

# **v2e: From Video Frames to Realistic DVS Events**

## ***Supplementary Material***

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

### **1. Overview**

- `video_presentation.mp4` is a short video that summarizes v2e's features and results of the paper.
- Folder `mvsec-night-val-labels` includes two examples of labels and the plotting script for demonstration.
- Section 2 presents additional v2e technical details that may be useful to readers.
- Section 3 presents additional training details and statistics.
- Section 4 presents additional comparison between the v2e MVSEC day events and the real MVSEC day events.
- Section 5 discussed two other DVS non-idealities that are not currently included in the v2e toolbox.

## 2. Additional Details for the v2e toolbox

**Linear to logarithmic mapping** To model the DVS sensor, we first convert the light intensity value from linear to logarithmic scale using a lin-log mapping. Equation 1 converts the digital number (DN) intensity sample luma intensity value  $Y \in [0, 255]$  into the logarithmic value  $L$ :

$$L = f(Y) = \begin{cases} \frac{Y}{20} \ln 20 & \text{if } Y < 20 \\ \ln Y & \text{if } Y \geq 20 \end{cases} \quad (1)$$

Dark pixels output small DNs, even down to DN 0. By default, computer vision uses 8-bit values, which limits dynamic range to  $255 = 48 \text{ dB}$ . To deal with this limited range and quantization,  $f(Y)$  in Eq. 1 is a piece-wise linear-logarithmic function, since the log function is sensitive to small values near zero. For example, the logarithmic change from DN 0 to DN 1 is infinite, and the change from DN 1 to DN 2 is a factor of 2. These huge changes of log intensity could create a huge number of unrealistic noise events. Therefore, for the range less than DN=20, we use a linear mapping from exposure value (intensity) to log intensity. The mapping is joined at  $Y = 20$  DN. The maximum output value is  $L_{\max} = \ln(255) = 5.54$ . The linearizing part of the conversion function means that small DNs will be converted linearly, reducing noise in the synthetic DVS output.

**Finite intensity-dependent photoreceptor bandwidth** Since the real DVS pixel has finite analog bandwidth, an optional low-pass filter filters the input  $L$  values. Although the photoreceptor circuit is technically a 2nd-order system consisting of a cascade of two filters, one pole is nearly always dominant in practice and so we model it with a 1st-order lowpass. This cutoff models the DVS pixel response under low illumination. DVS pixel bandwidth is proportional to intensity, at least for low photocurrents [1]. v2e models this effect for each pixel by making the filter bandwidth (BW) that increases monotonically with the intensity value.

This filter is implemented by an IIR lowpass on pixel values in the interpolated brightness values. The transfer function of the filter in continuous time form is  $H(s) = \frac{1}{(\tau s + 1)}$  where  $\tau = 1/f_{3dB}$ ,  $f_{3dB}$  is the cutoff frequency, and  $s = 2\pi f$ . The shape of this transfer function is illustrated in Fig. 5.

This first-order RC lowpass filter has a nominal cutoff frequency of  $f_{3dB\max}$  for full white pixels. This bandwidth is proportional to the luma  $Y_s$ . To avoid nearly zero bandwidth for small DN pixels, an additive constant limits the minimum bandwidth to about 1/10 of the maximum. The update is done by the steps in (2-6):

$$f_{3dB} = ((Y_s + 20)/275) \times f_{3dB\max} \quad (2)$$

$$\tau = 1/(2\pi f_{3dB}) \quad (3)$$

$$\Delta t = t_{t+1} - t_t \quad (4)$$

$$\epsilon = \min(\Delta t/\tau, 1) \quad (5)$$

$$L_{lp} \leftarrow (1 - \epsilon)L_{lp} + \epsilon L_{in} \quad (6)$$

where  $L_{lp}$  is the lowpass-filtered brightness output, and  $L_{in}$  is the input brightness value. The value 275 is chosen to result in  $f_{3dB\max}$  for  $Y_s = 255$ .

**Event generation model:** Given that a pixel has a memorized brightness value  $L_{\text{mem}}$ , and that the new low pass filtered brightness value is  $L_{lp}$ , then the basic model of event generation generates a signed integer quantity  $N_e$  of positive ON or negative OFF events by the recipe (7):

$$\begin{aligned} \Delta L &= L_{lp} - L_{\text{mem}} \\ \theta &= \begin{cases} \theta_{\text{ON}} & \text{if } \Delta L \geq 0 \\ \theta_{\text{OFF}} & \text{if } \Delta L < 0 \end{cases} \\ N_e &= \text{floor}\left(\frac{\Delta L}{\theta}\right) \\ L_{\text{mem}} &\leftarrow L_{\text{mem}} + N_e \times \theta \end{aligned} \quad (7)$$

In (7),  $N_e$  denotes the signed number of generated ON or OFF events. If  $N_e$  is positive it means ON events, and if negative it means OFF events. The floor( $x$ ) function takes the value closer to zero for both signs, e.g. floor(1.3) = 1 and floor(-2.6) = -2.

