Events-to-Video: Bringing Modern Computer Vision to Event Cameras

Henri Rebecq † René Ranftl ‡ Vladlen Koltun ‡ Davide Scaramuzza †

Abstract

Event cameras are novel sensors that report brightness changes in the form of asynchronous "events" instead of intensity frames. They have significant advantages over conventional cameras: high temporal resolution, high dynamic range, and no motion blur. Since the output of event cameras is fundamentally different from conventional cameras, it is commonly accepted that they require the development of specialized algorithms to accommodate the particular nature of events. In this work, we take a different view and propose to apply existing, mature computer vision techniques to videos reconstructed from event data. We propose a novel recurrent network to reconstruct videos from a stream of events, and train it on a large amount of simulated event data. Our experiments show that our approach surpasses state-of-the-art reconstruction methods by a large margin ($> 20\%$) in terms of image quality. We further apply off-the-shelf computer vision algorithms to videos reconstructed from event data on tasks such as object classification and visual-inertial odometry, and show that this strategy consistently outperforms algorithms that were specifically designed for event data. We believe that our approach opens the door to bringing the outstanding properties of event cameras to an entirely new range of tasks. A video of the experiments is available at https://youtu.be/IdYrC4cUO0I

1. Introduction

Event cameras are bio-inspired vision sensors that work radically differently from conventional cameras. Instead of capturing intensity images at a fixed rate, event cameras measure *changes* of intensity asynchronously at the time they occur. This results in a stream of *events*, which encode the time, location, and polarity (sign) of brightness changes (Fig. 2). Event cameras such as the Dynamic Vision Sensor (DVS) [24] possess outstanding properties when compared to conventional cameras. They have a very high dynamic range (140 dB versus 60 dB), do not suffer from motion

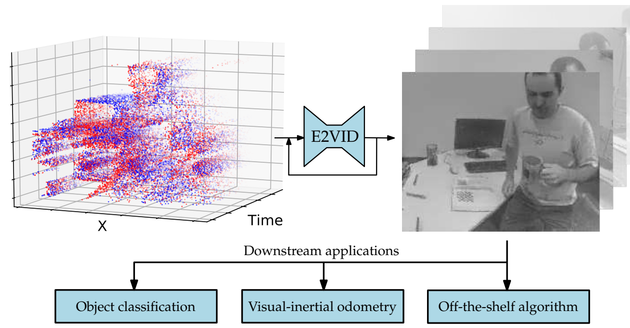


Figure 1. Our network converts a spatio-temporal stream of events (left) into a high-quality video (right). This enables direct application of off-the-shelf computer vision algorithms such as object classification (Section 5.1) and visual-inertial odometry (Section 5.2), yielding state-of-the-art results with event data in each case. This figure shows an actual output sample from our method, operating on a real event sequence from a publicly available dataset [30].

blur, and provide measurements with a latency as low as one microsecond. Event cameras thus provide a viable alternative, or complementary, sensor in conditions that are challenging for conventional cameras.

However, since the output of an event camera is an asynchronous stream of events (a representation that is fundamentally different from natural images), existing computer vision techniques cannot be directly applied to this data. As a consequence, custom algorithms need to be specifically tailored to leverage event data. Such specialized algorithms have demonstrated impressive performance in applications ranging from low-level vision tasks, such as visual odometry [18, 39, 53, 38, 41], feature tracking [20, 52, 15] and optical flow [5, 3, 47, 55, 46], to high-level tasks such as object classification [35, 21, 45] and gesture recognition [2].

While some works [9, 10, 5, 29, 18, 14] focused on exploiting the low latency of the sensor by processing the data event-by-event, significant progress has been made by mapping a set of events into an image-like 2D representation prior to processing. Examples are the integration of events on the image plane [39, 26, 27] as well as time surfaces [21, 45, 55, 51, 55]. However, neither event images nor time surfaces are natural images, meaning that much of the existing computer vision toolbox cannot be applied effectively. Most importantly, deep networks trained on real image data

† Dept. Informatics, Univ. of Zurich and Dept. Neuroinformatics, Univ. of Zurich and ETH Zurich
‡ Intel Labs

cannot be directly transferred to these representations.

In this paper, we propose to establish a bridge between vision with event cameras and conventional computer vision. Specifically, we learn how to reconstruct natural videos from a stream of events ("events-to-video"), *i.e*. we learn a mapping between a stream of events and a stream of images (Fig. 1). This allows us to apply off-the-shelf computer vision techniques to event cameras.

Our work differs from previous image reconstruction approaches [3, 31, 43] in two essential ways. First, instead of embedding handcrafted smoothness priors into our reconstruction framework, we directly learn video reconstruction from events using a large amount of simulated event data. Second, instead of focusing mainly on the quality of the reconstructions, we build our approach with the goal of applying standard computer vision techniques to the reconstructions. To this end, we encourage the reconstructed images to share the statistics of natural images through a perceptual loss that operates on mid-level image features.

To further validate the quality of our approach, we use our reconstructions to solve two popular problems with event cameras: (i) object classification from a stream of events, and (ii) visual-inertial odometry. We apply off-the-shelf computer vision algorithms that were built to process conventional images to the reconstructed videos for both tasks. We show that this strategy outperforms state-of-the-art methods that had been specifically designed for event data.

In summary, our contributions are:

- A novel recurrent network architecture to reconstruct a video from a stream of events, which outperforms the state-of-the-art in terms of image quality by a large margin.
- We establish that the network can be trained from simulated event data and generalizes remarkably well to real events.
- The application of our method to two problems with event cameras: object classification and visual-inertial odometry from event data. Our method outperforms state-of-the-art algorithms designed specifically for event data in both applications.

We believe that the most alluring characteristic of our method is that it acts as a bridge between conventional cameras and event cameras, thus bringing the main stream of computer vision research to event cameras: mature algorithms, modern deep network architectures, and weights pretrained from large natural image datasets. We believe that our work will open the door to leveraging the benefits of event cameras – high temporal resolution, high dynamic range (Fig. 6), and no motion blur (Fig. 9) – to a broader array of applications.

