Neuromorphic Event-Based Spatio-temporal Attention using Adaptive Mechanisms

Amélie Gruel, Antonio Vitale, Jean Martinet, Michele Magno

Neuromorphic Event-Based Spatio-temporal Attention using Adaptive Mechanisms

Amelie Gruel
I3S / CNRS
Université Côte d'Azur
Sophia Antipolis, France
amelie.gruel@univ-cotedazur.fr

Antonio Vitale
PBL Center
ETH Zürich
Zürich, Switzerland
antonio.vitale@pbl.ee.ethz.ch

Jean Martinet
I3S / CNRS
Université Côte d'Azur
Sophia Antipolis, France
jean.martinet@univ-cotedazur.fr

Michele Magno
PBL Center
ETH Zürich
Zürich, Switzerland
michele.magno@pbl.ee.ethz.ch

***Abstract*—Contrary to RGB cameras, Dynamic Vision Sensor (DVS) output visual data in the form of an asynchronous events stream by recording pixel-wise luminance changes at microsecond resolution. While conventional computer vision approaches utilise frame-based input data, thus failing to take full advantage of the high temporal resolution, novel approaches use spiking neural networks Spiking Neural Networks (SNNs) which are more compatible to handle event-based data since these bio-inspired neural models intrinsically encode information in a sparse manner using activation spikes trains. This paper presents an attentional mechanism which detects regions with higher event density by using inherent SNN dynamics combined with online weight and threshold adaptation. We implemented the network directly on Intel's research neuromorphic chip Loihi and evaluate our proposed method on the open DVS128 Gesture Dataset. Our system is able to process 1 ms of event-data in 6 ms and reject more than 50% of incoming unwanted events occurring only 20 ms after activity onset.**

***Index Terms*—Event-Based Vision, Online Adaptation, Neuromorphic Hardware, Spiking Neural Networks**

I. Introduction

Event-based vision is an emerging technology based on bio-inspired vision sensors that output pixel-level brightness changes instead of standard intensity frames and promises to significantly improve energy efficiency and latency in next generation of computer vision applications [1]. However, classic computer vision algorithms which rely on frame-based data for applications such as object classification [2] and scene segmentation [3] are unable to handle sparse, asynchronous event data coming from Dynamic Vision Sensors (DVSs). Approaches combining classic methods with event-based data as input rely on pre-processing this data at the cost of losing information from the data stream [4]. Recent work has shown that SNNs can operate on raw event data and achieve similar performance as conventional methods [5].

Neuromorphic chips, such as Intel's Loihi [6], implement SNNs directly on hardware. These ultra-fast, low-power processors can be interfaced directly to DVSs and are a promising alternative for handling event-based data. However the maturity of SNN-based vision algorithms is still far from state of the art conventional methods. [7] presented an overview of attention models which use events as input derived from classical computer vision. In [8] the authors interface two Asynchronous Time-based Image Sensor (ATIS) cameras to the neuromorphic platform SpiNNaker [9] to obtain optical flow and scene depth estimation. The work in [10] shows ultra-fast control of an Unmanned Aerial Vehicle achieved by interfacing a DVS camera directly to Intel's neuromorphic platform Kapoho Bay, the same one used in this work.

This paper presents a low-latency neuromorphic temporal attention model which detects a dynamic Region of Interest (ROI) in a scene recorded by a DVS, relying solely on the intrinsic dynamics of SNNs. A ROI is a part of the scene which contains useful information and visual attention can be defined as the behavioural and cognitive process of selectively focusing on a discrete aspect of sensory cues while disregarding other perceivable information [11].

The object tracking SNN implemented by [12] focuses on the same task as this work. There Dynamic Neural Fields (DNF) as a soft Winner-Takes-All (WTA) are used in order to detect the salient activity by manually increasing the activity in a pre-defined region. However, the preovious approach is not applicable in real scenarios where the ROI is not known before hand. In contrast, our proposed method aims at tracking a cluster of activity which occurs first in time. To the best of our knowledge this is the first work which presents this approach and demonstrates it with experimental evaluation.

Our ROI detection mechanism relies on finding activity originating within a spatio-temporal region where many pixels are active. This activity more likely corresponds to an object of interest relevant for the task at hand than to background noise. The contributions of the paper are as follows:

- we propose and evaluate the neuromorphic spatio-temporal attention model for DVS camera using the PyNN simulator
- we implement and evaluate the proposed approach on Intel's neuromorphic chip Loihi to demonstrate its benefits.

