$

where $\beta(t, s)$ is some temporal decay function and a_0 is an arbitrary constant value, typically 0. Eq. (9) can be used to incrementally estimate the event activity for each generated event e_i according to

$$a_i = \beta_i a_{i-1} + 1, \tag{10}$$

where $a_i \coloneqq a(t_i)$ is the event activity and $\beta_i \coloneqq \beta(t_i, t_{i-1})$ is the decay value, both at the timestamp of event e_i. Assuming the temporal decay function is appropriate, the event activity also encodes the event stream dynamics since it considers *all* events, whose generation process is described by Eq. (4). For example, scenes with more texture will have higher event activity than scenes with less texture, given the remaining dynamics are the same.

Event rate. The event rate is the number of events generated per unit of time, being defined in terms of the event activity according to $r(t, s) \coloneqq a(t, s)/(t - s)$.

4.3. Adaptive Decay

Instead of using a constant decay rate λ, we propose to construct a decay rate that varies in time $\lambda(t)$ according to the event stream dynamics

$$\frac{d\beta_a(t, t_p)}{dt} = -\lambda(t)\beta_a(t, t_p). \tag{11}$$

Note that the choice of $\lambda(t)$ influences the expression of $\beta_a(t, t_p)$, which in general is no longer of the same exponential form of Eq. (5). We consider an adaptive decay rate that is proportional to the event activity $\lambda(t) \propto a(t)$. Besides simplicity, there are two main reasons for this choice: 1) since the event activity implicitly encodes the event stream dynamics, by linearity, the adaptive decay rate will also encode the event stream dynamics, and 2) as discussed previously, the decay rate should scale with the event stream dynamics. Substituting the considered adaptive decay rate and only considering the decay term from Eq. (9), *i.e.*, $a(t) = \beta(t, s)a(s)$, Eq. (11) can be written as

$$\frac{da(t)}{dt} = -a^2(t). \tag{12}$$

The solution of Eq. (12) is given by

$$a(t) = \frac{1}{1 + a(0)t} a(0), \tag{13}$$

from which a recursive expression w.r.t. to an arbitrary previous timestamp $t_p \leq t$ can be obtained

$$a(t) = \frac{1}{1 + a(t_p)(t - t_p)} a(t_p), \tag{14}$$

and consequently the adaptive decay is given by

$$\boxed{\beta_a(t, t_p) = \frac{1}{1 + a(t_p)(t - t_p)}.} \tag{15}$$

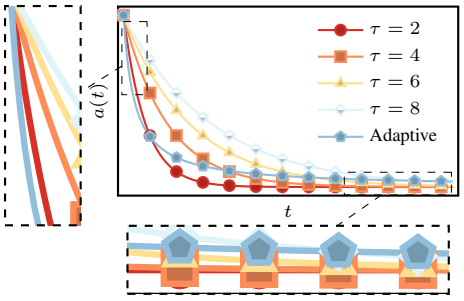


Figure 3. **Constant *vs*. adaptive global decay processes.** For the constant decay process, the different time constants τ used are provided in the legend in milliseconds.

Fig. 3 depicts the constant and adaptive global decay processes. The adaptive process exhibits a sharper decrease when the event activity is higher (left zoom) but a smoother decrease when the event activity is lower (bottom zoom). The proposed adaptive decay process thus adapts to the event stream dynamics in accordance to the assertion made regarding how the decay rate should behave based on Fig. 2.

Algorithm. A fully event-based algorithm that estimates the adaptive decay and the event activity is presented in Algorithm 1. The algorithm is simple, and it only comprises two steps that are executed for each event e_i in the event stream, namely: 1) to compute the current adaptive decay $\beta_{a,i}$ based on the previous event activity a_{i-1}, according to Eq. (15), and 2) to compute the current event activity a_i based on the current adaptive decay $\beta_{a,i}$ and the previous event activity a_{i-1}, according to Eq. (10).

