Are High-Resolution Event Cameras Really Needed?

Daniel Gehrig and Davide Scaramuzza

Dept. of Informatics, Univ. of Zurich and
Dept. of Neuroinformatics, Univ. of Zurich and ETH Zurich
{dgehrig,sdavide}@ifi.uzh.ch

Abstract. Due to their outstanding properties in challenging conditions, event cameras have become indispensable in a wide range of applications, ranging from automotive, computational photography, and SLAM. However, as further improvements are made to the sensor design, modern event cameras are trending toward higher and higher sensor resolutions, which result in higher bandwidth and computational requirements on downstream tasks. Despite this trend, the benefits of using high-resolution event cameras to solve standard computer vision tasks are still not clear. In this work, we report the surprising discovery that, in low-illumination conditions and at high speeds, low-resolution cameras can outperform high-resolution ones, while requiring a significantly lower bandwidth. We provide both empirical and theoretical evidence for this claim, which indicates that high-resolution event cameras exhibit higher per-pixel event rates, leading to higher temporal noise in low-illumination conditions and at high speeds. As a result, in most cases, high-resolution event cameras show a lower task performance, compared to lower resolution sensors in these conditions. We empirically validate our findings across several tasks, namely image reconstruction, optical flow estimation, and camera pose tracking, both on synthetic and real data. We believe that these findings will provide important guidelines for future trends in event camera development.

Keywords: Methodology, and Theory, Low-level Vision

Multimedia Material

For videos and more, visit our project page at https://uzh-rpg.github.io/eres/.

1 Introduction

Event cameras are neuromorphic vision sensors that work radically different than standard frame-based cameras. Instead of measuring synchronous intensity frames at fixed time intervals, event cameras only measure the changes in logarithmic intensity and do this asynchronously and with microsecond-level temporal resolution. The resulting output is a stream of asynchronous events,
We study the impact of camera resolution on downstream task performance and bandwidth requirements. Intuitively, in ideal settings (top curve) using higher resolution sensors leads to higher performance, but also higher event rates, which cause higher bandwidths. However, in realistic noise conditions (bottom curves) this is no longer the case. We find that especially at high speeds and in low-illumination conditions, using high-resolution cameras significantly decreases the task performance. In these settings, lower resolutions are more viable in terms of performance and bandwidth.

where each event encodes the pixel location, time and sign (polarity) of the intensity change. Due to their working principle, event cameras have several advantages over standard cameras, including a higher dynamic range (140 dB vs. 60 dB), low power consumption (milliwatts instead of watts), low motion blur, and microsecond-level latency and temporal resolution.

Due to these properties, applications and research using event cameras has spread to various fields, including robotics\cite{19, 10, 34, 40, 21}, tactile sensing\cite{2, 41}, high-speed control\cite{38, 43}, driving\cite{17, 36, 45}, space \cite{8} and computational photography\cite{32, 42, 4, 44}. Since their surge in popularity, event-camera technology has continuously improved, leading to several commercially available sensors, summarized in Tab.1 (borrowed from the survey in \cite{12}). These improvements include better noise characteristics and higher sensor resolution. While early event cameras such as the DVS128 had a resolution $128 \times 128$\cite{25}, modern event cameras now go up to $1280 \times 960$\cite{39}.

While higher sensor resolutions have enabled new applications, such as computational photography \cite{32, 42}, they are increasingly placing a burden on the end-user both in terms of data bandwidth and computational requirements of downstream systems\cite{28, 29}. This is because, as resolution increases, so does the resulting event rate. Common issues that arise with modern high-resolution event cameras are slow readout rates and output bus saturation\cite{12}, which both lead to event timestamp perturbations and loss of events. While readout rates have been steadily increasing from 2 MHz \cite{25} to 1200 MHz \cite{39}, they require sophisticated readout schemes to reduce overhead. For this reason, some solutions use hardware-integrated filters to reduce the event rate\cite{11}. However, while existing strategies reduce the overall bandwidth, they are not lossless, and, thus, either skip events or introduce systematic noise on the event timestamps. Both can have deleterious effects on the downstream task.
This raises the question, of what the true benefits of high-resolution cameras are. In this work, we shed light on this question by empirically and theoretically studying the effect of event camera resolution on three downstream tasks, namely (i) image reconstruction, (ii) optical flow estimation, and (iii) camera pose tracking. For each task, we investigate the trade-off between task performance and event rate in ideal, daytime and nighttime conditions (Fig.1). We find both on synthetic and real data that higher resolution event cameras exhibit higher per-pixel event rates, which makes them less robust to high-speed motion and more susceptible to temporal noise induced by low illumination conditions. This leads to the surprising discovery that, in these conditions, lower resolution event cameras exhibit better performance than their high-resolution counterparts, while also producing fewer events. We demonstrate that this conclusion generalizes across several tasks and both learning- and model-based methods. Additionally, we find that, when including noisy nighttime data in the training set, learning-based methods can overcome this trend. Moreover, contrast-maximization-based methods [15] only show this trend at high speeds, while photometric-based methods [14] already degrade at low speeds. We believe that these findings will act as important guidelines for future trends in event camera manufacturing, especially when moving toward higher resolution sensors. Our contributions are:

- We theoretically and empirically study the effect of higher resolutions on per-pixel event rates and show that event rates with increasing sensor resolution.
- We study the effect of event camera resolution on three important computer vision tasks. We show that, both in daytime and nighttime conditions and during high-speed motions, tasks performed with high-resolution cameras show a significant performance drop, which is smaller for lower resolutions.
- We thus identify event-camera resolutions which outperform high-resolution event cameras in terms of task performance, while requiring less bandwidth and do so over a variety of model-based and learning-based methods, as well as in simulation and on real data.

2 Related Work

Since the first commercial event camera in 2008[25], several new sensors appeared, (Tab. 1) which drove the development of different application domains. While early works used the DVS[25], ATIS[30] and DAVIS240[5] to solve simple tasks like low degree-of-freedom SLAM or object recognition, more modern approaches use higher resolution event cameras such as the Prophesee Gen3, Rhino 3, and Samsung Gen3 which provide more fine-grained detail in the event stream. Most recently, a race to increase sensor resolution culminated in the Celex-V, Prophesee Gen4, and Samsung DVS-Gen4, all featuring approximately one-megapixel resolution. These sensors enable applications in computational photography, where higher resolutions directly translate to more fine-grained detail. However, higher resolutions result in higher event rates, which place burdens on the data bus, readout systems, and downstream tasks. These include readout congestion and bus saturation, both of which can introduce significant
Table 1. Event camera comparison. Courtesy of [12].

