Video to Events: Recycling Video Datasets for Event Cameras

Daniel Gehrig* Mathias Gehrig* Javier Hidalgo-Carrió Davide Scaramuzza
Dept. Informatics, Univ. of Zurich and
Dept. of Neuroinformatics, Univ. of Zurich and ETH Zurich

Abstract

Event cameras are novel sensors that output brightness changes in the form of a stream of asynchronous "events" instead of intensity frames. They offer significant advantages with respect to conventional cameras: high dynamic range (HDR), high temporal resolution, and no motion blur. Recently, novel learning approaches operating on event data have achieved impressive results. Yet, these methods require a large amount of event data for training, which is hardly available due the novelty of event sensors in computer vision research. In this paper, we present a method that addresses these needs by converting any existing video dataset recorded with conventional cameras to synthetic *event data. This unlocks the use of a virtually unlimited number of existing video datasets for training networks designed for real event data. We evaluate our method on two relevant vision tasks, i.e., object recognition and semantic segmentation, and show that models trained on synthetic events have several benefits: (i) they generalize well to real event data, even in scenarios where standard-camera images are blurry or overexposed, by inheriting the outstanding properties of event cameras; (ii) they can be used for fine-tuning on real data to improve over state-of-the-art for both classification and semantic segmentation.*

Multimedia Material

This project's code is available at
https://github.com/uzh-rpg/rpg_vid2e.
Additionally, qualitative results are available in this video:
https://youtu.be/uX6XknBGg0w

1. Introduction

Event cameras, such as the Dynamic Vision Sensor [22] (DVS), are novel sensors that work radically differently from conventional cameras. Instead of capturing intensity images at a fixed rate, they 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.

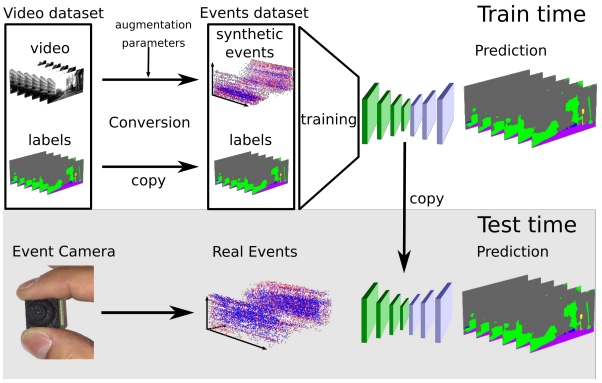


Figure 1. Our method converts any large scale, high quality video dataset, to a synthetic event camera dataset. This unlocks the great wealth of existing video datasets for event cameras, enabling new and exciting applications, and addressing the shortage of high quality event camera datasets. Networks trained on these synthetic events generalize surprisingly well to real events. By leveraging the high dynamic range and lack of motion blur of event cameras these networks can generalize to situations where standard video frames over exposed or blurred. Best viewed in color.

Event cameras 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 blur, and provide measurements with a latency on the order of microseconds. Thus, they are a viable alternative, or complementary sensor, in conditions that are challenging for standard cameras, such as high-speed and high-dynamic-range (HDR) scenarios [19, 36, 42, 33].[1] However, because the output of event cameras is asynchronous, existing computer vision algorithms developed for standard cameras cannot be directly applied to these data but need to be specifically tailored to leverage event data (for a survey on event cameras and the field of event-based vision, we

*Equal contribution

[1] https://youtu.be/0hDGFFJQfmA

refer the reader to [12]).

Recently, novel learning approaches operating on event data have achieved impressive results in scenarios where networks operating on standard cameras fail [26, 41, 1, 42, 33, 15, 34]. Notably, in [34] it was shown that a network trained to reconstruct grayscale intensity frames solely from events can synthesize high framerate videos ($> 5,000$ frames per second) of high-speed phenomena (e.g., a bullet shot by gun hitting an object) and can as well render HDR video in challenging lighting conditions (e.g., abrupt transition from dark to bright scene). It was also shown that off-the-shelf deep learning algorithms trained on large-scale standard camera datasets can be applied to these synthesized HDR, high-framerate videos and that, by doing so, they consistently outperforms algorithms that were specifically trained only on event data.[2] These results highlight that the *event data contain all the visual information* that is needed to carry out the same tasks that can be accomplished with standard cameras and that it should be possible to design *efficient learning algorithms* that process the event data *end to end without passing through intermediate image representations*.

Unfortunately, the design of efficient, end-to-end learning methods requires a large amount of event data for training, which is hardly available because of the novelty of event sensors: event cameras were first commercialized in 2008 and research on event-based vision has made most progress only in the past five years.

A viable alternative to the lack of large scale datasets are event camera simulators [32]; however, an open research question is how well neural networks trained on synthetic events will generalize to real event cameras. Moreover, simulated scenarios still suffer from lack of realism.

To address these issues, we propose a method to generate synthetic, large-scale event-camera data from existing real-world, video datasets recorded with conventional cameras. On the one hand, our method addresses the shortage of event-camera data by leveraging the virtually unlimited supply of existing video datasets and *democratizing* this data for event camera research. The availability of these new datasets can unlock new and exciting research directions for event cameras and spark further research in new fields, previously inaccessible for event cameras. On the other hand, since our method directly relies on video sequences recorded in real-world environments, we show that models trained on synthetic events generated from video generalize surprisingly well to real event data, even in challenging scenarios, such as HDR scenes or during fast motions. To conclude, our contributions are:

- We present a framework for converting existing video datasets to event datasets, thus enabling new applications for event cameras.
- We show that models trained on these synthesized event datasets generalize well to real data, even in scenarios where standard images are blurry or overexposed, by inheriting the outstanding properties of event cameras.
- We evaluate our method on two relevant vision tasks, i.e., object recognition and semantic segmentation, and show that models trained on synthetic events can be used for fine-tuning on real data to improve over state of the art.