Two additional values need to be set, which correspond to the initial event activity a_0 and the initial timestamp t_0. Both can be set to 0 for typical sequences. From a conceptual standpoint, the event stream is initially empty, and the first event timestamp is typically set to 0. From a more practical view, we have observed that a few hundred events are needed for the adaptive decay process to converge, making the initial conditions close to irrelevant for typical sequences, which have millions of events.

4.4. Event Weight

We can construct a temporally adaptive event weight $w_k(t)$ inspired by the properties of the adaptive decay given by Eq. (15), with which each event can be augmented, *i.e.*,

$$\hat{e}_k \coloneqq (e_k, w_k(t)) \,, \tag{16}$$

where e_k represents each event, as defined in Eq. (1). The temporally adaptive event weight is defined according to

$$w_k(t) \coloneqq \frac{1}{1 + a(t)(t - t_k)}, \tag{17}$$

where $a(t)$ is the event activity at timestamp t, given by Eq. (14). We recall that the event activity $a(t)$ is a global quantity that is updated for each incoming event e_i, according to Algorithm 1. The event weight $w_k(t)$ is thus a quantity that temporally varies according to the global event stream dynamics. There are two main utilities for this quantity, namely: 1) to determine whether an event is considered active, and 2) to weigh the event contribution [14]. An event is considered active if its contribution is still relevant. Using Eq. (17), we can consider that an event e_k is active if its weight is higher than a threshold, *i.e.*, $w_k(t) > \mathsf{w}$; otherwise, it is considered inactive. These utilities can be used separately or jointly, and, as shown in Sec. 5, they can be applied to event-based and batch-based processing paradigms.

Algorithm 1 Event-based Adaptive Global Decay Process

Input: Event stream $\{e_i\}$, event activity a_0 and timestamp t_0.
Output: Event activity a_i, adaptive decay $\beta_{a,i}$.
Procedure:

1: **for** each event e_i **do**
2: Set $\beta_{a,i} \leftarrow \frac{1}{1+a_{i-1}(t_i - t_{i-1})}$, Eq. (15).
3: Set $a_i \leftarrow \beta_{a,i} a_{i-1} + 1$, Eq. (10).
4: **end for**

Adaptive event batches. We can adaptively gather consecutive events into batches for later processing, using the event weight defined in Eq. (17). Let us consider an arbitrary event e_j of the event stream, which is augmented with the temporally adaptive weight $w_j(t)$. From Eq. (17), it is straightforward to show that the inequality $w_j(t) \leq w_k(t)$ holds for $t_j \leq t_k \leq t$, meaning that the weight of an event will never be higher than the weight of subsequent events. Thus, to efficiently form event batches, we only need to keep gathering incoming events until the first event e_j is considered inactive, *e.g.*, $w_j(t) \leq \mathsf{w}$. Let us assume that N events were gathered. The adaptive batch is then formed by the N gathered events. The process continues with the first event of the next batch being the first incoming event after forming the current batch, *i.e.*, e_{j+N}.

5. Experiments and Results

We evaluate the proposed adaptive decay process in several event-based vision problems, namely: asynchronous spatial convolution (Sec. 5.2), global motion estimation (Sec. 5.3), pattern classification (Sec. 5.4) and learning-based image intensity reconstruction (Sec. 5.5). For each problem, we compare the performance between the original method, *i.e.*, without adaptive decay, and the corresponding method using the proposed adaptive decay process. We refer to the original works as the *Baseline* since they achieve state-of-the-art performance. Except where mentioned otherwise, the implementation of the original methods follows the description and hyper-parameters setup reported in the respective papers. Additional results are provided in the supplementary material.

5.1. Datasets

DAVIS 240C Dataset [24][1]. It consists of approximately 1 minute of real sequences captured by a DAVIS240 camera [4] with increasing motion magnitude. The camera's pose is provided at 200Hz by a motion-capture system, and a built-in Inertial Measurement Unit (IMU) provides acceleration and angular velocity measurements at 1000Hz. We use this dataset to evaluate the proposed method in asynchronous spatial convolution, global motion estimation and image intensity reconstruction.