| Supplier   | Camera         | Year, Reference | Resolution Interface | Max. Bandwidth (Mev/s) |
|------------|----------------|-----------------|----------------------|------------------------|
| iniVation  | DVS128         | 2008[25]        | 128 × 128            | USB 2 1               |
|            | DAVIS240       | 2014[5]         | 240 × 180            | USB 2 12              |
|            | DAVIS346       | 2017            | 346 × 260            | USB 3 12              |
|            | DVXplorer       | 2020            | 640 × 480            | USB 3 165             |
| Prophesee  | ATIS           | 2011[30]        | 304 × 240            | - -                   |
|            | Gen3 CD        | 2017[31]        | 640 × 480            | USB 3 66              |
|            | Gen3 ATIS      | 2017[31]        | 480 × 360            | USB 3 66              |
|            | Gen4 CD        | 2020[11]        | 1280 × 720           | USB 3 1066            |
| Samsung    | DVS-Gen2       | 2017[37]        | 640 × 480            | USB 2 300             |
|            | DVS-Gen3       | 2018[33]        | 640 × 480            | USB 3 600             |
|            | DVS-Gen4       | 2020[39]        | 1280 × 960           | USB 3 1200            |
| CelePixel  | CeleX-IV       | 2017[20]        | 768 × 640            | - 200                 |
|            | CeleX-V        | 2019[7]         | 1280 × 800           | - 140                 |
| Insightness| Rhino 3        | 2018[23]        | 320 × 262            | USB 2 50              |

delays or noise on event timestamps, and even dropped events[12]. In the next section, we discuss strategies to overcome this challenge.

**Event Filtering and Compression:** Higher event rates directly translate to higher storage requirements. To counteract these, lossy and lossless event compression techniques were introduced[3, 24]. However, these methods do not work online and are thus not able to mitigate event distortions at recording time.

Modern event cameras such as the Prophesee Gen4[11] include sophisticated read-out schemes such as row-wise arbitration which reduce per-event data transmission but introduce systematic errors in the event timestamps. Recently, event rate controllers were introduced which aim at reducing the event rate while recording. They work by randomly skipping events [11] or by tuning camera parameters during recording[9]. While these controllers can alleviate bandwidth requirements, they do so by directly affecting the events. Thus algorithms and event rate controllers need to be co-designed, thereby increasing complexity which inhibits scaling to more complex tasks. Instead of filtering the event stream we seek to understand the benefits of high-resolution cameras and their trade-offs in terms of downstream task performance and bandwidth.

**Power-Performance Analysis:** The tradeoff between task performance and data bandwidth was initially studied in [6] in the context of a Power-Performance analysis. They compared frame- and event-based sensors in terms of their ability to reconstruct the continuous-time intensity profile at a single pixel (Performance), and their required bandwidth (Power). However, this analysis was limited to a simple simulated toy example which did not consider sensor noise. In this work, we generalize this analysis to event cameras at different resolutions and in different conditions and study them in terms of their

---

1 The studied intensity signal was modeled as a random process that did not resemble real data and did not model sensor noise.
To understand the event rate at different resolutions we study the log brightness derivative \( L_t \) of a given scene (a). Brightness changes are mostly confined to edges, making them semi-dense. We consider the brightness derivative centered at an edge pixel of varying size (b), and find that as pixel size decreases, more details become visible which increase \( L_t \) (b), leading to a power law with slope \( \gamma^* < 0 \) (related to the fractal dimension[26] at small pixel sizes). Thus, to satisfy Eq. (2) high resolution event cameras must trigger events at higher rates (c, top), making them more sensitive to temporal effects, such as pixel cutoff frequency or high-speed motion (c, bottom).

**Fig. 2.**

3 Approach

In the first step, we will review the event generation model before studying events at higher resolution. Finally, we will introduce baselines which we will use to quantify task performance.

**Event Generation:** Event cameras have independent pixels \( x = (x, y) \) which trigger an event whenever their internal voltage \( L(x, t) \), caused by the illumination incident on that pixel, exceeds a certain threshold \( C \), called the contrast threshold. The resulting events are four tuples \( e_k = (x_k, t_k, p_k) \), each encoding the location \( x_k \), time \( t_k \) and sign (polarity) \( p_k \in \{-1, 1\} \) of this change. The event trigger condition can be formalized as follow

\[
p_k(L(x_k, t_k) - L(x_k, t - \Delta t_k)) > C,
\]

which describes the ideal event generation model [13, 14]. Here \( \Delta t_k \) is the time since the last event at pixel \( x_k \). By linearizing \( L \), Eq. (1) can be rewritten as

\[
p_k L_t(x_k, t_k) \Delta t_k > C,
\]

with \( L_t = \frac{\partial L}{\partial t} \). While in the ideal setting the internal voltage level is equated with the logarithmic brightness \( L(x, t) \equiv \log I(x, t) \), more recently, a more realistic event-camera model has been introduced which models this voltage as a second order system, dependent on the cutoff frequency \( f_{\text{cutoff}} \) [22]. It determines the rate at which event pixels track an input brightness signal, and depends on the
ambient illumination conditions. In darker settings $f_{\text{cutoff}}$ is lower, causing event pixels to become slow and generate characteristic “motion blur patterns”, which can be viewed in Fig. 3 of [22].

**Event Generation at Different Resolutions:** While Eq. (2) assumes that per-pixel scene irradiance is measured in a single direction, it is integrated over a steric angle, which spans a single pixel. As camera resolution increases, this pixel area decreases, leading to a more fine-grained perception of the environment. This integration is followed by a subsampling of the image. Let $E$ denote the irradiance map at a high resolution and $\mathbf{x}$ denote the coordinates at this high resolution.

$$L(\mathbf{x}, t) = \log I(\mathbf{x}, t) = \log[(k * E)(\mathbf{x}, t)],$$  
with \[ k(\mathbf{x}) = \begin{cases} \frac{1}{p^2}, & -\frac{p}{2} \leq x, y \leq \frac{p}{2} \\ 0, & \text{else} \end{cases} \]

where the scene irradiance $E$ is convolved with a box filter $k$ with sides $p$ corresponding with the pixel size. Followed by a subsampling to a lower resolution with coordinates $\mathbf{x}_l$.

$$L_l(\mathbf{x}_l, t) = L(p\mathbf{x}_l, t)$$

We consider the event rate at a fixed pixel $\mathbf{x}$, with coordinates $\mathbf{x}_l = \frac{\mathbf{x}}{p}$ for pixel size $p$, by studying the temporal derivative $L_t$, at that pixel:

$$L_t(\mathbf{x}, t) = \frac{\partial \log I(\mathbf{x}, t)}{\partial t} = \frac{I_t(\mathbf{x}, t)}{I(\mathbf{x}, t)} = (k * E_t)(\mathbf{x}, t) \quad (k * E)(\mathbf{x}, t).$$

We observe that $E_t$ is sparse, with high values close to occlusions and contrast changes (Fig. 2 (a), top). Thus aggregations of larger areas diminish, leading to lower values at higher pixel sizes. Plotting $L_t$ centered at an individual pixel, reveals a power-law for low pixel sizes, *i.e.* high resolutions (Fig. 2 b)

$$L_t(\mathbf{x}, t) \propto \frac{1}{p^{\gamma^*}(\mathbf{x})} \quad \text{as} \quad p \to 0. \quad (6)$$