2. Related Work

Events-to-video reconstruction is a popular topic in the event camera literature. Early approaches did not reconstruct videos, but focused on the reconstruction of a single image from a large set of events collected by an event camera moving through a static scene. These works exploit the fact that every event provides one equation relating the intensity gradient and optic flow through brightness constancy [15]. Cook *et al*. [10] used bio-inspired, interconnected networks to simultaneously recover intensity images, optic flow, and angular velocity from an event camera performing small rotations. Kim *et al*. [17] developed an Extended Kalman Filter to reconstruct a 2D panoramic gradient image (later upgraded to a full intensity frame by 2D Poisson integration) from a rotating event camera, and later extended it to a 3D scene and 6 degrees-of-freedom (6DOF) camera motion [18] (albeit in a static scene only). Bardow *et al*. [3] proposed to estimate optic flow and intensity *simultaneously* from sliding windows of events through a variational energy minimization framework. They showed the first video reconstruction framework from events that is applicable to dynamic scenes. However, their energy minimization framework requires multiple hand-crafted regularizers, which can result in severe loss of detail in the reconstructions.

Recently, methods based on direct event integration have emerged. These approaches do not rely on any assumption about the scene structure or motion dynamics, and can naturally reconstruct videos at arbitrarily high framerates. Munda *et al*. [31] cast intensity reconstruction as an energy minimization problem defined on a manifold induced by the event timestamps. They combined direct event integration with total variation regularization and achieved real-time performance on the GPU. Scheerlinck *et al*. [43] proposed to filter the events with a high-pass filter prior to integration, and demonstrated video reconstruction results that are qualitatively comparable with [31] while being computationally more efficient. While these approaches currently define the state-of-the-art, both suffer from artifacts which are inherent to direct event integration. The reconstructions suffer from "bleeding edges" caused by the fact that the contrast threshold (the minimum brightness change of a pixel to trigger an event) is neither constant nor uniform across the image plane. Additionally, pure integration of the events can in principle only recover intensity up to an unknown initial image $\mathcal{I}_0$, which causes "ghosting" effects where the trace of the initial image remains visible in the reconstructed images.

Barua *et al*. [4] proposed a learning-based approach to reconstruct intensity images from events. They used K-SVD [1] on simulated data to learn a dictionary that maps small patches of integrated events to an image gradient and used Poisson integration to recover the intensity image. In

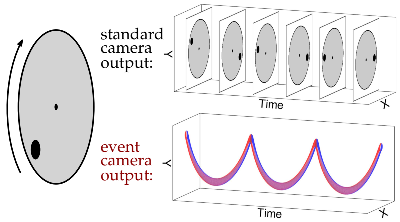


Figure 2. Comparison of the output of a conventional camera and an event camera looking at a black disk on a rotating circle. While a conventional camera captures frames at a fixed rate, an event camera transmits the brightness changes continuously in the form of a spiral of events in space-time (red: positive events, blue: negative events). Figure inspired by [29].

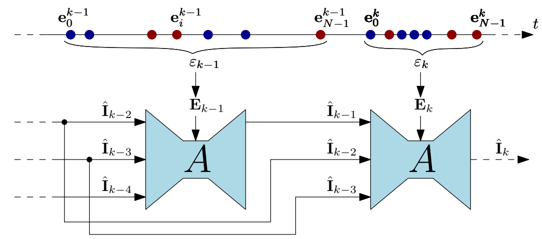


Figure 3. Overview of our approach. The event stream (depicted as red/blue dots on the time axis) is split into windows ε_k with N events in each. Each window is converted into a 3D event tensor $\mathbf{E}_k$ and passed through the network together with the last K reconstructed images to generate a new image reconstruction $\hat{\mathcal{I}}_k$. In this example, $K = 3$ and $N = 7$.

contrast to [4], we do not reconstruct individual intensity images from small windows of events, but instead reconstruct a temporally consistent video from a long stream of events (several seconds) using a recurrent network. Instead of mapping event patches to a dictionary of image gradients, we learn pixel-wise intensity estimation directly.

Despite the body of work on events-to-video reconstruction, further downstream vision applications based on the reconstructions have, to the best of our knowledge, never been demonstrated prior to our work.

3. Video Reconstruction Approach

An event camera consists of independent pixels that respond to changes in the spatio-temporal brightness signal $L(\mathbf{x}, t)$[1] and transmit the changes in the form of a stream of asynchronous events (Fig. 2). For an ideal sensor, an event $\mathbf{e}_i = (\mathbf{u}_i, t_i, p_i)$ is triggered at pixel $\mathbf{u}_i = (x_i, y_i)^T$ and time t_i when the brightness change since the last event at the pixel reaches a threshold $\pm C$, which can be fixed by the user. However, C is in reality neither constant nor uniform across the image plane. Instead, it strongly varies depending on various factors, such as the sign of the brightness change [14], the event rate (because of limited pixel bandwidth) [8], and the temperature [24]. Consequently, events cannot by directly integrated to recover accurate intensity images in practice.

3.1. Overview

Our goal is to translate a continuous stream of events into a sequence of images $\{\hat{\mathcal{I}}_k\}$, where $\hat{\mathcal{I}}_k \in [0,1]^{W \times H}$. To achieve this, we partition the incoming stream of events into sequential (non-overlapping) spatio-temporal windows of events $\varepsilon_k = \{\mathbf{e}_i\}$, for $i \in [0, N-1]$, each containing a fixed number N of events. For each new event sequence ε_k, we generate a new image $\hat{\mathcal{I}}_k$ by fusing the K previous reconstructed images $\{\hat{\mathcal{I}}_{k-K}, ..., \hat{\mathcal{I}}_{k-1}\}$ with the new events ε_k (see Fig. 3). The reconstruction function is implemented by a recurrent convolutional neural network. We train the network in supervised fashion, using a large quantity of simulated event sequences with corresponding ground-truth images. Because we process windows with a constant number of events, the output framerate is proportional to the event rate, making our approach fully data-driven. While our method introduces some latency due to processing events in windows, it nonetheless captures the major advantages of event cameras: our reconstructions have a high dynamic range (Fig. 6) and are free of motion blur, even at high speeds (Fig. 9).

[1] Event cameras respond in fact to logarithmic brightness changes, i.e. $L = \log E$ where E is the *irradiance*.