Poker DVS [37][2] **and N-MNIST** [29][3] **Datasets**. The Poker DVS dataset consists of 4 classes, each representing a suit of a set of regular playing cards. The N-MNIST dataset is a conversion of the MNIST dataset to event-based data. We use these datasets to evaluate the proposed method in classification tasks using HOTS [17].

UZH-FPV Drone Racing Dataset [5][4]. It consists of approximately 15 minutes of real sequences captured by a mini DAVIS346 camera onboard a flying racing quadrotor. We use this dataset to train a deep neural-network architecture [30, 34] for image intensity reconstruction from events.

5.2. Asynchronous Spatial Convolution

We consider the event-based convolution method [35] as the baseline. A high-pass filter in the frequency domain is proposed to attenuate low-frequency noise, which boils down to employing a constant decay process, as described in Sec. 4.1. To compare the baseline with the proposed adaptive decay process (15), we take snapshots at certain timestamps of the state of applying the asynchronous spatial convolution with an identity kernel. We then measure the corresponding image similarity with the real log-frames gradient magnitude, in terms of Structural Similarity Index (SSIM) [40] and Multi-Scale SSIM (MSSIM) [41], to demonstrate that the proposed method globally adapts to event stream dynamics by yielding sharp edge-like images.

Tab. 2 presents the similarity results averaged over all the `rotation` and `translation` sequences [24]. The proposed adaptive decay process achieves the best overall results for both similarity measures. As previously discussed in Sec. 4.1, a single time constant can not appropriately handle event streams with varying dynamics. To quantitatively assess this, we present in Fig. 4 the box plots of the similarity measures averaged over all the `rotation` and `translation` sequences [24] in sub-intervals with increasing motion dynamics. Globally, using a lower time constant (*e.g.*, $\tau = 1$), better performance is achieved in later sub-intervals, *e.g.*, 3 and 4, since higher event rates are more reliably captured. Conversely, using a higher time constant (*e.g.*, $\tau = 100$), better performance is achieved in earlier sub-intervals, *e.g.*, 1. A time constant in the order of tens of milliseconds (*e.g.*, $\tau = 10$) represents a compromise between the previous two, being a common choice in works that rely on time constants [1, 35, 42]. The proposed decay process achieves an almost constant performance across the sub-intervals since it adapts to the event stream dynamics.

[1] https://rpg.ifi.uzh.ch/davis_data.html.
[2] http://www2.imse-cnm.csic.es/caviar/POKERDVS.html.
[3] https://www.garrickorchard.com/datasets/n-mnist.
[4] https://fpv.ifi.uzh.ch/.

Method	rotation SSIM	rotation MSSIM	translation SSIM	translation MSSIM
Baseline ($\tau = 1$)	**0.36**	0.51	0.39	0.51
Baseline ($\tau = 10$)	0.35	0.54	0.41	**0.59**
Baseline ($\tau = 100$)	0.20	0.38	0.29	0.45
Adaptive	**0.36**	**0.56**	**0.42**	**0.59**

Table 2. Image similarity averaged over all the `rotation` and `translation` sequences [24]. The time constants τ are in milliseconds. Higher is better.

