# Frame2DVS Techinical Report

Zhe He

September 2019

## 1 Overview

Our goal is to generate a event stream with high sampling rate from a conventional video, which contains frames $\{I_0, I_1 ... I_i ... I_N\}$. To this end, we first interpolate the intermediate frames between two consecutive frames. Given frames $I_t$ and $I_{t+1}$, a interpolated frame $I_{t+\delta_t}$ at arbitrary time $t + \delta_t$ is generated from a generator $G$, such that $I_{t+\delta_t} = G(I_t, I_{t+1}, \delta_t)$. Second, after a series of interpolated frames $\{I_{t+\delta_t^0}, I_{t+\delta_t^1}, ... I_{t+\delta_t^i} ... I_{t+\delta_t^{N-1}}\}$ are generated, we use a converter to convert the brightness changes between two consecutive interpolated frames into events.

## 2 Frames Interpolation

For a real DVS sensor, whether an event will be triggered or not only depends on the intensity changes. Thus, for the interpolation stage, we first convert the input RGB frames into luminosity frames:

$$Y = 0.2126 \times R + 0.7152 \times G + 0.0722 \times B \tag{1}$$

In Eq. 1, $R, G, B$ are the values of three channels of a RGB pixel, and $Y$ is the output luminosity value.

The luminosity frames are then interpolated by the generator. Here, we adopt the generator network proposed in [2]. The generator $G$ first takes two consecutive luminosity frames as input and produces the backward and forward optic flow. Then, the bi-directional optic flow are linearly interpolated at an arbitrary time between the two input frames. Finally, the interpolated frame is produced by warping both input frames with the predicted bi-directional optic flow. For our task, we modified the official implementation [1] of [2] and trained it on Adobe240FPS [5] dataset.

[1] https://github.com/avinashpaliwal/Super-SloMo

# 3 Events Generation

After we get the interpolated frames, the next step is to generate events from these frames. We adopted the method introduced in [3]. For a real DVS sensor [1, 4], an event is triggered when the change of log intensity exceeds the threshold. Therefore, to model the DVS sensor, we first need to convert the intensity value from linear scale into logarithmic scale,

$$x_{log} = f(x) = \begin{cases} x & \text{if x ¡ 20} \\ \log(x) & \text{otherwise} \end{cases} \tag{2}$$

We used Eq. 2 to convert the intensity value $x \in [0, 255]$ of each pixel into logarithmic value. Note that $f(x)$ in Eq. 2 is actually a piece-wise function, since the log function is very sensitive to the small values near zero.

Positive and negative events are generated as follows,

$$e = \begin{cases} \text{+A} & \text{if } x^t_{log} - x^{t-1}_{log} > \sigma \\ \text{-A} & \text{if } x^t_{log} - x^{t-1}_{log} < -\sigma \end{cases} \tag{3}$$

In Eq. 3, $e$ denotes the generated event. When the positive change of log intensity is greater than the threshold $\sigma$, an ON event $+A$ is triggered. Similarly, a OFF event is triggered when the negative change of log intensity is greater than the threshold $\sigma$.

The real DVS sensor is completely asynchronous, which means an event can be triggered at any time. However, for our conversion based on conventional videos, the timestamps of the interpolated frames are discrete, and they are also fully determined by the frame rate of input video and the slow-motion factor. Thus, we used the following strategy to assign the timestamps:

Given two consecutive interpolated frames $I_{t+\delta_i}$ and $I_{t+\delta_{i+1}}$,

- If there is only one event triggered at a pixel, the timestamp of this event will be assigned as $t + (\delta_i + \delta_{i+1})/2$
- If there are $n$ events triggered at a pixel, the timestamps will be evenly distributed between $t+\delta_i$ and $t+\delta_{i+1}$, e.g. $\{t+\delta_i+\Delta, t+\delta_i+2\Delta, \ldots, t+\delta_i+n\Delta\}$, $\Delta = (\delta_{i+1} - \delta_i)/(n+1)$

# References


[1] Tobi Delbrück, Bernabe Linares-Barranco, Eugenio Culurciello, and Christoph Posch. Activity-driven, event-based vision sensors. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pages 2426–2429. IEEE, 2010.

[2] 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 *Proceedings of the*

*IEEE Conference on Computer Vision and Pattern Recognition*, pages 9000–9008, 2018.

[3] Matthew L Katz, Konstantin Nikolic, and T Delbruck. Live demonstration: Behavioural emulation of event-based vision sensors. In *2012 IEEE International Symposium on Circuits and Systems*, pages 736–740. IEEE, 2012.

[4] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128 $\times$ 128 120 db 15$\mu$s latency asynchronous temporal contrast vision sensor. *IEEE journal of solid-state circuits*, 43(2):566–576, 2008.

[5] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1279–1288, 2017.