v2e: From Video Frames to Realistic DVS Events

Yuhuang Hu Shih-Chii Liu Tobi Delbruck
Institute of Neuroinformatics, University of Zürich and ETH Zürich, Switzerland
{yuhuang.hu, shih, tobi}@ini.uzh.ch

Abstract

To help meet the increasing need for dynamic vision sensor (DVS) event camera data, this paper proposes the v2e toolbox that generates realistic synthetic DVS events from intensity frames. It also clarifies incorrect claims about DVS motion blur and latency characteristics in recent literature. Unlike other toolboxes, v2e includes pixel-level Gaussian event threshold mismatch, finite intensity-dependent bandwidth, and intensity-dependent noise. Realistic DVS events are useful in training networks for uncontrolled lighting conditions. The use of v2e synthetic events is demonstrated in two experiments. The first experiment is object recognition with N-Caltech 101 dataset. Results show that pretraining on various v2e lighting conditions improves generalization when transferred on real DVS data for a ResNet model. The second experiment shows that for night driving, a car detector trained with v2e events shows an average accuracy improvement of 40% compared to the YOLOv3 trained on intensity frames.

1. Introduction

A Dynamic Vision Sensor (**DVS**) outputs brightness change events [17, 7]. Each pixel holds a memorized brightness value (log intensity value) and continuously monitors if the brightness changes away from this stored value by a specified event threshold. The high dynamic range, fine time resolution, and quick, sparse output make DVS attractive sensors for machine vision under difficult lighting conditions and limited computing power. Since the first DVS cameras, subsequent generations of DVS-type event cameras have been developed; see [6, 18, 25, 7] for surveys.

With the growing commercial development of event cameras and the application of deep learning to the camera output, large DVS datasets are needed for training these networks. Although the number of DVS datasets is growing (see [34]), they are still far fewer than frame-camera datasets. Thus, DVS simulators[15, 21, 26] and transfer learning methods such as [26, 8] were developed to exploit existing intensity and mixed modality [11] frame datasets.

Computer vision papers about event cameras have made incorrect claims such as "event cameras [have] no motion blur" and have "latency on the order of microseconds" [29, 26, 20], perhaps fueled by the titles of papers like [17, 2, 30], which report their best metrics obtained under lab conditions. Recent reviews like [7] are not explicit about the actual behavior under low light conditions. DVS cameras must obey the laws of physics like any other vision sensor: Their output is based on counting photons. Under low illumination conditions, photons become scarce, and therefore counting them becomes noisy and slow.

This paper introduces the v2e toolbox that is aimed at realistic modeling of these conditions and crucial for the deployment of event cameras in uncontrolled lighting conditions. The main contributions of this work are as follows:

1. A description of the operation of the DVS pixel for the computer vision community, together with the behavior of DVS pixels under low illumination (Sec. 3);
2. A demystification of claims in the computer vision literature about lack of motion blur and DVS latency (Secs. 3.2, 3.3);
3. The v2e toolbox[1], which introduces the first DVS pixel model that includes temporal noise, leak events, finite intensity-dependent bandwidth, and Gaussian threshold distribution (Sec. 4);
4. A newly labeled dataset MVSEC-NIGHTL21[2] from a subset of the MVSEC dataset for car detection in night driving conditions (Sec. 5.2).
5. Sec. 6 shows that a network benefits from training with low-light v2e synthetic events. Network generalization is improved when transferring to real event data.

In addition to realistic v2e visual examples in Sec. 6.1, we use v2e toolbox in two computer vision tasks. We first study object recognition in Sec. 6.2 using N-Caltech 101 dataset [23]. The result shows that by training on v2e events

[1]Link: https://github.com/SensorsINI/v2e
[2]Link: http://sensors.ini.uzh.ch/databases.html

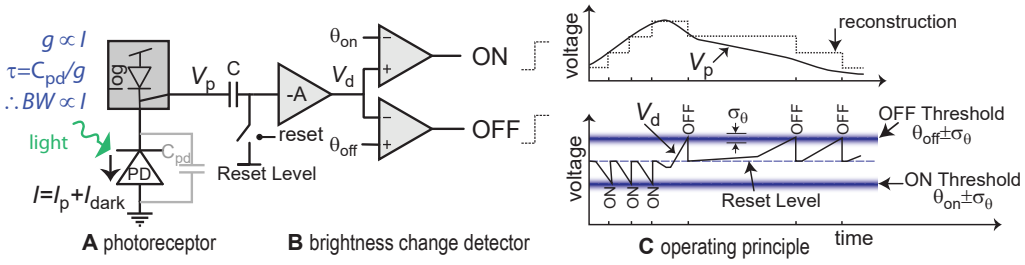

Figure 1. Simplified DVS pixel with principle of operation. Light falling on the photodiode (PD) creates a photocurrent, I_p. More details in text. BW=pixel bandwidth, θ_{ON}=ON threshold, θ_{OFF}=OFF thresholds, V_p=photoreceptor output voltage, V_d= amplified voltage output. Figure adapted from [17, 22].

synthesized in multiple lighting conditions, our classification network surpasses the supervised baseline accuracy after fine-tuning. The second task is a car detection task using the MVSEC dataset [35]. By combining the Network Grafting Algorithm (**NGA**) [11] and synthetic events, we trained a detection network that works well in the day condition. In the night condition, this detector exhibits up to 40% accuracy improvement compared to YOLOv3, which is an intensity frame-based detector (Sec. 6.3).