Here $\gamma^*$ is related to the fractal dimensions of $E_t$ and $E$ [26]. When an event is triggered, the constraint (2) must be satisfied, and thus we find that the time between consecutive events at a single pixel, $\Delta t_k$, must satisfy

$$\Delta t_k = \frac{p_k C}{L_t(\mathbf{x}_k, t_k)} \propto p^{\gamma^*(\mathbf{x}_k, t_k)} \quad \text{as} \quad p \to 0. \quad (7)$$

This means that as sensor resolution increases, the time window between consecutive events at a specific pixel must decrease. This assumes a constant contrast threshold. If the contrast threshold is simultaneously increased, this effect can be counteracted but results in a loss of detail. However, in this work we do not study the effect of contrast threshold variations, and leave it for future work. We verify this relation in Fig. 2 (c), where we show histograms of $\Delta t_k$ across common event
camera resolutions. We see that as for smaller pixels, i.e. higher resolutions, the histograms shift to lower $\Delta t_k$. The mode of the distribution shifts high $\Delta t$ to low $\Delta t$ as $p$ decreases. Due to the increased event rate, small perturbations of event timestamps, caused by slow pixel response times lead to more significant effects on following events. This indicates that high-resolution event cameras are more susceptible to temporal noise effects caused by cutoff frequency and high speed motions (see Fig. 2 (c) bottom).

### 3.1 Tasks

**Image Reconstruction** We use simple event integration as a baseline for image reconstruction, directly leveraging Eq. (1). Given a high resolution image $L_{h,0}(x)$ at time $t_0$ and low resolution events $E_l = \{e_i\}_{i=1}^{N}$ we can produce a prediction of the image at $t_1$ as follows:

$$
\hat{L}_{1,h}(x) = L_{0,h}(x) + U(\Delta L_l)(x)
$$

$$
\Delta L_l(x_i) = \sum_{i=0}^{N} p_i C \delta(x_i - x)
$$

Where $\delta$ denotes the Kronecker delta and $U$ denotes the upsampling operator, which upsamples the low-resolution brightness increment to the same resolution as the high-resolution images. In what follows, we will fix the scale of the high-resolution image and vary the resolution of the events. This way, we can compare predictions across event camera scales, without changing the ground-truth image $L_{1,h}(x)$. Since this method depends on $C$ which is typically unknown we will only use it in simulation. However, due to its direct relationship to the event generation model, it serves as a good baseline for evaluating the quality of the events. In the following sections, we will refer to it as the Event Integration (EI) baseline.

**Optical Flow Estimation** Following the event generation model in Eq. (1) we see that consecutive events at the same pixel measure a brightness change of $\pm C$. We design a flow method that enforces this constraint. Let $L_l(x)$ be a low resolution image at time $t_0$ and let us focus on low resolution events triggered in a small patch and time window $\Delta T$ after $t_0$. We assume that the optical flow $v$ is approximately constant within this spatio-temporal volume. Therefore, locally the brightness at any time $t$ can be approximated with

$$
L(x_l, t) = L_l(x_l - vt)
$$

Where for simplicity we measure $t$ relative to $t_0$. By inserting Eq. (9) into the generative model in Eq. (1) and enforcing equality we have for a single event

$$
p_k(L_l(x_k - vt_k) - L_l(x_k - v(t_k - \Delta t_k))) = C
$$

Eq. (10) forces the difference of the intensities at coordinates $x_k - vt_k$ and $x_k - v(t_k - \Delta t_k)$ to be equal to $\pm C$. We denote $\Delta L_l^k(v) \doteq L_l(x_k - vt_k) - L_l(x_k - vt_k)$
\(v(t_k - \Delta t_k))\) and then stack the above equation for several events resulting in
\[
\hat{v} = \arg \min_v \| \Delta L_i(v) \odot p - 1C \|^2
\] (11)
With \(\odot\) denoting element-wise multiplication. We minimize Eq. (11) in two steps. We first solve for the optimal contrast threshold, \(\hat{C}\), with the closed form solution
\[
\hat{C} = \frac{p^T \Delta L_i(v)}{p^T p},
\] (12)
before resubstituting and minimizing the original objective for \(v\) using a first-order method. We call this method \textit{Event-based Photometric Flow} (EPF).

**Camera Pose Tracking** Camera pose tracking estimates the camera pose given a photometric depth map, as in [14]. We develop a pose tracker, based on [14] and [27]. We parametrize the camera trajectory using a cumulative B-spline \(T(t)\), parametrized by coordinates \(\xi\) and use the events, together with the photometric depth map to constrain this spline. Again, we use Eq. (1), to set up a constraint for each event, but modify Eq. (9) to fit into a pose tracking setting. We use a low resolution frame \(L_l(x)\) together with a depthmap \(D(x)\) at time \(t_0\), and low resolution events. Here, \(L_l(x)\) plays the role of reference view. The intensity at each event is found by backprojecting the event at position \(x_k\) and camera pose \(T(t_k)\) and reprojecting into the reference view.
\[
L(x, t) = L_l(x')
\]
\[
x' = \pi(X, K, T_0)
\]
\[
X = \pi^{-1}(x, Z, K, T(t))
\]
Where \(\pi\) projects and backprojects pixel coordinates according to camera matrix \(K\), pose \(T(t)\), and depth \(Z\). Since \(T\) is a spline, it can be sampled at arbitrary times. \(T_0\) stands for the camera pose and \(X\) denotes the 3-D point corresponding to the event. We assume planar scenes, to avoid the need for raytracing, and thus find a closed-form solution for the depth \(Z\) for each event. Thus for a single event
\[
p_k(L_l(x_k^r(\xi) - L_l(x_k(\xi))) = C,
\] (14)
where each point \(x_{0/1}^r\) depends on the spline \(\xi\). Stacking these residuals we get
\[
\xi = \arg \min_\xi \| \Delta L_i(\xi) \odot p - 1\hat{C} \|^2
\] (15)
which we solve using the non-linear least squares solver Ceres[1]. In what follows, we will refer to this method as \textit{Event-based Photometric Pose Tracker} (EPPT).

**4 Experiments**
First, we evaluate the introduced baselines and state-of-the-art methods on a synthetic multi-scale dataset, where we show a tradeoff between resolution, task performance, and event rate. Finally, we show similar conclusions for real data.
Are High-Resolution Event Cameras Really Needed?

Fig. 3. Each plot compares the event rate and task-performance on the rocks sequence in different settings. Each point corresponds to the median task performance and event rate of a specific setting. Equally colored points correspond sequences recorded at the same resolution, while connected dots correspond to the same camera trajectory, traversed at different speeds (the more right, the faster the sequence, resulting in a higher event rate). Finally, each plot is separated into three gray bands which correspond to different cutoff frequency settings. These settings correspond with ideal ($f_{cutoff} = \infty$), daytime ($f_{cutoff} = 200$Hz) and nighttime ($f_{cutoff} = 50$Hz) conditions. As $f_{cutoff}$ decreases, so does task performance. For each task, condition (ideal/daytime/nighttime), and camera velocity, the ideal camera is the one that is closest to the “goal”, maximizing task performance, while minimizing event rate.