Our work is structured as follows: First, we review relevant literature in event camera research and deep learning techniques as well as available datasets in Sec. 2. We then present the method for converting video datasets to events in Sec. 3. Section 4.1 validates and characterizes the realism of events generated by our approach in the setting of object recognition (Sec. 4.1). Finally, we apply our method to the challenging task of per-pixel semantic segmentation in Sec. 4.2.

2. Related Work

2.1. Event Camera Datasets for Machine Learning

The number of event camera datasets tailored to benchmarking of machine learning algorithms is limited. The earliest such datasets are concerned with *classification* and are counterparts of their corresponding image-based datasets. Both Neuromorphic (N)-MNIST and N-Caltech101 [31] were generated by mounting an event camera on a pan-and-til unit in front of a monitor to reproduce the saccades for generating events from a static image. Later, Sironi et al. [38] introduced N-CARS, a binary classification dataset but with events from dynamic scenes rather than static images. The most recent classification dataset [4], termed American Sign Language (ASL)-DVS, features 24 handshapes for american sign language classification. Closely related to neuromorphic classification is neuromorphic action recognition. This task has been targeted by the DVS-Gesture dataset [2] which contains 11 different gestures recorded by the DVS128 event camera.

The first and so far only neuromorphic human pose dataset, DAVIS Human Pose Dataset(DHP19), has been recently introduced by [7]. It features four event cameras with resolution of 260×346 recording 33 different movements simultaneously from different viewpoints.

The DAVIS Driving Dataset (DDD17) [5] and Multi-Vehicle Stereo Event Camera (MVSEC) dataset [40] are two driving datasets. The former provides data about vehicle speed, position, steering angle, throttle and brake besides a single event camera. The latter dataset features

[2] https://youtu.be/eomALySSGVU

multiple vehicles in different environments and also provides ego-motion and LIDAR data together with frames and events from a stereo DAVIS setup. A subset of DDD17 was later extended [1] with approximate semantic labels to investigate semantic segmentation with event cameras.

2.2. Deep Learning with Event Cameras

The applicability of deep learning to event camera data was first explored in the context of classification. Neil et al. [28] designed a novel recurrent neural network architecture applied to classification on the N-MNIST dataset. Later, Maqueda et al. [26] proposed an event-frame representation and designed a CNN architecture for steering angle regression on the DDD17 dataset. The same dataset has been modified by Alonso et al. [1] to perform semantic segmentation. The availability of MVSEC has spurred research in optical flow [41, 42, 15] and depth estimation [42, 39]. In contrast to aforementioned work, [33, 34] trained a convolutional recurrent neural network entirely on simulated events to perform image reconstruction.

2.3. Synthetic Events

This section reviews work in the domain of generative modeling for events from event cameras. Early work in this domain has been performed by Kaiser et al. [18]. They generate events simply by applying a threshold on the image difference. Depending on the pixel's intensity difference a positive or negative event is generated. Pix2NVS [3] computes per-pixel luminance from conventional video frames. The technique generates synthetic events with inaccurate timestamps clustered to frame timestamps. To the best of our knowledge, the two first simulators attempting to generate events accurately are [27] and [21]. Both works render images at high frame-rate and linearly interpolate the intensity signals to generate events. Rebecq et al. [32] additionally introduces an adaptive sampling scheme based on the maximum displacement between frames. This leads to improved accuracy for very fast motion and lower computation in case of slow motion. The generative model used in [27, 32] has been formalized in previous work [22, 13, 14].

3. Methodology

In this section, we describe our method for converting video to synthetic events. This conversion can be split into two steps: event generation and frame upsampling, covered in Sec. 3.1 and Sec. 3.2, respectively. Fig. 2 illustrates these individual steps. In a first step, we leverage a recent frame interpolation technique [17] to convert low frame rate to high frame rate video using an adaptive upsampling technique. This video is then used to generate events using the generative model by leveraging a recent event camera simulator (ESIM) [32]. To facilitate domain adaptation between synthetic and real events, we further introduce two domain adaptation techniques. Finally, we make use of [15] to convert the sparse and asynchronous events to tensor-like representations, which enables learning with traditional convolutional neural network (CNN) architectures.

3.1. Event Generation Model

Event cameras have pixels that are independent and respond to changes in the continuous log brightness signal $L(\mathbf{u}, t)$. An event $e_k = (x_k, y_k, t_k, p_k)$ is triggered when the magnitude of the log brightness at pixel $u = (x_k, y_k)^T$ and time t_k has changed by more than a threshold C since the last event at the same pixel.

$$\Delta L(\mathbf{u}, t_k) = L(\mathbf{u}, t_k) - L(\mathbf{u}, t_k - \Delta t_k) \geq p_k C. \quad (1)$$

Here, Δt_k is the time since the last triggered event, $p_k \in \{-1, +1\}$ is the sign of the change, also called polarity of the event. Equation (1) describes the generative event model for an ideal sensor [14, 12].

