Stakes of Neuromorphic Foveation: a promising future for embedded event cameras

Amélie Gruel, Dalia Hareb, Antoine Grimaldi, Jean Martinet, Laurent Perrinet, Bernabé Linares-Barranco, Teresa Serrano-Gotarredona

Stakes of Neuromorphic Foveation: a promising future for embedded event cameras

Amélie Gruel[1*], Dalia Hareb[1], Antoine Grimaldi[2], Jean Martinet[1], Laurent Perrinet[2], Bernabé Linares-Barranco[3] and Teresa Serrano-Gotarredona[3]

[1]SPARKS, Université Côte d'Azur, CNRS, I3S, 2000 Rte des Lucioles, Sophia-Antipolis, 06900, France.
[2]NeOpTo, Université Aix Marseille, CNRS, INT, 27 Bd Jean Moulin, Marseille, 13005, France.
[3]Neuromorphic Group, Instituto de Microelectrónica de Sevilla IMSE-CNM, 28. Parque Científico y Tecnológico Cartuja, Sevilla, 41092, State, Country.

*Corresponding author(s). E-mail(s): amelie.gruel@univ-cotedazur.fr;
Contributing authors: dalia.hareb@univ-cotedazur.fr; antoine.grimaldi@univ-amu.fr;
jean.martinet@univ-cotedazur.fr; laurent.perrinet@univ-amu.fr;
bernabe@imse-cnm.csic.es; terese@imse-cnm.csic.es;

Abstract

Foveation can be defined as the organic action of directing the gaze towards a visual region of interest, to acquire relevant information selectively. With the recent advent of event cameras, we believe that taking advantage of this visual neuroscience mechanism would greatly improve the efficiency of event-data processing. Indeed, applying foveation to event data would allow to comprehend the visual scene while significantly reducing the amount of raw data to handle.

In this respect, we demonstrate the stakes of neuromorphic foveation theoretically and empirically across several computer vision tasks, namely semantic segmentation and classification. We show that foveated event data has a significantly better trade-off between quantity and quality of the information conveyed than high or low resolution event data. Furthermore, this compromise extends even over fragmented datasets. Our code is publicly available online at: github.com/amygruel/FoveationStakes_DVS/.

Keywords: Foveation, event cameras, spiking neural networks, saliency, neuromorphic, semantic segmentation, classification.

1 Introduction

The joint use of silicon retinas (Dynamic Vision Sensors, DVS) and Spiking Neural Networks (SNNs) is a promising combination for dynamic visual data processing. Both technologies have recently emerged separately about a decade ago from electronics and neuroscience communities, sharing many features: biological inspiration, temporal dimension, model sparsity, aim for a higher energy efficiency, etc.

However, traditional and neuromorphic computer vision models can have difficulties handling a great amount of data simultaneously while minimising their energy consumption, especially at a

This article is published as part of the Special Issue on "What can Computer Vision learn from Visual Neuroscience?"

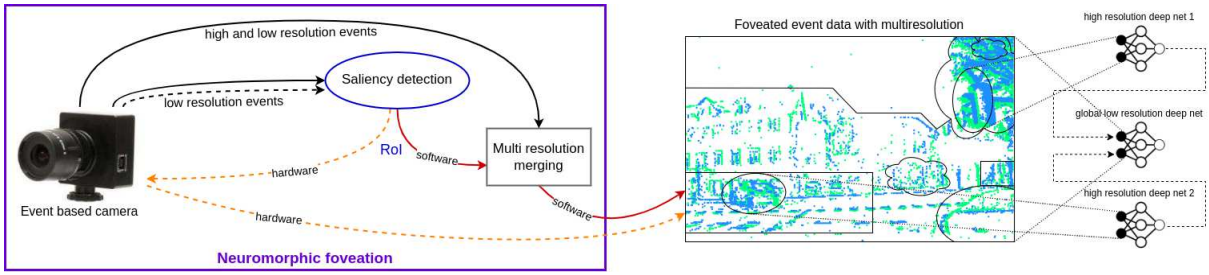


Fig. 1: Overview of the neuromorphic foveation concept as defined in this paper. The results presented below corresponds to the software neuromorphic foveation (red full arrows), as a validation to a future hardware neuromorphic foveation (orange dotted arrows). The multi-resolution event sample used as an example of output of such a process is extracted from the DDD17 dataset [Binas et al., 2017].

high temporal resolution. A recent study shows that in certain lightning conditions, high resolution event cameras produce data susceptible to temporal noise and with an increasingly high *per pixel event rate*, thus leading to the decreased performance of some traditional computer vision tasks [Gehrig and Scaramuzza, 2022]. Some event camera manufacturers recently tried to prevent this issue using event rate controllers, but those reduce the event rate by randomly dropping events [Finateu et al., 2020] or tuning camera parameters [Delbrück et al., 2021] during the recording, which alter significantly the visual data therefore is not a viable solution. Another remedy for such an issue could be found in event data downscaling (see [Gruel et al., 2022a]) — however the trade-off between information retention and data reduction with existing methods is not yet ideal.

We thus believe that foveation, a visual neuroscience mechanism allowing the complex eye to selectively acquire relevant information, is a more appropriate approach to optimise the on- and off-line processing of event data.

To demonstrate the interest of applying foveation to event data, we define in this work the concept of neuromorphic foveation (see Fig. 1): a retro-action loop between an event camera and a neuromorphic saliency detector, merging events at multiple resolutions according to the detected regions of interest. Such a process should be allowed by a foveated DVS as described in [Serrano-Gotarredona et al., 2022]; however, as this sensor is not yet available, we validate here the neuromorphic foveation concept with a first software implementation (red pathway in Fig. 1) before extending this feedback loop system to hardware (orange dotted pathway) in future works.

We study the respective evolution of the amount of event data processed in a computer vision task and its accuracy when software neuromorphic foveation as described above is applied. In order to simulate the foveation, the event data will be processed at a higher or lower resolution, depending on the relevance of the spatial regions in the image at different coordinates. Our proposed model goes beyond biology by allowing multiple RoI of arbitrary size and shape.

To the best of our knowledge, this work and our results are the first to show that foveation offers a significantly better compromise between quantity and quality of the information than the high or low resolution, especially on fragmented datasets. Furthermore, we demonstrate that our saliency detector is specifically efficient on data reduced using the *event count* method.

The following sections establish a detailed outline of the different hardware and mechanisms involved in this work; a complete description of the software neuromorphic foveation methodology; and the experimental evaluation procedure.

1.1 Event cameras

Event cameras (or silicon retinas) represent a new kind of sensors that measure pixel-wise changes in brightness and output asynchronous events accordingly [Posch et al., 2014]. This novel technology is inspired by the spatio-temporal filtering that happens in the horizontal and bipolar cells of the biological human retina. The filtered information is coded as asynchronous spikes as transmitted in the retinal optic nerve. The spatio-temporal filtering eliminates data redundancy over time and space, so that this technology allows for an energy-efficient recording and storage of data evolving over time and space. Furthermore, each event is recorded punctually and asynchronously with no redundancy; as opposed to traditional frame-based cameras, where each pixel outputs data in all frames, in a synchronous manner.

In the last two decades, different technological developments of spiking silicon retinas implementing spatial and temporal filtering have been published. However, they have not reached the maturity of commercial applications due to several limitations as high fixed pattern noise, low fill factor, and complex circuitry resulting in low sensor resolution. Recently, dynamic vision sensors (DVS) have been proposed. Each DVS pixel computes autonomously the relative temporal difference of the illumination received on its photosensor. When pixel illumination increases and its relative change goes over a certain threshold, the pixel would generate a positive ON output spike. Similarly, if the illumination decreases and its relative change goes over a certain negative threshold the pixel generates a negative OFF output spike. DVS circuitry can be implemented with compact circuitry which results in low fixed pattern noise and higher resolution sensors. Furthermore, DVS exhibit high temporal resolution (below 1 microsecond) and intrascene dynamic range as high as 120dB [Lichtsteiner et al., 2008]. Due to these features advanced megapixel DVS sensors have been developed [Li et al., 2019, Suh et al., 2020, Kubendran et al., 2021, Guo et al., 2017] and the DVS technology have reached the commercialisation stage. New high speed vision applications and systems based on DVS sensors are emerging.