3.2. Event Representation

In order to be able to process the event stream using a CNN, we need to convert ε_k into a fixed-size tensor representation $\mathbf{E}_k$. A natural choice is to encode the events in a spatio-temporal voxel grid [56]. The duration $\Delta T = t^k_{N-1} - t^k_0$ spanned by the events in ε_k is discretized into B temporal bins. Every event distributes its polarity p_i to the two closest spatio-temporal voxels as follows:

$$\mathbf{E}(x_l, y_m, t_n) = \sum_{\substack{x_i = x_l \\ y_i = y_m}} p_i \max(0, 1 - |t_n - t^*_i|), \qquad (1)$$

where $t^*_i \triangleq \frac{B-1}{\Delta T}(t_i - t_0)$ is the normalized event timestamp. We use $N = 25{,}000$ events per window and $B = 10$ temporal bins, unless specified otherwise.

3.3. Training Data

Our network requires training data, *i.e.* a large amount of event sequences with corresponding ground-truth image sequences. Formally, if we let $\mathbf{E}^S = \{\mathbf{E}_0, ..., \mathbf{E}_{T-1}\}$ be a sequence of event tensors, and $\mathbf{I}^S = \{\mathcal{I}_0, ..., \mathcal{I}_{T-1}\}$ be the corresponding sequence of images, we need to generate a large dataset of mappings $\{\mathbf{E}^S \leftrightarrow \mathbf{I}^S\}$. However, there exists no such large-scale dataset with event data and corresponding ground-truth images. Furthermore, images acquired by a conventional camera would provide poor ground truth in scenarios where event cameras excel, namely high dynamic range and high-speed scenes. For these reasons, we propose to train the network on synthetic event data, and show subsequently that our network generalizes to real event data in Section 4.

We use the event simulator ESIM [37], which allows simulating a large amount of event data reliably. ESIM renders images along the camera trajectory at high framerate, and interpolates the brightness signal at each pixel to approximate the continuous intensity signal needed to simulate an event camera. Consequently, ground-truth images $\mathcal{I}$ are readily available. We map MS-COCO images [25] to a 3D plane, and simulate the events triggered by random camera motion within this simple 3D scene. Examples of generated synthetic event sequences are presented in the appendix. We enrich the training data by simulating a different set of positive and negative contrast thresholds for each simulated scene (sampled according to a normal distribution with mean 0.18 and standard deviation 0.03; these parameters were chosen based on empirical data). This prevents the network from learning to simply integrate events, which would work on noise-free, simulated data, but would generalize poorly to real event data (for which the assumption of a fixed contrast threshold does not hold). The camera sensor size is set to 240×180 pixels (to match the resolution of the DAVIS240C sensor [7] used in our evaluation). Using MS-COCO images allows capturing a much larger variety of scenes than is available in any existing event camera dataset. We generate $1{,}300$ sequences of 2 seconds each, which results in approximately 45 minutes of simulated event data. Note that the simulated sequences contain only globally homographic motion (*i.e.* there is no independent motion in the simulated sequences). Nevertheless, our network generalizes surprisingly well to scenes with arbitrary motions, as will be shown in Section 4.

3.4. Network Architecture and Training

The main module of our recurrent network is a UNet [40] architecture similar to the one introduced by Zhu *et al.* [55] in the context of optical flow estimation. The input tensor (obtained by concatenating $\mathbf{E}_k, \hat{\mathcal{I}}_{k-K}, ..., \hat{\mathcal{I}}_{k-1}$), of size $(B+K) \times H \times W$, is passed through 4 strided convolution layers (the number of output channels doubling each time), followed by two residual blocks [16] and four upsampling transposed convolution layers. The resulting activation is convolved depthwise to obtain a final image reconstruction. Following [55], we use skip connections between symmetric convolution layers. Additional details of the architecture are provided in the appendix. On top of this basic module (labeled "A" in Fig. 3), we introduce a recurrent connection to propagate intensity information forward in time; in other words, the network does not need to reconstruct a new image from scratch at every time step, but only to incrementally update the previous reconstructions using the new sequence of events. During training we unroll the network for L steps. At test time, the preceding K reconstructed images are fed into the network (Fig. 3). We found that $L = 8$ and $K = 3$ provide a good trade-off between reconstruction quality and training time.

Loss: We use the calibrated perceptual loss (LPIPS) [49], which passes the reconstructed image and the target image through a VGG network [44] trained on ImageNet [42], and averages the distances between VGG features computed across multiple layers. By minimizing LPIPS, our network effectively learns to endow the reconstructed images with natural statistics (*i.e.* with features close to those of natural images). The total loss $\mathcal{L}_k$ is computed as $\mathcal{L}_k = \sum_{l=0}^{L} d_L(\hat{\mathcal{I}}_{k-l}, \mathcal{I}_{k-l})$, where d_L denotes the LPIPS distance.

Training Procedure: We split the synthetic sequences into $1{,}270$ training sequences and 30 validation sequences, and implement our network using PyTorch [34]. We use ADAM [19] with an initial learning rate of 0.0001, subsequently decayed by a factor of 0.9 every 10 epochs. We use a batch size of 16 and train for 40 epochs.

4. Evaluation

In this section, we present both quantitative and qualitative results on the fidelity of our reconstructions, and compare to recent methods [3, 31, 43]. We focus our evaluation on real event data. An evaluation on synthetic data can be found in supplementary material.

We use event sequences from the Event Camera Dataset [30]. These sequences were recorded using a DAVIS240C sensor [7] moving in various environments. It contains events as well as ground-truth grayscale frames at a rate of $20\,\mathrm{Hz}$. We remove the redundant sequences (*e.g.* that were captured in the same scene) and those for which the frame quality is poor, leaving seven sequences in total that amount to $1{,}670$ ground-truth frames. For each sequence, we reconstruct a video from the events with our method and each baseline. For each ground truth frame, we query the reconstructed image with the closest timestamp to the ground truth (tolerance of $\pm 1\,\mathrm{ms}$).