II. Background

A. Spiking Neural Networks

SNNs are artificial neural network which mimic the dynamics of biological neuronal circuits by receiving and processing

This work was supported by the European Union's ERA-NET CHIST-ERA 2018 research and innovation programme under grant agreement ANR-19-CHR3-0008.

information in the form of spike trains. We selected the "Leaky Integrate-And-Fire" SNN model because of its simplicity: the membrane potential is at rest when there is no input; otherwise, it increases according to the incoming spikes, and it slowly decays towards the resting value when the input stops (leak). If the membrane potential overcomes a threshold, an output spike is produced and the membrane potential is reset.

B. Neuromorphic Hardware: Intel's Loihi

Intel Loihi is a recent neuromorphic research chip [6] that has been used in this work as processing platform. In particular, the Kapoho Bay platform, which comes in a USB form factor with 2 Loihi chips with a total of 256 neuro-cores able to simulate 262.144 neurons and up to 260Mn synapses. It can be easily interfaced for live communication with a host system[1] by programming the three embedded x86 processors, which are used for monitoring and I/O spike management embedded on the Loihi chip.

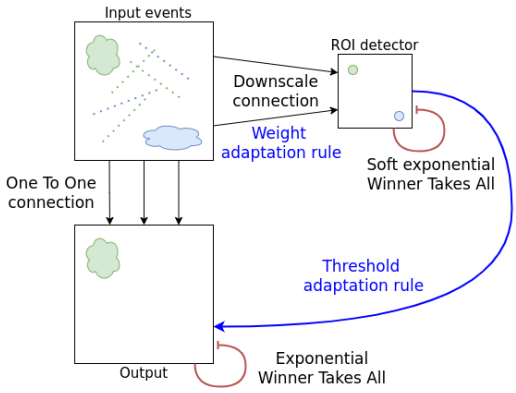

Fig. 1: Architecture of the spatio-temporal attention model.

C. DVS Data

An event camera consists of a pixel array which responds to luminance changes in a sparse and asynchronous fashion. An event $e_i = \{\mathbf{x_i}, p_i, t_i\}$, with polarity $p_i \in \{+, -\}$, is triggered at pixel location $\mathbf{x_i} = \{x_i, y_i\}$ at time t_i whenever the change of luminance at the pixel exceeds above or decreases below the contrast sensitivity threshold [1]. To design and evaluate the proposed neuromorphic model, an existing dataset has been used. The DVS128 Gesture Dataset is a commonly used neuromorphic classification dataset of different gestures performed by multiple participants in front of a 128×128 DVS [13]. Our target is the hand(s) as the objects of interest. The samples from 3 classes were used in various combinations for the demonstration of this spatio-temporal attention model: "arm roll" (1), "hand clap" (2) and "right hand clockwise" (7).

The samples are separated into two sub-types, either "large" or "small", according to the number of events in the sample and the ratio of active pixels in the sensor: classes 1 and 2 are "small" and class 7 is "large". New samples with size 128×256 were generated by concatenating samples from both types. We balance our dataset by allowing the first gesture to start either on the right or on the left. A varying time shift to the start of the second sample is added, using shifts of $0.5ms$, $1ms$, $5ms$, $10ms$, $20ms$ and $50ms$, to demonstrate the accuracy of the filtering increasing with the time shift.

[1]We used an UP board featuring a 64-bit Intel®ATOM™x5-Z8350 processor, 1.92GHz, 4GB RAM; it runs Ubuntu 18.04, Python 3.7 and version 0.9.9 of Intel's NxSDK.

III. Neuromorphic Model

This section describes the main contribution of the paper that is the proposed SNN architecture for the temporal filtering of DVS data. This whole mechanism relies solely on intrinsic SNN dynamics and dynamic adaptation rules applied to synaptic weights and population thresholds. This is a crucial features as this leads to minimise the latency as it doesn't require the conversion of spiking event in a frame. These are modified dynamically over time according to the layers' activation, which allows for a good generalisability of the network, since the ROI detection is not specialised for any specific context. The proposed architecture, shown in Fig. 1, is designed to be lightweight to enable running in real-time.

A. Input layer

The input layer translates sensor events into spikes. The spikes produced by the input layer are sent to ROI detector via an excitatory down-scaling connection. This corresponds to a convolutional layer with a kernel size $S \times S$, a stride S, without padding. The input neurons are segregated into non-overlapping square regions of size S. The input data is of size 256×128 pixels wide, thus S was arbitrarily set at 5. Each neuron at the Input layer's subregions are connected to one neuron in the ROI detector layer. Here, the ROI detector is a 2D layer of size 52×26 pixels.