4.1 Synthetic Data:

We use the event camera simulator in [22] to generate a synthetic multi-scale event camera dataset. Each sequence features a random camera trajectory in a planar scene with varying textures. We select resolutions $r_i = (w, w)$ with $w \in \{128, 240, 346, 640, 1280\}$, to ensure the same field of view across resolutions. The selected widths were selected following commonly used resolutions in Tab. 1. We use the event camera simulator in [22] to generate events for each resolution. In addition to varying the resolution, we also select cutoff frequency $f_{cutoff} \in \{\infty, 200, 50\}$ Hz to simulate ideal, daytime, and nighttime settings. These values represent realistic settings and were calibrated in [22]. We also vary the camera speed along a given trajectory by scaling it by a factor $s \in \{1, 1.3, 2\}$. We choose a realistic contrast threshold of 0.2 for all experiments.

Image Reconstruction For the image reconstruction task, we use the method from Sec. 3.1. We use a 50 ms sliding window of events at resolution $r \times r$ with $r \leq 1280$ together with the left keyframe at resolution 1280 × 1280 to generate predictions of the right keyframe according to Eq.(8). The window is reduced appropriately for higher camera speeds. We compute the peak signal-to-noise ratio (PSNR) between the predicted and ground truth right keyframe and the event rate, reporting their median across all windows in Fig. 3 (a). We summarized these results in Tab. 2 (left), and report the full table in the appendix.

Discussion In ideal settings ($f = \infty$), the highest resolution event cameras achieve the highest performance, 3.18 dB on average higher than the next highest resolutions, but require a roughly 8 times higher event rate. The same holds
for daytime settings ($f = 200$ Hz), but this time with a significantly lower margin of 0.86 dB. Interestingly, in this setting, we observe that task performance degrades more at higher event camera resolution than at lower resolution. The event rate is also reduced, but there remains an approximate factor of 6 between resolutions $1280 \times 1280$ and $640 \times 640$. Finally, in nighttime settings, ($f = 50$ Hz) we observe a peak at resolutions $240 \times 240$, $346 \times 346$ and $640 \times 640$. This shows that for image reconstruction, high-resolution cameras are outperformed by lower resolution cameras, especially in nighttime conditions.

**Comparison with other Methods:** We validate these conclusions, by evaluating the image reconstruction performance of the learning-based method E2VID\[32\] and model-based method HF[35], following the protocol described in [32]. We report the image reconstruction performance of the two methods in Tab. 3 (left). Similar to Tab. 2, for both methods lower resolution cameras become more attractive in nighttime settings.

**Optical Flow** Here we use the method introduced in Sec. 3.1. We extract 20 features from a target image and reproject them into frames spaced at 0.5 seconds in time. For each frame, we extract a spatio-temporal volume of events around each feature of size $61 \times 61$ and time window $\Delta T = 50$ ms. We adapt the size and length of this window based on the resolution and speed of the camera,
Are High-Resolution Event Cameras Really Needed? 11

Table 3. Comparison on the carpet sequence of state-of-the-art methods[32] and [35] on image reconstruction and [15] and [18] on optical flow estimation.

| Task Metric | Image Reconstruction | Optical Flow |
|-------------|----------------------|--------------|
| Method      | Image Reconstruction | Optical Flow |
|             | PSNR [dB]↑           | RNEPE [px]↓  |
| f_{cutoff} [Hz] | ∞ 200 50 | ∞ 200 50 |
| 128 × 128  | 11.53 11.54 11.39 11.14 11.15 10.97 | 5.06 4.89 4.76 2.44 2.32 2.56 |
| 240 × 240  | 12.19 12.17 11.63 11.57 11.58 11.14 | 2.51 2.59 2.57 1.01 1.10 1.59 |
| 346 × 346  | 12.25 12.21 11.36 11.61 11.60 10.98 | 1.74 1.69 1.72 0.94 0.93 1.24 |
| 640 × 640  | 12.31 12.03 10.61 11.65 11.55 10.43 | 0.71 0.67 0.69 0.79 0.84 0.86 |
| 1280 × 1280| 13.38 12.33 10.05 12.63 12.04 9.93 | 0.32 0.25 0.37 0.61 0.63 0.68 |

Table 4. Task performance of optical flow method CM [15] and our Event-based Photometric Pose Tracker (EPPT), in nighttime conditions (f_{cutoff} = 50Hz) depending on the camera velocity. While PPT with high resolution cameras degrades already at 1.8 m/s, CM degrades at much higher speeds of 9m/s.

Discussion As before, Fig. 3 (b) suggests, a trade-off between accuracy and event camera resolution, especially in nighttime and daytime conditions, and at high speeds. Here we observe that already in clean conditions and at high speeds, lower resolution event cameras perform best, and with a significantly lower event rate. This conclusion is supported by Tab. 2 (middle), where lower resolution camera outperform higher resolution ones by between 0.24-2.09 px. On room, the highest resolution camera outperforms others by a small margin of 0.09, but using the stated values for the highest resolution and lowest camera speed. We select L_t at the same resolution as the events and located at the beginning of the spatio-temporal window, and estimate optical flow by minimizing the objective in Eq. (10), using a first-order method. We initialize the flow to the ground truth flow and perturb it with Gaussian noise with σ=3. At lower resolutions we reduce σ to create comparable conditions. We compare estimated and ground truth flow using a resolution independent normalized end-point-error (RNEPE)

\[ \text{RNEPE} = \frac{H_{\text{max}}}{H} \| \hat{v} - v \|_2. \]  

Where \( \hat{v} \) and \( v \) are estimated and ground truth optical flow respectively, \( H \) is the sensor height and \( H_{\text{max}} = 1280 \). We report the RNEPE in Fig. 2 (b), summarize results in Tab. 2 (middle), and include the full table in the appendix.

Discussion As before, Fig. 3 (b) suggests, a trade-off between accuracy and event camera resolution, especially in nighttime and daytime conditions, and at high speeds. Here we observe that already in clean conditions and at high speeds, lower resolution event cameras perform best, and with a significantly lower event rate. This conclusion is supported by Tab. 2 (middle), where lower resolution camera outperform higher resolution ones by between 0.24-2.09 px. On room, the highest resolution camera outperforms others by a small margin of 0.09, but
it features a roughly three times higher event rate (Fig. 3 (b)).