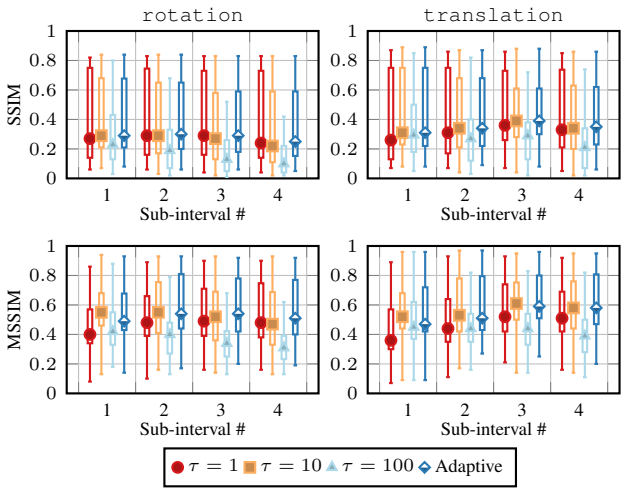


Figure 4. Image similarity box plots by intervals of 15 seconds averaged over all the `rotation` and `translation` sequences [24]. The time constants τ are in milliseconds.

5.3. Global Motion Estimation

We compare each baseline method with two adaptive variants that use the proposed decay method, namely: 1) variant that also detects event inactivity based on the event weight (17) as described in Sec. 4.4, and 2) variant that additionally weighs each event contribution. The variants are identified by prepending the method's name by **A**daptive and **W**eighted, respectively. Note that a fixed number of events is *not* provided to the adaptive variants; the weight threshold w is fixed to 0.1 instead, *e.g.*, from which the batches are adaptively formed in batch-based processing.

Evaluation metrics. Similar to [25, 27], the evaluation metrics are: velocity error (e_x, e_y, e_z), respective standard deviation (σ), Root-Mean Squared (RMS) error, and RMS er-

	Method	e_x	e_y	e_z	σ	RMS	RMS%
boxes	Baseline	9.46	8.99	**11.15**	**13.31**	**13.37**	**1.82**
	AIncP	**7.50**	8.08	11.73	13.56	13.59	1.85
	WIncP	7.86	**7.72**	14.01	13.96	13.99	1.91
dynamic	Baseline	8.38	11.18	17.81	20.51	20.59	3.97
	AIncP	6.06	7.13	**7.97**	10.97	11.06	2.13
	WIncP	**5.89**	**6.56**	8.05	**10.03**	**10.13**	**1.95**
poster	Baseline	14.84	10.43	**14.02**	18.03	18.09	1.93
	AIncP	**10.46**	**8.51**	15.80	**16.80**	**16.87**	**1.80**
	WIncP	12.11	8.98	18.48	18.39	18.49	1.97
shapes	Baseline	74.36	102.0	179.6	158.9	158.9	21.1
	AIncP	19.13	32.39	61.98	54.36	54.54	7.25
	WIncP	**14.02**	**22.76**	**42.97**	**36.10**	**36.18**	**4.81**
Average	Baseline	26.76	33.16	55.63	52.69	52.74	7.21
	AIncP	10.79	14.03	24.37	23.92	24.01	3.26
	WIncP	**9.97**	**11.50**	**20.88**	**19.62**	**19.70**	**2.66**

Table 3. **Event-based angular velocity estimation.** Accuracy comparison on the `rotation` sequences [24]. Lower is better.

ror w.r.t. the maximum excursions of ground truth in percentage (%). We evaluate angular velocity estimates in deg/s and linear velocity estimates in m/s on the `rotation` and `translation` sequences [24], respectively.

Event-based. We consider the Incremental Potential (IncP) method [27][5] as the baseline. The hyper-parameters are set as follows: first-in-first-out (FIFO) queue with the most recent 10000 events, *i.e.*, ≈ 0.23 events per pixel, and 9×9 spatial neighborhood centered at the incoming event.

Tab. 3 presents the accuracy performance on the `rotation` sequences [24]. Overall, the best adaptive variant (**W**eighted) outperforms the baseline by 63% in terms of RMS error, mainly due to the improvement of 80% on the `shapes` sequence. Compared to the other sequences, this sequence is much sparser, exhibiting lower spatial dynamics, which explains why considering a single number of events for all sequences is not adequate. The proposed decay process adapts to the dynamics of all sequences.