Each reconstruction is then compared to the corresponding ground-truth frame according to several quality metrics. We equalize the histograms of every ground-truth frame and reconstructed frame prior to computing the error metrics (this way the intensity values lie in the same intensity range and are thus comparable). Note that the camera speed gradually increases in each sequence, leading to significant motion blur on the ground-truth frames towards the end of the sequences; we therefore exclude these fast sections in

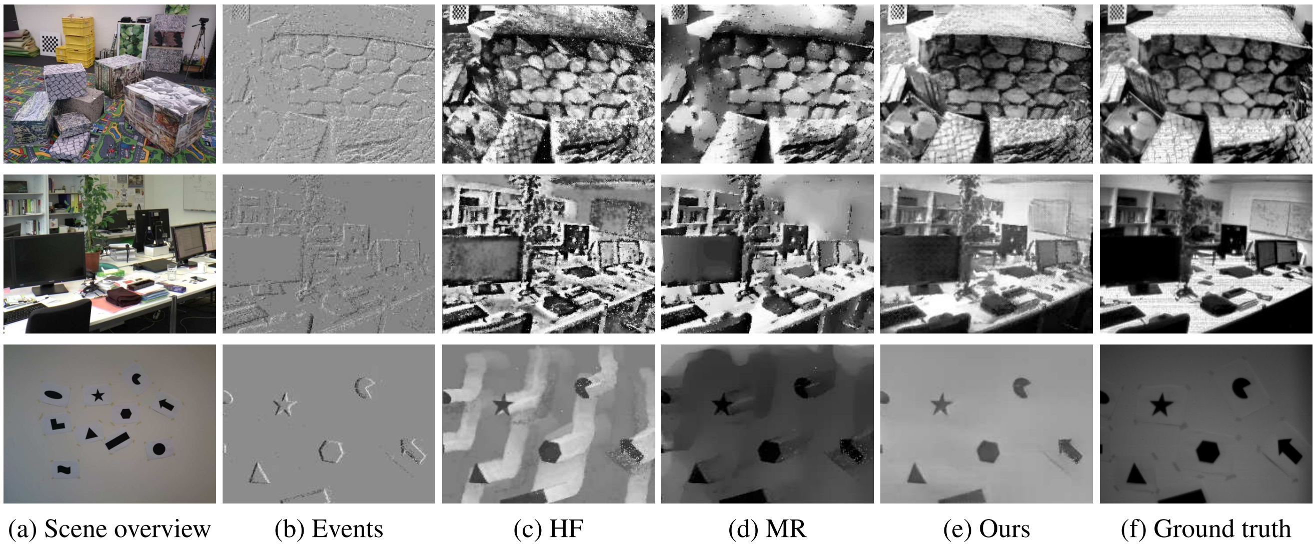


Figure 4. Comparison of our method with MR and HF on sequences from [30]. Our network reconstruct fine details well (textures in the first row) compared to the competing methods, while avoiding their artifacts (e.g. the "bleeding edges" in the third row).

our quantitative evaluation. We also omit the first few seconds from each sequence, which leaves enough time for the baseline methods that are based on event integration to converge. Note that this works in favor of the baselines, as our method converges almost immediately (the initialization phase is analyzed in the supplementary material).

We compare our approach against several state-of-the-art methods: [3] (which we denote as SOFIE for "Simultaneous Optic Flow and Intensity Estimation"), [43] (HF for "High-pass Filter"), and [31] (MR for "Manifold Regularization"). For HF and MR, we used the code that was provided by the authors and manually tuned the parameters on the evaluated sequences to get the best results possible. For HF, we also applied a bilateral filter to the reconstructed images (with filter size $d = 5$ and $\sigma = 25$) in order to remove high-frequency noise, which improves the results of HF in all metrics. For SOFIE, we report qualitative results instead of quantitative results since we were not able to obtain satisfying reconstructions on our datasets using the code provided by the authors. We report three image quality metrics: mean squared error (MSE; lower is better), structural similarity (SSIM; higher is better) [48], and the calibrated perceptual loss (LPIPS; lower is better) [49].

Results and Discussion: The main quantitative results are presented in Table 1, and are supported by qualitative results in Figs. 4 and 5. Additional results are available in the supplementary material. We also encourage the reader to watch the supplementary video, which conveys these results in a better form than still images can.

On all datasets, our reconstruction method outperforms the state-of-the-art by a large margin, with an average 21% increase in SSIM and a 23% decrease in LPIPS. Qualitatively, our method reconstructs small details remarkably well compared to the baselines (see the boxes in the first row of Fig. 4, for example). Furthermore, our method does not suffer from "ghosting" or "bleeding edges" artifacts that are present in other methods (particularly visible in the third row of Fig. 4). These artifacts result from (i) incorrectly estimated contrast thresholds and (ii) the fact that these methods can only estimate the image intensity up to some unknown initial intensity $\mathcal{I}_0$, whose ghost can remain visible. We also compare our method to HF, MR, and SOFIE qualitatively using datasets and image reconstructions directly provided by the authors of [3], in Fig. 5. Once again, our network generates higher quality reconstructions, with finer details and less noise. Finally, we show that our network is able to leverage the outstanding properties of events to reconstruct images in low light (Fig. 6) and during high speed motions (Fig. 9), two scenarios in which conventional cameras fail.

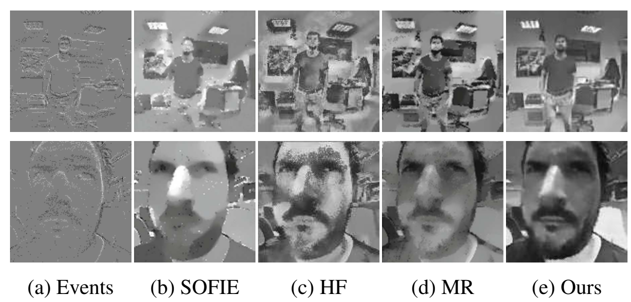
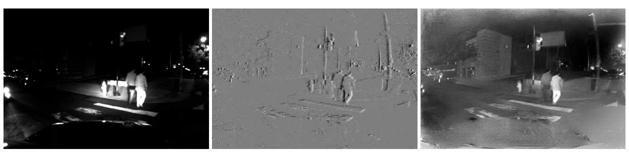


Figure 5. Qualitative comparison on the dataset introduced by [3]. Our method produces cleaner and more detailed results.