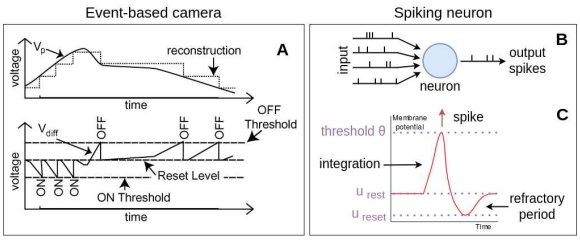


Fig. 2: (A) Principle of operation of an event-based camera, from [Lichtsteiner et al., 2008]. (B) Behavior of a spiking neuron, which receives spike trains as input and processes this information to produce a new sequence of activations. (C) Evolution of the neuron's membrane potential over time when activated by input spikes.

1.2 Spiking neural networks

SNNs [Paugam-Moisy and Bohte, 2012] represent an asynchronous type of artificial neural network closer to biology than traditional artificial networks, mainly because they seek to mimic the dynamics of neural membrane and action potentials over time. SNNs receive and process information in the form of spike trains, meaning as a non-monotonous sequence of activations, as represented in Fig. 2AB. Therefore, they make for a suitable candidate for the efficient processing and classification of incoming event patterns measured by event-based cameras as each event can be assimilated to an activation spike between two spiking neurons. This spatio-temporal model allows capturing and processing the dynamics of a scene. Moreover, since this model is sparse, it enables energy efficient implementations.

A SNN is constructed using populations of neurons linked together with connections, according to certain rules and a certain architecture. By definition, a spiking neuron follows a model based on parameters describing its internal state and its reaction to the input current (as pictured in Fig. 2B). Many models exist; from this set we chose to use the Leaky Integrate-and-Fire (LIF) model within the Spiking Neural Network Pooling method. The dynamics of the LIF neuron's membrane potential u are described by the equations 1 and 2:

$$\tau_m \frac{du}{dt} = u_{rest} - u(t) + RI(t) \tag{1}$$

where τ_m is the membrane's time constant and I the input current modulated by a resistance R.

Without any input current, the membrane potential is at rest and is of value u_{rest}. When activated, it increases according to the input current. Moreover, at each timestep a slow decrease towards u_{rest} is driven by the time constant τ_m, thus modeling the voltage leakage. The firing time t_f is defined by:

$$\begin{cases} u(t_f) = \theta \\ u'(t_f) > 0 \end{cases} \Rightarrow u(t_f) = u_{reset} \qquad (2)$$

Once the membrane potential u crosses the threshold θ with a positive slope, a spike is produced and the membrane potential is reset at u_{reset}. This is coherent with a biological neuron's behaviour when an action potential occurs: those two steps corresponds respectively to the neuron's depolarisation (or overshoot) and hyperpolarisation (or undershoot) [Bear et al., 2007].

1.3 Foveation

Most artificial sensors, such as the CMOS chip that powers the cameras in the average smartphone, have evenly spaced sensors. This is also the case for all event cameras. This is an optimal choice when considering the trade-off between the increasing miniaturization of each pixel and the growing demand for higher image resolutions. However, the majority of biological visual sensors have very irregular sensor grids. Some insects, for example, have a peripheral vision grid used for navigation, while another is specialized to a very specific location in the visual space and dedicated to mating behavior. Most predatory mammals have a central visual area on the retina with a high density of photoreceptors and the ability to move their eyes, and thus the localization of this area of high acuity around the center of the gaze. This action is driven by visual attention mechanisms [Gruel and Martinet, 2021] and can be learned by means of a saccadic mechanism [Daucé et al., 2020] that has been shown to optimize the efficiency of information gain at each saccade [Daucé and Perrinet, 2020]. The wide variety of anatomical configurations illustrates that this arrangement is closely related to the behavioral repertoire of each animal [Land, 2018], and the introduction of such an irregularity, which we refer to here as *foveation* for simplicity, can provide significant improvements in information processing in event cameras.

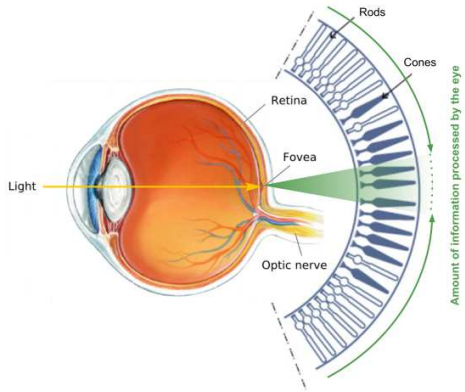


Fig. 3: Biological foveation mechanism, adapted from [Bear et al., 2007].

A foveated sensor that mimics the retinotopic layout of a biological retina will generally use fewer pixels than a conventional sensor, and will therefore be more energy efficient. Indeed, this would allow to maintain a high accuracy about the meaningful information, while significantly reducing the amount of raw data to be processed. In the particular case of neuromorphic architectures such as SpiNNaker or Intel Loihi, the power consumption is directly proportional to the number of spikes/events processed, and reducing this number is an efficient way to reduce power consumption, a heavy constraint for embedded applications. However, we believe that the development of a mechanism mimicking foveation would greatly improve the processing of event-driven data, beyond the energy efficiency aspect. Indeed, recent studies have shown that the particular geometrical layout of the log-polar mapping observed in the human retina has several advantages [Hao et al., 2021]. In particular, a rotation or a zoom is transformed into translations into a log-polar mapping [Traver and Pla, 2003]. Further rotation and zoom-invariant processing can for example be easily implemented in a convolutional neural network. Moreover, foveation can lead to several improvements in image compression [Araujo and Dias, 1997] or in the efficiency of image registration [Sarvaiya et al., 2009]. Yet, it is still not known whether such foveation could be beneficial for event-driven imaging.

1.4 Foveated event-based camera

The introduction of foveation at the sensor level for event-driven sensors should reduce the energy and bandwidth consumption from the sensor level up to the computing system by dynamically allocating more bandwidth and pixel resolution to the regions of interest and reducing the information of peripheral (non interesting) regions. Recently, an electronically foveated DVS have been proposed [Serrano-Gotarredona et al., 2022] where DVS pixels can be dynamically configured in high-resolution foveal regions or grouped into low resolution regions with arbitrary sizes. The sensor can attend in parallel an arbitrary number of foveal high-resolution regions. Furthermore, the electronically reconfiguration of foveal regions makes it faster and lower energy compared with the configuration of the center of the region of interest using mechanical control of the sensor position.

However this sensor has not yet been associated with a feedback loop saliency detection mechanism and validated on a computer vision task — this work is a first step towards this direction.

2 Neuromorphic foveation methodology

2.1 Saliency detection

The detection of RoI to foveate on is a little-explored issue regarding event-data. In this work, we propose to use part of the SNN presented in [Gruel et al., 2022b]: this saliency detector integrates the events produced by each pixel at a low resolution and outputs a set of coordinates for one or multiple RoI. In this case, our RoI would be a region where the amount of events received over a certain amount of time is more important than elsewhere over the whole scene. The visual attention mechanism implemented here is thus *bottom up* and *covert* (see [Gruel and Martinet, 2021]).

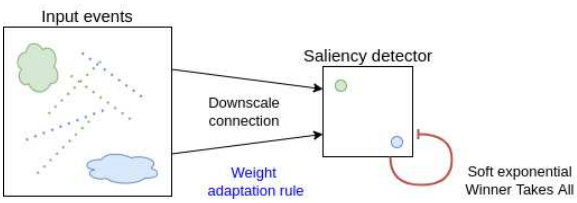


Fig. 4: Spiking neural network model used to detect saliency by event density, adapted from [Gruel et al., 2022b]

This whole mechanism relies solely on intrinsic SNN dynamics and dynamic adaptation rules applied to synaptic weights and population thresholds. This is a crucial feature as it leads to minimising the latency since it does not require the conversion of spiking events into a frame. The saliency detection is not specialised for any specific context or any specific shape, which allows for a good generalization ability of the network. The proposed architecture, shown in Fig. 4, is designed to be lightweight enough to enable running in real-time.

We use 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.

