PIX2NVS: PARAMETERIZED CONVERSION OF PIXEL-DOMAIN VIDEO FRAMES TO NEUROMORPHIC VISION STREAMS

Yin Bi and Yiannis Andreopoulos

Dept. of Electronic and Electrical Engineering
University College London (UCL), London, U.K.

ABSTRACT

We propose and make available a generic pixel-to-neuromorphic vision stream (PIX2NVS) framework in order to allow for the generation of neuromorphic data streams from conventional pixel-domain video frames. In order to quantify the accuracy of our framework against experimentally-derived NVS data from previous work, we also propose and validate two metrics, the Chamfer distance and ϵ-repeatability. The most important application of PIX2NVS will be in the generation of artificial NVS from large annotated video frame collections used in machine learning research, e.g., YouTube-8M, YFCC100m, YouTube-BoundingBoxes, thereby transferring these datasets to the neuromorphic domain.

Index Terms— dynamic vision sensing, neuromorphic vision, software, video analysis

1. INTRODUCTION

Due to the unattainable bandwidth and energy requirements of conventional video cameras, hardware designs of neuromorphic vision sensors, a.k.a., dynamic vision sensors (DVS) or silicon retinas [1, 2], have been proposed recently. As shown in the example of Fig. 1, silicon retina cameras produce a stream of coordinates and timestamps of reflectance events triggering on or off in an asynchronous manner, i.e., when the logarithm of the intensity value of a CMOS sensor grid position changes beyond a threshold. The key advantages of such cameras are [1]: *(i)* sensing at very low latency and very high speed, e.g., microsecond-level latency and tens of thousands of triggers per second; *(ii)* low power requirements, e.g., 20mW versus hundreds of mW for conventional frame-based video cameras; *(iii)* robustness to uncontrolled lighting conditions, as no synchronous global shutter is used.

One of the major obstacles in developing neuromorphic-based advanced machine learning algorithms for recognition, classification and retrieval is the lack of widely-available event-based neuromorphic vision streams with reliable annotations to train and test with. Recent work has attempted to resolve this issue by recording limited-scale annotated datasets in controlled conditions [3–6], i.e., video frames displayed in a monitor under controlled frame-rate and brightness/contrast conditions and are recorded with a DVS camera. While such experimental approaches provided for the first available annotated video datasets in neuromorphic vision stream (NVS) format, their three issues are that: *(i)* the recording is affected by environmental and monitor conditions (e.g., lighting, monitor flicker, vibrations, etc.); *(ii)* high-accuracy synchronization between the played-out video frames and the corresponding NVS may be difficult to resolve because of drift between the timing of the playout device and the DVS camera; *(iii)* due to their hardware nature, such measurement-based approaches cannot scale to large datasets containing millions of videos, such as the recently-released Youtube-8M dataset [7]. To this end, recent work [8–10] proposed models to generate NVS events using piecewise linear interpolation of the pixel intensity given by successively rendered images. However, these approaches have one or more of the following detriments: use of custom bias settings, requirement to have pixel-domain frames captured by a co-existing active pixel sensor (AES) camera, such as the bundled AES of the DAVIS240C device, and lack of distortion metrics to quantify the accuracy of the generated NVS events.

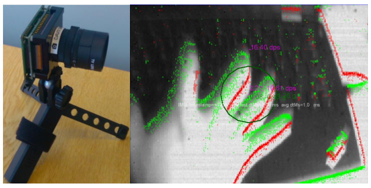

Fig. 1. Left: iniLabs DAVIS240C camera used in our experiments. Right: a captured video frame with NVS events superimposed on it (green/red points: trigger ON/OFF).

In this paper we propose and make available online[1] the pixel-to-NVS (PIX2NVS) framework, which is a software codebase that can be used to generate neuromorphic vision streams from any pixel-domain video format. We also propose and verify two new metrics, Chamfer distance and ϵ-repeatability, to quantify the accuracy of the model-generated NVS against ground-truth event streams available from experimental setups, such as aedat files from DAVIS camera

[1] http://www.github.com/pix2nvs

deployments. Beyond its full parameterization, our framework is deployable at scale for different datasets widely-used within the machine learning and computer vision communities and can be extended by other researchers in the area in order to fit the needs of various application domains.