If the change is multiple times the threshold, then multiple DVS events are generated. The memorized brightness value is updated by an  $N_e$  multiple of the threshold. These events are spread over the time between this input frame and the next one (see below for details).

**Leak noise events** DVS pixels emit spontaneous ON events called *leak events* [3]. They are caused by junction leakage and parasitic photocurrent in the change detector reset switch. They occur at a rate typically about 0.1 Hz. v2e adds these leak events by decreasing the memorized brightness value, by using (8-10):

$$\Delta t = t_{t+1} - t_t \quad (8)$$

$$\delta_{\text{leak}} = \Delta t R_{\text{leak}} / \theta_{\text{ON}} \quad (9)$$

$$L_{\text{mem}} \leftarrow L_{\text{mem}} - \delta_{\text{leak}} \quad (10)$$

where  $R_{\text{leak}}$  is the nominal leak rate and  $\theta_{\text{ON}}$  is the nominal scalar ON threshold. Even if  $L_{\text{lp}}$  does not change, eventually  $L_{\text{mem}}$  drifts away from  $L_{\text{lp}}$  by  $\theta_{\text{ON}}$ , and the pixel emits an ON event. The paper Fig. 5 illustrates one of these leak events being generated by the gradual change of  $L_{\text{mem}}$ .  $\theta_{\text{ON}}$  is the individual pixel ON-event threshold. This way, the leak rate varies according to the random variation of the event threshold and the leak events become desynchronized. To make leak events appear from start of simulation, if the leak rate is nonzero, then each pixel's  $L_{\text{mem}}$  is initialized to a uniformly-distributed random fraction of  $\theta_{\text{ON}}$  below the initial value of  $L_{\text{lp}}$ .

**Temporal noise:** The quantal nature of photons results in *shot noise*: If on the average  $N$  photons are accumulated in each integration period, then the variance around the average will also be  $N$ . Shot noise appears in all vision sensors. In conventional imagers that accumulate for a fixed integration time, shot noise gets larger as the signal gets larger, but its effect on contrast shrinks. That is because as  $N$  grows, the standard deviation only grows as  $\sqrt{N}$ , so the relative noise shrinks as  $1/\sqrt{N}$ .

DVS pixels are different. At low light intensities, DVS pixel integration time is approximately proportional to the intensity, which means that a DVS pixel integrates over a constant number of photons. It means that DVS pixel photoreceptors have total noise power that is constant with intensity. As the intensity increases, the total noise is spread over more bandwidth. It is often observed that DVS recordings show more noise in the dark parts of the scene. The reason for this is that more of the total noise power is concentrated in lower frequencies that lie within the passband of the subsequent change detector.

v2e models temporal noise using a Poisson process. It generates ON and OFF temporal noise events to match an observed noise event rate  $R_n$ . Fig. 5 shows how it works: For each sample, a uniformly-distributed number in range 0-1 is compared with two thresholds to decide if an ON or OFF noise event is generated.

To model the increase of temporal noise with reduced intensity, observed noise rate  $R_n$  for dark parts of the scene is multiplied by a linear function of luma  $0 < Y \leq 1$  that reduces noise in bright parts by a factor  $0 < F < 1$  (default  $F = 0.25$ ). This modified rate  $r$  is multiplied by the timestep  $\delta t$  to obtain the probability  $p < 1$  that will applied to the next sample. The complete steps are (11-15):

$$r = ((F - 1) \times Y + 1) \times R_n \quad (11)$$

$$p = r \times \delta t \quad (12)$$

$$u = \text{uniformly distributed sample in } [0, 1) \quad (13)$$

$$u < p: \text{Generate OFF event} \quad (14)$$

$$u > (1 - p): \text{Generate ON event} \quad (15)$$

These noise events are injected to the output, and the pixel is reset the same way that a ‘real’ input would reset it. This way, the noise events do not discard changes in the input  $Y$  signal.

**v2e output** v2e outputs a variety of formats. The basic output is a stream of events  $e(t, x, y, p)$  (either in text or jAER .aeadat format), and a DVS AVI video that accumulates the signed DVS events starting from a gray image, at a specified frame rate (*constant-duration*), or with two variable-frame rate count-based exposure strategies, *constant-count* and *area-event* [2].