Input layer

The input layer translates sensor relative changes in the illumination (or events) into spikes. The spikes produced by the input layer are sent to the saliency detector via an excitatory downscaling connection. This corresponds to a convolutional layer with a kernel size $S \times S$, a stride S, without padding. The input neurons are separated into non-overlapping square regions of size $S \times S$. Each neuron in the input layer's subregions is connected to one corresponding neuron in the saliency detector layer.

Saliency detector

The saliency detection aggregates the active regions into distinct segments using a soft Winner-Takes-All (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 RoI are to be detected by the network, the soft WTA weight has been set experimentally to 0.02.

In the case of the saliency detector, a specific exponential WTA is implemented according to the radial basis function Eq. 3 in order to allow RoI of arbitrary sizes:

$$W_{WTA} = max(\frac{e^d}{w \times h}, w_{max}) \tag{3}$$

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 saliency in this layer is enabled by a dynamic weight adaptation rule between the input layer and the saliency detector, inspired by Hebb's rule: "cells that fire together wire together" [Hebb, 1949]. This rule is implemented by increasing or decreasing the weights of synapses that have recently fired, as described in Eq. 4.

$$\omega(t+1) = \begin{cases} \omega(t) + \Delta\omega & \text{if } ft_{synapse} \geq t \\ \omega_{init} & \text{if } ft_{synapse} < t - t_d \end{cases} \tag{4}$$

where $\omega(t)$ is the weight at the simulation step t of the synapse to which is applied the dynamic weight adaptation rule, $\Delta\omega$ the positive weight variation at each simulation step, ω_{init} the initial weight of the synapse, $ft_{synapse}$ the firing time of the last spike transmitted by the synapse and t_d the delay before the synaptic weight decays back to ω_{init}.

2.2 Reconstitution of foveated data

In this work, we consider the foveation process akin to the combination of a sample's events in high resolution and low resolution using a mask, as presented by the Fig. 5. This binary combination is a software simplification of the neuromorphic foveation concept as described above, and discriminates the fovea (events in high resolution — purple region in Fig. 5) from the retinal periphery (low resolution; i.e. spatially downscaled – yellow region in Fig. 5). The RoI (in red in Fig. 5) detected by the saliency detector mentioned earlier is thus assimilated to the fovea.

Let (x_{min}, y_{min}) and (x_{max}, y_{max}) be the coordinates of the delimiting points of the area of foveation detected by the saliency detector (as seen on Fig. 5),

$$\begin{aligned} \text{Fovea} = \{(x,y)| \\ x \in [x_{min}, x_{max}], y \in [(y_{min}, y_{max}]\} \\ \text{Periphery} = \{(x,y)| \\ x \notin [x_{min}, x_{max}], y \notin [(y_{min}, y_{max}]\} \end{aligned} \tag{5}$$

where $Fovea$ and $Periphery$ correspond to the coordinates of the set of salient and non-salient events respectively, in different resolutions.

2.3 Event data reduction

As explained above, the software implementation of neuromorphic foveation used in this work is akin to a binary foveation process merging event data of two different resolutions together. As it is more difficult to produce higher resolution data of an original dataset, we chose to use the spatial event downscaling methods described in [Gruel et al., 2022a]. In order to remain consistent with a future conversion of this software foveation model into hardware, where the saliency is detected on aggregated pixels before refining the resolution at the relevant areas in order to minimize the bandwidth, we follow the process described in Fig. 5. The original dataset, corresponding thereafter to the denomination "high resolution", is spatially reduced before being given as input to the saliency detector. A subsequent "multi resolution merging" process then combine both datasets into foveated event data according to the detected RoI.

Event data reduction is not trivial, as explained in [Gruel et al., 2022a]. Many different approaches can be used to produce the spatial downscaling depicted in Fig. 5. We decided to compare each method described in [Gruel et al., 2022a] in our experimental validation. It is to be noted that in order to process high and low resolution events using the same frame of reference, an expansion was applied to the spatially reduced data so that a reduced pixel physically corresponds to the size of $factor \times factor$ original pixels.

The following section provides a short description of each spatial reduction method used in this work, as well as the corresponding strengths and

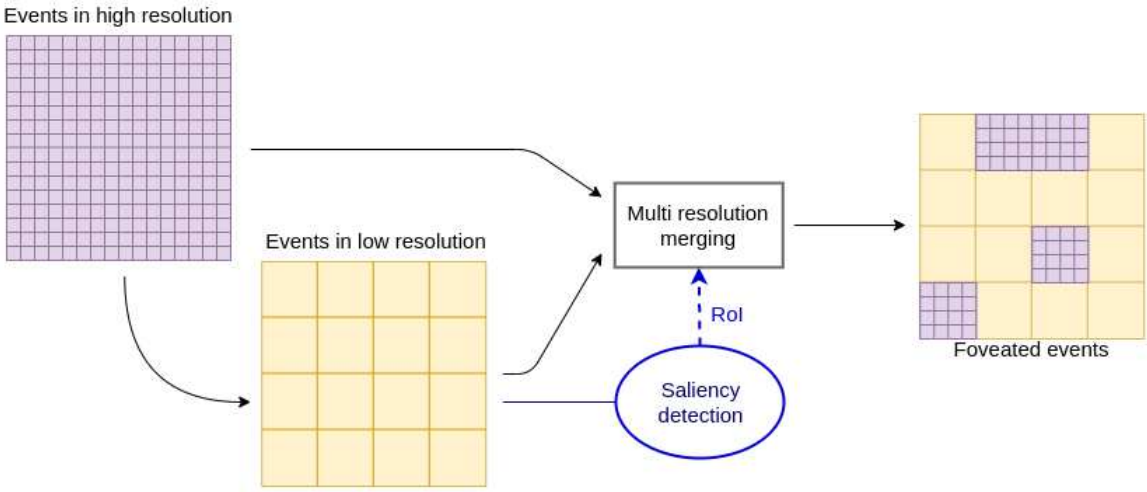


Fig. 5: Binary foveation of events using the corresponding high and low resolution (spatially downscaled by $factor$) and based on a known region of interest.

weaknesses. These methods reduce the data by downscaling the x, y coordinates of pixels, bringing an original $width \times height$ sensor size to a target $(width/ratio) \times (height/ratio)$ size, where $ratio$ is the downscaling ratio.

Event funnelling

The event *funnelling* method simply consists in dividing all the spatial coordinates of the events by the dividing factor (and removing any duplicate) to obtain the spatially reduced events. From a computational point of view, this downscaling method consists simply in updating the memory address of the event's x, y coordinates. Since this relies on one elementary operation repeated n times, with n the number of events processed, it has a complexity $O(n)$. This process can easily be implemented with a low resource usage given its simplicity. Other advantages lie in speed and absence of significant resource usage. However, a main drawback is the increased spatial density in the event data, as nearly every event is kept, which may have an impact on the target task.

Event count

The *event count* method consists in estimating the normalised value reached by the log-luminance related to the (larger) pixels in each the target size. This normalised *event count* is updated every time a new event is triggered. Since its complexity relies on its number n of events, it is therefore also $O(n)$. By definition, this method waits for the next event to be produced before it is able to trigger the next output event. As previously, its benefits include low computational resource consumption and speed.

Linear and cubic log-luminance reconstruction

The log-luminance reconstruction method aims to recreate the log-luminance curves seen by the pixels in the target sensor size, then extrapolating the events produced by the average of these curves (see [Gruel et al., 2022a] for more details). The curves can be estimated with a linear or cubic interpolation.

In contrast to both previous spatial reduction methods, this log-luminance reconstruction needs the information of when in the future will be the next event, which if obviously unknown in the current timestamp. Therefore it requires an adaptation for real time operation. Furthermore, even thought the real time processing needs to adjust the algorithm, the log-luminance reconstruction has the best optical coherence out of the existing event downscaling methods.

3 Experimental validation

To validate our proposed model, we apply two traditional computer vision tasks, semantic segmentation and classification, to foveated and spatially reduced datasets. All datasets are spatially downscaled by a dividing factor 4.

This section describes the datasets and the different models used to perform such tasks, as well as the comparative results.