2. MODEL DESCRIPTION

NVS devices like the iniLabs DAVIS [1] and Pixium ATIS sensors [2] output asynchronous events indicating temporal intensity contrast changes. Events are recorded in pixel coordinates, timestamped with microsecond resolution and labeled as ON or OFF [1]. They are produced in a format compliant with the address event representation protocol (AER) [11]. Based on the FFMPEG library [2] [14], our model retrieves a pixel-domain video that may be encoded and wrapped in any standard format container (e.g., MP4, MKV, etc.) and extracts a series of pixel-domain video frames, based on which it produces a stream of NVS events and stores them in text or AEDAT format (AER data file). The aim of the model is to produce NVS events that are as similar as possible to the ones that would have been generated if the equivalent scene would have been captured with an NVS-generating device like iniLabs DAVIS or Pixium ATIS hardware. Similarity is assessed based on the metrics of the next section. The operation of our model is shown in Algorithm 1, and the details of the operation are described in the following parts of this section.

2.1. Converting pixels to log-intensity/contrast-enhanced (LICE) values

For every spatial position (i,j) of each frame F_n, the RGB pixel values $(r_{i,j}, g_{i,j}, b_{i,j})$, typically ranging between 0 to 255, are first converted into luminance values via $y_{i,j} = 0.299 r_{i,j} + 0.587 g_{i,j} + 0.114 b_{i,j}$ or, if `hue = TRUE`, hue values via $h_{i,j} = b_{i,j}/(r_{i,j} + g_{i,j})$. Without loss of generality, for the remainder of this work we shall be focusing on luminance values. These values are then converted into *log-intensity* values via

$$l_{i,j} = \begin{cases} y_{i,j}, & y_{i,j} \leq T_{\text{log}} \\ \ln(y_{i,j}), & y_{i,j} > T_{\text{log}} \end{cases} \tag{1}$$

with T_{log} the threshold used to control the switch between the linear and the log mapping. For log intensity, $T_{\text{log}} = 0$, while for lin-log intensity, the threshold is set to a value close to 10% of the maximum value, e.g., $T_{\text{log}} = 20$. Alternatively, we can determine the *contrast-enhanced* intensity values [15] by first defining the perceptual luminance of each pixel as $l'_{i,j} = 100 \times \sqrt{(y_{i,j}/255)^{\gamma}}$, with $\gamma = 2.2$, and calculate the contrast-enhanced intensity at coordinate (i,j) by $l_{i,j} = \frac{\sum_{p=0}^{1} |l'_{i,j} - l'_{i+2p-1,j}| + \sum_{p=0}^{1} |l'_{i,j} - l'_{i,j+2p-1}|}{4}$. The choice between log-intensity and contrast enhancement (LICE) is controlled by setting parameter `LICE_mode` $\in \{$`LI,CE`$\}$.

```
Input: Pixel-domain video frames $F_0, F_1, ..., F_N$ extracted from a video format container using FFMPEG, parameters: hue, LICE_mode, $T_{\text{log}}, \gamma, T_{\text{map}}$, dif, new, tstamp, fps
Output: Event tuples $E_e = \langle x_e, y_e, t_e, P_e \rangle$ stored in a text file and/or in an AEDAT stream
Operation: Read $F_0$, convert pixels to LICE values, produce event tuples and optionally update LICE values
for $n = 1 : 1 : N$ do
    Read $F_n$ and convert the RGB pixel values to LICE values using Section 2.1
    Find differences of LICE values of successive frames using (2) and (3)
    If the difference is equal or exceeds threshold $T_{\text{map}}$, then output ON and OFF events with coordinate, polarity and timestamp using (4) and (5)
    Optionally update LICE values using (6)
end for
```

Algorithm 1: Conversion of video frame pixels to NVS.