3.2. Frame Upsampling

While the event generative model provides a tool for generating events for a given brightness signal, it requires that this signal be known at high temporal resolution. In particular for event cameras this timescale is on the order of microseconds. Event camera simulators, such as ESIM, can address this problem by adaptively rendering virtual scenes at arbitrary temporal resolution (Section 3.1 of [32]). However, video sequences typically only provide intensity measurements at fixed and low temporal resolution on the order of milliseconds.

We therefore seek to recover the full intensity profile $I(\mathbf{u}, t)$ given a video sequence of N frames $\{I(\mathbf{u}, t_i)\}_{i=0}^{N}$ captured at times $\{t_i\}_{i=0}^{N}$. A subproblem using only two consecutive frames has been well studied in frame interpolation literature. We thus turn to [17], a recent technique for frame interpolation which is finding wide spread use in smart-phones. Compared to other frame interpolation techniques such as [23, 24, 29, 30] the method in [17] allows to reconstruct frames at arbitrary temporal resolution, which is ideal for the posed task. The number of intermediate frames, must be chosen carefully since too low values lead to aliasing of the brightness signal (illustrated in [32], Fig. 3) but too high values impose a computational burden. The following adaptive sampling strategy, inspired by [32], uses bidirectional optical flow (as estimated internally by [17]) to compute the number of intermediate samples. Given two consecutive frames $I(t_i)$ and $I(t_{i+1})$ at times t_i and t_{i+1}, we generate K_i equally spaced intermediate frames. K_i is chosen such that the relative displacement between intermediate frames is at most 1 pixel for all pixels:

$$K_i = \max_{\mathbf{u}} \max\{\|\mathbf{F}_{i \to i+1}(\mathbf{u})\|, \|\mathbf{F}_{i+1 \to i}(\mathbf{u})\|\} - 1, \quad (2)$$

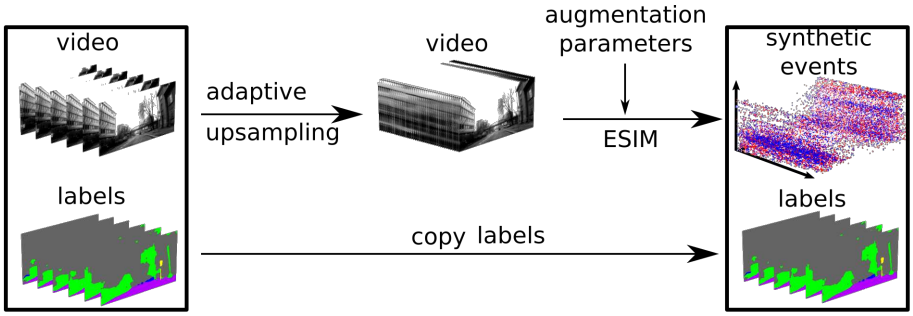


Figure 2. Overview of the method. Low frame-rate video is first adaptively upsampled using the method proposed in [17]. This upsampled video is fed to the event camera simulator (ESIM) [32] which produces asynchronous and sparse events with high temporal resolution.

where $\mathbf{F}_{i\to j}(\mathbf{u})$ is the optical flow from frame i to j at pixel location $\mathbf{u}$. We use this strategy to adaptively upsample between pairs of video frames, resulting in an adaptively upsampled video sequence (Fig. 2 middle).

3.3. Event Generation from High Frame Rate Video

The next step is to generate events from the high frame rate video sequence generated in Sec. 3.2. We generate events by employing the algorithm described in [32] (Sec 3.1). For each pixel the continuous intensity signal in time is approximated by linearly interpolating between video frames. Events are generated at each pixel whenever the magnitude of the change in intensity exceeds the contrast threshold, C (defined in (1)) which is a parameter of ESIM. Since the contrast threshold in (1) is typically not known for real sensors and can vary from sensor to sensor and between positive and negative events, we propose to randomize it at train-time. Before generating a sequence of events we randomly sample contrast thresholds for positive and negative events, C_p, C_n from the uniform distribution $\sim \mathcal{U}(C_{\text{min}}, C_{\text{max}})$. A similar procedure was used in [33, 34] where randomization was shown to improve domain adaptation between simulated and real data. In this work we chose C_{min} =0.05 and C_{max} =0.5.

3.4. Event Representation and Learning

As a next step, The synthetic events and original labels are used to train a network. To do this, we consider a window of events leading up to the time stamped ground truth label and train a model to predict it. Note that this works for general datasets with precisely timestamped images and labels. We take advantage of existing CNN architectures designed for standard images by converting the asynchronous and sparse event streams into tensor-like representation. We chose the Event Spike Tensor (EST) [15] since it was shown to outperform existing representations on both high- and low-level tasks. The EST is generated by drawing the events with positive and negative polarity into two separate spatio-temporal grids of dimensions $H \times W \times C$ and stacking them along the channel dimension. Here H and W are the sensor resolution and C is a hyper-parameter which controls the number of temporal bins used to aggregate events. In this work we chose $C = 15$.

4. Experiments

In this section, we present an evaluation of the method described in 3 on two tasks: object classification (Sec. 4.1) and semantic segmentation (Sec. 4.2). In each case we show that models that are trained on synthetic events have the following benefits: (i) they generalize well from synthetic to real events (ii), can be used to fine tune on real event data, leading to accelerated learning and improvements over the state of the art, and (iii) can generalize to scenarios where standard frames are blurry or underexposed.