Limitations: Our method introduces some latency due to the fact that we process events in windows as opposed to

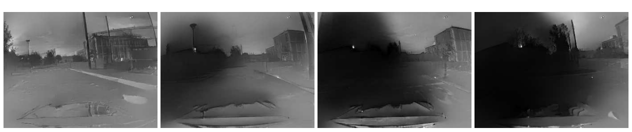
Dataset	MSE			SSIM			LPIPS		
	HF	MR	Ours	HF	MR	Ours	HF	MR	Ours
dynamic_6dof	0.10	0.11	**0.08**	0.39	0.44	**0.50**	0.53	0.53	**0.43**
boxes_6dof	0.09	0.07	**0.04**	0.45	0.47	**0.63**	0.51	0.54	**0.36**
poster_6dof	0.06	0.05	**0.04**	0.52	0.55	**0.68**	0.44	0.50	**0.32**
shapes_6dof	0.11	0.14	**0.10**	0.34	0.43	**0.44**	0.63	0.64	**0.53**
office_zigzag	0.09	0.06	**0.05**	0.36	0.43	**0.50**	0.54	0.55	**0.44**
slider_depth	0.08	0.08	**0.06**	0.48	0.51	**0.61**	0.50	0.55	**0.42**
calibration	0.07	0.06	**0.04**	0.41	0.41	**0.52**	0.55	0.57	**0.47**
Mean	0.09	0.08	**0.06**	0.42	0.46	**0.56**	0.53	0.55	**0.42**

Table 1. Comparison to state-of-the-art image reconstruction methods on the Event Camera Dataset [30]. Our approach outperforms prior such methods on all datasets by a large margin, with an average 21% increase in structural similarity (SSIM) and a 23% decrease in perceptual distance (LPIPS) compared to the best prior methods, respectively MR [31] and HF [43].



(a) VI Sensor (b) Events (c) Our reconstruction

Figure 6. A high-dynamic-range reconstruction from an event camera mounted on a car driving at night [54]. Because of low light, the conventional camera image is severely degraded (a). In contrast, the events (b) capture the whole dynamic range of the scene, which our method successfully converts into an image (c), recovering details that are lost in the conventional frame.



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

Figure 7. An example failure case on a sequence from the MVSEC dataset [54]. The sun (top left of (a)) generates noisy events which cause our network to make a local reconstruction error (b), which gets amplified in subsequent reconstructions (c,d).

event-per-event [43], ranging from $1\,\mathrm{ms}$ to $200\,\mathrm{ms}$ depending on the event rate and value of N (details in the appendix). Also, in some cases (extreme electronic noise in the events not modelled in simulation), our method can fail to reconstruct correctly some parts of the images, and may propagate the error to the next reconstructions (Fig. 7).

5. Downstream Applications

In this section, we demonstrate the potential of our method as a bridge between conventional computer vision and vision with event cameras, for both low-level and high-level tasks. First, we focus on object classification from events (Section 5.1) and then turn to camera pose estimation with events and inertial measurements (Section 5.2).

5.1. Object Classification

Pattern recognition from event data is an active research topic.[2]. While one line of work focuses on spiking neural architectures (SNNs) to recognize patterns from a stream of events with minimal latency (H-FIRST [33]), conventional machine learning techniques combined with novel event representations such as time surfaces (HOTS [21]) have shown the most promising results so far. Recently, HATS [45] addressed the problem of object classification from a stream of events. They proposed several modifications to HOTS, and achieved large improvements in classification accuracy, outperforming all prior approaches by a large margin.

We propose an alternative approach to object classification based on a stream of events. Instead of using handcrafted event representations, we directly train a classification network on images reconstructed from events.

We compare our approach against several recent methods: HOTS, and the state-of-the-art HATS, using the datasets and metric (classification accuracy) used in the HATS paper. The N-MNIST (Neuromorphic-MNIST) and N-Caltech101 datasets [32] are event-based versions of the MNIST [22] and Caltech101 [12] datasets. To convert the images to event sequences, an event camera was placed on a motor, and automatically moved while pointing at images from MNIST (respectively Caltech101) that were projected onto a white wall. The N-CARS dataset [45] proposes a binary classification task: deciding whether a car is visible or not using a $100\,\mathrm{ms}$ sequence of events. Fig. 8 shows a sample event sequence from each of the three datasets.

Our approach follows the same methodology for each dataset. First, for each event sequence in the training set, we use our network to reconstruct an image from the

[2]A list of related works can be found at: `https://github.com/uzh-rpg/event-based_vision_resources`

(a) N-MNIST (b) N-CARS (c) N-Caltech101

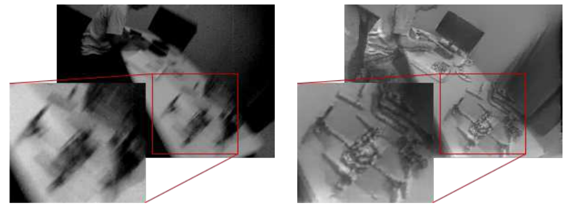
Figure 8. Samples from each dataset used in the evaluation of our object classification approach based on events (Section 5.1). Top: preview of the event sequence. Bottom: our image reconstruction.

	N-MNIST	N-CARS	N-Caltech101
HOTS	0.808	0.624	0.210
HATS/linear SVM	**0.991**	0.902	0.642
HATS/ResNet-18	n.a.	0.904	0.700
Ours (transfer learning)	0.807	n.a.	0.821
Ours (fine-tuned)	0.983	**0.910**	**0.866**

Table 2. Classification accuracy compared to recent approaches, including HATS [45], the state-of-the-art.

events (Fig. 8, bottom row). Then, we train an off-the-shelf CNN for object classification using the reconstructed images from the training set. For N-MNIST, we use a simple CNN (details in the supplement) and train it from scratch. For N-Caltech101 and N-CARS, we use ResNet-18 [16], initialized with weights pretrained on ImageNet [42], and fine-tune the network for the dataset at hand. Once trained, we evaluate each network on the test set (images reconstructed from the events in the test set) and report the classification accuracy. Furthermore, we perform a transfer learning experiment for the N-MNIST and N-Caltech101 datasets (for which corresponding images are available for every event sequence): we train the CNN on the conventional image datasets, and evaluate the network directly on images reconstructed from events without fine-tuning.