**Comparison with other Methods** Here we compare task performance for model-based method Contrast Maximization [15] and learning-based method E-RAFT [18]. Surprisingly, we observe that CM is more robust than EPF at high camera speeds and in nighttime conditions (Tab 3) (right). However, at higher speeds of 9 m/s, the $640 \times 640$ camera outperforms $1280 \times 1280$ by 0.2 px (Tab. 4). Interestingly, this suggests that estimators which optimize contrast instead of photometric consistency are more robust, especially at high speeds and in low light. Finally, E-RAFT always benefits from high resolution and does not degrade harshly in noisier conditions. Since E-RAFT was trained on real data both in daytime and nighttime conditions, it most likely generalizes to these conditions. This indicates another important lesson that can be drawn from these experiments: We may overcome the limitations of noise at high resolutions, by adopting a learning-base approach, and training on noisy night-time data, robustifying the network against these error sources. Note, that noisy training-data is the key here, since learning-based method E2VID exhibits the opposite trend, and in fact E2VID is trained with synthetic noise-less event data.

**Camera Pose Tracking** Finally, we evaluate our pose tracker EPPT (Sec. 3.1). In steps of 0.5 seconds, we initialize a spline with 10 control poses spaced at 2 ms with ground truth and then perturb the translation component with Gaussian noise with $\sigma = 20$cm. We minimize Eq. (15) to recover the original poses, repeating this process 20 times. At each time we compute the mean translation error between ground truth and estimated spline. We compute the median error over samples which we report in Fig. 3 (c), summarize in Tab. 2 (right). The full table is in the appendix. We omit room, as it is not a planar scene.

**Discussion** Similar to previous tasks, pose tracking degrades in nighttime conditions and at high speeds (Fig.3). Performance degrades strongly at high resolution, especially in nighttime conditions, where the error increases from 0.953 to 16.4 mm for resolutions $1280 \times 1280$ on the rocks dataset. High resolutions perform well on the carpet sequence, even in nighttime conditions, but break down as soon as high speeds are introduced (Tab. 4).

**4.2 Real Data**

Here we show how the conclusions drawn from the experiments on synthetic data transfer to the real world. To this end, we propose an experimental setup to record event camera data at different resolutions, observing the same scene. We evaluate image reconstruction, optical flow estimation, and pose tracking on the recorded dataset.

We use a Prophesee Gen4 event camera[11] with a resolution of $1280 \times 720$ installed on a motorized linear slider (Fig. 4 a) in front of a planar wall, featuring a picture of a template image with ArUco markers[16]. With this setup, we record
events at three depths $d \in \{20, 40, 60\}$ cm from the wall, and at three different camera speeds $V \in \{0.2, 0.4, 0.6\}$ m/s. We leverage the slider and markers to recover the camera pose with respect to the printed image. We use this pose to crop events to a common field of view, resulting in events at different resolutions (Fig. 4 (b), (c)) and speeds (Fig. 4 (d)). More details on how the camera pose was recovered are provided in the appendix.

**Image Reconstruction** As before, we generate image reconstructions using E2VID [32] and HF [35]. To ensure equal conditions across resolutions we pass 30 ms events windows to each method, adjusting the window size to the camera speed. To generate ground truth, we map the template image into the current camera frame, according to the current camera pose. We report the results in Tab. 5 (left). As before, at higher speeds higher resolution cameras degrade more than lower resolution cameras. Indeed, already at moderate speeds of 0.4 m/s, a resolution of $640 \times 360$ outperforms higher resolutions by an average of 0.2 dB.

Fig. 5 visualizes image reconstruction patches by [32] in different settings. At low resolutions, reconstructions show common blurring artefacts (c). By contrast at high resolutions and speeds, reconstructions show ghosting (d), which cause significant performance degradations at high speeds. This is caused by slow cutoff frequencies in the event pixels, which lead to slower response times.

**Optical Flow and Camera Pose Tracking** We evaluate the model-based method EPF, described in Sec. 3.1, CM [15] and E-RAFT [18] on patch-based
Table 5. Comparison of image reconstruction, optical flow estimation and camera pose tracking performance on the forest, pebbles and split_rocks sequences. While at slow speeds, the highest resolutions are almost always best, at higher speeds lower resolutions outperform the highest. We see that across tasks a resolution of 640 × 360 offers better task performance than the full resolution.

optical flow. For this, we extract corners from the template image and reproject these corners into the current event camera frame. By taking differences between projections at consecutive timestamps, we can generate ground truth flow. We use this flow to evaluate the methods described. We generate volumes of 30 ms and patch size 61 and report the resolution normalized RNEPE in Tab. 5 (middle). Similarly, for camera tracking we follow the same procedure as Sec. 4.1, but this time perturb the splines with \( \sigma = 5 \text{cm} \). We report the median position error in mm in Tab. 5 (right). For both EPF and EPPT we use image reconstructions from [32] as reference images. Similar conclusions hold as for the experiments in simulation. For speeds above 0.4 m/s, pose tracking and optical flow degrade at the highest resolution. In both cases, a resolution of 640 × 360 yields better results and lower event rates. While pose tracking performance increases by 31 mm, optical flow performance increases by 0.1 px for EPF. As before, CM and E-RAFT are more robust to high speeds, performing best at the highest resolution. Again this indicates that adopting CM-based objectives can in fact significantly improve the robustness to high speed, especially at high resolution. Moreover, adopting a learning-based approach, which is trained on noisy night-time events also helps.

5 Conclusion

As event cameras increasingly become the sensor of choice in challenging applications, their development will inexorably trend toward higher resolution sensors. However, higher resolution sensors significantly increase the required data bandwidth, while burdening downstream tasks. In this work, we showed, that in addition to these challenges, higher resolution cameras are also more sensitive to temporal effects such as slow pixel response times. We showed across the tasks of image reconstruction, optical flow estimation and camera pose tracking that high-resolution event cameras do not always give the best task performance. Especially when used in challenging high-speed scenarios and in low light, lower-resolution sensors often show a better performance while using lower bandwidth.

It is thus important that additional effort is made to solve these challenges before we can progress to higher resolution sensors. We believe that this work will act as a useful guide for future trends in event camera development.

6 Acknowledgment

This work was supported by Huawei, and as a part of NCCR Robotics, a National Centre of Competence in Research, funded by the Swiss National Science Foundation (grant number 51NF40_185543).

7 Appendix

In Sec. 7.1 we add additional details to explain how the real-world multi-scale event dataset was collected. Tables and figures in the main manuscript will have an "M-" prefix. In Sec. 7.2 we report the full table, which was used to generate Fig. M-3, split by tasks, supplementing Tab. M-2 in the main manuscript.

7.1 Generation of the Multi-scale Dataset

We install a Prophesee Gen4 event camera on a motorized linear slider (Fig. M-4 (a)), placed in a front-parallel configuration to a wall, featuring a printed image with ArUco markers[16] surrounding it (Fig.6 (b)). We leverage the markers and the linear slider to recover the camera pose with respect to the bottom left corners of the printed page, which we designate as the origin.