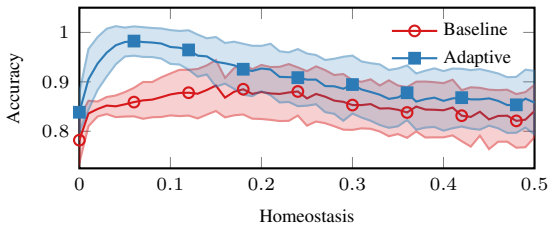
Batch-based. We consider the Contrast Maximization (CMax) method [8, 9] and the Approximate Potential (ApproxP) method [25, 27] as the baselines. The batch size and the spatial neighborhood size are set to 20000 events, *i.e.*, ≈ 0.46 events per pixel, and 3×3, respectively.

Tab. 4 presents the average accuracy performance on the `rotation` and `translation` sequences [24]. On the `rotation` sequences, in terms of RMS error, the CMax and the ApproxP baselines are outperformed by the best adaptive variants by 50% and 43%, respectively. On the `translation` sequences, in terms of RMS error, the CMax and the ApproxP baselines are outperformed by the best adaptive variants by 30% and 29%, respectively.

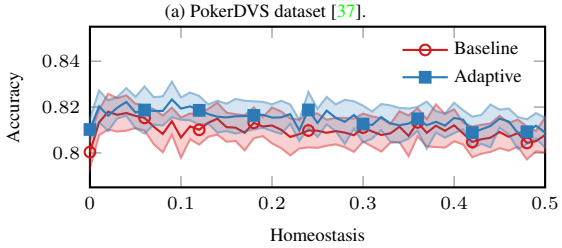
[5] https://github.com/ImperialCollegeLondon/EventEMin.

	Method	e_x	e_y	e_z	σ	RMS	RMS%
rotation	Baseline [8]	17.99	17.52	25.20	31.69	31.75	4.29
	ACMax	7.32	14.39	13.17	15.51	15.66	2.16
	WCMax	**7.30**	**14.38**	**13.16**	**15.34**	**15.50**	**2.14**
	Baseline [27]	17.39	17.02	25.01	29.55	29.62	3.99
	AApproxP	**7.62**	**15.57**	**14.74**	**16.43**	**16.60**	**2.28**
	WApproxP	8.27	16.02	14.99	16.69	16.86	2.33
translation	Baseline [8]	0.45	0.42	0.35	0.56	0.56	18.30
	ACMax	**0.31**	**0.24**	**0.34**	**0.39**	**0.40**	**12.87**
	WCMax	**0.31**	**0.24**	0.36	0.40	0.41	13.06
	Baseline [27]	0.44	0.37	0.36	0.54	0.54	17.67
	AApproxP	**0.30**	**0.24**	**0.32**	**0.40**	**0.40**	**12.51**
	WApproxP	**0.30**	**0.24**	0.34	**0.40**	0.40	12.74

Table 4. **Batch-based angular and linear velocity estimation.** Average accuracy comparison on the `rotation` and `translation` sequences [24]. Lower is better.



(a) PokerDVS dataset [37].



(b) N-MNIST dataset [29].

Figure 5. Classification accuracy using HOTS [17] in function of homeostasis [12] on the (a) PokerDVS [37] and (b) N-MNIST [29] datasets. The line and surrounding shaded area represent the mean and standard deviation over 100 trials, respectively.

5.4. Pattern Classification

We consider HOTS [17][6] as the baseline feature extractor and a simple K-Nearest Neighbours classifier with $K = 12$ [22]. We compare the baseline with an adaptive variant, whose time-surfaces around an event are given by the adaptive decay (15). Fig. 5 presents the accuracy performance for both approaches in function of the homeostatic parameter [12], which balances the activity among the time-surfaces in a layer. Overall, the proposed adaptive variant outperforms the baseline: in terms of maximum accuracy performance, there is an absolute increase of 9% and 0.5% on the PokerDVS [37] and N-MNIST [29] datasets, respec-

[6] https://github.com/neuromorphic-paris/cpphots.