4.1. Object Recognition

Object recognition using standard frame-based cameras remains challenging due to their low dynamic range, high latency and motion blur. Recently, event-based object recognition has grown in popularity since event cameras address all of these challenges. In this section we evaluate the event generation method proposed in 3 in this scenario. In particular, we provide an analysis of each component of the method, frame upsampling and event generation. In our evaluation we use N-Caltech101 (Neuromorphic-Caltech101) [31], the event-based version of the popular Caltech101 dataset [11] which poses the task of multi class recognition. This dataset remains challenging due to a large class imbalance. The dataset comprises 8,709 event sequences from from 101 object classes each lasting for the duration of 300 ms. Samples from N-Caltech101 were recorded by placing an event camera in front of a screen and projecting various examples from Caltech101, while the event camera underwent three saccadic movements.

4.1.1 Implementation

To evaluate our method we convert the samples of Caltech101 to event streams, thus generating a replica (sim-N-Caltech101) of the N-Caltech101 dataset. We then

aim at quantifying how well a network trained on sim-N-Caltech101 generalizes to events in the real dataset, N-Caltech101. To convert samples from Caltech101 to event streams we adopt the strategy for converting still images to video sequences outlined in [33, 34]. We map the still images onto a 2D plane and simulate an event camera moving in front of this plane in a saccadic motion, as was done for the original N-Caltech101 dataset [31]. Note that, since the camera is moved virtually, video frames can be rendered at arbitrary temporal resolution, making it possible to simulate video cameras with different frame rates. Once a high frame rate video is rendered, we use this video to generate events. In a first step we fix the contrast threshold in ESIM to 0.06 but randomize this value later. Some examples from sim-Caltech101 as well as corresponding samples from N-Caltech101 and Caltech101 are shown in Fig. 4.

In a next step we train a classifier on data from sim-N-Caltech101. We chose an off-the-shelf classifier based on ResNet-34 [16] which has been pretrained on RGB images from ImageNet [37]. We choose a batch size of 4 and a learning rate of 10^-6 and trained the network to convergence. We then compute the test score of this network on a held out set on the real dataset which is reported in the first row of Tab. 2. As a baseline we compare against a network which was trained on real data and evaluated on the same held out test set. We can observe that the network trained on synthetic events leads to a lower score (75.1%) than one trained on real events (86.3%) leading to a gap of 11.2%. To address this gap we apply a form of domain randomization by randomly sampling the contrast threshold during training, as was described in 3.3. This is done for two reasons: On the one hand this step helps to add robustness to the network by exposing it to a larger variety of event streams, which benefits generalizability. On the other hand the true contrast threshold it typically not known during training, so randomization eliminates the need for hand tuning this parameter. By employing this technique we achieve an improved result of 78.2% reducing the gap to 8.1%.

We propose to further generalizability through dataset extension. It is well known that Caltech101 is unbalanced. For example, while the most common class (airplanes) has 800 samples, the least common class (inline skate) has only 31 samples. To address this imbalance, we exploit the fact that our method does not require real events. We downloaded images from the Internet (google images) to find additional examples for each class. We filtered wrong samples by using a ResNet-34 classifier [16], pretrained on Caltech101 images. By employing this strategy without contrast threshold randomization we achieve a test score of 76.9% and if we include both techniques we achieve a score of 80.7%, effectively reducing the gap to real to 5.6%. While this gap still remains, this result shows that the synthetic events generated by our method effectively capture most of the visual appearance of the real event stream thus achieve a high level of realism.

Fine Tuning In this section we show that a network pretrained on simulated data described in the previous section can be used to fine tune on real data, which leads to a large performance increase. We fine tune the best model obtained in the previous experiment network obtained by training on real events from N-Caltech101 with a reduced learning rate of 10^{-7} and train until convergence. The test score is reported in Tab. 2 where we see that fine tuning has a large impact on network performance. Not only does the test score surpass baseline on real data, it also beats existing state of the art event-based approaches, summarized in Tab. 3, such as [38, 15, 33] and approaches state of the art methods using standard images [25] with 94.7%.

4.1.2 Effect of Frame Upsampling

In this section we present an ablation study which aims at characterizing the effect of frame upsampling on the generated events. This is crucial since our method relies on video sequences, which typically only record visual information at low temporal resolution. In particular, we show that adaptive frame upsampling leads to improvements in the events in the case of low frame rate video. To understand this relationship we propose the following controlled experiment, illustrated in Fig. 3. We first generate a reference dataset from Caltech101 samples by rendering video frames at 530 Hz (Fig. 3 a), for 300 ms, such that the maximal displacement between consecutive frames is below 1 pixel (0.13 pixel). We simulate the low frame rate of conventional video cameras by downsampling these frames by factors of (4, 16, and 80) leading to maximal pixel displacements of 0.55, 2.11 and 9.4 respectively (Fig. 3 c). To recover high frame rate video we apply the frame interpolation technique described in [17], which results in frames at the same temporal resolution as the original video. To understand the effect of video quality on events we generate datasets for each of these three cases, fixing the settings for event generation, and varying the downsampling factor. This way changes in the events are reflected by the changes in video quality. To assess these differences we train three classifiers with the same training and network parameters as described in the previous section, and compare their test scores on events generated from the original high frame rate video. The test scores for different downsampling factors is reported in Tab. 1. While a network trained on events from high frame rate video (Tab. 1 top row) achieves a high score of 88.6% on this test set, we see that reducing the framerate (Tab. 1 second row) by a factor of 80, drastically reduces this score to 61.8%. In fact, artifacts caused by the low frame rate become apparent at these low frame rates. One such artifact is called ghosting and is caused when there is