We do this in three steps. First we convert events to using E2VID[32] to generate sharp grayscale images which we use to extract the marker corner positions (Fig.6 (c)). The 3-D position of the corners are fully known and fixed, and thus we solve a pose-only global bundle adjustment problem that minimizes the
reprojection error between 2-D detections and known landmark positions. The full problem we solve is written as

\[
\{T_i\} = \arg \min_{\{T_i\}} \sum_{i=1}^{N} \sum_{k=1}^{M} v_{ik} \|x_{ik} - \pi(X_k, K, T_i)\|_2^2,
\]  

(17)

where \(T_i\) denote the camera poses with \(i = 1, ..., N\), \(X_k\) denote the fixed landmark positions with \(k = 1, ..., M\) and \(K\) captures the camera intrinsics. The parameter \(v_{ik} = 1\) if the landmark \(k\) is in view \(i\) and 0 otherwise. Finally, \(x_{ik}\) denote the detection of landmark \(k\) in view \(i\). The linear slider additionally provides the \(x\)-coordinate along the path, which is used to reduce the number of unknowns. We thus reduce the number of unknowns to 9 parameters. Six describe the relative orientation and translation between the linear slider base at position \(x = 0\) and the last three describe the relative rotation of the camera frame with respect to the linear slider frame. We collect measurements for the bundle adjustment problem by following the camera speed profile in Fig.6 (d), which drives the slider back and forth a total of three times, at increasing speed. This is done because at high speeds image reconstruction becomes blurry and therefore few detections are made. By inserting a slow-speed segment between each right-moving segment, we ensure that accurate detections can still be made, to constrain bundle adjustment. This allows us to have accurate poses even, where there are no detections.

**Ground Truth Generation:** Finally, we use the recovered poses for ground truth generation for all three tasks: for image reconstructions, a template image (Fig. 6 (b)) is used, to which all image reconstructions are mapped using homographies. By mapping images at all speeds and resolutions to the same template, we ensure an equal evaluation. For optical flow we use the poses to project points into poses spaced at \(\Delta T\), resulting in accurate flow vectors. For camera pose tracking we use the poses themselves.

### 7.2 Additional Results in Simulation

Here we summarize additional results in simulation, which for brevity were omitted in Tab. M-2.

**Image Reconstruction** In Tab. 6 we report the full table, comparing the event integration method, described in Sec. M-3.1 on our synthetic multi-scale dataset. It completes the results in Tab. M-2 (left) and was used to generate Fig. M-3 (a). We report scores for different camera resolutions, cutoff frequencies, and speeds. Moreover, we show in Tab. 7 the full table comparing the image reconstruction methods E2VID [32] and HF [35] on the carpet sequence. Both tables, show, that the conclusions hold across datasets and methods that higher resolution cameras are outperformed by lower resolution ones, especially in nighttime conditions and at high speeds.
Optical Flow Estimation Similar to image reconstruction, we show the complete table which complements Tab. M-2 (middle) and Fig. M-3 (b). In Tab. 8 we show the optical flow method EPF on all synthetic datasets and for different camera resolutions, speeds and cutoff frequencies. Tab. 9 shows the performance of state-of-the-art flow methods Contrast Maximization (CM)[15] and E-RAFT[18] on the carpet sequence. Tab. 8 indicates the same conclusion of Tab. M-2 (left). The faster the camera and the lower the cutoff frequency (i.e. in nighttime conditions), the worse high-resolution cameras perform. Indeed, already in daytime conditions, sensors below a resolution of $346 \times 346$ show better performance than those at a resolution of $1280 \times 1280$. Tab. 9 reiterates the conclusion made in the main text: For EPF fast camera motion and nighttime conditions lead to significant degradation of the optical flow method at high resolutions. Instead, using lower resolution sensors is better. In nighttime conditions, all resolutions perform poorly, leading to noisy values. CM remains stable at high resolutions, high speeds, and low cutoff frequencies. However, at speeds of 9 m/s in night-time conditions, CM at high resolution is outperformed by a sensor of resolution $640 \times 640$ by 0.12 px. E-RAFT at the highest resolution remains stable at all speeds but shows significant degradation when going to high speeds and nighttime conditions. We believe that this is because E-RAFT trains on low-light events and can thus generalize better to this setting.

Camera Pose Tracking Also for camera pose tracking, we show the complete table, complementing Tab. M-2 (right) and Fig. M-3 (c). Here we also report the position error for three different camera speeds. We see that especially in nighttime conditions and at high speeds, pose tracking using high-resolution sensors is outperformed by higher resolution sensors.

References

1. Agarwal, A., Mierle, K., Others: Ceres solver. http://ceres-solver.org
2. Baghaei Naeini, F., AlAli, A.M., Al-Husari, R., Rigi, A., Al-Sharman, M.K., Makris, D., Zweiri, Y.: A novel dynamic-vision-based approach for tactile sensing applications. IEEE Transactions on Instrumentation and Measurement 69(5), 1881–1893 (2020). https://doi.org/10.1109/TIM.2019.2919354
3. Banerjee, S., Wang, Z.W., Chopp, H.H., Cossairt, O., Katsaggelos, A.K.: Lossy event compression based on image-derived quad trees and poisson disk sampling. In: IEEE ICIP. pp. 2154–2158 (2021). https://doi.org/10.1109/ICIP42928.2021.9506546
4. Bardow, P., Davison, A.J., Leutenegger, S.: Simultaneous optical flow and intensity estimation from an event camera. In: CVPR. pp. 884–892 (2016). https://doi.org/10.1109/CVPR.2016.102
5. Brandli, C., Berner, R., Yang, M., Liu, S.C., Delbruck, T.: A 240x180 130dB 3\(\mu\)s latency global shutter spatiotemporal vision sensor. IEEE JSSC 49(10), 2333–2341 (2014). https://doi.org/10.1109/JSSC.2014.2342715
6. Censi, A., Mueller, E., Frazzoli, E., Soatto, S.: A power-performance approach to comparing sensor families, with application to comparing neuromorphic to traditional vision sensors. In: ICRA (2015)
### Table 6. Effect of resolution on median image reconstruction accuracy (PSNR [dB]) in ideal ($f = \infty$ Hz), daytime ($f = 200$ Hz) and nighttime ($f = 50$ Hz) settings, and at three camera speeds. While highest resolution event cameras achieve the best task performance in ideal and clean conditions, they do this with only small margins (0.1 dB), and at the cost of significant event-rates. In noisy conditions, lower resolution event cameras such as the 346 outperform the highest resolution.

| Method  | Setting | Velocity [m/s] | 0.9 | 1.2 | 1.8 | 0.9 | 1.2 | 1.8 | 0.9 | 1.2 | 1.8 |
|---------|---------|----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| EI (ours) | 346 | 1280 | 15.389 | 15.39 | 15.389 | 15.343 | 15.314 | 15.233 | 14.892 | 15.141 | 14.627 |
| E2VID [32] | 346 | 1280 | 12.375 | 12.375 | 12.375 | 12.375 | 12.375 | 12.375 | 12.375 | 12.375 | 12.375 |
| HF [35] | 346 | 1280 | 11.605 | 11.605 | 11.605 | 11.541 | 11.532 | 11.501 | 11.385 | 11.268 | 10.972 |