3.1 Event-based datasets

DAVIS Driving Dataset 2017

The DAVIS Driving Dataset 2017 (DDD17) [Binas et al., 2017] contains 40 different driving sequences of event data captured by an event camera. However, since the original dataset provides only both grayscale images and event data without semantic segmentation labels, we used the segmentation labels provided in [Alonso and Murillo, 2019] that uses 20 different sequence intervals taken from 6 of the original DDD17 sequences. Furthermore, as only multi-channel representation of the events (normalised sum, mean and standard deviation for each polarity) are made available, we extracted the original events from DDD17 with the traditional $< x, y, p, t >$ structure using DDD20 tools[1] and selected the ones corresponding to the frames that have a ground truth. The resulting dataset is split into a training dataset consisting of 15,950 frames and a testing one consisting of 3,890 frames.

DVS 128 Gesture

The DVS128 Gesture dataset [Amir et al., 2017] has now become a standard benchmark in event data classification. It features 29 subjects recorded (with a 128×128 pixels DVS128 camera) performing 11 different hand gestures under 3 kinds of illumination conditions. A total of about 133 samples are available for each gesture, each composed roughly of 400K events, for a duration of 6 seconds approximately. The dataset is split in two sub-datasets to facilitate training: the train set contains 80% of the recorded samples and the test set contains the remaining 20%, with an even distribution of the 11 gestures in both parts.

[1] https://github.com/SensorsINI/ddd20-utils

As presented in the Fig. 6a, the event data's properties were compared for sample in high resolution (original dataset), low resolution (spatially downscaled with factor 4) and foveated (binary combination of the previous two).

3.2 Semantic segmentation

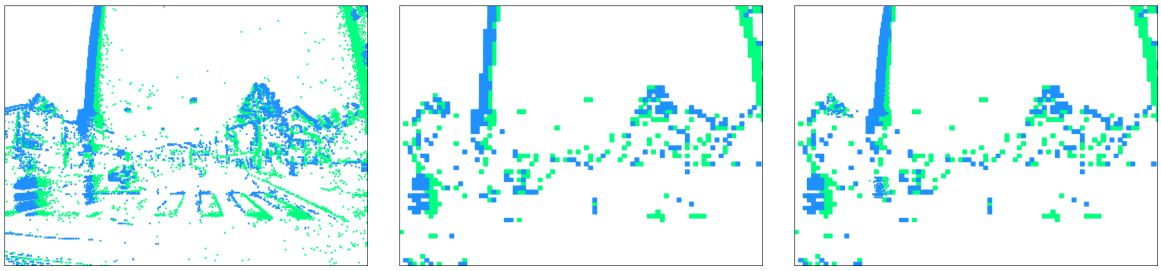
To validate the neuromorphic foveation, we apply it first to the computer vision task, semantic segmentation. It is a visual classification problem which consists of assigning a label, corresponding to a given class, for each pixel in the image. It is a key task in scene understanding that has been extensively studied using artificial neural networks, more specifically, Convolutional Neural Network (CNN) model with either frames or events as input. This task is often solved using an encoder-decoder CNN architecture, where the encoder downsamples the input image and the decoder upsamples the result returned by the encoder until the original size of the image is reached.

3.2.1 Ev-segNet

The semantic segmentation was performed using the model *Ev-SegNet* built by [Alonso and Murillo, 2019] as it outperforms all existing studies in this kind of task using event cameras. This model is inspired from current state-of-the-art semantic segmentation CNNs, slightly adapted to use the event data encoding. As shown in Fig. 9, it consists of an encoder-decoder architecture: an encoder represented by Xception model in which all the training is concentrated, and a light decoder connected to the encoder via skip connections to help deep neural architecture to avoid the vanishing gradient problem and also to make the fine-grained details learned in the encoder part used in the decoder to construct the initial image. Moreover, [Alonso and Murillo, 2019] use an auxiliary loss which increases convergence speed.

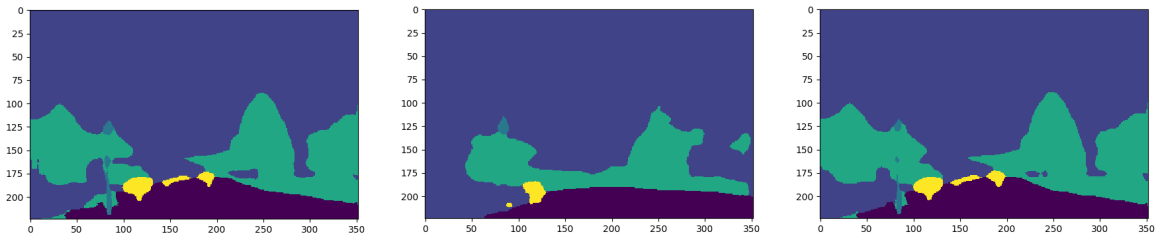
The model takes as input 6 channels representing the count, mean and standard deviation of the normalised timestamps of events happening at each pixel, within an interval of $50ms$ for the positive and negative polarities. It is applied to the DDD17 dataset described previously. Finally, the training is performed via backpropagation in order

(a) Accumulated events in the sample.



$N_{events} = 20,199$ $N_{events} = 882$ $N_{events} = 5129$

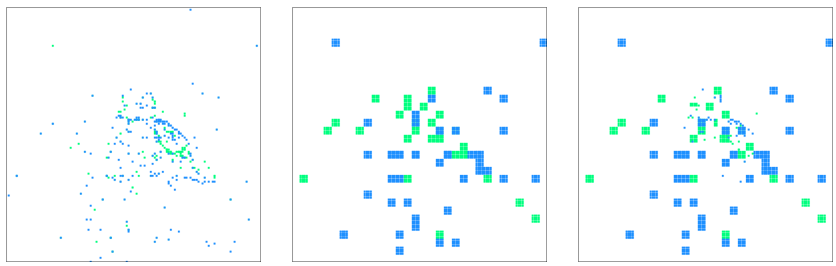
(b) Segments and performance values identified by Ev-SegNet in the sample.



$Accuracy = 86.06\%$ $Accuracy = 78.92\%$ $Accuracy = 84.75\%$

$MIoU = 0.5$ $MIoU = 0.3508$ $MIoU = 0.4648$

Fig. 6: Visual representation of the events and the different segments identified by Ev-SegNet [Alonso and Murillo, 2019] in the sample `rec1487417411_export_1467` from the DDD17 dataset [Binas et al., 2017], after various processes. The subplot on the left corresponds to the original data and in the middle to the same event data spatially reduced by 4 using the *event count* method. The right subplot corresponds to the sample foveated according to the RoI detected with *event count* method. **Top** Each frame corresponds to the accumulation of the events over 50ms, the sample's time-window. Green and blue pixels correspond respectively to positive and negative events.



$N_{events} = 308267$ $N_{events} = 13786$ $N_{events} = 190238$

Fig. 7: Visual representation of the events in the sample `user25_led`, corresponding to the gesture "left hand clockwise" (class 6) from the DVS128 Gesture dataset [Amir et al., 2017], after various process. The subplot on the left corresponds to the original data and in the middle to the same event data spatially reduced by 4 using the *event count* method. The right subplot corresponds to the sample foveated according to the RoI detected with *event count* method. Each frame corresponds to the accumulation of the events occurring during the sample's first 10 ms. Green and blue pixels correspond respectively to positive and negative events.

to minimise the soft-max cross-entropy loss measured by summing the error between the estimated pixels' classes and the true ones.

The semantic segmentation performance is measured thanks to standard metrics of semantic segmentation: the Accuracy (Eq. 6) and the Mean Intersection over Union (MIoU) (Eq. 7).

$$\begin{aligned}\text{Accuracy}(y,\widehat{y}) &= \frac{1}{N}\sum_{i=1}^{N}\delta(y_i,\widehat{y}_i)\\ &= \frac{TP+TN}{TP+TN+FP+FN}\end{aligned} \tag{6}$$

$$\begin{aligned}\text{MIoU}(y,\widehat{y}) &= \frac{1}{C}\sum_{j=1}^{C}\frac{\sum_{i=1}^{N}\delta(y_{i,c},1)\delta(y_{i,c},\widehat{y_{i,c}})}{\sum_{i=1}^{N}max(1,\delta(y_{i,c},1)\delta(\widehat{y_{i,c}},1))}\\ &= \frac{TP}{TP+TN+FP+FN}\end{aligned} \tag{7}$$

where y and $\widehat{y}$ are the desired output and the system output respectively. C is the number of classes. N is the number of pixels and δ denotes the Kronecker delta function. TP, TN, FP, and FN respectively stand for: true positive, true negative, false positive, and false negative.