2.2. Event generation

For all frames beyond the first one, we can derive the NVS events by establishing the difference between the LICE values and a function that utilizes corresponding LICE values from the previous frame. To this end, we propose three approaches: *(i)* co-located LICE differencing between successive frames (which is enabled by parameter setting `dif=0`),

$$d_{i,j} = l_{i,j}[n] - l_{i,j}[n-1] \tag{2}$$

(ii) two variants of LICE differencing that utilize the average or the minimum value of the weighted-neighborhood of LICE values in the previous frame (`dif=avg`, `dif=min`),

$$\begin{cases} d_{i,j} = l_{i,j}[n] - \frac{\sum_{p=0}^{1} l_{i+2p-1,j}[n-1] + \sum_{p=0}^{1} l_{i,j+2p-1}[n-1]}{4} \\ d_{i,j} = l_{i,j}[n] - \min_{p \in 0,1}(l_{i+2p-1,j+2p-1}[n-1] \end{cases} \tag{3}$$

The eth NVS event of frame n (out of $e_{\text{tot}}[n]$ events detected in that frame) is generated if and only if $|d_{i,j}| \geq T_{\text{map}}$; in such a case, the type of event is:

$$P_e = \begin{cases} \text{ON}, & \text{sgn}(d_{i,j}) = 1 \\ \text{OFF}, & \text{sgn}(d_{i,j}) = -1 \end{cases} \tag{4}$$

the coordinates of the event are $(x_e, y_e) = (i,j)$. Concerning the timestamp of the event, we can generate it as: *(i)* a random number between the timestamps of frames $n-1$ and n; *(ii)* a linearly-scaled value between the timestamps of frames $n-$

[2] MPEG coding frameworks and FFMPEG are chosen because of their wide availability and support, the proposed framework can also be extended to support formats of non-MPEG codecs [12, 13].

1 and n; *(iii)* fixed to the timestamp of frame n (parameter `tstamp`$\in$\{`RAND`, `LINEAR`, `FRAME`\} controls this):

$$t_e = \begin{cases} U([n-1,n]) \times \text{fps} \\ (n-1+\frac{e}{e_{\text{tot}}[n]}) \times \text{fps} \\ n \times \text{fps} \end{cases} \quad (5)$$

where fps stands for the frame-rate of the video and $U([a,b])$ returns a uniformly-distributed number within $[a,b]$.

2.3. Optional LICE update

In their comparison for change detection, current NVS-generating devices only utilize the log-scaled values of recently-detected positions, otherwise they utilize the original values. If we follow this approach (which occurs when parameter `new = FALSE`), for all positions (i,j) that no NVS event was detected, we must copy the LICE value found in the previous frame, i.e., $\forall(i,j) \notin \{(x_1,y_1),\ldots,(x_{e_{\text{tot}}},y_{e_{\text{tot}}})\}$.

$$l_{i,j}[n] = l_{i,j}[n-1]. \quad (6)$$

However, given that our model begins with the pixel representation of frames, we have the option to omit this update, i.e., by setting `new = TRUE`, to always use the LICE value derived based on the pixels of frame F_n.

3. DISTANCE METRICS

To evaluate the performance of PIX2NVS against ground truth NVS data generated by hardware experiments, we propose to use the Chamfer distance and the ϵ-repeatability, two metrics that allow us to quantify correspondences between model-generated NVS events and experimentally-derived events. First, the model and experimentally-derived events are grouped into "frames" F_n^{mod} and F_n^{exp} ($n = 0,1,2...,N$). Specifically, they are allocated to the n frame if their timestamps fall within $t_{\{\text{mod,exp}\}} \in [n, n+1) \times \text{fps}$. While this frame grouping is artificial, it allows for the introduction of quantization in time and, together with the inherent quantization in the spatial coordinates of NVS events, facilitates the establishment of metrics that represent the average spatio-temporal correspondence between two sets of NVS data. In what follows, we present the two metrics in reference to the model-generated events; expressing these metrics in reference to experimentally-derived events can be derived analogously.