### Table 7. Image reconstruction evaluation on the carpet sequence, at varying camera speeds and for model-based method HF [35] and learning-based method E2VID [32]. In daytime and nighttime settings, increasing camera speed leads to a significant drop in performance, especially for high-resolution sensors. This is why, in these settings, lower resolutions like $346 \times 346$, $240 \times 240$ and $128 \times 128$ have higher performance, while having lower event-rate.
| Dataset | Setting | \(f_{\text{cutoff}}\) [Hz] | \(\infty\) | 200 | 50 |
|---------|---------|-----------------|-------|-----|-----|
|         |         | Velocity        | 1.2   | 1.6 | 2.4 |
|         |         | 128 \times 128  | 2.36  | 2.31| 2.62 |
|         |         | 240 \times 240  | 1.26  | 1.25| 1.37 |
|         |         | 346 \times 346  | 1.03  | 1.03| 1.04 |
|         |         | 640 \times 640  | 0.96  | 0.92| 0.93 |
|         |         | 1280 \times 1280| 1.33  | 1.34| 1.37 |
|         |         | Carpet          | 0.7   | 1.0 | 1.4 |
|         |         | 128 \times 128  | 2.09  | 1.89| 2.11 |
|         |         | 240 \times 240  | 0.91  | 0.93| 0.93 |
|         |         | 346 \times 346  | 0.58  | 0.62| 0.61 |
|         |         | 640 \times 640  | 0.36  | 0.37| 0.36 |
|         |         | 1280 \times 1280| 0.48  | 0.45| 0.46 |
|         |         | Rocks           | 0.7   | 1.0 | 1.4 |
|         |         | 128 \times 128  | 1.90  | 1.56| 1.64 |
|         |         | 240 \times 240  | 0.74  | 0.79| 0.76 |
|         |         | 346 \times 346  | 0.53  | 0.47| 0.50 |
|         |         | 640 \times 640  | 0.38  | 0.37| 0.34 |
|         |         | 1280 \times 1280| 0.42  | 0.41| 0.42 |

|         |         | Nighttime       | 1.2   | 1.6 | 2.4 |
|         |         | 128 \times 128  | 2.21  | 3.12| 4.35 |
|         |         | 240 \times 240  | 1.76  | 2.39| 3.97 |
|         |         | 346 \times 346  | 1.84  | 2.72| 4.09 |
|         |         | 640 \times 640  | 1.70  | 2.40| 4.21 |
|         |         | 1280 \times 1280| 1.95  | 2.68| 3.98 |
|         |         | Carpet          | 0.7   | 1.0 | 1.4 |
|         |         | 128 \times 128  | 1.57  | 2.19| 2.19 |
|         |         | 240 \times 240  | 0.75  | 0.95| 1.72 |
|         |         | 346 \times 346  | 0.65  | 1.01| 1.82 |
|         |         | 640 \times 640  | 0.85  | 1.26| 1.91 |
|         |         | 1280 \times 1280| 0.94  | 1.32| 2.06 |
|         |         | Rocks           | 0.7   | 1.0 | 1.4 |
|         |         | 128 \times 128  | 2.10  | 2.36| 2.67 |
|         |         | 240 \times 240  | 1.68  | 2.03| 2.60 |
|         |         | 346 \times 346  | 1.74  | 2.13| 2.85 |
|         |         | 640 \times 640  | 1.47  | 1.92| 2.71 |
|         |         | 1280 \times 1280| 1.43  | 1.93| 3.04 |

**Table 8.** Optical flow accuracy of our Event-based Photometric Flow (EPF) following Eq. M-3.1. We report the resolution normalized end-point-error (RNEPE). All methods prefer higher resolution cameras in ideal and clean settings at almost all speeds. In noisy settings, EPF using high resolution event-cameras degrades, leading to lower resolution cameras being more accurate.
Table 9. Optical flow evaluation on the carpet sequence, at varying camera speeds, and for model-based method CM [15] and learning-based method E-RAFT[18]. In daytime and nighttime settings, higher speeds significantly reduce performance for EPF, especially for high resolution cameras, making low resolution cameras a better choice. Nonetheless, both [15] and [18] remain robust, but with diminishing returns at higher resolutions.