For the baselines, we report directly the accuracy provided in [45]. To make the comparison with HATS as fair as possible, we also provide results of classifying HATS features with a ResNet-18 network (instead of the linear SVM used originally). The results are presented in Table 2, where the datasets are presented in increasing order of difficulty from left to right. Despite the simplicity of our approach, it outperforms all baselines, and the gap between our method and the state-of-the-art increases as the datasets get more difficult. While we perform slightly worse than HATS on N-MNIST (98.3% versus 99.1%), this can be attributed to the synthetic nature of N-MNIST, for which our approach does not bring substantial advantages compared to a hand-crafted feature representation such as HATS. Note that, in contrast to HATS, we did not perform hyperparameter tuning. On N-CARS (binary classification task with natural event data),



(a) DAVIS frame (b) Our reconstruction

Figure 9. Comparison of DAVIS frames and reconstructed frames on a high-speed portion of the 'dynamic_6dof' sequence. Our reconstructions from events do not suffer from motion blur, which leads to increased pose estimation accuracy (Table 3).

our method performs better, though the improvement is minor (91% versus 90.4% for HATS). However, N-CARS is almost saturated in terms of accuracy.

On N-Caltech101 (the most challenging dataset, requiring classification of natural event data into 101 object classes), our method truly shines, outperforming HATS by a large margin (86.6% versus 70.0%). This significant gap can be explained by the fact that our approach leverages decades of computer vision research and datasets. Lifting the event stream into the image domain with our events-to-video approach allows us to use a mature CNN architecture pretrained on existing large, labeled datasets, thus leveraging powerful hierarchical features learned on a large amount of image data – something that is not possible with event data, for which labeled datasets are scarce. Finally, and perhaps even more strikingly, we point out that our approach, in the pure transfer learning setup (*i.e.* feeding images reconstructed from events to a network trained on real image data) performs better than all other methods, while not using the event sequences from the training set. To the best of our knowledge, this is the first time that direct transfer learning between image data and event data has been achieved.

We also point out that our approach is real-time capable. On N-Caltech101, end-to-end classification takes less than $10\,\mathrm{ms}$ (sequence reconstruction: $\leq 8\,\mathrm{ms}$, object classification: $\leq 2\,\mathrm{ms}$) on an NVIDIA RTX 2080 Ti GPU. More details about performance can be found in the appendix.

5.2. Visual-Inertial Odometry

The task of Visual-inertial odometry (VIO) is to recover the 6-degrees-of-freedom (6-DOF) pose of a camera from a set of visual measurements (images or events) and inertial measurements from an inertial measurement unit (IMU) that is rigidly attached to the camera. Because of its importance in augmented/virtual reality and mobile robotics, VIO has been extensively studied in the last decade and is relatively mature today [28, 23, 6, 13, 36]. Yet systems based on conventional cameras fail in challenging conditions such as high-speed motions or high-dynamic-range environments. This has recently motivated the development of VIO systems with event data (EVIO) [53, 38, 41].

Inputs	Ours E+I	U.SLAM E+I	U.SLAM E+F+I	HF E+I	MR E+I	VINS-Mono F+I
shapes_translation	0.18	0.32	**0.17**	failed	2.00	0.93
poster_translation	**0.05**	0.09	0.06	0.49	0.15	failed
boxes_translation	**0.15**	0.81	0.26	0.70	0.45	0.22
dynamic_translation	**0.08**	0.23	0.09	0.58	0.17	0.13
shapes_6dof	1.09	0.09	**0.06**	failed	3.00	1.99
poster_6dof	**0.12**	0.20	0.22	0.45	0.17	1.99
boxes_6dof	0.62	0.41	**0.34**	1.71	1.17	0.94
dynamic_6dof	0.15	0.27	**0.11**	failed	0.55	0.76
hdr_boxes	0.34	0.44	0.37	0.64	0.66	**0.32**
Mean	0.31	0.32	**0.19**	0.76	0.92	0.91
Median	**0.15**	0.27	0.17	0.61	0.55	0.84

Table 3. Mean translation error (in meters) on the sequences from [30]. Our method outperforms all other methods that use events and IMU, including UltimateSLAM (E+I). Surprisingly, it even performs on par with UltimateSLAM (E+F+I), while not using additional frames. Methods for which the mean translation error exceeds $5\,\mathrm{m}$ are marked as "failed".

The state-of-the-art EVIO system, UltimateSLAM [41], operates by independently tracking visual features from pseudo-images reconstructed from events and optional images from a conventional camera, and fusing the tracks with inertial measurements using an existing optimization backend [23]. Here, we go one step further and directly apply an off-the-shelf VIO system (specifically, VINS-Mono [36], which is state-of-the-art [11]) to videos reconstructed from events using either our approach, MR, or HF, and evaluate against UltimateSLAM. As is standard [53, 38, 41], we use sequences from the Event Camera Dataset [30], which contain events, frames, and IMU measurements from a DAVIS240C [7] sensor. Each sequence is 60 seconds long, and contains data from a hand-held event camera undergoing a variety of motions in several environments. All sequences feature extremely fast motions (angular velocity up to $880\,^{\circ}/\mathrm{s}$ and linear velocity up to $3.5\,\mathrm{m/s}$), which leads to severe motion blur on the frames (Fig. 9). We compare our approach against the two operating modes of UltimateSLAM: UltimateSLAM (E+I) which uses only events and IMU, and UltimateSLAM (E+F+I) that uses the events, the IMU, and additional frames. We run a publicly available VIO evaluation toolbox [50] on raw trajectories provided by the authors of UltimateSLAM, which ensures that the trajectories estimated by all methods are evaluated in the exact same manner. For completeness, we also report results from running VINS-Mono directly on the frames from the DAVIS sensor.