2. Prior Work for DVS Simulators

Katz et al. [15] used a 200 Hz frame rate camera to synthesize DVS events with a 5 ms time resolution using a simple model of the DVS pixel operation that generated DVS events from the camera intensity samples. The *Event Camera Dataset and Simulator* [21] and the newer *ESIM* [26] toolboxes can be used to generate synthetic DVS events from synthetic video (*e.g.*, using Blender) or image datasets, and thus enabled many recent advances in processing DVS output based on transfer learning. An extension to *ESIM* called *rpg_vid2e* drives *ESIM* from interpolated video frames [8]. *rpg_vid2e* used the same idealistic model of DVS pixels as *ESIM*, that is, it assumed that the DVS pixel bandwidth is at least as large as the upsampled video rate, there is no temporal noise, no leak DVS events, and that the threshold mismatch is uniformly distributed — all of which are invalid for real cameras. Thus *rpg_vid2e* simulated ideal DVS pixels under good lighting, but not realistic DVS pixels under bad lighting, which is an important use case for DVS. The v2e toolbox proposed in this paper is a step towards incorporating a more realistic DVS model in the simulator. By enabling explicit control of the noise and nonideality 'knobs', v2e enables the generation of synthetic datasets covering a range of illumination conditions.

3. DVS Pixel Operation and Biases

Fig. 1 shows a simplified schematic of the DVS pixel circuit. The continuous-time process of generating events is illustrated in Fig. 1C. The DVS pixel bias current parameters control the pixel event threshold and analog bandwidth. In Fig. 1A, the input photocurrent generates a continuous logarithmic photoreceptor output voltage V_{p}. The change amplifier in Fig. 1B produces an inverted and amplified output voltage V_{d}. When V_{d} crosses either the ON or OFF threshold voltage, the pixel emits an event (via a shared digital output that is not shown). The event reset memorizes the new log intensity value across the capacitor C.

The logarithmic response of the photoreceptor comes from the exponential current versus voltage relationship in the feedback diode (gray box in Fig. 1A). The smaller the photocurrent I, the longer the time constant $\tau = C_{\mathrm{pd}}/g$, where C_{pd} is the photodiode capacitance and $g \propto I$ is the conductance of the feedback diode. The event thresholds θ_{ON} and θ_{OFF} are nominally identical across pixels but statistically vary by σ_θ because of transistor mismatch.

3.1. DVS under low lighting

Fig. 2 shows a behavioral simulation based on logarithmic photoreceptor dynamics of a DVS pixel operating under extremely low illumination conditions when a grating consisting of alternating gray and white strips passes over the pixel. Note that in the absence of light, a continuous dark current I_{dark} flows through the photodiode. During the initial "moderately bright" cycles, the signal photocurrent, $I_p \gg I_{\mathrm{dark}}$ and the bandwidth of the photoreceptor, which depends on I_p, is high enough so that V_p can follow the input current fluctuations. The contrast of the signal was set to 2 so that the white part of the grating produced twice the photocurrent compared to the gray part. The pixel makes about five events for each rising and falling edge (the change threshold was set to 0.1 units), but these are spread over time due to the rise and fall time of V_p. In the shadowed "very dark" section, the overall illumination is reduced by 10X. The contrast of the *signal* is still unchanged (the reflectance of the scene is the same as before), but now I_p is comparable to I_{dark}, thus reducing the actual contrast of the current fluctuations. Because I is so small, the bandwidth decreases to the point where the photoreceptor can no longer follow the input current fluctuations and the edges become

extremely motion blurred. Both effects reduce the number of generated brightness change events (to about 2 per edge) and increase their timing jitter. v2e models these effects to produce realistic low-light synthetic DVS events.

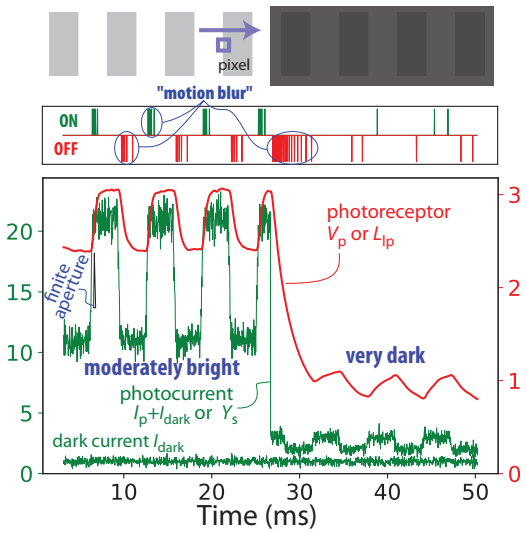

Figure 2. Simulated DVS pixel photoreceptor and resulting ON and OFF events under moderate and extremely low illumination. Both photocurrent and dark current include shot noise which is proportional to the mean current.
Code to reproduce: https://git.io/JOWbG

3.2. Motion blur

For frame-based video, motion blur is simply a low-pass box filter imposed by the finite integration time for the frame. It should be obvious from Fig. 2 that a DVS pixel does not respond instantly to an edge: The finite response time of the photoreceptor blurs the edge. The transition from one brightness level to another is like the response of an RC lowpass filter. The bigger the step, the longer it takes for the pixel to settle to the new brightness value. The result is that a passing edge will result in an extended series of events as the pixel settles down to the new value. This finite response time over which the pixel continues to emit events is the equivalent "motion blur" of DVS pixels. Under bright indoor illumination, typical values for the pixel motion blur are on the order of 1 ms. Under very low illumination, the equivalent pixel motion blur can extend for tens of milliseconds.

Fig. 3 shows measured DAVIS346 [33] DVS motion blur of a moving edge under bright and dark conditions. Users typically view DVS output as 2D frames of histogrammed event counts collected over a fixed integration time (Fig. 3A and C). The frame integration time low-pass filters the DVS output stream just like conventional video cameras. The additional DVS motion blur can be easily observed by lining up the events in a 3D space-time view of the event cloud that

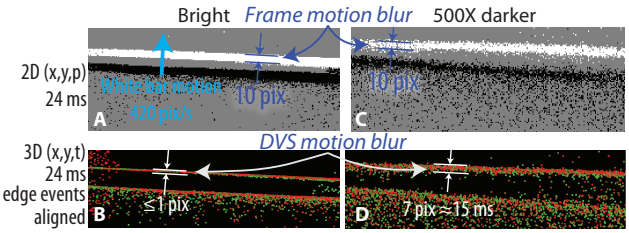

Figure 3. Measured motion blur of real DVS outputs for a moving white bar (speed: 420 pixels/s) on a dark background.

compensates for the motion of the edge (Figs. 3B and D). In this view, the DVS motion blur appears as a thickened edge. In Fig. 3B, the motion blur of the leading white edge is less than 1 pixel (*i.e.*, less than 2 ms), but in Fig. 3D, the blur is about 7 pixels or 15 ms.