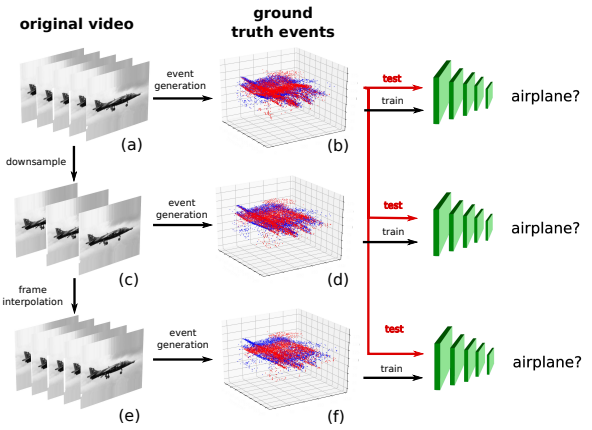


Figure 3. Evaluation of the effect of frame interpolation on event quality. We render a high frame rate video of a Caltech101 [11] image (a) by sliding a virtual camera in front of a 2d planes, following three saccadic movements as described in [31] which we use to generate ground truth events (b). We then downsample the video (c) which leads to a distortion of the event stream (d). By applying the interpolation technique in [17] we can reconstruct the original video (e) which leads to improved event quality. To quantify this quality, we train three classifiers, one on each dataset, and compare test scores on the ground truth events.

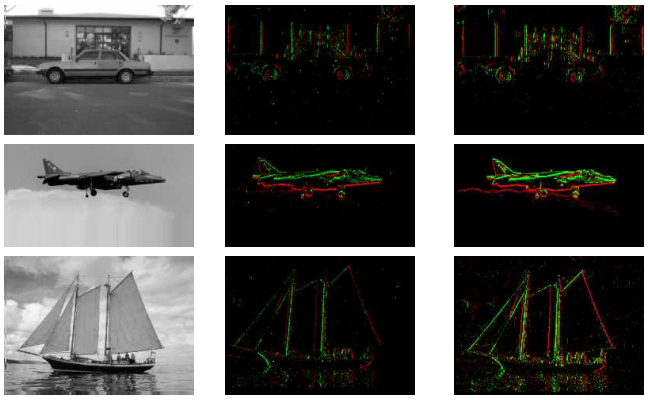
video	downsampling factor			
	1	4	16	80
original	0.887		-	
downsampled	0.887	0.882	0.867	0.618
interpolated	0.887	0.881	0.877	0.687
average interframe displacement [px]	0.13	0.55	2.11	9.4

Table 1. Ablation study on the effect of downsampling. Test score of networks trained on events generated from different video streams and evaluated on events from high frame rate video.

a large displacement between consecutive frames. In this case linear interpolation of the intensity values over time results in the appearance and disappearance of parts of the scene which cause events to be generated in an unrealistic fashion. By using frame interpolation we reduce these effects, as indicated by the increased performance (68.7%).

4.2. Semantic Segmentation

Semantic segmentation is a recognition task which aims at assigning a semantic label to each pixel in an image. It has numerous applications, including street lane and pedestrian detection for autonomous driving. Nonetheless, semantic segmentation using standard images remains challenging especially in edge-case scenarios, where their quality is greatly reduced due to motion blur or over- and under-exposure. Event-based segmentation promises to address these issues by leveraging the high dynamic range, lack of motion blur and low latency of the event camera.



(a) preview (b) real events (c) synthetic events

Figure 4. A side-by-side comparison of a samples from Caltech101 (a), N-Caltech101 (b) and our synthetic examples from (c) sim-N-Caltech101. While the real events were recorded by moving an event camera in front of a projector, the synthetic events were generated using ESIM by moving a virtual camera in front of a 2D projection of the sample in (a).

contrast threshold randomization	dataset extension	fine tuning on real	test score
			0.751
	✓		0.769
✓			0.782
✓	✓		0.807
		✓	0.856
	✓	✓	0.852
✓		✓	0.904
✓	✓	✓	**0.906**
real data			0.863
images [25]			0.947

Table 2. Effect of randomization on test accuracy. For comparison we report the test scores when trained on real events and also the state of the art [25] on the original Caltech101 images.

Method	Training Data	Test Score
HATS [38]	real	0.642
HATS+ResNet-34 [38]	real	0.691
RG-CNN [4]	real	0.657
EST [15]	real	0.817
E2VID [33]	real	0.866
ours	synthetic	0.807
ours	synthetic + real	**0.906**

Table 3. Comparison of classification accuracy for state of the art classification methods on N-Caltech101 [31]. Our method uses a ResNet-34[16] architecture.