Table 3 presents the mean translation error of each method, for all datasets (additional results are presented in the supplement). First, we note that our method performs better than UltimateSLAM (E+I) on all sequences, with the exception of the 'shapes_6dof' sequence. This sequence features a few synthetic shapes with very few features (≤ 10), which cause VINS-Mono to not properly initialize, leading to high error (note that this is a problem with VINS-Mono and not our image reconstructions). Overall, the median error of our method is $0.15\,\mathrm{m}$, which is almost twice smaller than UltimateSLAM (E+I) ($0.27\,\mathrm{m}$), which uses the exact same data. Indeed, while UltimateSLAM (E+I) uses coarse pseudo-images created from a single, small window of events, our network is able to reconstruct images with finer details, and higher temporal consistency – both of which lead to better feature tracks, and thus better pose estimates. Even more strikingly, our approach performs on par with UltimateSLAM (E+F+I), while the latter requires additional frames which we do not need. The median error of both methods is comparable ($0.15\,\mathrm{m}$ for ours versus $0.17\,\mathrm{m}$ for UltimateSLAM (E+F+I)).

Finally, we point out that running the same VIO (VINS-Mono) on competing image reconstructions (MR and HF) leads to significantly larger tracking errors (e.g. median error three times as large for MR), which further highlights the superiority of our image reconstructions for downstream vision applications. We acknowledge that our approach is not as fast as UltimateSLAM. Since the main difference between both approaches is how they convert events into "image-like" representations, a rough estimate of the performance gap can be obtained by comparing the time it takes for each method to synthesize a new image: UltimateSLAM takes about $1\,\mathrm{ms}$ on a CPU, in comparison to $\leq 4\,\mathrm{ms}$ on a high-end GPU for our method. Nevertheless, our events-to-video network allows harnessing the outstanding properties of events for VIO, reaching even higher accuracy than state-of-the-art EVIO designed specifically for event data.

6. Conclusion

We presented a novel events-to-video reconstruction framework based on a recurrent convolutional network trained on simulated event data. In addition to outperforming state-of-the-art reconstruction methods on real event data by a large margin ($> 20\%$ improvement), we showed the applicability of our method as a bridge between conventional cameras and event cameras on two vision applications, namely object classification from events and visual-inertial odometry. For each of these tasks, we applied an off-the-shelf computer vision algorithm to videos reconstructed from events by our network, and showed that the result outperforms state-of-the-art algorithms tailored for event data in each case. This validates that our approach allows to readily apply decades of computer vision research to event cameras: mature algorithms, modern deep architectures, and weights pretrained from large image datasets.

Acknowledgements

This work was supported by the the Swiss National Center of Competence Research Robotics (NCCR) and the SNSF-ERC Starting Grant.

References

[1] Michal Aharon, Michael Elad, and Alfred M. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006. 2

[2] Arnon Amir, Brian Taba, David Berg, Timothy Melano, Jeffrey McKinstry, Carmelo Di Nolfo, Tapan Nayak, Alexander Andreopoulos, Guillaume Garreau, Marcela Mendoza, Jeff Kusnitz, Michael Debole, Steve Esser, Tobi Delbruck, Myron Flickner, and Dharmendra Modha. A low power, fully event-based gesture recognition system. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017. 1

[3] Patrick Bardow, Andrew J. Davison, and Stefan Leutenegger. Simultaneous optical flow and intensity estimation from an event camera. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2016. 1, 2, 4, 5

[4] Souptik Barua, Yoshitaka Miyatani, and Ashok Veeraraghavan. Direct face detection and video reconstruction from event cameras. In *IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, 2016. 2, 3

[5] Ryad Benosman, Charles Clercq, Xavier Lagorce, Sio-Hoi Ieng, and Chiara Bartolozzi. Event-based visual flow. *IEEE Trans. Neural Netw. Learn. Syst.*, 25(2):407–417, 2014. 1

[6] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart. Robust visual inertial odometry using a direct EKF-based approach. In *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2015. 7

[7] Christian Brandli, Raphael Berner, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck. A 240x180 130dB 3us latency global shutter spatiotemporal vision sensor. *IEEE J. Solid-State Circuits*, 49(10):2333–2341, 2014. 4, 8

[8] Christian Brandli, Lorenz Muller, and Tobi Delbruck. Real-time, high-speed video decompression using a frame- and event-based DAVIS sensor. In *IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2014. 3

[9] Jörg Conradt, Matthew Cook, Raphael Berner, Patrick Lichtsteiner, Rodney J. Douglas, and Tobi Delbruck. A pencil balancing robot using a pair of AER dynamic vision sensors. In *IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2009. 1

[10] Matthew Cook, Luca Gugelmann, Florian Jug, Christoph Krautz, and Angelika Steger. Interacting maps for fast visual interpretation. In *Int. Joint Conf. Neural Netw. (IJCNN)*, 2011. 1, 2

[11] Jeffrey Delmerico and Davide Scaramuzza. A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots. *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2018. 8

[12] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):594–611, 2006. 6

[13] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. On-manifold preintegration for real-time visual-inertial odometry. *IEEE Trans. Robot.*, 33(1):1–21, 2017. 7

[14] Guillermo Gallego, Jon E. A. Lund, Elias Mueggler, Henri Rebecq, Tobi Delbruck, and Davide Scaramuzza. Event-based, 6-DOF camera tracking from photometric depth maps. *IEEE Trans. Pattern Anal. Machine Intell.*, 40(10):2402–2412, 2018. 1, 3

[15] Daniel Gehrig, Henri Rebecq, Guillermo Gallego, and Davide Scaramuzza. Asynchronous, photometric feature tracking using events and frames. In *Eur. Conf. Comput. Vis. (ECCV)*, 2018. 1, 2

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2016. 4, 7

[17] Hanme Kim, Ankur Handa, Ryad Benosman, Sio-Hoi Ieng, and Andrew J. Davison. Simultaneous mosaicing and tracking with an event camera. In *British Machine Vis. Conf. (BMVC)*, 2014. 2

[18] Hanme Kim, Stefan Leutenegger, and Andrew J. Davison. Real-time 3D reconstruction and 6-DoF tracking with an event camera. In *Eur. Conf. Comput. Vis. (ECCV)*, 2016. 1, 2

[19] Diederik P. Kingma and Jimmy L. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. 4

[20] Beat Kueng, Elias Mueggler, Guillermo Gallego, and Davide Scaramuzza. Low-latency visual odometry using event-based feature tracks. In *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2016. 1