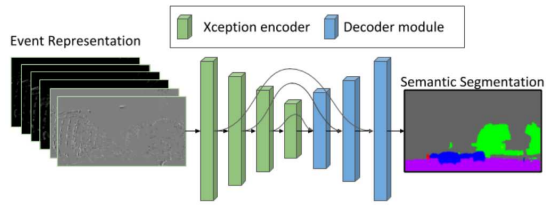


Fig. 9: CNN architecture of Ev-SegNet, from [Alonso and Murillo, 2019].

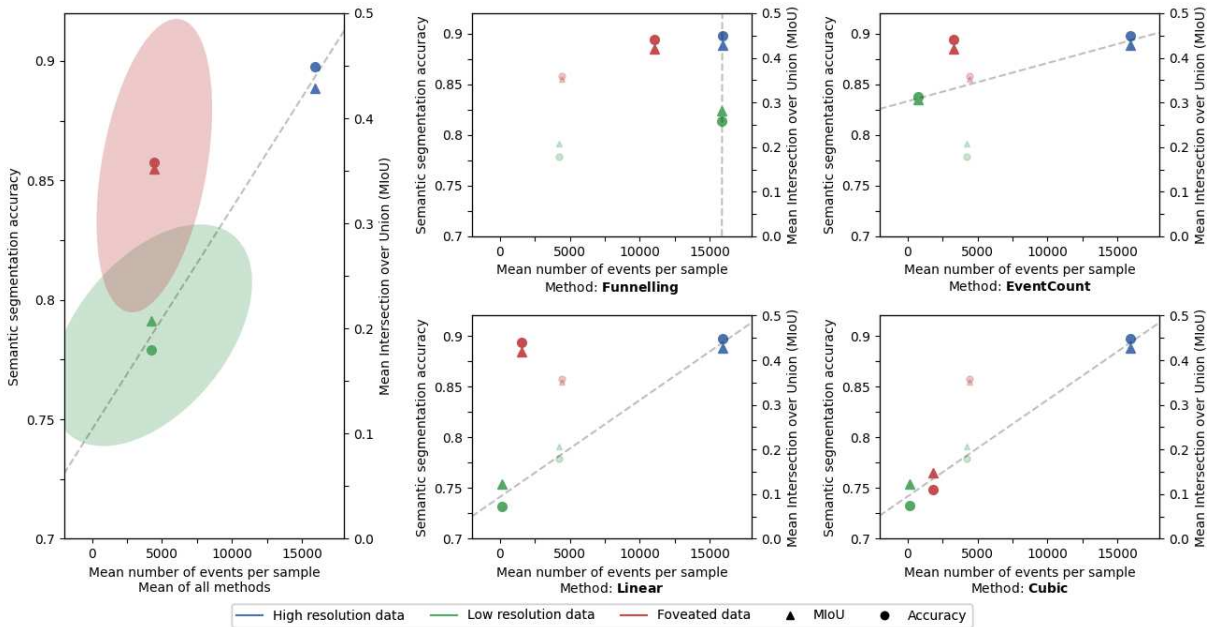


Fig. 8: Semantic segmentation performance according to the number of events in the dataset after processing for the event data in high resolution (in blue), low resolution (in green) and after foveation (in red). The subplot on the left depicts the mean values and the corresponding confidence ellipsis; it shows that foveated data managed to keep in average the same performance as the original dataset, while decreasing by two third the mean number of events to reach the low resolution's value. The mean values are plotted as an overlay in the subsequent subplots, to assess the quality of the foveation and the reduction compared to the others. This highlights the clear advantage of using the methods *linear* and *event count* compared to *funnelling*.

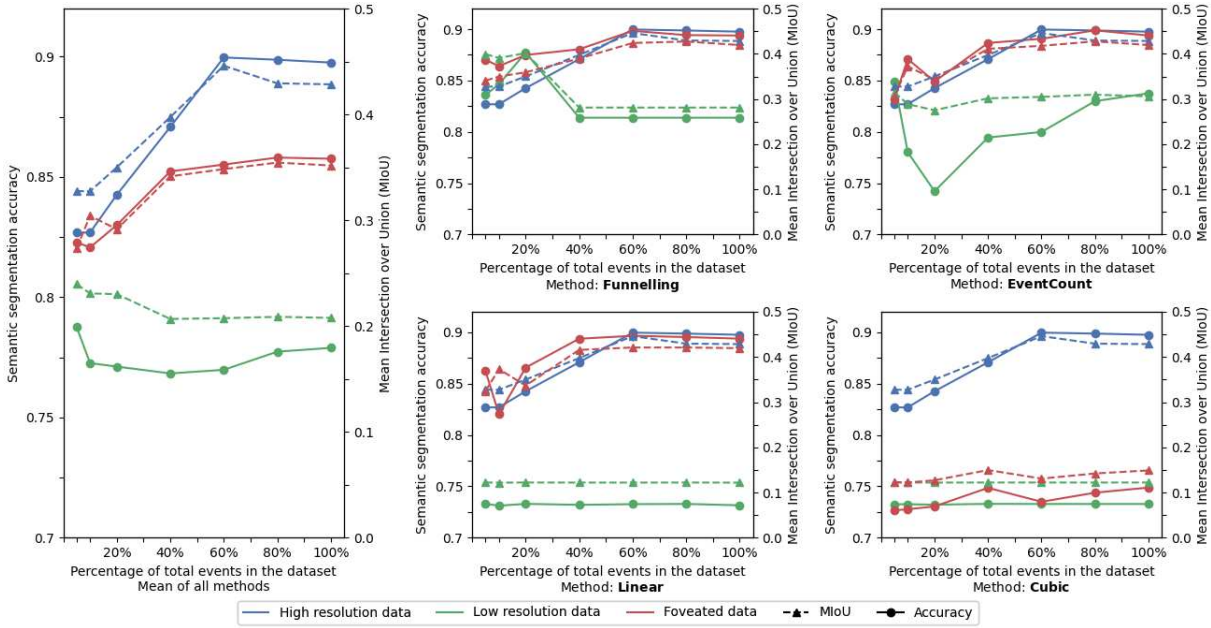


Fig. 10: Semantic segmentation performance evolution according to the percentage of total events in the dataset after processing for the event data in high resolution (in blue), low resolution (in green) and after foveation (in red). The subplot on the left depicts the mean values, and shows while all three types of data show the same behaviour, the foveated data outperforms in average the high resolution data from a 60% decrease and downwards. According to the subsequent subplots, the *funnelling* method is the one maintaining a highest accuracy the longest, either in the foveated or reduced dataset.

3.2.2 Segmentation results

Fig. 8 and 10 present a comparison between the different versions of the DDD17 dataset [Binas et al., 2017], i.e. in high (blue marks) and low (green marks) resolutions and after foveation (red marks), according to the semantic segmentation model *Ev-SegNet*'s performance. Fig. 8 depicts the trade-off between the accuracy, the MIoU and the mean number per sample for each reduction method — the first subplot corresponding to the mean and confidence ellipse for all methods combined. Fig. 10 shows the evolution of the segmentation performance for an increasingly subsampled input dataset, for each pre-process. Each dataset is reduced structurally by sub-sampling events in a stochastic way: events are filtered with a probability p, corresponding to the values in abscissa of Fig. 10.

In those first two graphs, the foveation is obtained by detecting the saliency on the data downscaled using the specified method, then merging it with the dataset reduced using the same method. This allows us to stay as close as possible to the hardware foveation design, by simulating the feedback loop by using twice the same reduced data.

Fig. 11 for its part studies the qualitative aspect of the saliency detection by presenting the semantic segmentation according to the mean number of events per sample. Each foveation method, i.e. each method on which the saliency was detected, is indicated in different colours. The average values for all datasets foveated using one method are presented in opaque, while each specific dataset's performance and number of events are overlayed.

