Event driven bio-inspired attentive system for the iCub humanoid robot on SpiNNaker

DOI:
10.1088/2634-4386/ac6b50

Citation for published version (APA):
D’Angelo, G., Perrett, A., Iacono, M., Furber, S., & Bartolozzi, C. (2022). Event driven bio-inspired attentive system for the iCub humanoid robot on SpiNNaker. *Neuromorphic Computing and Engineering*. https://doi.org/10.1088/2634-4386/ac6b50

Published in:
Neuromorphic Computing and Engineering

Event driven bio-inspired attentive system for the iCub humanoid robot on SpiNNaker

NEUROMORPHIC
Computing and Engineering

RECEIVED
22 October 2021

REVISED
26 April 2022

ACCEPTED FOR PUBLICATION
28 April 2022

PUBLISHED
20 May 2022

PAPER

Event driven bio-inspired attentive system for the iCub humanoid robot on SpiNNaker

Giulia D'Angelo[1,2,3,*], Adam Perrett[1,3], Massimiliano Iacono[2], Steve Furber[1] and Chiara Bartolozzi[2]

[1] The University of Manchester, Manchester, United Kingdom
[2] Istituto Italiano di Tecnologia, Genoa, Italy
[*] Author to whom any correspondence should be addressed.
[3] These authors contributed equally to this work.

E-mail: giulia.dangelo@iit.it, adam.perrett@manchester.ac.uk, massimiliano.iacono@iit.it, steve.furber@manchester.ac.uk and chiara.bartolozzi@iit.it

Keywords: attention, neuromorphic computing, bioinspired, event-driven, robotics

Abstract

Attention leads the gaze of the observer towards interesting items, allowing a detailed analysis only for selected regions of a scene. A robot can take advantage of the perceptual organisation of the features in the scene to guide its attention to better understand its environment. Current bottom–up attention models work with standard RGB cameras requiring a significant amount of time to detect the most salient item in a frame-based fashion. Event-driven cameras are an innovative technology to asynchronously detect contrast changes in the scene with a high temporal resolution and low latency. We propose a new neuromorphic pipeline exploiting the asynchronous output of the event-driven cameras to generate saliency maps of the scene. In an attempt to further decrease the latency, the neuromorphic attention model is implemented in a spiking neural network on SpiNNaker, a dedicated neuromorphic platform. The proposed implementation has been compared with its bio-inspired GPU counterpart, and it has been benchmarked against ground truth fixational maps. The system successfully detects items in the scene, producing saliency maps comparable with the GPU implementation. The asynchronous pipeline achieves an average of 16 ms latency to produce a usable saliency map.

1. Introduction

Visual attention guides the perception of the environment [1]. It is a mechanism that selects relevant parts of the scene to sequentially allocate the limited available computational resources to smaller regions of the field of view. In the animal world, this is coupled with eye movements, aimed to sequentially centre the selected region within the highest resolution region of the retina [2]. The detailed analysis only of salient regions of the visual field can dramatically reduce the computational load of processing the full visual field at once. In a similar manner, a robot working in real-time can exploit visual attention advantageously to optimise the use of computational resources. The motivation of this work is to produce an analogous reduction in computational loads for autonomous systems. Robots, such as the humanoid robot iCub [3], need to generate fast and precise response to autonomously interact with the environment reacting to external stimuli. Recent studies in computer vision have exploited the concept of attention for different tasks: classifying MNIST handwritten numbers only on regions of interests (ROIs) of the visual field with the 1.07% error [4], fixation prediction adding audio cues [5], visual search [6], and object recognition, where it has been demonstrated that attentional selection (based on saliency) increases the number of regions where objects are identified with random ROI selection [7].