3.3. Latency

Quick response time is a clear advantage of DVS cameras, and they have been used to build complete visually servoed robots with total closed-loop latencies of under 3 ms [4, 3]. But it is important to realize the true range of achievable response latency. For example, high-speed USB computer interfaces impose a minimum latency of a few hundred microseconds [4].

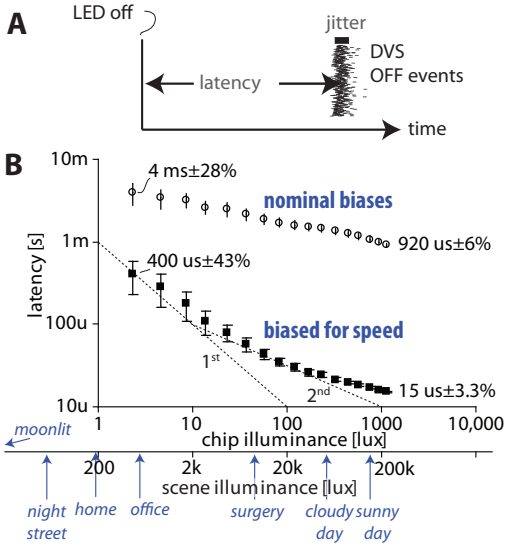

Figure 4. Real DVS latency measurements to turning off blinking LED. **A:** definition of response latency. **B:** measured data. Adapted from [17], with scene illumination axis based on [5].

Added to these computer and operating system latencies are the DVS sensor chip latencies, which are illustrated in Fig. 4 with real DVS data. This experiment recorded the response latency to a blinking LED turning off. The horizontal axis in Fig. 4B is in units of lux (visible photons/area/time): The upper scale is for chip illumination,

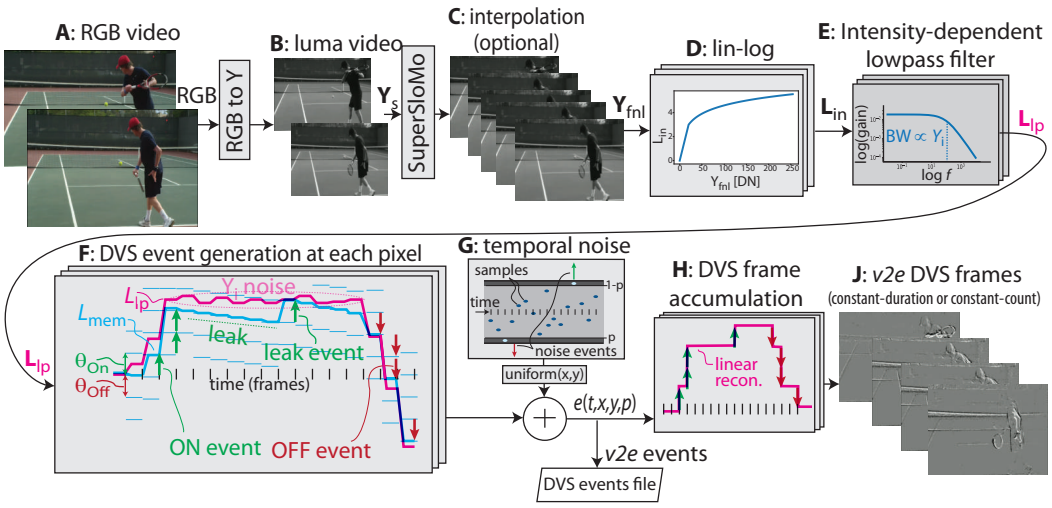

Figure 5. Steps of the v2e DVS event generation (Sec. 4; see lettered headers).

and the lower scale is for scene illumination, assuming 20% scene reflectance and a $f/2.8$ lens aperture ratio [5]. Typical scenarios are listed below the scene illuminance axis. The DVS was biased in two different ways: The "nominal biases" setup used settings that are meant for everyday use of the DVS. With these settings, the DVS pixel bandwidth is limited by the photoreceptor and source follower biases, and thus the DVS latency is only a soft function of intensity. This choice limits noise at low light intensities. The "biased for speed" setup uses higher bias currents for the photoreceptor and source follower to optimize the DVS for the quickest possible response, with the tradeoff of additional noise from a shorter integration time. With this setup, we see that the latency decreases with the reciprocal of intensity. Typical users of DVS will experience real-world latencies in the order of about one to a few ms and latency jitter in the order of 100 μs to 1 ms. The absolute minimum latency is reported in papers as a figure of merit for such sensors (as is customary in the electronics community), but it clearly does not reflect real-world use. Additional discussion on DVS latency can be found in [12]. The v2e lowpass filtering (Sec 4) models these effects.

4. The v2e Toolbox

Fig. 5 shows the steps of the DVS emulation starting from RGB pixel intensity samples.

A-B: Color to luma conversion: v2e starts from a source video that is T seconds long. The frames of color video are automatically converted into M luma frames, $\mathbf{Y}_s = \{\mathbf{Y}_s^{(i)}\}_{i=1}^{M}$, using the ITU-R recommendation *BT. 709* digital video (linear, non-gamma-corrected) color space conversion [13]. Each frame $\mathbf{Y}_s^{(i)}$ is associated with a timestamp t_i where $0 = t_1 < \ldots < t_i < \ldots < t_M = T$. For grayscale video frames, the pixel value is treated as the luma value. After conversion to luma, frames are optionally scaled to the desired output height and width in pixels.