As we want to optimize the performance while minimizing the number of events, the goal to be reached is situated on the upper left corner of this plot. *Event count* emerges as the method for optimal salience detection.

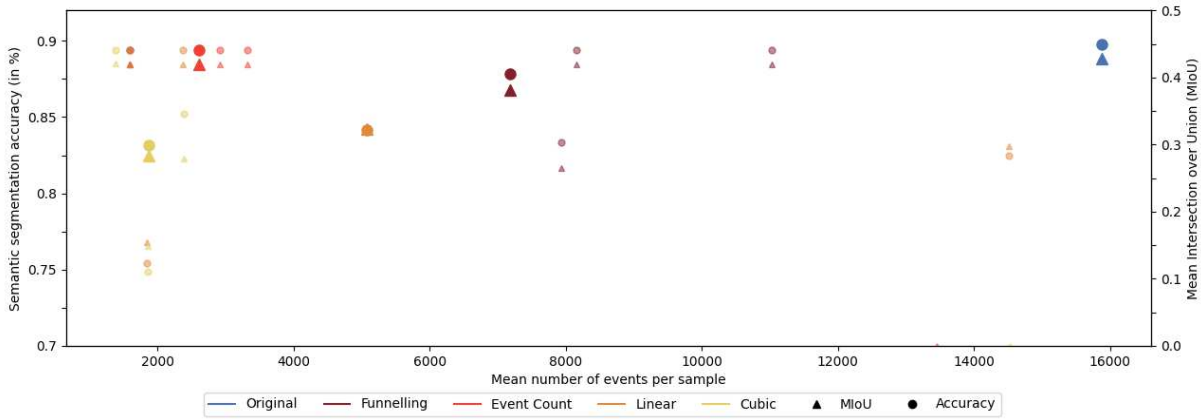


Fig. 11: Study of the qualitative aspect of the saliency detection, based on the semantic segmentation performance. The plot depicts the average semantic segmentation performance according to the method on which the saliency was detected. The overlayed dots correspond to different reduction methods with which the multi-resolution merging was achieved, using the RoI detected with one specific method. The goal is to minimize the mean number of events per sample while increasing the performance, thus it is to tend to the subplot's upper left corner. Foveated data using RoI detected on *event count* are the closest to this goal.

3.3 Classification

Furthermore, we test the impact of the neuromorphic foveation on a classification task, which assigns a label to each sample of a dataset.

3.3.1 Classification model

To test for the impact of the different event reduction methods on classification performances, we use an existing event-based algorithm [Grimaldi et al., 2022]. This online classification algorithm is an extension of a previous study entitled *HOTS: A Hierarchy of Event-Based Time-Surfaces for Pattern Recognition* [Lagorce et al., 2016]. In this work, they make object recognition on a stream of events through a feedforward hierarchical architecture using *time surfaces*, an event-driven analog representation of the local dynamics of a scene. Using a form of Hebbian learning, the network is able to learn, in an unsupervised way, progressively more complex spatio-temporal features which appear in the event stream. Once trained, one layer of this network transforms any incoming event from the stream of events into a novel event as it is selected in a layer of spiking neurons. Using it as a building block, such layers can be stacked together, each layer's output address space defining a novel input address space for the next layer. Inspired from this dynamical processing of the visual information, [Grimaldi et al., 2022] added a Multinomial Logistic Regression (MLR) as an online classifier and transformed the previous *post hoc* classification process, that counted the activity of the neurons of the last layer, into an always-on decision process. The MLR layer also takes *time surfaces* as input, and a formal demonstration was made to assimilate this algorithm to a SNN with Hebbian learning.

We test this method on a widely used and challenging event-based dataset for gesture recognition: DVS128 Gesture, described above.

3.3.2 Classification results

Fig. 12 presents a comparison between the different versions of the DVS 128 Gesture dataset [Amir et al., 2017], i.e. in high (blue marks) and low (green marks) resolutions and after foveation (red marks), according to the classification model *HOTS*'s performance. Fig. 12 depicts the trade-off between the accuracy, the MIoU and the mean number per sample for each reduction method — the first subplot corresponding to the mean and confidence ellipse for all methods combined. The

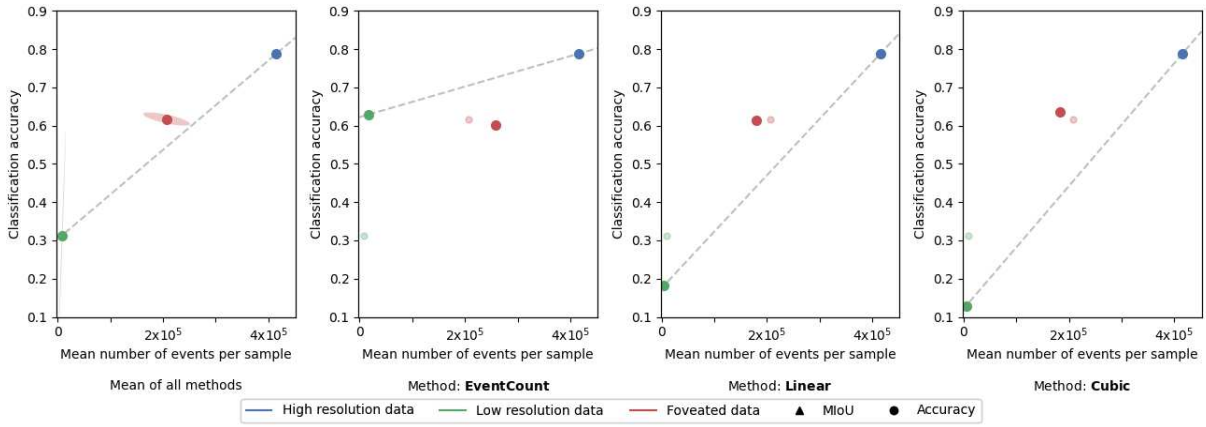


Fig. 12: Classification performance on DVS 128 Gesture according to the number of events in the dataset after processing for the event data in high resolution (in blue), low resolution (in green) and after foveation (in red). The subplot on the left depicts the mean values and the corresponding confidence ellipsis; it shows that the foveation applied to classification leads in average to a good trade-off between number of events and performance. The mean values are plotted as an overlay in the subsequent subplots, to assess the quality of the foveation and the reduction compared to the others. This highlights .

foveation is obtained using the same process as for the segmentation results.

3.4 Quantitative assessment of the saliency detection

Fig. 13 presents the spatial and temporal density according to the mean number of RoI detected per method, to discuss the quantitative aspect of the saliency detection.

The temporal density D_t corresponds to the activation probability of pixels averaged over the whole sensor, and is defined in Eq. 8.

$$D_t = \frac{\sum_{x=0}^{w}\sum_{y=0}^{h} P_{x,y}}{w \cdot h} \tag{8}$$

with w and h respectively the width and height of the sensor. The activation probability $P_{x,y}$ is calculated as the number of events (positive or negative) occurring at a given pixel divided by the time length of the sample:

$$P_{x,y} = \frac{\sum_{t=t_{min}}^{t_{max}} \delta(x_t, x) \cdot \delta(y_t, y)}{t_{max} - t_{min}} \tag{9}$$

with $P_{x,y}$ the activation probability of one pixel of coordinates (x, y), t_{min} and t_{max} respectively the minimum and maximum timestamp of the sample, and δ the Kronecker delta function, which returns 1 if the variables are equal, and 0 otherwise.