Table 10. Effect of event camera resolution on median camera position accuracy (mm) with changing speed, and setting.
7. Chen, S., Guo, M.: Live demonstration: CeleX-V: A 1M pixel multi-mode event-based sensor. In: CVPRW (2019)
8. Cohen, G., Afshar, S., Moreale, B., Bessell, T., Wabnitz, A., Rutten, M., van Schaik, A.: Event-based sensing for space situational awareness. The Journal of the Astronautical Sciences pp. 1–17 (2019)
9. Delbruck, T., Graca, R., Paluch, M.: Feedback control of event cameras. In: CVPRW (2021)
10. Falanga, D., Kleber, K., Scaramuzza, D.: Dynamic obstacle avoidance for quadrotors with event cameras. Science Robotics 5(40), eaa9712 (2020). https://doi.org/10.1126/scirobotics.aaz9712, https://www.science.org/doi/abs/10.1126/scirobotics.aaz9712
11. Finateu, T., Niwa, A., Matolin, D., Tsuchimoto, K., Mascheroni, A., Reynaud, E., Mostafalu, P., Brady, F., Chotard, L., LeGoff, F., Takahashi, H., Wakabayashi, H., Oike, Y., Posch, C.: A 1280x720 back-illuminated stacked temporal contrast event-based vision sensor with 4.86µm pixels, 1.066geps readout, programmable event-rate controller and compressive data-formatting pipeline. In: IEEE Intl. Solid-State Circuits Conf. (ISSCC) (2020)
12. Gallego, G., Delbruck, T., Orchard, G., Bartolozzi, C., Taba, B., Censi, A., Leutenegger, S., Davison, A., Conradt, J., Daniilidis, K., Scaramuzza, D.: Event-based vision: A survey. IEEE T-PAMI. (2020). https://doi.org/10.1109/TPAMI.2020.3008413
13. Gallego, G., Forster, C., Mueggler, E., Scaramuzza, D.: Event-based camera pose tracking using a generative event model (2015), arXiv:1510.01972
14. Gallego, G., Lund, J.E.A., Mueggler, E., Rebecq, H., Delbruck, T., Scaramuzza, D.: Event-based, 6-DOF camera tracking from photometric depth maps. IEEE T-PAMI. 40(10), 2402–2412 (Oct 2018). https://doi.org/10.1109/TPAMI.2017.2769655
15. Gallego, G., Rebecq, H., Scaramuzza, D.: A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation. In: CVPR. pp. 3867–3876 (2018). https://doi.org/10.1109/CVPR.2018.00407
16. Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F., Marín-Jiménez, M.: Automatic generation and detection of highly reliable fiducial markers under occlusion. Pattern Recognition 47(6), 2280–2292 (2014). https://doi.org/10.1016/j.patcog.2014.01.005, https://www.sciencedirect.com/science/article/pii/S0031320314000235
17. Gehrig, M., Aarents, W., Gehrig, D., Scaramuzza, D.: Dsec: A stereo event camera dataset for driving scenarios. In: IEEE RA-L (December 2021)
18. Gehrig, M., Millhäuser, M., Gehrig, D., Scaramuzza, D.: Dense optical flow from event cameras. In: IEEE Int. Conf. 3D Vis. (3DV) (2021)
19. Glover, A., Bartolozzi, C.: Robust visual tracking with a freely-moving event camera. In: IROS. pp. 3769–3776 (2017). https://doi.org/10.1109/IROS.2017.8206226
20. Guo, M., Huang, J., Chen, S.: Live demonstration: A 768 x 640 pixels 200meps dynamic vision sensor. In: IEEE Int. Symp. Circuits Syst. (ISCAS) (2017). https://doi.org/10.1109/ISCAS.2017.8050397
21. Hagenmaars, J.J., Paredes-Vallés, F., Bohté, S.M., de Croon, G.C.H.E.: Evolved neuromorphic control for high speed divergence-based landings of mavs. IEEE Robotics and Automation Letters 5(4), 6239–6246 (2020). https://doi.org/10.1109/LRA.2020.3012129
22. Hu, Y., Liu, S.C., Delbruck, T.: v2e from video frames to realistic dvs events. In: CVPRW (2021)
23. Insightness Event-based Sensor Modules: http://www.insightness.com/technology/ (2020)
24. Khan, N., Iqbal, K., Martini, M.G.: Lossless compression of data from static and mobile dynamic vision sensors—performance and trade-offs. IEEE Access 8, 103149–103163 (2020). https://doi.org/10.1109/ACCESS.2020.2996661
25. Lichtsteiner, P., Posch, C., Delbruck, T.: A 128×128 120 dB 15 µs latency asynchronous temporal contrast vision sensor. IEEE JSSC 43(2), 566–576 (2008). https://doi.org/10.1109/JSSC.2007.914337
26. Messikommer, N.A., Gehrig, D., Loquercio, A., Scaramuzza, D.: Event-based asynchronous sparse convolutional networks. In: ECCV (2020)
27. Mueggler, E., Gallego, G., Rebecq, H., Scaramuzza, D.: Continuous-time visual-inertial odometry for event cameras. IEEE TRO 34(6), 1425–1440 (Dec 2018). https://doi.org/10.1109/tro.2018.2858287
28. Muglikar, M., Gallego, G., Scaramuzza, D.: ESL: Event-based structure light. In: IEEE International Conference on 3D Vision.(3DV) (December 2021)
29. Muglikar, M., Moews, D., Scaramuzza, D.: Event-guided depth sensing. In: IEEE International Conference on 3D Vision.(3DV) (December 2021)
30. Posch, C., Matolin, D., Wohlenannt, R.: A QVGA 143 dB dynamic range frame-free PWM image sensor with lossless pixel-level video compression and time-domain CDS. IEEE JSSC 46(1), 259–275 (Jan 2011). https://doi.org/10.1109/JSSC.2010.2085952
31. Prophesee Evaluation Kits: https://www.prophesee.ai/event-based-evk/ (2020)
32. Rebecq, H., Ranftl, R., Kolstun, V., Scaramuzza, D.: High speed and high dynamic range video with an event camera. IEEE T-PAMI. (2019). https://doi.org/10.1109/TPAMI.2019.2963386
33. Ryu, H.E.: Industrial DVS design; key features and applications. http://rpg.ifi.uzh.ch/docs/CVPR19workshop/CVPRW19_Eric_Ryu_Samsung.pdf
34. Sanket, N., Parameshwarar, C.M., Singh, C., Kuruttukulam, A.V., Fermüller, C., Scaramuzza, D., Aloimonos, Y.: Evdodgenet: Deep dynamic obstacle dodging with event cameras. pp. 10651–10657 (05 2020). https://doi.org/10.1109/ICRA40945.2020.9196877
35. Scheerlinck, C., Barnes, N., Mahony, R.: Continuous-time intensity estimation using event cameras. In: ACCV (2018)
36. Sironi, A., Brambilla, M., Bourdis, N., Lagorce, X., Benosman, R.: HATS: Histograms of averaged time surfaces for robust event-based object classification. In: CVPR. pp. 1731–1740 (2018)
37. Son, B., Suh, Y., Kim, S., Jung, H., Kim, J.S., Shin, C., Park, K., Lee, K., Park, J., Woo, J., Roh, Y., Lee, H., Wang, Y., Ovsiannikov, I., Ryu, H.: A 640x480 dynamic vision sensor with a 9µm pixel and 300Meps address-event representation. In: IEEE Intl. Solid-State Circuits Conf. (ISSCC) (2017). https://doi.org/10.1109/ISSCC.2017.7870263
38. Sugimoto Dimitrova, R., Gehrig, M., Brescianini, D., Scaramuzza, D.: Towards low-latency high-bandwidth control of quadrotors using event cameras. In: ICRA (2020)
39. Suh, Y., Choi, S., Ito, M., Kim, J., Lee, Y., Seo, J., Jung, H., Yeo, D.H., Namgung, S., Bong, J., seok Kim, J., Park, P.K.J., Kim, J., Ryu, H., Park, Y.: A 1280x960 Dynamic Vision Sensor with a 4.95-µm pixel pitch and motion artifact minimization. In: IEEE Int. Symp. Circuits Syst. (ISCAS) (2020)
40. Sun, S., Cioffi, G., de Visser, C., Scaramuzza, D.: Autonomous quadrotor flight despite rotor failure with onboard vision sensors: Frames vs.
Are High-Resolution Event Cameras Really Needed? 23

events. IEEE Robotics and Automation Letters 6(2), 580–587 (2021). https://doi.org/10.1109/LRA.2020.3048875

41. Taunyazov, T., Sng, W., Lim, B., See, H.H., Kuan, J., Ansari, A.F., Tee, B., Soh, H.: Event-Driven Visual-Tactile Sensing and Learning for Robots. In: Proceedings of Robotics: Science and Systems. Corvalis, Oregon, USA (July 2020). https://doi.org/10.15607/RSS.2020.XVI.020

42. Tulyakov, S., Gehrig, D., Georgoulis, S., Erbach, J., Gehrig, M., Li, Y., Scaramuzza, D.: Time lens: Event-based video frame interpolation. In: CVPR. pp. 16155–16164 (2021)

43. Vitale, A., Renner, A., Nauer, C., Scaramuzza, D., Sandamirskaya, Y.: Event-driven vision and control for uavs on a neuromorphic chip. ICRA (2021)

44. Zhang, X., Liao, W., Yu, L., Yang, W., Xia, G.S.: Event-based synthetic aperture imaging with a hybrid network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 14235–14244 (June 2021)

45. Zhu, A.Z., Yuan, L., Chaney, K., Daniilidis, K.: EV-FlowNet: Self-supervised optical flow estimation for event-based cameras. In: Robotics: Science and Systems (RSS) (2018). https://doi.org/10.15607/RSS.2018.XIV.062