C: Synthetic slow motion: The luma frames are then optionally interpolated using the *Super-SloMo* video interpolation network [14] to increase the temporal resolution of the input video. Super-SlowMo predicts the bi-directional optic flow vectors from consecutive luma frames that are then used to linearly interpolate new frames at arbitrary times between the two input frames. To better estimate flow for luma frames, we retrained Super-SloMo on the Adobe240FPS [32] dataset after converting its RGB frames to luma frames.

If upsampling is not needed, we define the upsampling ratio $U_{\text{fnl}} = 1$. Alternatively, to determine U_{fnl}, the user can choose both a maximum DVS timestamp step $\Delta t_{\max}$ and whether to activate automatic upsampling that was introduced in [8]. The manual upsampling ratio U_{man} is computed from the source video frame rate f_s, $U = \lceil 1/(f_s \times \Delta t_{\max}) \rceil$. For example, $f_s = 60$Hz and $\Delta t_{\max} = 1$ms result in $U_{\text{man}} = 17$. If automatic upsampling is activated, v2e computes the maximum optic flow F in pixels over a batch of frames from the Super-SloMo optic flow estimate. It then computes $U_{\text{fnl}} = \max(U_{\text{man}}, \lceil F \rceil)$ to limit the maximum flow per interframe to 1 pixel. The frame rate after the optional upsampling is $f_{\text{fnl}} = f_s \times U_{\text{fnl}}$. The upsampled frames $\mathbf{Y}_{\text{fnl}} = \{\mathbf{Y}_{\text{fnl}}^{(j)}\}_{j=1}^{M \times U_{\text{fnl}}}$ corresponds to timestamps $0 = t_1 < \ldots < t_j < \ldots < t_{M \times U_{\text{fnl}}} = T$.

For simplicity, the following discusses synthetic event generation for a single DVS pixel. We denote Y as the pixel's luma intensity value in a luma frame $\mathbf{Y}$. Similarly, we use L to represent the pixel's log intensity values in a log intensity frame $\mathbf{L}$.

D: Linear to logarithmic mapping: The method to

generate events from frames is based on [15]. Standard digital video usually represents intensity linearly, but DVS pixels detect changes in log intensity. By default, computer vision uses 8-bit values, equivalent to a limited dynamic range (DR) of $255 = 48$dB. To deal with this limited DR and quantization, we use a lin-log mapping between Y and log intensity value L as illustrated in Fig. 5D. For luma intensity value $Y < 20$ digital number (**DN**), we use a linear mapping from exposure value (intensity) to log intensity. The linearizing part of the conversion function means that small Y values will be converted linearly, reducing quantization noise in the synthetic DVS output.

E: Finite intensity-dependent photoreceptor bandwidth: Since the real DVS pixel has finite analog bandwidth, an optional lowpass filter filters the input L value. This filter models the DVS pixel response under low illumination as discussed in Sec. 3. The DVS pixel bandwidth is proportional to intensity, at least for low photocurrents [17]. v2e models this effect for each pixel by making the filter bandwidth (BW) increase monotonically with the intensity value. Although the photoreceptor and source follower form a 2nd-order lowpass, one pole is usually dominant, and so this filter is implemented by an infinite impulse response (IIR) first-order lowpass filter. The nominal cutoff frequency is $f_{3\text{dBmax}}$ for full white pixels. The filter's bandwidth is proportional to the luma intensity values Y. We denote the filtered L value L_{lp}. The shape of the filter's transfer function is illustrated in Fig. 5E.

To avoid nearly zero bandwidth for small DN pixels, an additive constant limits the minimum bandwidth to about 10% of the maximum value. The update is done by the steps in the supplementary material, along with details of the filter.

Logarithm and temporal contrast threshold: We define the pixel event thresholds for generating ON and OFF events as $\theta_{\text{ON}} > 0$, $\theta_{\text{OFF}} < 0$. Typically the magnitudes of θ_{ON} and θ_{OFF} are quite similar and take on values from $0.1 < |\theta| < 0.4$, *i.e.*, the typical range of adjustable DVS thresholds is approximately from 10% to 50% light intensity change. That is, the change of the logarithmic value $\Delta L = L_{\text{new}} - L_{\text{old}} = \ln(Y_{\text{new}}/Y_{\text{old}})$ corresponds to the intensity change ratio, $Y_{\text{new}}/Y_{\text{old}}$.

The event thresholds are dimensionless and represent a threshold for relative intensity change, *i.e.*, a threshold on the change of the intensity by a ratio relative to the memorized value. These relative intensity changes are produced by scene reflectance changes, which is why this representation is useful for producing events that are informative about the visual input.

F: Event generation model: We assume that the pixel has a memorized brightness value L_{mem} in log intensity and that the new low pass filtered brightness value is L_{lp}. The model then generates a signed integer quantity N_{e} of positive ON or negative OFF events from the change $\Delta L = L_{\text{lp}} - L_{\text{mem}}$ where $N_{\text{e}} = \left\lfloor \frac{\Delta L}{\theta} \right\rfloor$. Details are in the supplementary material.

If ΔL is a multiple of the ON and OFF thresholds, multiple DVS events are generated. The memorized brightness value is updated by N_{e} multiples of the threshold.

Threshold mismatch: The typical value of DVS contrast threshold is about $|\theta_{\text{nominal}}| = 0.3$. Measurements show that the threshold varies with a Gaussian [17] distribution $\sigma_\theta \approx 3\%$ contrast, *i.e.*, before starting the DVS event generation, we store a 2D array of θ_{ON} and θ_{OFF} values drawn from $\theta_{\text{nominal}} + \mathcal{N}(0, \sigma_\theta)$ with $\sigma_\theta = 0.03$.

Hot pixels: DVS sensors always have some 'hot pixels', which continuously fire events at a high rate even in the absence of input. Hot pixels can result from abnormally low thresholds or reset switches with a very high dark current. Hot pixels are created by the frozen threshold sampling, but v2e limits the minimum threshold to 0.01 to prevent too many hot pixel events.