In this section we evaluate our method for semantic segmentation by generating a large scale synthetic event dataset from the publicly available DAVIS Driving Dataset (DDD17) [5]. It features grayscale video with events from the Dynamic and Activate Vision Sensor (DAVIS) [6] and semantic annotations provided by [1] for a selection of sequences. In [1] a network trained on Cityscapes [9] was used to generate labels for a total of 19840 grayscale im-

ages (15950 for training and 3890 for testing). The combination of grayscale video and events allows us to generate synthetic events and evaluate against real events from the event camera. We show that training solely on synthetic events generated by our method yields competitive performance with respect the state of the art trained on real events. Furthermore, we improve on the state of the art [1] by training on synthetic events and fine tuning on real events.

4.2.1 Implementation

The annotated version of DDD17 [1] provides segmentation labels which are synchronized with the frames and thus appear at 10-30 Hz intervals. For each label we use the events that occurred in a 50 ms time window before the label for prediction, as was done in [1]. We consider two event-based input representations: the EST, which was already used in Sec. 4.1 and the 6-channel representation proposed by [1]. In [1] a six channel tensor is constructed from the events, with three channels for both the positive and negative events. The first channel is simply the histogram of events; that is the number of events received at each pixel within a certain time interval. The second channel is the mean timestamp of the events while the third channel is the standard deviation of the timestamps.

We use the network architecture proposed in [1] which consists of a U-Net architecture [35] with an Xception encoder [8] and a light decoder architecture. We use a batchsize of 8 and use ADAM [20] with a learning rate of 10^-3 and train until convergence.

4.2.2 Quantitative Results

As was done in [1], we use the following two evaluation metrics: *Accuracy* and *Mean Intersection over Union (MIoU)*. Given predicted semantic labels $\hat{y}$, ground-truth labels y, N the number of pixels and C the number of classes, *accuracy* is defined as

$$\text{Accuracy} = \frac{1}{N}\sum_{n=1}^{N}\delta(y_n, \hat{y}_n) \tag{3}$$

and simply measures the overall proportion of correctly labelled pixels. MIoU, defined as

$$\text{MIoU} = \frac{1}{C}\sum_{c=1}^{C}\frac{\sum_{n=1}^{N}\delta(y_{n,c},1)\delta(y_{n,c},\hat{y}_{n,c})}{\sum_{n=1}^{N}\max(1,\delta(y_{n,c},1)+\delta(\hat{y}_{n,c},1))} \tag{4}$$

is an alternative metric that takes into account class imbalance [10] in an image through normalization and is thus a more robust metric compared to accuracy.

We train two neural networks on synthetic events generated from video, one using the event representation in [1] and one using the EST [15]. We evaluate these networks on the test set, and vary the size of the window of events between 10, 50 and 250 ms which was also done in [1]. The results from this experiment are summarized in Tab. 4. We compare against the state of the art method in [1], represented in the last row. Tab. 4 indicates that the overall accuracy (on 50 ms) for both representations remains within 4% of the 89.8% correctly classified pixels. The difference on the MIoU metric is slightly larger with 45.5% for EST and 48.2% for Alonso et al.'s representation compared to 54.8% if trained on real events. These results indicate that training only on synthetic events yields good generalization to real events, though slightly lower than training on real event data directly. In the next step we want to quantify the gain in when we fine tune on real data.

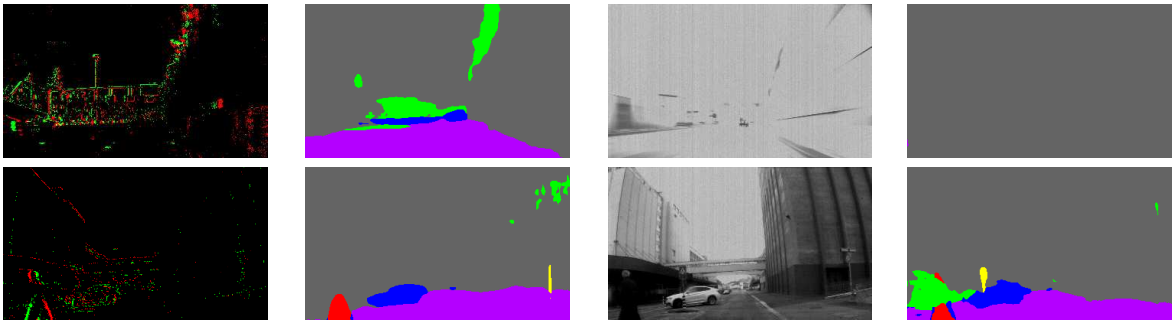
In a next step we fine tune these models on real data. We do this with a lower learning rate of 10^-4, and after only two epochs of training we observe large improvements leading to state of the art performance, as captured in Tab. 4. In fact, our method outperforms existing approaches consistently by an average of 1.2%. In addition, we see that our method remains moderately robust even with large variations on the event window size.

4.2.3 Edge-Cases

In previous sections we have demonstrated that event datasets generated using our method generalize well to real data and networks trained on these datasets can be fine tuned on real data to enhance performance above state of the art. In this section we investigate how networks trained on synthetic events alone generalize to scenarios in which traditional frames corrupted due to motion blur or over- and under-exposure. In this experiment we use the model trained with EST inputs from synthetic events. Fig. 5 illustrates two edge cases where frame-based segmentation fails, due to over-exposure (top row) and low contrast (bottom row). We see that in the first case the segmentation network only predicts the background class (top right) since the image is overexposed. In the second case the frame-based segmentation wrongly classifies a person as vegetation which is due to the low contrast in the lower left part of the image. The network using events handles both cases gracefully thanks to the high contrast sensitivity of the event camera. It is important to note that the network never saw real events during training, yet generalizes to edge-case scenarios. This shows that networks trained on synthetic events can generalize beyond the data they were trained with, and do this by inheriting the outstanding properties of events.