A contrario, the spatial density D_s is the activation probability of the whole sensor over a limited time-window averaged over the temporal length of the sample, as described in Eq. 10.

$$D_s = \frac{\sum_{t=t_{window}}^{T} P_{[t-t_w,t]}}{N_w} \tag{10}$$

with t_w and T respectively the length of the time-window and the length of the sample in time. The results in Fig. 13 are presented for $t_w = 50\mu s$. N_w corresponds to the number of successive time-windows in the sample, as presented in Eq. 11.

$$N_w = \lceil \frac{T}{t_w} \rceil \tag{11}$$

The activation probability $P_{[t-t_w,t]}$ is calculated as the ratio between the number of pixels activated during the time-window $[t - t_w, t]$ and the overall number of pixels in the sensor:

$$P_{[t-t_w,t]} = \frac{\sum_{t=t-t_w}^{t}\sum_{x=0}^{w}\sum_{y=0}^{h} \delta(t_x, t) \cdot \delta(t_y, t)}{w \cdot h} \tag{12}$$

with w and h respectively the width and height of the sensor, t_x and t_y the timestamp of any events occuring at coordinates (x, y) and δ the Kronecker

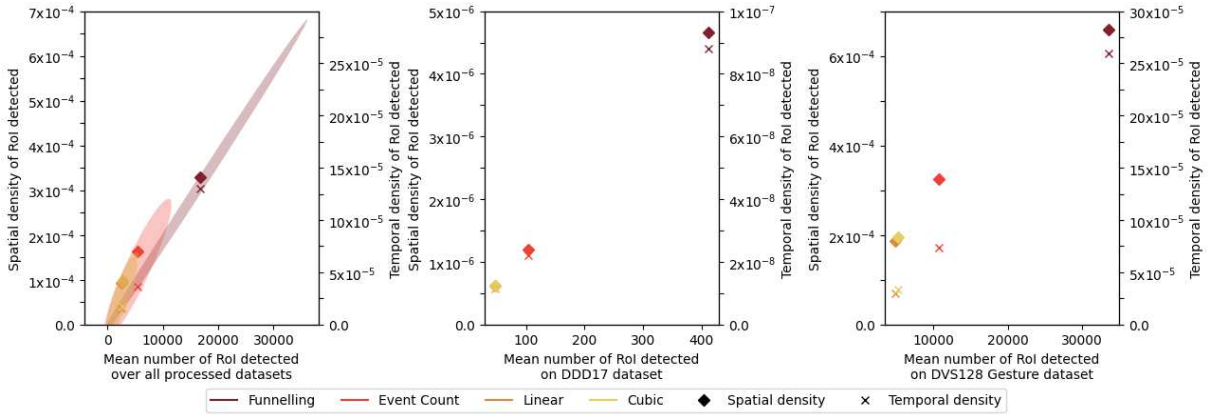


Fig. 13: Study of the quantitative aspect of the saliency detection, by comparing the RoI's spatial and temporal density according to the mean number of RoI detected per sample. No goal is to be reached here. However this graph shows that the *funnelling* method has a significantly higher temporal and spatial density, as well as a highest mean number of RoI detected.

delta function, which returns 1 if the variables are equal, and 0 otherwise.

4 Discussion

To validate our initial hypothesis, the foveation would have to produce a number of events significantly closer to the low resolution's while allowing for a performance closer to the high resolution's. In other terms, the foveated results should be above the dotted grey line on Fig. 8 and Fig. 12. We do observe a striking decrease in the number of events between pre- (high resolution) and post-processing (low resolution and foveation) of the dataset. Concerning the DDD17 dataset (Fig. 8), in average, the spatial downscaling and the foveation keep 30% of the original events. The most important drop of event is seen with the *linear* method: the reduced data drops to only 1% of the original size, and the foveated data to 10%. Similarly, the foveation's semantic segmentation performance is averaged over all methods are remarkably close to the high resolution's performance. A similar behaviour is found with the DVS128 Gesture dataset and its classification performance (Fig. 12): the average number of events per sample drops to 1% of the original metric after reduction, and to 50% after foveation, while the performance decreases respectively by 63% and only by 25%.

Fig. 8 pictures an outlier, the *cubic* method. The trade-off between performance and data size displayed by the foveation is not as good as in other methods; this can be explained by the fact that the method produces too sparse data to offer a coherent saliency detection.

All in all, those observations combined do confirm our core thesis, that neuromorphic foveation leads to the best trade-off between information quantity and quality, at least on a software level.

Furthermore, it is interesting to note that when comparing the proportional decrease of the number of events in the dataset post-process in Fig.10 while all three types of data show the same behaviour, the averaged foveated data outperforms the high resolution data from an 60% decrease and downwards. This is explained by the fact that the majority of events kept in the foveated dataset provides relevant information to the semantic segmentation model, while a significant part of the events in the original dataset is not as useful.

The less important drop of the data size when applying foveation to the classification of DVS 128 Gesture (Fig. 12) compared to the one presented in Fig. 8 is explained by the inner properties of the DVS 128 Gesture dataset: the lower spatial density of this dataset compared to DDD17, due to a static recording and a non-moving camera, leads to the detection of more densely aggregated RoI already containing most of the sample's event

data. In other words, the principal object of interest in DVS 128 Gesture is the hand, which will be the main subject of the saliency detection due to its constant movement. This is confirmed in Fig. 7 where the high resolution is only visible on the hand in the foveated sample. As the movement of the hand is the main cause of event production while recording the scene, the corresponding region contains the most events — thus the foveation on the DVS 128 Gesture dataset do not decrease the data size as much as when applied to a moving camera recording.

This theoretical reasoning is confirmed by the results presented in Fig. 13: the mean number of RoI detected on DVS 128 Gesture and its corresponding temporal and spatial densities (right sub-graph) are all significantly greater than those detected on DDD17 (middle sub-graph). Fig. 13 also highlights the important quantitative variations of the saliency detection according to the foveation method used: *funnelling* produces a great amount of RoI, due to its lack of event drop; while *linear* detects the least saliency. The choice of the method on which to detect the salience is not trivial, each one having its interests and its disadvantages. One must thus take time to think upstream about the physical properties of the dataset and the desired intensity of the foveation. All in all, when applying the foveation to classification of DVS 128 Gesture and semantic segmentation of DDD17, *event count* seems like a good trade-off between the three proposed methods.

The qualitative study of the saliency detection displayed in Fig. 11 using different methods (i.e. detecting saliency on data reduced using those methods) reinforces the interest of using the *event count* method for foveation. Indeed this figure highlights the overall advantage of *event count*, as its use leads to a minimised data quantity for a maximised semantic segmentation performance.

To meet our initial goal of finding the ideal process to reduce the number of events while maintaining the quality of the information transmitted, a final aspect can be discussed here to complete the results presented above: that of the generation time. Indeed the spatial reduction of events as well as the neuromorphic detection of salience requires a non-negligible generation time. The first has been discussed in [Gruel et al., 2022a], and depends strongly on the selected reduction method. The second varies with the technical material used. Indeed the SNN model can be run either on CPU, using a SNN simulator such as PyNN [Davison et al., 2009], or on GPU with adapted simulators such as Norse [Pehle and Pedersen, 2021]. On both cases the simulation time increases with the number of neurons in the model and the size of the input data. A third option could be to use neuromorphic chips, such as Human Brain Project's SpiNNaker [Furber and Bogdan, 2020] or Loihi [Davies et al., 2018], which enable fast and low power simulations and which simulation time only relies on the input data size. All in all, it is very unlikely that these two processes (reduction and saliency detection) combined can be done in real time.

However we seek to convert this software neuromorphic foveation process into a hardware implementation, using the foveated sensor introduced by [Serrano-Gotarredona et al., 2022]. With such a sensor that implements the spatial reduction and foveation electronically, we can disregard the generation time.

5 Conclusion

In this work, we demonstrate the stakes of foveation applied to event data for semantic segmentation. Such a strategy does concurrently preserve the accuracy of event data processing and greatly reduce the amount of data needed for the task.

Further work will implement the hardware neuromorphic foveation feedback process by concurrently using the foveated DVS presented in [Serrano-Gotarredona et al., 2022] and a neuromorphic chip, e.g. Human Brain project's SpiNN-3 [Furber and Bogdan, 2020], in order to record foveated event data in an embedded fashion.

Acknowledgments. 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.

The authors are grateful to the OPAL infrastructure from Université Côte d'Azur for providing resources and support.

Statements and Declarations

Funding

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.

Conflict of interest

Not applicable.

Ethics approval

Not applicable.

Consent to participate

Not applicable.

Consent for publication

Not applicable.

Availability of data and materials

Not applicable.

Code availability

The code is publicly available online at: github.com/amygruel/FoveationStakesDVS/.