Leak noise events: DVS pixels emit spontaneous ON events called *leak events* [22] with typical rates ≈ 0.1 Hz. They are caused by junction leakage and parasitic photocurrent in the change detector reset switch [22]. v2e adds these leak events by continuously decreasing the memorized brightness value L_{mem} as shown in Fig. 5F. The leak rate varies according to random variations of the event threshold, which decorrelates leak events from different pixels.

G: Temporal noise: The quantal nature of photons results in *shot noise*: If, on average, K photons are accumulated in each integration period, then the average variance will also be K. At low light intensities, the effect of this shot noise on DVS output events increases dramatically, resulting in balanced ON and OFF shot noise events at above 1 Hz per pixel rate. v2e models temporal noise using a Poisson process. It generates ON and OFF temporal noise events to match a noise event rate R_n (default 1 Hz). To model the increase of temporal noise with reduced intensity, the noise rate R_n is multiplied by a linear function of luma $0 < Y \leq 1$ that reduces noise in bright parts by a factor $0 < c < 1$ (default $c = 0.25$). This modified rate r is multiplied by the time step Δt to obtain the probability $p = r \times \Delta t \ll 1$ that will be applied to the next sample. For each sample, a uniformly distributed number in the range 0-1 is compared against two thresholds $[p, 1-p]$ as illustrated in Fig. 5G to decide if an ON or OFF noise event is generated. These noise events are added to the output and reset the pixels. The complete steps are described in the supplementary material.

Event timestamps: The timestamps of the interpolated frames are discrete. Given two consecutive interpolated frames, the timestamps of events are evenly distributed in between (t_j, t_{j+1}).

5. Data Preparation and Datasets

5.1. Event voxel grid representation

Our experiments use the event voxel grid method to convert N events into a 3D representation with size $H \times W \times D$ [27, 11] to use as the network input for the Sec. 6 results. H and W are sensor height and width dimensions. D is a hyperparameter that defines the number of slices of the output voxel grid. These slices are effectively frames of histogrammed DVS events where each slice has an exposure time of $(N/R)/D$, where R is the average event rate.

5.2. Datasets

N-Caltech 101 [23] is an event-based object recognition dataset generated from the Caltech 101 object recognition dataset [16]. Each image in the Caltech 101 dataset was recorded by a DVS for 300 ms using three 100 ms-long triangular saccades. The dataset contains 8,709 object samples over 101 object categories. For experiments in Sec. 6.2, we synthesized v2e synthetic datasets under three different conditions, namely, Ideal (no non-ideality), Bright (modest amount of noise), and Dark (most noise). The synthesis parameters are in Table. 1. Each recording of the synthetic dataset has the same duration and number of saccades as in the original N-Caltech 101 dataset. When preparing event voxel grids, we choose $D = 15$ slices and use all events of each recording. Each slice in an event voxel grid corresponds to 20 ms. (If the DVS motion blur is smaller than the slice duration, it would not be easily noticeable.)

MVSEC [35] is a stereo driving dataset captured from two DAVIS346 event cameras. The cameras recorded both intensity frames and events. We used the intensity frames in `outdoor_day_2` to synthesize the events for the day condition (parameters in Table 1). We set $N = 25,000$ events and $D = 10$ when preparing event voxel grids. This setting is identical to [11]. There are 5,900 intensity frame and event voxel grid pairs generated from the v2e day recording. These pairs are used as training samples for NGA. For experiments in Sec. 6.3, we also used the training and validation datasets from [11] that were generated from the real events. Additionally, we generate 2,000 pairs from the MVSEC `outdoor_night_1` recording. The first 1,600 pairs are used as the night training samples. The night recording in MVSEC is not labeled. We hand-labeled the remaining 400 pairs for cars to create a night validation set.

6. Results

6.1. Qualitative demonstrations

We use a `chair` example from the N-Caltech 101 dataset to demonstrate qualitatively that v2e can realistically synthesize DVS events under bright and dark illumination. Fig. 6 compares 20 ms snapshots of the DVS output. We displayed the 300 ms chair video at a high frame rate on a monitor (after ensuring the monitor had no backlight flicker) and recorded the DAVIS346 DVS output with the lens aperture fully open (Fig. 6A) and closed down to 450X lower luminance (Fig. 6C). Then we modeled DVS events from the chair video with v2e using bright (Fig. 6B) and dark settings (Fig. 6D). The motion blurring and noise under low illumination are clearly visible and v2e produces qualitatively similar effects as reduced lighting.

Table 1. v2e synthesis parameters for N-Caltech 101 and MVSEC datasets. θ and σ_θ represent event threshold and threshold variation respectively.

Dataset	**N-Caltech 101**			**MVSEC**
Condition	Ideal	Bright	Dark	Day
θ	0.05-0.5	0.05-0.5	0.05-0.5	$(0.73_{\text{ON}}, 0.43_{\text{OFF}})$
σ_θ	$\times^\dagger$	$\mathcal{U}(\cdot,\cdot)^\ddagger$	$\mathcal{U}(\cdot,\cdot)$	0.03
Shot Noise	$\times$	$\times$	1-10 Hz	2 Hz
Leak Events	$\times$	0.1-0.5 Hz	$\times$	0.5 Hz
Cutoff Freq.	$\times$	200 Hz	10-100 Hz	200 Hz

† '$\times$' sign means not applicable.
‡ $\mathcal{U}(\cdot,\cdot) = \text{Uniform}(\min(0.15\,\theta, 0.03), \min(0.25\,\theta, 0.05))$.