5. Known Limitations

One apparent shortcoming is the occurrence of blurry frames in a video dataset. Blurr typically persists in interpolated frames and thus yields suboptimal results when used



(a) events (b) prediction from events (c) DAVIS frame (d) prediction from frame

Figure 5. Edge cases for semantic segmentation (violet: street; green: vegetation; red: person; blue: car; yellow: object; gray: background). The first row depicts a scenario in which the conventional camera is over-exposed. This results in deteriorated frame-based segmentation performance. In contrast, the event-based segmentation network is able to predict the road labels accurately. The second row showcases a scenario in which frame-based segmentation wrongly classifies a person as vegetation. This is due to the low contrast in the lower left part of the image. The event camera gracefully handles this case thanks to its superior contrast sensitivity. Best viewed in color.

Representation	Fine tuned	Acc. [50 ms]	MIoU [50 ms]	Acc. [10 ms]	MIoU [10 ms]	Acc. [250 ms]	MIoU [250 ms]
Alonso et al. [1]		86.03	48.16	77.25	31.76	84.24	40.18
EST [15]		85.93	45.48	81.11	30.70	84.49	40.66
Alonso et al.	✓	89.36	55.17	86.06	39.93	87.20	47.85
EST	✓	**90.19**	**56.01**	**87.20**	45.82	**88.64**	**51.61**
Alonso et al.	trained on real	89.76	54.81	86.46	**45.85**	87.72	47.56

Table 4. Semantic segmentation performance of different input representations on the test split of [1]. *Accuracy* and *MIoU* (*Mean Intersection over Union*). The models are trained on representations of 50 milliseconds (ms) of events and evaluated with a representations of 10 ms and 250 ms of events. The results reported in the last row are taken from [1]. This model was trained directly on the real events.

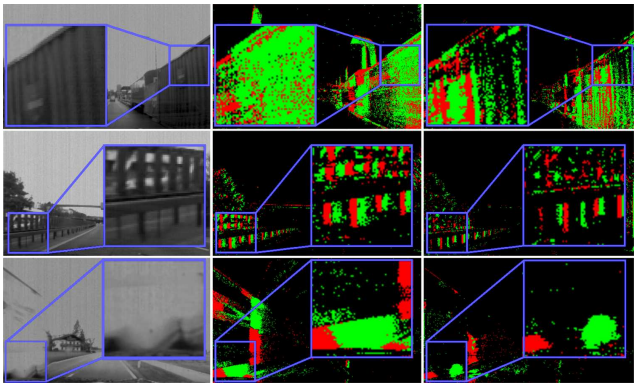


Figure 6. Interpolation artefacts and events in three scenarios (rows). Interpolated frames (left column), real events (middle) and synthetic events (right). Positive and negative events colored in green and red respectively. *Row 1 & 2*: At high optic flows repetitive structures are copied instead of interpolated, leading to missing/wrong events. *Row 3*: Car tire is incorrectly interpolated (collapses to linear interpolation) due to large optical flow. *Best viewed in PDF format*.

in combination with the generative model. Furthermore, the generative model does not account for noise that exists in real event cameras. We consider noise modelling an interesting direction of future work that could be incorporated into the proposed framework. Finally, this work builds on frame interpolation methods. While they also have limitations, see Fig. 6, it is an active area of research. Consequently, the proposed method can directly benefit from future improvements in frame interpolation techniques.

6. Conclusion

Over the years, the computer vision community has collected a large number of extensive video datasets for benchmarking novel algorithms. This stands in contrast to the relatively few datasets available to researchers on event-based vision. This work offers a simple, yet effective solution to this problem by proposing a method for converting video datasets into event datasets. The availability of these new synthetic dataset offers the prospect of exploring previously untouched research fields for event-based vision.

The proposed method utilizes a combination of neural network based frame interpolation and widely used generative model for events. We highlight the generalization capability of models trained with synthetic events in scenarios where only real events are available. On top of that, we show that finetuning models (trained with synthetic events) with real events consistently improves results in both object recognition and semantic segmentation.

7. Acknowledgements

This work was supported by the Swiss National Center of Competence Research Robotics (NCCR), through the Swiss National Science Foundation, the SNSF-ERC starting grant, and Prophesee.

References

[1] Iñigo Alonso and Ana C Murillo. EV-SegNet: Semantic segmentation for event-based cameras. In *IEEE Conf. Comput. Vis. Pattern Recog. Workshops (CVPRW)*, 2019. 2, 3, 6, 7, 8

[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)*, pages 7388–7397, 2017. 2

[3] Yin Bi and Yiannis Andreopoulos. PIX2NVS: Parameterized conversion of pixel-domain video frames to neuromorphic vision streams. In *IEEE Int. Conf. Image Process. (ICIP)*, pages 1990–1994, Sept. 2017. 3

[4] Yin Bi, Aaron Chadha, Alhabib Abbas, , Eirina Bourtsoulatze, and Yiannis Andreopoulos. Graph-based object classification for neuromorphic vision sensing. In *Int. Conf. Comput. Vis. (ICCV)*, 2019. 2, 6