With respect to Chamfer distance, for each model event $E_i^{\text{mod}} = \langle x_i^{\text{mod}}, y_i^{\text{mod}}, t_i^{\text{mod}}, P_i^{\text{mod}}\rangle$ (with $E_i^{\text{mod}} \in F_n^{\text{mod}}$), we first search for event $E_j^{\text{exp}} = \langle x_j^{\text{exp}}, y_j^{\text{exp}}, t_j^{\text{exp}}, P_j^{\text{exp}}\rangle$ (with $E_j^{\text{exp}} \in F_n^{\text{exp}}$) with the minimium Euclidean distance, calculated based on their spatial coordinates, i.e., $\forall i$:

$$j^* = \arg\ \min_{\forall j} \left\|(x_i^{\text{mod}}, y_i^{\text{mod}}) - (x_j^{\text{exp}}, y_j^{\text{exp}})\right\| \quad (7)$$

Then the Chamfer distance for the $e_{\text{tot}}[n]$ model events corresponding to frame F_n^{mod} is defined as

$$C(n) = \frac{\sum_{i=1}^{e_{\text{tot}}[n]} \left\|(x_i^{\text{mod}}, y_i^{\text{mod}}) - (x_{j^*}^{\text{exp}}, y_{j^*}^{\text{exp}})\right\|}{e_{\text{tot}}[n]} \quad (8)$$

The Chamfer distance for the entire video sequence is

$$C = \frac{\sum_{n=0}^{N} C(n)}{N+1} \quad (9)$$

The ϵ-repeatability metric is defined as the number of events in F_n^{mod} repeated in F_n^{exp} within ϵ distance with respect to the total events. For each model event E_i^{mod} ($E_i^{\text{mod}} \in F_n^{\text{mod}}$), we first find whether at least one event E_j^{exp} exists in F_n^{exp} with spatial coordinates that have Euclidean distance smaller or equal to ϵ. Thus, for each model frame, F_n^{mod}, we get a new model event set:

$$E_i^{\text{mod},\epsilon} = \begin{cases} E_i^{\text{mod}}, & \left\|(x_i^{\text{mod}}, y_i^{\text{mod}}) - (x_j^{\text{exp}}, y_j^{\text{exp}})\right\| \le \epsilon \\ \oslash, & \text{otherwise} \end{cases} \quad (10)$$

Then the ϵ-repeatability rate for F_n^{mod} is defined by the normalized l_0 "norm" (i.e., counting the fraction of the total number of non-zero elements):

$$r^\epsilon[n] = |E_i^{\text{mod},\epsilon}|/e_{\text{tot}}[n] \quad (11)$$

The final ϵ-repeatability for a video sequence is the mean of $r^\epsilon[n]$ over all n.

4. EXPERIMENTAL RESULTS

We use an iniLabs DAVIS240C camera to record pixel-domain video frames and experimental NVS events simultaneously, the latter serving as ground truth. We then deploy our PIX2NVS model based on the captured video frames in order to generate artificial NVS events to compare against the ground truth. Beyond this, we also validate the accuracy of our model based on a recently-released dataset [9]. The parameters used for the reported experiments were: `hue=FALSE`, `LICE_mode=LI`, $T_{\text{log}} = 20, T_{\text{map}} = 0.4$, `dif=0`, `new=TRUE`, `tstamp=FRAME`, and `fps` is set according to the frame rate of the utilized video content.

4.1. Display of Generated Events

A qualitative comparison between the real and model-generated events is shown in Fig.2. It is evident that the model-generated NVS events are clustered around frame times (since we use `tstamp=FRAME`). In addition, real NVS events contain flicker noise due to the underlying electronics, while our model-generated NVS datasets do not include such noise since they are based on thresholded differencing of LICE values. Beyond these effects, the qualitative comparison shows that our model appears to be generating events that resemble the spatio-temporal structure of real NVS events from the DAVIS240C.