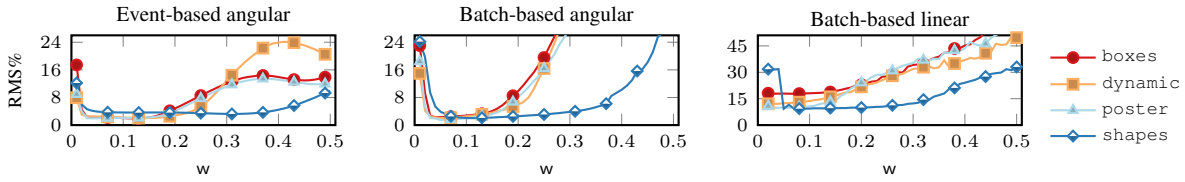


Figure 6. Angular and linear velocity estimation in function of the weight threshold w on the `rotation` and `translation` sequences [24] for (left) event-based and (middle)-(right) batch-based processing. Note the plateau for all the sequences around $w = 0.1$.

Method	MSE (↓)	SSIM (↑)	MSSIM (↑)
Baseline [30]	0.095	0.36	0.50
Proposed	**0.085**	0.36	**0.51**

Table 5. **Image intensity reconstruction evaluation**. Average performance comparison on DAVIS [24] sequences.

Weight	MSE (↓)	SSIM (↑)	MSSIM (↑)
$w = 0.01$	0.130	0.23	0.36
$w = 0.05$	**0.085**	**0.36**	**0.51**
$w = 0.10$	0.095	0.35	0.48
$w = 0.15$	0.120	0.30	0.35
$w = 0.20$	0.167	0.15	0.09
$w = 0.25$	0.132	0.25	0.26

Table 6. Image reconstruction evaluation in function of the weight threshold w on DAVIS [24] sequences.

tively. Most notably, this is achieved by requiring 2 fewer hyper-parameters, namely the time constants for each layer. In general, for an HOTS architecture with L layers, the proposed adaptive method requires L fewer hyper-parameters compared to the original implementation [17].

5.5. Image Intensity Reconstruction

We consider SSL-E2V [30][7] as the baseline image reconstruction method by training the network with a fixed-size input event stream of 20000, *i.e.*, ≈ 0.46 events per pixel. We compare it with another network that was trained with an adaptive-size input event stream, whereby the weight threshold w was fixed to 0.05. As presented in Tab. 5, the proposed adaptive-size training achieves slightly better average performance in terms of Mean Squared Error (MSE) and MSSIM, and similar performance in terms of SSIM compared to the baseline.

Event weight influence. For some experiments conducted, one may question whether we have just replaced one parameter, *i.e.*, the number of events or equivalently the decay rate, for another, *i.e.*, the weight threshold. In this section, we empirically assess the influence of the weight threshold.

Fig. 6 presents the performance on global motion estimation in function of the weight threshold w. For all the sequences, there is minimum plateau around 0.1. Tab. 6 reports the performance on image reconstruction in function of the weight threshold w. The best performances were obtained using a weight threshold of 0.05 and 0.1, respectively, independently of the scene dynamics.

[7] https://github.com/tudelft/ssl_e2vid.

Dataset	Resolution	Min	Max	Mean ± SD	Median
DAVIS [24]	240 × 180	2.43	7.86	4.33 ± 1.00	4.37
DSEC [11]	640 × 480	0.79	3.46	2.02 ± 2.00	2.03

Table 7. Runtime per event in nanoseconds over 100 trials.

Runtime. Tab. 7 presents the runtime per event of the proposed adaptive decay process. The runtime was estimated using a single thread on a PC with an Intel Core i7-7700 CPU 3.6GHz and 32GB of RAM. The runtime per event is almost three orders of magnitude lower than the typical temporal resolution of event cameras. The latency introduced is thus almost negligible, and it does not depend on the camera resolution, *i.e.*, the number of pixels of the camera.