B. ROI detector

The ROI detection aggregates the active regions into distinct segments using a soft WTA by laterally inhibiting the neurons in the same layer: each neuron activation leads to the inhibition of the others, without autapses (self-connections). Since a strong WTA leads to the activation of only one neuron in the layer and multiple ROIs are to be detected by the network, the soft WTA weight has been set experimentally to 0.02.

In the case of the ROI detector, a specific exponential WTA is implemented according to the radial basis function Eq. 1 in order to allow ROIs of arbitrary sizes:

$$W_{WTA} = max(\frac{e^d}{w \times h}, w_{max}) \quad (1)$$

where d corresponds to the Euclidean distance in number of neurons between the active and target neuron subject to inhibition, and w and h to the width and height of the layer. The weight W_{WTA} has an upper bound of $w_{max} = 50$.

Finally, the adaptive detection of ROIs by this layer is enabled by a dynamic weight adaptation rule between the input layer and the ROI detector, inspired by Hebb's rule: "cells that fire together wire together" [14]. This rule is implemented by

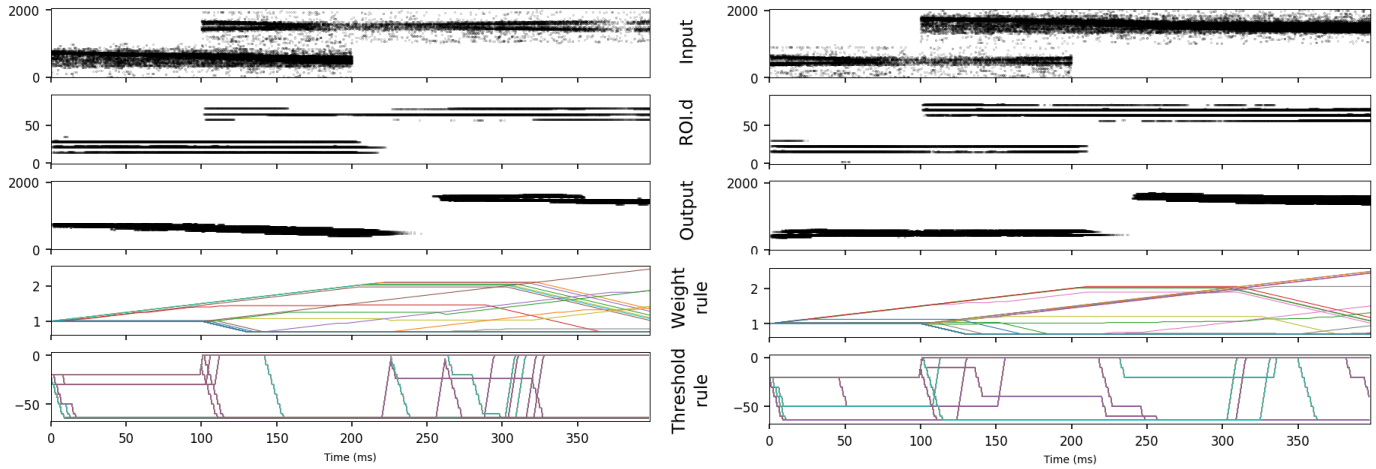

Fig. 2: Simulation of the attention model applied to two samples with a time shift of 100 ms. The Input, ROI.d and Output plots in the 3 first rows represent the neurons emitting a spike at time t (x axis) in the Input, ROI detector and Output layers respectively. The Weight and Threshold Rule correspond to the evolution of the weights and thresholds at the output layer.

increasing or decreasing the weights of synapses that have recently fired.

C. Output layer

The output layer is connected to the input layer via excitatory one-to-one connection. The spikes filtering is implemented a threshold adaptation rule. The thresholds are initialised at a high value to reduce the firing rate. The thresholds are updated during runtime according to the activation of the neurons in the ROI detector layer: the thresholds of the neurons in the $S \times S$ region in the output layer are decreased whenever the corresponding neurons spike in the ROI detector, while they are increased in the absence of ROI activity. We implement this threshold adaptation rule in order to make the neurons in the Output layer more susceptible to be activated while keeping a correct outline of the object of interest, according to the ROIs detected by the ROIs detector. This mechanism needs to know which neurons correspond to the ROIs in the Output layer before lowering the corresponding thresholds. Thus the initial thresholds are set at a high value in order to let the ROIs detector activate first, meanwhile preventing excessive unwanted output.