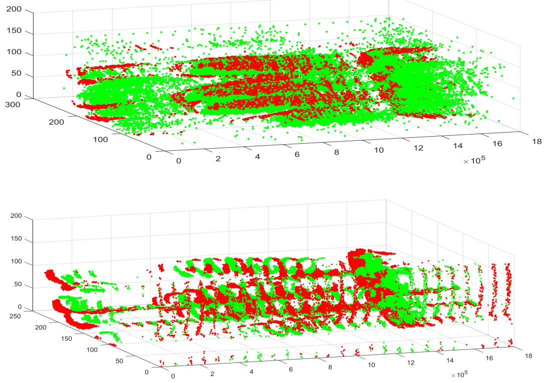

Fig. 2. Experimental NVS events (top) and model-generated ones (bottom). Green/Red points: Trigger ON/OFF.

4.2. Experimental Validation of the Proposed Metrics

Because of the presence of such flicker noise in the experimentally derived NVS, we measure the proposed metrics in reference to the model-generated data. In order to evaluate the suitability of the proposed Chamfer distance and ϵ-repeatability in the domain of NVS data, before measurement with each metric, we impose artificial spatio-temporal distortions in the model-generated events by: *(i)* spatial downsampling (SD) of the events' coordinates; *(ii)* temporal downsampling (TD) by reduction of the `fps` value used for the grouping of NVS events into frames; *(iii)* pseudo-random injection of additive noise (AN) NVS events at 1% to 7% of the possible spatial coordinates within each model NVS frame. Experiments are conducted using the dataset of Mueggler *et al.* [9] and real DVS events and video frames captured with a DAVIS240C camera in our laboratory. For the cases of SD and TD, measurement is carried out by first upscaling the downscaled NVS events to the original spatio-temporal resolution before using the process described in Section 3. If the proposed metrics are appropriate for the utilized NVS data, we expect that, as we impose such SD/TD/AN distortions: *(i)* the Chamfer distance will increase; *(ii)* the ϵ-repeatability will decrease. Indeed, the results, shown in Table 1, validate this expectation for all cases. Therefore, we conclude that these two metrics are appropriate for the quantification of the accuracy of model-generated NVS events.

4.3. Initial Validation of Model Options

Having validated the suitability of the proposed metrics for quantification of the accuracy of NVS data, we can now begin to evaluate the multitude of options of the proposed PIX2NVS approach on actual datasets. To this end, we used the dataset of Muegler *et al.* [9]. Table 2 presents the obtained results. Evidently, for the examined dataset, log-intensity provides for better performance in comparison to contrast enhancement. In terms to intensity change check, `dif=0` (i.e., differencing between corresponding pixels) have the similar performance to the `dif=avg`. As to LICE update, using the new frames as map values (`new = TRUE`) vs. updating the map only when events are generated (`new = FALSE`) is found to offer lower Chamfer distance but slightly higher ϵ-repeatability. Further experimentation with larger datasets will provide for more evidence on what are the best conversion options to use.

Table 1. Average Chamfer distance / ϵ-repeatability ($\epsilon = 2.5$) w.r.t. spatial downsampling (SD), temporal downsampling (TD) and additive noise (AN) from 5 videos in lab tests with DAVIS240C and the Mueggler *et al.* dataset [9].

Dataset		Lab Tests	Mueggler [9]
Original data		1.81 / 0.86	1.27 / 0.89
SD	120×90	2.71 / 0.73	1.86 / 0.80
	80×60	2.74 / 0.72	1.90 / 0.78
	60×45	3.12 / 0.66	2.17 / 0.72
TD	fps/2	2.07 / 0.82	1.38 / 0.87
	fps/3	2.24 / 0.78	1.44 / 0.87
AN	1%	2.31 / 0.80	1.76 / 0.85
	3%	3.11 / 0.74	2.11 / 0.81
	5%	3.28 / 0.70	2.34 / 0.78
	7%	3.55 / 0.67	2.52 / 0.69

Table 2. Chamfer distance / ϵ-repeatability ($\epsilon = 2.5$) w.r.t. different options. The results are the mean of 5 videos.