6. Conclusion

We propose a global adaptive decay process for event cameras and demonstrate its usefulness across several event-based vision problems. By considering the proposed process, the baseline methods' performances are shown to be consistently improved. More importantly, the proposed method removes a parameter usually tuned according to prior assumptions while introducing negligible latency. We believe this method can have a broad positive impact across most event-based applications. This work opens several avenues for future research, including additional empirical evaluation regarding the weight threshold value and considering other forms of temporally varying decay rates. Since the proposed method works globally, *i.e.*, it assumes that events come from a single motion, it should fail in scenarios whereby, *e.g.*, objects are independently moving. Considering multi-scale approaches or explicitly modeling more than one motion are also possible avenues for future research.

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

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. 3, 6, 12

[2] Ignacio Alzugaray and Margarita Chli. Asynchronous Corner Detection and Tracking for Event Cameras in Real Time. *IEEE Robotics and Automation Letters*, 3(4):3177–3184, Oct. 2018. 3

[3] 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. 3

[4] 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. 6

[5] Jeffrey Delmerico, Titus Cieslewski, Henri Rebecq, Matthias Faessler, and Davide Scaramuzza. Are We Ready for Autonomous Drone Racing? The UZH-FPV Drone Racing Dataset. In *International Conference on Robotics and Automation (ICRA)*, pages 6713–6719, 2019. 6

[6] Tobias Fischer and Michael Milford. Event-Based Visual Place Recognition With Ensembles of Temporal Windows. *IEEE Robotics and Automation Letters*, 5(4):6924–6931, 2020. 3

[7] 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, 11

[8] 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. 3, 7

[9] Guillermo Gallego and Davide Scaramuzza. Accurate Angular Velocity Estimation With an Event Camera. *IEEE Robotics and Automation Letters*, 2(2):632–639, 2017. 1, 2, 3, 7

[10] Daniel Gehrig, Henri Rebecq, Guillermo Gallego, and Davide Scaramuzza. Asynchronous, Photometric Feature Tracking Using Events and Frames. In *Computer Vision – ECCV*, Lecture Notes in Computer Science, pages 766–781. Springer International Publishing, 2018. 3

[11] Mathias Gehrig, Willem Aarents, Daniel Gehrig, and Davide Scaramuzza. DSEC: A Stereo Event Camera Dataset for Driving Scenarios. *IEEE Robotics and Automation Letters*, 6(3):4947–4954, 2021. 8

[12] Antoine Grimaldi, Victor Boutin, Laurent Perrinet, Sio-Hoi Ieng, and Ryad Benosman. A Homeostatic Gain Control Mechanism to Improve Event-driven Object Recognition. In *International Conference on Content-Based Multimedia Indexing (CBMI)*, pages 1–6. IEEE, 2021. 7

[13] Cheng Gu, Erik Learned-Miller, Daniel Sheldon, Guillermo Gallego, and Pia Bideau. The Spatio-Temporal Poisson Point Process: A Simple Model for the Alignment of Event Camera Data. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13495–13504, 2021. 3

[14] Javier Hidalgo-Carrio, Guillermo Gallego, and Davide Scaramuzza. Event-aided Direct Sparse Odometry. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5771–5780. IEEE, 2022. 5

[15] 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. 2

[16] 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

[17] 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, 2, 6, 7, 8, 12

[18] 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

[19] 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, 3

[20] Weng Fei Low, Zhi Gao, Cheng Xiang, and Bharath Ramesh. SOFEA: A Non-iterative and Robust Optical Flow Estimation Algorithm for Dynamic Vision Sensors. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 368–377, June 2020. 3

[21] Ana I. Maqueda, Antonio Loquercio, Guillermo Gallego, Narciso García, and Davide Scaramuzza. Event-Based Vision Meets Deep Learning on Steering Prediction for Self-Driving Cars. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5419–5427, 2018. 2