Another exponential WTA is applied to ensure the detection of one ROI at the time. This relies on the WTA having a strong weight value to prevents other neurons in the output layer to spike when one segment is already filtered.

D. Implementation on Neuromorphic Hardware

As depicted in Fig. 1, the key elements in the presented SNN are both the weight and threshold adaptation mechanisms. Currently the Loihi platform does not support these features thus the SNN had to be modified to implement it on the neuromorphic chip. The input and the ROI layer were connected to the output in a way such that at any given timestep the neurons in the output layer would spike only if the corresponding neurons were active in both previous layers. The connections between input layer and ROI detector layer, as well as the WTA mechanisms were kept the same. The resulting 3-layer network occupies 68 neurocores and consists of 4.224 neurons and 81.716 synapses.

IV. Experimental Results

The ground truth $GT(x, y, t)$ for each experiment was defined as the event stream of the first gesture only and all other events are $GT'(x, y, t)$. We treat events in the same way irrespective of their polarity.

The input, output and ground truth sequence were split into 100 ms windows and the active neurons, corresponding to x, y event coordinates, were stored in arrays with same dimensions as the neuron populations. The weighted Root Mean Square Error (RMSE) was calculated according to Eq. 2:

$$RMSE = \frac{RMSE(R_{O \cap GT}, GT) + RMSE(R'_{O \cap GT'}, GT)}{N_{GT}} \tag{2}$$

with O the output active region, $R_{O \cap GT}(x, y, t)$ the intersection with the ground truth, $R'_{O \cap GT'}(x, y, t)$ the respective intersection with regions not belonging to the ground truth and N_{GT} the total number of events in the ground truth. Eq. 2 defines both $RMSE_{input}$ and $RMSE_{output}$. In order to penalise wrongly activated regions, $RMSE'_{\cap}$ was multiplied with the total number of wrongly active events. The total $RMSE$ for the output sequence was obtained by adding $RMSE_{\cap}$ and $RMSE'_{\cap}$. The final error value is reported as the ratio between $RMSE_{output}$ and $RMSE_{input}$ such that a lower error value indicates better filtering performance.

A. Temporal filtering

Both results produced using PyNN and Loihi show that this attention model filters the region of primary interest as

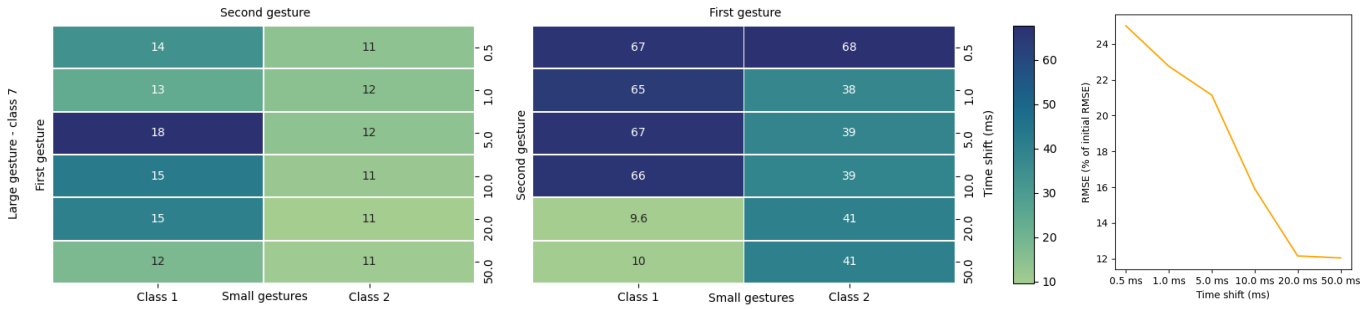

Fig. 3: Study of size and spatial invariance of the attention model. **Left and Center:** Error variation of different combinations between two "small" gestures (classes 1 and 2, x-axis) and one "large" gesture (class 7, y-axis left side), according to different time shifts (y-axis right size). Irrespective of the gesture size, the error decreases as the time shift increases. **Right:** Evolution of the error as the time shift increases: the larger time shit, the better filtering from the attention model.

a function of time with significant performance. As depicted in Fig. 2, when alternating between two overlapping gestures the network focuses only on the one that occurs first, independently of their size. The output is not significantly influenced depending on the size of the different input ROIs: the first gesture is filtered out (see Fig. 2) and the error stays low (see Fig. 3) independently of the characteristics of the first and second occurring input gestures.