[21] Xavier Lagorce, Garrick Orchard, Francesco Gallupi, Bertram E. Shi, and Ryad Benosman. HOTS: A hierarchy of event-based time-surfaces for pattern recognition. *IEEE Trans. Pattern Anal. Machine Intell.*, 39(7):1346–1359, 2017. 1, 6

[22] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 6

[23] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale. Keyframe-based visual-inertial SLAM using nonlinear optimization. *Int. J. Robot. Research*, 2015. 7, 8

[24] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128×128 120 dB 15 μs latency asynchronous temporal contrast vision sensor. *IEEE J. Solid-State Circuits*, 43(2):566–576, 2008. 1, 3

[25] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Eur. Conf. Comput. Vis. (ECCV)*, 2014. 4

[26] Min Liu and Tobi Delbruck. Adaptive time-slice block-matching optical flow algorithm for dynamic vision sensors. In *British Machine Vis. Conf. (BMVC)*, 2018. 1

[27] 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 Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018. 1

[28] Anastasios I. Mourikis and Stergios I. Roumeliotis. A multi-state constraint Kalman filter for vision-aided inertial navigation. In *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2007. 7

[29] Elias Mueggler, Basil Huber, and Davide Scaramuzza. Event-based, 6-DOF pose tracking for high-speed maneu-

vers. In *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2014. 1, 3

[30] Elias Mueggler, Henri Rebecq, Guillermo Gallego, Tobi Delbruck, and Davide Scaramuzza. The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM. *Int. J. Robot. Research*, 36:142–149, 2017. 1, 4, 5, 6, 8

[31] Gottfried Munda, Christian Reinbacher, and Thomas Pock. Real-time intensity-image reconstruction for event cameras using manifold regularisation. *Int. J. Comput. Vis.*, 2018. 2, 4, 5, 6

[32] Garrick Orchard, Ajinkya Jayawant, Gregory K. Cohen, and Nitish Thakor. Converting static image datasets to spiking neuromorphic datasets using saccades. *Front. Neurosci.*, 9:437, 2015. 6

[33] Garrick Orchard, Cedric Meyer, Ralph Etienne-Cummings, Christoph Posch, Nitish Thakor, and Ryad Benosman. HFirst: A temporal approach to object recognition. *IEEE Trans. Pattern Anal. Machine Intell.*, 37(10):2028–2040, 2015. 6

[34] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS Workshops*, 2017. 4

[35] José A. Perez-Carrasco, Bo Zhao, Carmen Serrano, Begoña Acha, Teresa Serrano-Gotarredona, Shouchun Chen, and Bernabé Linares-Barranco. Mapping from frame-driven to frame-free event-driven vision systems by low-rate rate coding and coincidence processing–application to feedforward ConvNets. *IEEE Trans. Pattern Anal. Machine Intell.*, 35(11):2706–2719, 2013. 1

[36] Tong Qin, Peiliang Li, and Shaojie Shen. VINS-Mono: A robust and versatile monocular visual-inertial state estimator. *arXiv:1708.03852*, 2017. 7, 8

[37] Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza. ESIM: An open event camera simulator. In *Conf. on Robotics Learning (CoRL)*, 2018. 4

[38] Henri Rebecq, Timo Horstschaefer, and Davide Scaramuzza. Real-time visual-inertial odometry for event cameras using keyframe-based nonlinear optimization. In *British Machine Vis. Conf. (BMVC)*, Sept. 2017. 1, 7, 8

[39] Henri Rebecq, Timo Horstschäfer, Guillermo Gallego, and Davide Scaramuzza. EVO: A geometric approach to event-based 6-DOF parallel tracking and mapping in real-time. *IEEE Robot. Autom. Lett.*, 2:593–600, 2017. 1

[40] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015. 4

[41] Antoni Rosinol Vidal, Henri Rebecq, Timo Horstschaefer, and Davide Scaramuzza. Ultimate SLAM? Combining events, images, and IMU for robust visual SLAM in HDR and high speed scenarios. *IEEE Robot. Autom. Lett.*, 3(2):994–1001, 2018. 1, 7, 8

[42] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Fei-Fei Li. ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis.*, 115(3):211–252, 2015. 4, 7

[43] Cedric Scheerlinck, Nick Barnes, and Robert Mahony. Continuous-time intensity estimation using event cameras. In *Asian Conf. Comput. Vis. (ACCV)*, 2018. 2, 4, 5, 6

[44] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015. 4

[45] 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 Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018. 1, 6, 7

[46] Timo Stoffregen, Guillermo Gallego, Tom Drummond, Lindsay Kleeman, and Davide Scaramuzza. Event-based motion segmentation by motion compensation. *arXiv preprint arXiv:1904.01293*, 2019. 1

[47] Timo Stoffregen and Lindsay Kleeman. Simultaneous optical flow and segmentation (SOFAS) using dynamic vision sensor. In *Australasian Conf. Robot. Autom. (ACRA)*, 2017. 1

[48] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, 2004. 5

[49] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018. 4, 5

[50] Zichao Zhang and Davide Scaramuzza. A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry. In *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2018. 8

[51] Yi Zhou, Guillermo Gallego, Henri Rebecq, Laurent Kneip, Hongdong Li, and Davide Scaramuzza. Semi-dense 3D reconstruction with a stereo event camera. In *Eur. Conf. Comput. Vis. (ECCV)*, 2018. 1

[52] Alex Zihao Zhu, Nikolay Atanasov, and Kostas Daniilidis. Event-based feature tracking with probabilistic data association. In *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2017. 1

[53] Alex Zihao Zhu, Nikolay Atanasov, and Kostas Daniilidis. Event-based visual inertial odometry. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017. 1, 7, 8

[54] Alex Zihao Zhu, Dinesh Thakur, Tolga Ozaslan, Bernd Pfrommer, Vijay Kumar, and Kostas Daniilidis. The multivehicle stereo event camera dataset: An event camera dataset for 3D perception. *IEEE Robot. Autom. Lett.*, 3(3):2032–2039, 2018. 6

[55] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. EV-FlowNet: Self-supervised optical flow estimation for event-based cameras. In *Robotics: Science and Systems (RSS)*, 2018. 1, 4

[56] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised event-based optical flow using motion compensation. In *Eur. Conf. Comput. Vis. Workshops (ECCVW)*, 2018. 3