[22] Jean-Matthieu Maro, Sio-Hoi Ieng, and Ryad Benosman. Event-Based Gesture Recognition With Dynamic Background Suppression Using Smartphone Computational Capabilities. *Frontiers in Neuroscience*, 14:275, 2020. 7

[23] Elias Mueggler, Christian Forster, Nathan Baumli, Guillermo Gallego, and Davide Scaramuzza. Lifetime Estimation of Events from Dynamic Vision Sensors. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4874–4881, 2015. ISSN: 1050-4729. 2, 3

[24] 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. 1, 3, 6, 7, 8, 11, 12, 13, 14, 15, 16, 17

[25] 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, 6, 7

[26] 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

[27] 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, 3, 6, 7

[28] A.V. Oppenheim, R.W. Schafer, and T.G. Stockham. Nonlinear Filtering of Multiplied and Convolved Signals. *Proceedings of the IEEE*, 56(8):1264–1291, Aug. 1968. 2, 11

[29] Garrick Orchard, Cedric Meyer, Ralph Etienne-Cummings, Christoph Posch, Nitish Thakor, and Ryad Benosman. HFirst: A Temporal Approach to Object Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(10):2028–2040, 2015. 6, 7, 12

[30] 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. 1, 3, 6, 8, 12

[31] 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. 1

[32] Henri Rebecq, Guillermo Gallego, Elias Mueggler, and Davide Scaramuzza. EMVS: Event-Based Multi-View Stereo—3D Reconstruction with an Event Camera in Real-Time. *International Journal of Computer Vision*, 126(12):1394–1414, 2018. 3

[33] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. High Speed and High Dynamic Range Video with an Event Camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(6):1964–1980, 2021. 3

[34] Fitsum Reda, Janne Kontkanen, Eric Tabellion, Deqing Sun, Caroline Pantofaru, and Brian Curless. FILM: Frame Interpolation for Large Motion. In *Computer Vision – ECCV*, Lecture Notes in Computer Science, pages 250–266. Springer, 2022. 6, 12

[35] Cedric Scheerlinck, Nick Barnes, and Robert Mahony. Asynchronous Spatial Image Convolutions for Event Cameras. *IEEE Robotics and Automation Letters*, 4(2):816–822, 2019. 1, 2, 3, 4, 6, 11, 12

[36] Yusuke Sekikawa, Kosuke Hara, and Hideo Saito. EventNet: Asynchronous Recursive Event Processing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3882–3891, 2019. 2

[37] Teresa Serrano-Gotarredona and Bernabé Linares-Barranco. Poker-DVS and MNIST-DVS. Their History, How They Were Made, and Other Details. *Frontiers in Neuroscience*, 9, 2015. 6, 7, 12

[38] Sumit Bam Shrestha and Garrick Orchard. SLAYER: Spike Layer Error Reassignment in Time. In *Advances in Neural Information Processing Systems (NIPS)*, volume 31. Curran Associates, Inc., 2018. 1

[39] Amos Sironi, Manuele Brambilla, Nicolas Bourdis, Xavier Lagorce, and Ryad Benosman. HATS: Histograms of Averaged Time Surfaces for Robust Event-Based Object Classification. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1731–1740, 2018. 2

[40] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 6

[41] Z. Wang, E.P. Simoncelli, and A.C. Bovik. Multiscale Structural Similarity for Image Quality Assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems Computers*, volume 2, pages 1398–1402, 2003. 6

[42] Yi Zhou, Guillermo Gallego, and Shaojie Shen. Event-Based Stereo Visual Odometry. *IEEE Transactions on Robotics*, 37(5):1433–1450, Oct. 2021. 6, 12

[43] 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. 3

[44] Alex Zihao Zhu, Yibo Chen, and Kostas Daniilidis. Realtime Time Synchronized Event-Based Stereo. In *Computer Vision – ECCV*, Lecture Notes in Computer Science, pages 438–452. Springer International Publishing, 2018. 1, 3

[45] 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, 3