**Throughput performance** v2e processes video about 20 to 100 times slower than real time on low-end GPU hardware. For example, a laptop Nvidia MX150 GPU on 2019 Hauwei Matebook Pro X running Ubuntu 18.04 with python 3.7 processed a source video shot at 50 Hz with 6X upsampling and all DVS pixel effects activated at about 1.35 frame/s, *i.e.*, a slowdown of 37. Processing time is dominated by frame interpolation, so faster inference hardware would speed up the processing. v2e v1.3 Batch mode and numba optimization provides speedup of about 2X over single frame and native python.

### 3. Additional Details for the N-Caltech 101 Experiments

The event rates of the original N-Caltech 101 dataset and the v2e datasets are presented in Table 1.

The raw data that was used for producing Fig. 7 in the paper is shown in Table. 2.

**Additional Training Details** When fine tuning the network pretrained on the v2e events with the real events, an additional dropout layer was added before the classification layer to prevent overfitting. The dropout rate is set to 0.85.

Table 1. v2e data statistics.

Dataset	Event Rate (events/second)
N-Caltech 101	$383.68 \pm 194.21$ K
v2e N-Caltech 101 (Ideal)	$1679.49 \pm 2389.07$ K
v2e N-Caltech 101 (Bright)	$1290.45 \pm 1770.25$ K
v2e N-Caltech 101 (Dark)	$906.02 \pm 989.02$ K

Table 2. Raw data for producing Fig. 7.

Training Condition(s)	Test Acc. (%)
Ideal	$82.42 \pm 0.28$
Bright	$81.92 \pm 0.74$
Dark	$81.69 \pm 0.60$
Bright+Add. Bright	$81.91 \pm 0.67$
Bright+Dark	$83.16 \pm 0.39$
All	<b><math>83.36 \pm 0.76</math></b>

### 4. Additional Details for the MVSEC Experiments

#### 4.1. Compare v2e-MVSEC-Day and MVSEC-Day

The noise occurs at 2.5Hz, therefore, the number of noise events per second is close to  $346 \times 260 \times 2.5 = 224900$  events.

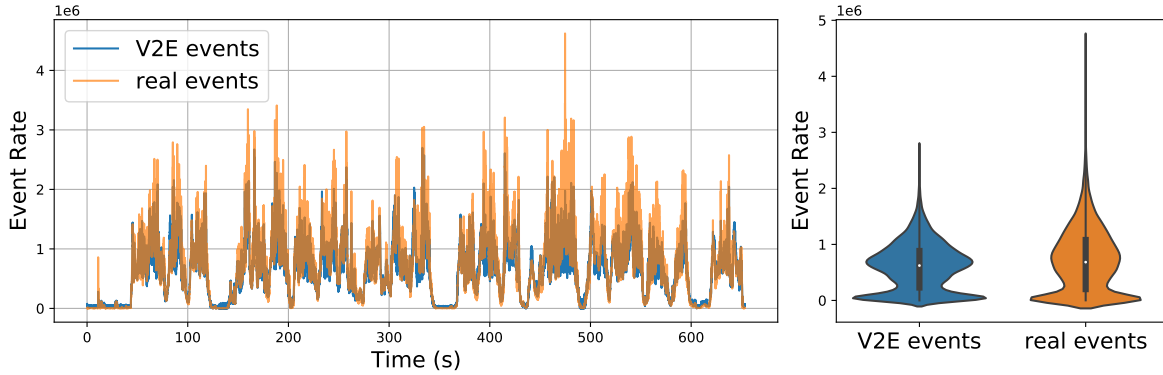


Figure 1. Compare v2e-MVSEC Day and MVSEC-Day events. This is after subtracting the noise events.

## 5. Other DVS non-idealities

**Refractory period** The real DVS pixel has an adjustable refractory period, which is used to limit the maximum pixel event rate. After each event is detected, the reset switch transistor is connected for a finite time  $T_{\text{refr}}$ . During this time, the change amplifier ignores changes in the log intensity. To model finite refractory period, a user could write code to ignore frames subsequent to an event for a period  $T_{\text{refr}}$  after the event is generated.

**Finite event output bandwidth** v2e does not model that DVS have a maximum output event rate, e.g. about 10 MHz for the DAVIS346 camera used here, which is determined by a combination of on-chip arbitration circuits and computer interface limitations.

## References

- [1] P. Lichtsteiner, C. Posch, and T. Delbruck. A  $128 \times 128$  120 db 15  $\mu\text{s}$  latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits*, 43(2):566–576, 2008. [2](#)
- [2] M. Liu and T. Delbruck. Adaptive Time-Slice Block-Matching optical flow algorithm for dynamic vision sensors. In *BMVC 2018*, pages 12–16, Nescatle upon Tyne, Sept. 2018. [3](#)
- [3] Y. Nozaki and T. Delbruck. Temperature and parasitic photocurrent effects in dynamic vision sensors. *IEEE Transactions on Electron Devices*, 64(8):3239–3245, 2017. [3](#)