B. Loihi vs PyNN simulator

1) Simulation time: The simulation of the model using PyNN and NEST is limited because it runs on a CPU[2]: around 3h26min58s were necessary to process 400 ms of the data used in Fig.2, which is a ratio of 1:12505, very far from real time.

On Loihi, the event data is collected for 1 ms on the embedded x86 cores before being sent to the corresponding input layer neurons. On average, measurements showed that it takes 6 ms to process 1 ms of input data. This ratio of 1:6 is much closer to real time.

These numbers show that processing event-based data on neuromorphic hardware is more efficient in terms of latency. This is an important advantage not only for running offline simulations, but also paves the way for potential real-time, closed-loop applications.

2) Adaptation rules: Currently, Loihi does not support online threshold adaptation. The results presented in Fig. 3 were obtained from the modified SNN running on the neuromorphic hardware. These experiments were not performed with the PyNN simulator due to the runtime limitations mentioned above. However, to demonstrate the effectiveness of the adaptation rules, we produced some limited results using shorter input data (400 ms instead of 1.5 s) on the PyNN simulator: the RMSE equals 0.03 on average in the context described in Fig. 2 Left, and 0.16 for Fig. 2 Right.

These results point towards significant improvement using adaptation rules. Incorporating such mechanisms in the design of future neuromorphic architectures will give rise to novel event-based attention applications.

[2]Intel 8-core i9-10885H at 2.4GHz.

V. Conclusion

This paper introduces a low-latency spatio-temporal attention SNN model, deployed on Intel's Loihi chip, which filters ROIs from a DVS stream. The architecture allows for a quick and efficient preprocessing of event data, which can be used prior to a task-related process focusing on a region of interest.

Future work will exploit this result as a basis for the implementation of real-time tasks such as multi-object tracking and scene segmentation by interfacing DVSs directly to the neuromorphic chip.

References

[1] G. Gallego *et al.*, "Event-based vision: A survey," *CoRR*, vol. abs/1904.08405, 2019.

[2] S. Song, Y. Zhu, J. Hou, Y. Zheng, T. Huang, and S. Du, "Improved convolutional neutral network based model for small visual object detection in autonomous driving," in *AICAS*, 2019.

[3] J. Peng, L. Tian, X. Jia, H. Guo, Y. Xu, D. Xie, H. Luo, Y. Shan, and Y. Wang, "Multi-task adas system on fpga," in *AICAS*, 2019.

[4] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "Ev-flownet: Self-supervised optical flow estimation for event-based cameras," *CoRR*, vol. abs/1802.06898, 2018.

[5] F. Paredes-Vallés, J. J. Hagenaars, and G. de Croon, "Self-supervised learning of event-based optical flow with spiking neural networks," *CoRR*, vol. abs/2106.01862, 2021.

[6] M. Davies *et al.*, "Advancing neuromorphic computing with loihi: A survey of results and outlook," *Pr. of the IEEE*, vol. 109, no. 5, 2021.

[7] A. Gruel and J. Martinet, "Bio-inspired visual attention for silicon retinas based on spiking neural networks applied to pattern classification," in *Content-Based Multimedia Indexing (CBMI)*, June 2021.

[8] G. Haessig, F. Galluppi, X. Lagorce, and R. Benosman, "Neuromorphic networks on the spinnaker platform," in *IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, pp. 86–91, 2019.

[9] C. Mayr, S. Höppner, and S. B. Furber, "Spinnaker 2: A 10 million core processor system for brain simulation and machine learning," *CoRR*, vol. abs/1911.02385, 2019.

[10] A. Vitale, A. Renner, C. Nauer, D. Scaramuzza, and Y. Sandamirskaya, "Event-driven vision and control for uavs on a neuromorphic chip," in *IEEE ICRA*, 2021.

[11] A. Borji and L. Itti, "State-of-the-art in visual attention modeling," *IEEE PAMI*, vol. 35, no. 1, pp. 185–207, 2013.

[12] A. Renner, M. Evanusa, and Y. Sandamirskaya, "Event-based attention and tracking on neuromorphic hardware," *IEEE CVPRW*, 2019.

[13] A. Amir *et al.*, "A Low Power, Fully Event-Based Gesture Recognition System," in *CVPR*, pp. 7388–7397, IEEE, 2017.

[14] D. Hebb, "The organization of behavior: A neuropsychological theory," *Journal of the American Medical Association*, vol. 143, no. 12, 1949.