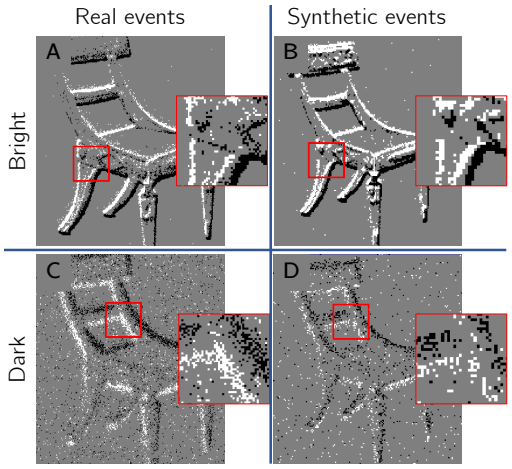

Figure 6. `chair` example under bright and dark conditions. **A** and **C** are generated from real events; **B** and **D** are from v2e synthetic events. Zoom-in views of the selected regions are also displayed.

6.2. N-Caltech 101 object recognition

This section presents experiments that show the use of v2e synthetic data in an N-Caltech 101 object recognition task. Unless otherwise mentioned, models in this section were trained for 100 epochs with a batch size of 4 using AdamW [19] optimizer. The initial learning rate is 10^{-4}, and the learning rate decreases by 10 at every 30 epochs. The recognition network is a ResNet34 [10] pretrained on ImageNet. We replaced the first layer to accommodate the voxel grid input and re-initialized the classification layer. For every class of each dataset, we randomly selected 45%

as the train samples, 30% as validation samples, and 25% as test samples. This data split is fixed for all datasets in this section. Every model was evaluated on the same test dataset that contains real event recordings from the original N-Caltech 101 dataset.

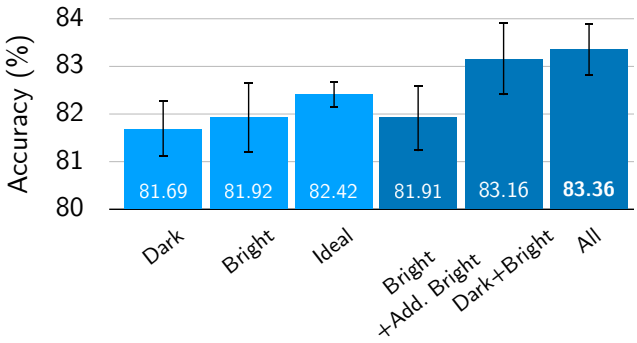

Figure 7. N-Caltech 101 test accuracy. The x-axis indicates different v2e synthetic training datasets and combinations. The five-run averaged accuracy is showed at the bottom of each bar.

Fig. 7 studies the effect of the event camera non-idealities. With training data from events synthesized under the three conditions: Ideal, Bright, and Dark, we found that the accuracy of the ResNet34 network is highest for the Ideal condition because there is no noise. Because the original N-Caltech 101 was recorded in bright conditions, this result was expected. However, by combining synthesized events from two (Bright+Dark) or three conditions for the training data, the test accuracy is better than when training on a single condition. For instance, 83% for All *vs*. 82% for Ideal. To make the accuracy scores in Fig. 7 comparable, the number of epochs for the combined datasets were reduced proportionally so that the same total number of samples were presented during training. We also synthesized additional v2e samples under Bright condition (labeled as Add. Bright), which are added to the previously generated Bright dataset to form a new 'Bright' training set. From Fig. 7, it is clear that the accuracies are similar (Bright *vs*. Bright+Add. Bright). This result shows that by including a wide range of synthesis parameters, the gap between network accuracy from real and synthetic event data can be reduced.

Table 2 presents three groups of accuracies on the N-Caltech 101 dataset. In the first group, we first established the baseline (86.74%) by training a model using real events. By combining three synthesis conditions, the model trained only on synthetic data reaches 83.36%, which is only 3% lower than the baseline. After training with synthetic data, we fine-tuned this trained model on the real events until convergence with learning rate 10^{-6}. This fine-tuned model reaches 87.85% accuracy, which is significantly better than the baseline. This group of results shows that the pretraining on various v2e conditions improves model generalization on real event data. Compared to the second group of the table, our fine-tuned model accuracy is also higher than recent literature [31, 1, 9]. The most similar prior art is Ref. [8] and their results are summarized in the third group of Table. 2. With the same ResNet34 model, our baseline and synthetic data trained model accuracies are higher than theirs. However, our model accuracy after fine-tuning is lower than their model that reached 90.4%. (The Ref. [8]'s dataset split and training code are not available, so we could not replicate their results.)

Table 2. N-Caltech 101 test accuracy on the classification task. Reported results are averaged over five runs.

Method	Training Dataset	Accuracy (%)
ResNet34 (ours)	real events	86.74±0.54
ResNet34 (ours)	v2e-All	83.36±0.76
ResNet34 (ours)	+fine-tune	**87.85±0.12**
HATS [31]	real events	64.20
RG-CNNs [1]	real events	65.70
EST [9]	real events	81.70
ResNet34 [8]	real events	86.30
ResNet34 [8]	synthetic events	78.20
ResNet34 [8]	+fine-tune	**90.40**

6.3. MVSEC car detection

In this section, we used v2e synthetic data to train a car detection network using the Network Grafting Algorithm (**NGA**) [11]. The NGA algorithm enables training of an event-driven network using paired synchronous intensity frames and brightness change events (here generated by v2e). We followed the same setup and the training schedule as in [11] using the YOLOv3 detection network [28].

Table 3. MVSEC car detection results in average precision. The five-run averaged scores are shown. GN refers to grafted network based on YOLOv3.