Authors' contributions

The authors Teresa Serrano-Gotarredona, Jean Martinet, Amélie Gruel and Bernabé Linares-Barranco contributed to the conceptualisation and methodology design of the study. The project coordination and administration were handled by Amélie Gruel. Jean Martinet and Laurent Perrinet carried out the funding acquisition and supervision. Formal analysis and investigation were performed by Amélie Gruel, Dalia Hareb and Antoine Grimaldi. Results visualisation and presentation were realised by Amélie Gruel. The first draft of the manuscript was written by Amélie Gruel, Dalia Hareb and Jean Martinet, and Antoine Grimaldi, Laurent Perrinet and Teresa Serrano-Gotarredona added to a second draft by reviewing and editing. All authors read and approved the final manuscript.

References

[Alonso and Murillo, 2019] Alonso, I. and Murillo, A. (2019). EV-SegNet: Semantic Segmentation for Event-based Cameras. *CVPR W*.

[Amir et al., 2017] Amir, A., Taba, B., Berg, D., Melano, T., McKinstry, J., Di Nolfo, C., Nayak, T., Andreopoulos, A., Garreau, G., Mendoza, M., et al. (2017). A low power, fully event-based gesture recognition system. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7243–7252.

[Araujo and Dias, 1997] Araujo, H. and Dias, J. (1997). An introduction to the log-polar mapping. *Proceedings II Workshop on Cybernetic Vision*, (1):139–144.

[Bear et al., 2007] Bear, M. et al. (2007). The Human Eye. In *Neurosciences, Exploring the brain*, Wolters Kluwer Health.

[Binas et al., 2017] Binas, J., Neil, D., Liu, S.-C., and Delbruck, T. (2017). DDD17: End-To-End DAVIS Driving Dataset. *arXiv:1711.01458 [cs]*.

[Daucé et al., 2020] Daucé, E., Albiges, P., and Perrinet, L. U. (2020). A dual foveal-peripheral visual processing model implements efficient saccade selection. *Journal of Vision*, 20(8):22–22.

[Daucé and Perrinet, 2020] Daucé, E. and Perrinet, L. (2020). Visual Search as Active Inference. In Verbelen, T., Lanillos, P., Buckley, C. L., and De Boom, C., editors, *Active Inference*, Communications in Computer and Information Science, pages 165–178. Springer International Publishing.

[Davies et al., 2018] Davies, M. et al. (2018). Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro*.

[Davison et al., 2009] Davison, A. P., Brüderle, D., Eppler, J. M., Kremkow, J., Muller, E., Pecevski, D., Perrinet, L., and Yger, P. (2009). Pynn: a common interface for neuronal network simulators. *Frontiers in Neuroinformatics*, 0.

[Delbrück et al., 2021] Delbrück, T., Graca, R., and Paluch, M. (2021). Feedback control of event cameras. *CoRR*, abs/2105.00409.

[Finateu et al., 2020] Finateu, T., Niwa, A., Matolin, D., Tsuchimoto, K., Mascheroni, A., Reynaud, E., Mostafalu, P., Brady, F. T., Chotard, L., Legoff, F., Takahashi, H., Wakabayashi, H., Oike, Y., and Posch, C. (2020).

5.10 a 1280×720 back-illuminated stacked temporal contrast event-based vision sensor with 4.86µm pixels, 1.066geps readout, programmable event-rate controller and compressive data-formatting pipeline. *2020 IEEE International Solid- State Circuits Conference - (ISSCC)*, pages 112–114.

[Furber and Bogdan, 2020] Furber, S. and Bogdan, P. (2020). *Spinnaker - a spiking neural network architecture*. NOW Publishers INC.

[Gehrig and Scaramuzza, 2022] Gehrig, D. and Scaramuzza, D. (2022). Are high-resolution cameras really needed? *arXiv*.

[Grimaldi et al., 2022] Grimaldi, A., Boutin, V., Ieng, S.-H., Benosman, R., and Perrinet, L. (2022). A robust event-driven approach to always-on object recognition.

[Gruel and Martinet, 2021] Gruel, A. and Martinet, J. (2021). Bio-inspired visual attention for silicon retinas based on spiking neural networks applied to pattern classification. *CBMI*.

[Gruel et al., 2022a] Gruel, A., Martinet, J., Serrano-Gotarredona, T., and Linares-Barranco, B. (2022a). Event data downscaling for embedded computer vision. In *VISAPP*.

[Gruel et al., 2022b] Gruel, A., Vitale, A., Martinet, J., and Magno, M. (2022b). Neuromorphic event-based spatio-temporal attention using adaptive mechanisms. In *AICAS*.

[Guo et al., 2017] Guo, M., Huang, J., and Chen, S. (2017). Live demonstration: A 768 × 640 pixels 200meps dynamic vision sensor. In *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–1.

[Hao et al., 2021] Hao, Q., Tao, Y., Cao, J., Tang, M., Cheng, Y., Zhou, D., Ning, Y., Bao, C., and Cui, H. (2021). Retina-like Imaging and Its Applications: A Brief Review. *Applied Sciences*, 11(15):7058.

[Hebb, 1949] Hebb, D. (1949). The organization of behavior: A neuropsychological theory. *Journal of the American Medical Association*, 143(12).

[Kubendran et al., 2021] Kubendran, R., Paul, A., and Cauwenberghs, G. (2021). A 256x256 6.3pj/pixel-event query-driven dynamic vision sensor with energy-conserving row-parallel event scanning. In *2021 IEEE Custom Integrated Circuits Conference (CICC)*, pages 1–2.

[Lagorce et al., 2016] Lagorce, X., Orchard, G., Galluppi, F., Shi, B. E., and Benosman, R. B. (2016). Hots: a hierarchy of event-based time-surfaces for pattern recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(7):1346–1359.

[Land, 2018] Land, M. F. (2018). *Eyes to See: The Astonishing Variety of Vision in Nature*. Oxford University Press.

[Li et al., 2019] Li, C., Longinotti, L., Corradi, F., and Delbruck, T. (2019). A 132 by 104 10 micro m-pixel 250 micro w 1kefps dynamic vision sensor with pixel-parallel noise and spatial redundancy suppression. In *2019 Symposium on VLSI Circuits*, pages C216–C217.

[Lichtsteiner et al., 2008] Lichtsteiner, P., Posch, C., and Delbruck, T. (2008). A 128x128 120 db 15 ms latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits*, 43(2).

[Paugam-Moisy and Bohte, 2012] Paugam-Moisy, H. and Bohte, S. M. (2012). Computing with Spiking Neuron Networks. In *Handbook of Natural Computing*. Springer-Verlag.

[Pehle and Pedersen, 2021] Pehle, C. and Pedersen, J. E. (2021). Norse - A deep learning library for spiking neural networks. Documentation: https://norse.ai/docs/.

[Posch et al., 2014] Posch, C., Serrano-Gotarredona, T., Linares-Barranco, B., and Delbruck, T. (2014). Retinomorphic event-based vision sensors: Bioinspired cameras with spiking output. *Proceedings of the IEEE*, 102(10):1470–1484.

[Sarvaiya et al., 2009] Sarvaiya, J. N., Patnaik, S., and Bombaywala, S. (2009). Image registration using log-polar transform and phase correlation. In *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, pages 1–5.

[Serrano-Gotarredona et al., 2022] Serrano-Gotarredona, T., Faramarzi, F., and Linares-Barranco, B. (2022). Electronically foveated dynamic vision sensor. In *2022 IEEE International Conference on Omni-layer Intelligent Systems (COINS)*, pages 1–5.

[Suh et al., 2020] Suh, Y., Choi, S., Ito, M., Kim, J., Lee, Y., Seo, J., Jung, H., Yeo, D.-H., Namgung, S., Bong, J., Yoo, S., Shin, S.-H., Kwon, D., Kang, P., Kim, S., Na, H., Hwang, K., Shin, C., Kim, J.-S., Park, P. K. J., Kim, J., Ryu, H., and Park, Y. (2020). A 1280×960 dynamic vision sensor with a 4.95-micro m pixel pitch and motion artifact minimization. In *2020*

IEEE International Symposium on Circuits and Systems (ISCAS), pages 1–5.

[Traver and Pla, 2003] Traver, V. J. and Pla, F. (2003). Designing the Lattice for Log-Polar Images. *Discrete Geometry for Computer Imagery*, pages 164–173.