Comparison	Options	Actual Frame/Dataset	
		CD	ϵ-R
LICE Conversion	`LI`, $T_{\text{log}} = 0$	1.81/1.22	0.86/0.90
	`LI`, $T_{\text{log}} = 20$	2.14/1.83	0.82/0.83
	`CE`	2.54/2.70	0.78/0.78
LICE Checking	`dif=0`	1.81/1.22	0.86/0.90
	`dif=min`	2.51/1.48	0.78/0.86
	`dif=avg`	1.81/1.13	0.86/0.90
LICE Update	`new = TRUE`	1.81/1.22	0.86/0.90
	`new = FALSE`	1.90/1.24	0.83/0.89

5. CONCLUSION

We propose and make available online a parametric tool for software conversion of pixel-domain video frames into neuromorphic vision streams (http://www.github.com/pix2nvs). Our framework is also coupled with two new metrics for the quantification of accuracy of artificial NVS data in comparison to experimentally-available ones. Initial validation experiments with laboratory tests using an iniLabs DAVIS240C and publicly-available NVS measurements provide for the first results with our framework, demonstrating its suitability as a tool to convert conventional video frame datasets into neuromorphic streams.

6. REFERENCES

[1] T. Delbruck, "Neuromorphic vision sensing and processing," in *Proc. Europ. Solid-State Dev. Res. Conf.* IEEE, 2016, pp. 7–14.

[2] E Neftci, C Posch, and E Chicca, "Neuromorphic engineering," *Computational Intelligence-Volume II*, p. 278, 2015.

[3] J. A. Pérez-Carrasco et al., "Mapping from frame-driven to frame-free event-driven vision systems by low-rate rate coding and coincidence processing–application to feedforward ConvNets," *IEEE Trans. on Patt. Anal. Mach. Intel.*, vol. 35, no. 11, pp. 2706–2719, 2013.

[4] G. Orchard et al., "HFirst: a temporal approach to object recognition," *IEEE Trans. on Patt. Anal. Mach. Intel.*, vol. 37, no. 10, pp. 2028–2040, 2015.

[5] Y. Hu, H. Liu, M. Pfeiffer, and T. Delbruck, "DVS benchmark datasets for object tracking, action recognition, and object recognition," *Frontiers in Neuroscience*, vol. 10, 2016.

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

[7] S. Abu-El-Haija et al., "Youtube-8m: A large-scale video classification benchmark," *arXiv preprint arXiv:1609.08675*, 2016.

[8] M. L. Katz, K. Nikolic, and T. Delbruck, "Live demonstration: Behavioural emulation of event-based vision sensors," in *Proc. IEEE Int. Symp. on Circ. and Syst.* IEEE, 2012, pp. 736–740.

[9] 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," *Int. J. Robot. Res.*, vol. 36, no. 2, pp. 142–149, 2017.

[10] G. Pineda García, P. Camilleri, Q. Liu, and S. Furber, "pydvs: An extensible, real-time dynamic vision sensor emulator using off-the-shelf hardware," in *Proc. IEEE Symp. Ser. Comp. Intel. (SSCI), 2016*. IEEE, 2016, pp. 1–7.

[11] K. A. Boahen, "Point-to-point connectivity between neuromorphic chips using address events," *IEEE Trans. Circ. and Syst. II: Analog and Digital Sig. Process.*, vol. 47, no. 5, pp. 416–434, 2000.

[12] Y. Andreopoulos and M. Van der schaar, "Adaptive linear prediction for resource estimation of video decoding," *IEEE Trans. Circ. and Syst. for Video Technol.*, vol. 17, no. 6, pp. 751–764, 2007.

[13] J. Barbarien et al., "Scalable motion vector coding," in *Proc. IEEE Int. Conf. Image Process., 2004. ICIP'04*. IEEE, 2004, vol. 2, pp. 1321–1324.

[14] S. Tomar, "Converting video formats with ffmpeg," *Linux Journal*, vol. 2006, no. 146, pp. 10, 2006.

[15] K. Matkovic, L. Neumann, A. Neumann, T. Psik, and W. Purgathofer, "Global contrast factor-a new approach to image contrast.," *Computational Aesthetics*, vol. 2005, pp. 159–168, 2005.