Model	Training Dataset	AP_{50}
Test on real day events		
GN-B1 [11]	real day events	70.35±0.51
GN-D1 (ours)	v2e day events	62.52±1.15
GN-D2 (ours)	+fine-tune	69.82±0.64
GN-B2 [11]	real day events (10%)	47.88±1.86
GN-D3 (ours)	+fine-tune (10%)	68.55±0.28
Test on real night intensity frames		
YOLOv3	intensity frames	25.94
Test on real night events		
GN-B1 [11]	real day events	35.67
GN-N1 (ours)	real night events	29.38±1.08
GN-D1 (ours)	v2e day events	**36.41±2.90**

Table 3 summarizes our findings in two groups. In the first group, compared to the model trained with the real day events, our model GN-D1 trained with v2e day events gives

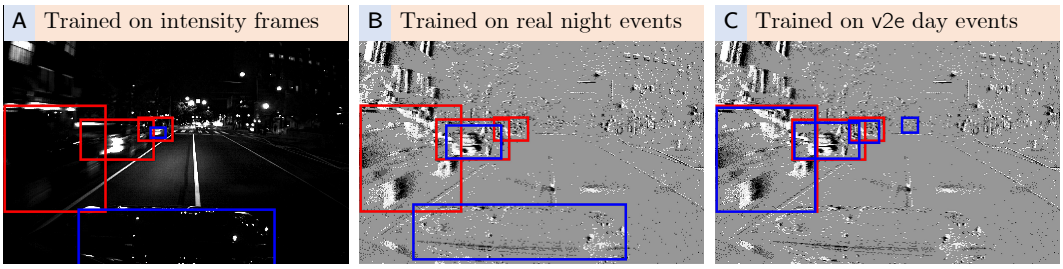

Figure 8. Car detection examples on the MVSEC night recording. The groundtruth bounding boxes are in red while the predicted boxes in blue. The detection quality in **C** is visually better than **B** and **A**.

an average precision (AP_{50}) that is 11% lower than baseline GN−B1's accuracy 70.35. We fine-tuned GN−D1 with the real day events for 10 epochs to reduce this accuracy gap. We find that the fine-tuned model GN−D2's accuracy of 69.82 is on par with the baseline GN−B1 [11]. We also fine-tuned GN−D1 with only 10% of real day event data (labelled as GN−D3). Compared to [11], GN−D3's accuracy of 68.55 is 43% higher than the baseline GN−B2's AP_{50} of 47.88. This result shows that pretraining on v2e synthetic events is beneficial for model generalization.

The second study (results in Table 3) is to determine whether the DVS allows the network output to be more invariant to different illumination levels compared to intensity cameras. A recent paper [24] only observed this phenomenon qualitatively. Here, we used our newly labeled real night validation set (see Sec. 5.2) to quantitatively validate this hypothesis. First, we evaluated the accuracy of the original YOLOv3 on the night intensity frames for the baseline AP_{50} of 25.94. Second, by NGA training on paired real night intensity frames and events, the grafted network GN−N1 reaches an improved accuracy of 29.38, showing that the higher DR of the event camera is useful. Finally, we used model GN−D1. The AP_{50} is 36.41, which is 40% better than 25.94 that is obtained by using the original YOLOv3. The GN−D1's accuracy is also higher than a baseline accuracy which is obtained by running the GN−B1 model.

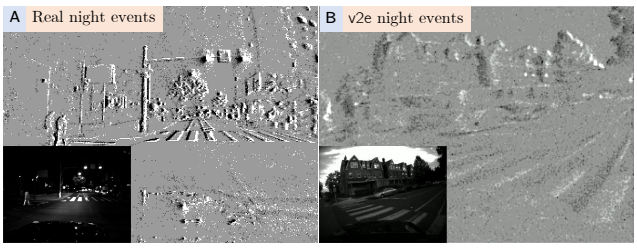

Figure 9. **A**: the headlight of the car illuminates mostly the center of the scene; **B**: v2e motion-blurred the entire scene uniformly.

The improved result of GN−D1 for night condition can be explained as follows: **1.** NGA works better for high-quality intensity frames and sharp DVS event inputs, but the real night intensity frames are severely underexposed and motion blurred (Fig. 8A, B); **2.** because the event camera is more robust to different lighting conditions than intensity cameras, GN−D1 trained with better exposed v2e day intensity frames transferred better to the night scenes (Fig. 8C).

We also used v2e to synthesize night events from the MVSEC day recording by setting the cutoff frequency to 10 Hz. We were surprised to find that by training with these synthetic night events, the accuracy is even lower than the AP_{50} of the original YOLOv3. This counterintuitive result can be understood with Fig. 9. In the night recording, the front area was illuminated by the car's headlight (Fig. 9A) while v2e uniformly motion-blurred the entire scene (Fig. 9B). The grafted network trained on the synthetic night events could not infer sharp features as available in the real night events.

7. Conclusion

This paper described the v2e toolbox that can synthesize realistic events from intensity frames. By modeling the noise and motion blur non-idealities of event cameras, we hope to debunk common myths about the event camera and bridge the gap between the simulation model and the real sensor. With this tool, we hope to stimulate more research in understanding and modeling these non-idealities.

Our experiments showed qualitative and quantitative evidence that v2e can generate realistic DVS events and that the synthetic events are useful. Along with simulators such as *ESIM* [26] and its extension [8], v2e provides practical ways to sample a large amount of diverse synthetic DVS events for building more robust event-driven algorithms.

Compared to [24] that only showed visual examples, this paper, for the first, time proved that the event camera is more robust under low lighting conditions for a car detection task. The results are encouraging for further research in using event cameras under difficult lighting conditions.

Acknowledgments This work was funded by the Swiss National Competence Center in Robotics (NCCR Robotics).

References

[1] Y. Bi, A. Chadha, A. Abbas, E. Bourtsoulatze, and Y. Andreopoulos. Graph-based object classification for neuromorphic vision sensing. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 491–501, 2019. 7

[2] C. Brandli, R. Berner, M. Yang, S. Liu, and T. Delbruck. A 240×180 130 db 3 μs latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits*, 49(10):2333–2341, 2014. 1

[3] J. Conradt, M. Cook, R. Berner, P. Lichtsteiner, R. J. Douglas, and T. Delbruck. A pencil balancing robot using a pair of aer dynamic vision sensors. In *2009 IEEE International Symposium on Circuits and Systems*, pages 781–784, 2009. 3

[4] T. Delbruck and M. Lang. Robotic goalie with 3 ms reaction time at 4% CPU load using event-based dynamic vision sensor. *Frontiers in Neuroscience*, 7:223, 2013. 3