[5] Jonathan Binas, Daniel Neil, Shih-Chii Liu, and Tobi Delbruck. DDD17: End-to-end DAVIS driving dataset. In *ICML Workshop on Machine Learning for Autonomous Vehicles*, 2017. 2, 6

[6] 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. 6

[7] Enrico Calabrese, Gemma Taverni, Christopher Awai Easthope, Sophie Skriabine, Federico Corradi, Luca Longinotti, Kynan Eng, and Tobi Delbruck. Dhp19: Dynamic vision sensor 3d human pose dataset. In *IEEE Conf. Comput. Vis. Pattern Recog. Workshops (CVPRW)*, 2019. 2

[8] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017. 7

[9] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 6

[10] Gabriela Csurka, Diane Larlus, Florent Perronnin, and France Meylan. What is a good evaluation measure for semantic segmentation?. In *BMVC*, volume 27, page 2013. Citeseer, 2013. 7

[11] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(4):594–611, 2006. 4, 6

[12] Guillermo Gallego, Tobi Delbruck, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew Davison, Jörg Conradt, Kostas Daniilidis, and Davide Scaramuzza. Event-based vision: A survey. *arXiv e-prints*, abs/1904.08405, 2019. 2, 3

[13] Guillermo Gallego, Christian Forster, Elias Mueggler, and Davide Scaramuzza. Event-based camera pose tracking using a generative event model. arXiv:1510.01972, 2015. 3

[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. Mach. Intell.*, 40(10):2402–2412, Oct. 2018. 3

[15] Daniel Gehrig, Antonio Loquercio, Konstantinos G. Derpanis, and Davide Scaramuzza. End-to-end learning of representations for asynchronous event-based data. In *Int. Conf. Comput. Vis. (ICCV)*, 2019. 2, 3, 4, 5, 6, 7, 8

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 770–778, 2016. 5, 6

[17] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super SloMo: High quality estimation of multiple intermediate frames for video interpolation. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018. 3, 4, 5, 6

[18] Jacques Kaiser, J. Camillo Vasquez Tieck, Christian Hubschneider, Peter Wolf, Michael Weber, Michael Hoff, Alexander Friedrich, Konrad Wojtasik, Arne Roennau, Ralf Kohlhaas, Rüdiger Dillmann, and J. Marius Zöllner. Towards a framework for end-to-end control of a simulated vehicle with spiking neural networks. In *IEEE Int. Conf. on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*, pages 127–134, 2016. 3

[19] Hanme Kim, Ankur Handa, Ryad Benosman, Sio-Hoi Ieng, and Andrew J. Davison. Simultaneous mosaicing and tracking with an event camera. In *British Mach. Vis. Conf. (BMVC)*, 2014. 1

[20] Diederik P. Kingma and Jimmy L. Ba. Adam: A method for stochastic optimization. *Int. Conf. Learn. Representations (ICLR)*, 2015. 7

[21] Wenbin Li, Sajad Saeedi, John McCormac, Ronald Clark, Dimos Tzoumanikas, Qing Ye, Yuzhong Huang, Rui Tang, and Stefan Leutenegger. InteriorNet: Mega-scale multi-sensor photo-realistic indoor scenes dataset. In *British Mach. Vis. Conf. (BMVC)*, 2018. 3

[22] 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

[23] Ziwei Liu, Raymond A. Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. *Int. Conf. Comput. Vis. (ICCV)*, pages 4473–4481, 2017. 3

[24] Gucan Long, Laurent Kneip, Jose M. Alvarez, Hongdong li, Xiaohu Zhang, and Qifeng Yu. Learning image matching by simply watching video. In *Eur. Conf. Comput. Vis. (ECCV)*, volume 9910, pages 434–450, Oct. 2016. 3

[25] Ammar Mahmood, Mohammed Bennamoun, Senjian An, and Ferdous Sohel. ResFeats: Residual network based features for image classification. In *IEEE Int. Conf. Image Process. (ICIP)*, Sept. 2017. 5, 6

[26] 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)*, pages 5419–5427, 2018. 2, 3

[27] 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(2):142–149, 2017. 3

[28] Daniel Neil, Michael Pfeiffer, and Shih-Chii Liu. Phased LSTM: Accelerating recurrent network training for long or event-based sequences. In *Conf. Neural Inf. Process. Syst. (NIPS)*, 2016. 3

[29] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive convolution. *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 2270–2279, 2017. 3

[30] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *Int. Conf. Comput. Vis. (ICCV)*, pages 261–270, 10 2017. 3

[31] 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. 2, 4, 5, 6

[32] Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza. ESIM: an open event camera simulator. In *Conf. on Robotics Learning (CoRL)*, 2018. 2, 3, 4

[33] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. Events-to-video: Bringing modern computer vision to event cameras. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019. 1, 2, 3, 4, 5, 6

[34] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. High speed and high dynamic range video with an event camera. *arXiv e-prints*, 2019. 2, 3, 4, 5

[35] 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. 7

[36] 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, Apr. 2018. 1

[37] 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, Apr. 2015. 5

[38] 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)*, pages 1731–1740, 2018. 2, 5, 6

[39] S. Tulyakov, F. Fleuret, M. Kiefel, P. Gehler, and M. Hirsch. Learning an event sequence embedding for event-based deep stereo. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1527–1537, 2019. 3

[40] 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, July 2018. 2

[41] 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. 2, 3

[42] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised event-based learning of optical flow, depth, and egomotion. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019. 1, 2, 3