[5] T. Delbruck and N. Mascarenhas. Notes on practical photometry. https://www.ini.uzh.ch/~tobi/wiki/doku.php?id=radiometry, 1997–2017. 3, 4

[6] T. Delbrück, B. Linares-Barranco, E. Culurciello, and C. Posch. Activity-driven, event-based vision sensors. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pages 2426–2429, 2010. 1

[7] G. Gallego, T. Delbruck, G. M. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza. Event-based Vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2020. 1

[8] D. Gehrig, M. Gehrig, J. Hidalgo-Carrió, and D. Scaramuzza. Video to Events: Recycling video datasets for event cameras. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3583–3592, 2020. 1, 2, 4, 7, 8

[9] D. Gehrig, A. Loquercio, K. Derpanis, and D. Scaramuzza. End-to-end learning of representations for asynchronous event-based data. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5632–5642, 2019. 7

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 6

[11] Y. Hu, T. Delbruck, and S-C. Liu. Learning to exploit multiple vision modalities by using grafted networks. In A. Vedaldi, H. Bischof, T. Brox, and J-M. Frahm, editors, *Computer Vision – ECCV 2020*, pages 85–101, Cham, 2020. Springer International Publishing. 1, 2, 6, 7, 8

[12] iniVation. Understanding the performance of neuromorphic event-based vision sensors. Technical report, iniVation, May 2020. 4

[13] International Telecomunication Union. Parameter values for the hdtv standards for production and international programme exchange. Technical report, International Telecomunication Union, 2015. BT.709. 4

[14] H. Jiang, D. Sun, V. Jampani, M. Yang, E. Learned-Miller, and J. Kautz. Super SloMo: High quality estimation of multiple intermediate frames for video interpolation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9000–9008, 2018. 4

[15] M. L. Katz, K. Nikolic, and T. Delbruck. Live demonstration: Behavioural emulation of event-based vision sensors. In *2012 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 736–740, 2012. 1, 2, 5

[16] Li Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 Conference on Computer Vision and Pattern Recognition Workshop*, pages 178–178, 2004. 6

[17] P. Lichtsteiner, C. Posch, and T. Delbruck. A 128× 128 120 db 15 μs latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits*, 43(2):566–576, 2008. 1, 2, 3, 5

[18] S-C. Liu, T. Delbruck, G. Indiveri, A. Whatley, and R. Douglas. *Event-Based Neuromorphic Systems*. John Wiley & Sons, Dec. 2014. 1

[19] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 6

[20] A. Mitrokhin, C. Ye, C. Fermüller, Y. Aloimonos, and T. Delbruck. EV-IMO: Motion segmentation dataset and learning pipeline for event cameras. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6105–6112, 2019. 1

[21] E. Mueggler, H. Rebecq, G. Gallego, T. Delbruck, and D. Scaramuzza. The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and slam. *The International Journal of Robotics Research*, 36(2):142–149, 2017. 1, 2

[22] Y. Nozaki and T. Delbruck. Temperature and parasitic photocurrent effects in dynamic vision sensors. *IEEE Transactions on Electron Devices*, 64(8):3239–3245, 2017. 2, 5

[23] G. Orchard, A. Jayawant, G. K. Cohen, and N. Thakor. Converting static image datasets to spiking neuromorphic datasets using saccades. *Frontiers in Neuroscience*, 9:437, 2015. 1, 6

[24] E. Perot, P. de Tournemire, D. Nitti, J. Masci, and A. Sironi. Learning to detect objects with a 1 megapixel event camera. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 16639–16652. Curran Associates, Inc., 2020. 8

[25] C. Posch, T. Serrano-Gotarredona, B. Linares-Barranco, and T. Delbruck. Retinomorphic Event-Based Vision Sensors: Bioinspired cameras with spiking output. *Proceedings of the IEEE*, 102(10):1470–1484, 2014. 1

[26] H. Rebecq, D. Gehrig, and D. Scaramuzza. ESIM: an open event camera simulator. In Aude Billard, Anca Dragan, Jan Peters, and Jun Morimoto, editors, *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pages 969–982. PMLR, 29–31 Oct 2018. 1, 2, 8

[27] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza. Events-To-Video: Bringing modern computer vision to event cameras. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3852–3861, 2019. 6

[28] J. Redmon and A. Farhadi. YOLOv3: An incremental improvement. *CoRR*, abs/1804.02767, 2018. 7

[29] C. Scheerlinck, H. Rebecq, D. Gehrig, N. Barnes, R. E. Mahony, and D. Scaramuzza. Fast image reconstruction with an event camera. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 156–163, 2020. 1

[30] T. Serrano-Gotarredona and B. Linares-Barranco. A 128 $\times$ 128 1.5% contrast sensitivity 0.9% fpn 3 μs latency 4 mw asynchronous frame-free dynamic vision sensor using transimpedance preamplifiers. *IEEE Journal of Solid-State Circuits*, 48(3):827–838, 2013. 1

[31] A. Sironi, M. Brambilla, N. Bourdis, X. Lagorce, and R. Benosman. HATS: Histograms of averaged time surfaces for robust event-based object classification. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1731–1740, 2018. 7

[32] S. Su, M. Delbracio, J. Wang, G. Sapiro, W. Heidrich, and O. Wang. Deep video deblurring for hand-held cameras. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 237–246, 2017. 4

[33] G. Taverni, D. Paul Moeys, C. Li, C. Cavaco, V. Motsnyi, D. San Segundo Bello, and T. Delbruck. Front and back illuminated dynamic and active pixel vision sensors comparison. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 65(5):677–681, 2018. 3

[34] Various contributors. Event-based vision resources. https://github.com/uzh-rpg/event-based_vision_resources, 2020. 1

[35] A. Z. Zhu, D. Thakur, T. Özaslan, B. Pfrommer, V. Kumar, and K. Daniilidis. The Multivehicle Stereo Event Camera Dataset: An event camera dataset for 3d perception. *IEEE Robotics and Automation Letters*, 3(3):2032–2039, 2018. 2, 6