Attention has attracted interest since the first psychological experiments where Yarbus [8] were recording the fixation points of subjects examining different pictures. Since then, attention has been modelled in order to understand its underlying neural implementation, and to equip artificial agents with similar capability to

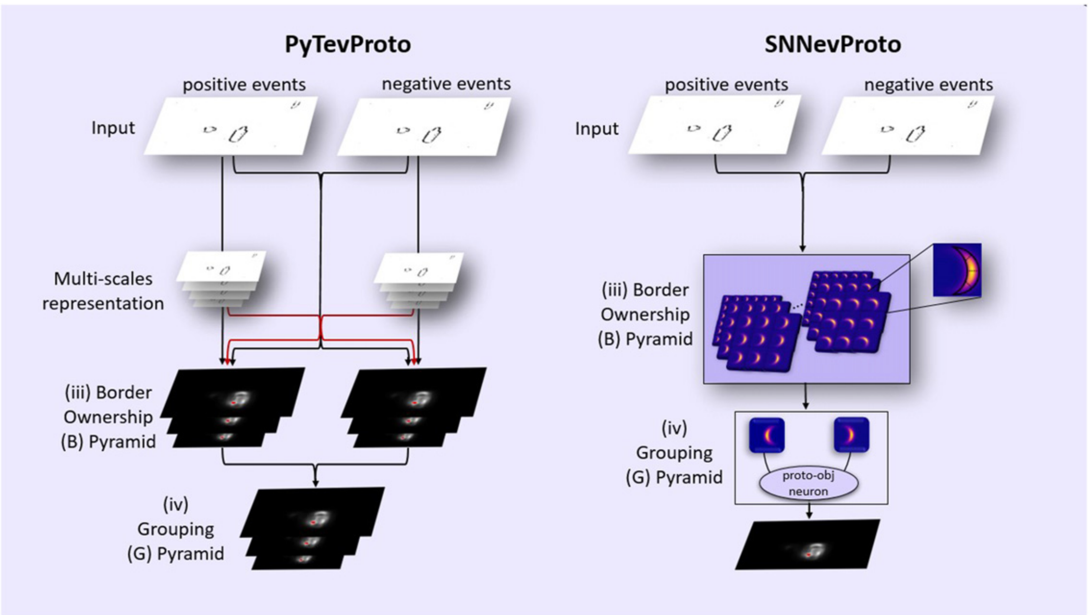
PyTevProto
positive events
negative events
Input
Multi-scales representation
(iii) Border Ownership (B) Pyramid
(iv) Grouping (G) Pyramid
SNNevProto
positive events
negative events
Input
(iii) Border Ownership (B) Pyramid
(iv) Grouping (G) Pyramid
proto-obj neuron

Figure 1. An overview of the model architectures for the PyTevProto (on the left) and the SNNevProto (on the right). The events are split based on the polarity and fed into the two models as input. The event-based model generates different scales by subsampling the 'event-frame' and creating a pyramid. The resulting scaled 'event-frames' are convolved with VM filters at four different orientations (border ownership pyramid) and grouped at the grouping layer directly processes the input with the two layers o border ownership and grouping pyramids. The red lines are inhibitory signals. The spike-based implementation processes the events asynchronously exploiting layers of VM shaped neurons at different scales and rotations. The proto-object neurons (grouping pyramid layer) integrate the response connecting VM filters with opposite side and pool the response from different scales. The outcome of both models is the saliency map.

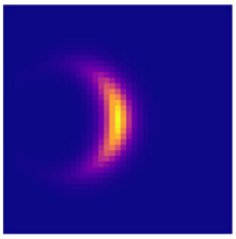

Figure 2. Representation of the VM filter described in equation (1) at 0°.

obtain a reasonable perception of the scene [9]. Attention is a complex mechanism that results from the interplay of a bottom–up process that is driven by the physical characteristic of the stimuli and top–down effects that depends on priors and goals [10]. Diverse studies tried to model the bottom–up components of attention. Some proposed the use of the saliency map formalism [11–13]. A saliency map is the representation of visual saliency in a scene, where each item appears to be interesting (salient) based on the observer visual exploration [14].

Specifically, selective attention extracts features from the environment and explains the situation as fast as possible filtering what is not necessary to understand the scene [15].

The widely used feature-based saliency model [9] extracts in parallel multiple different visual features and finds regions of high contrast within each feature channel. Their contribution defines the saliency of each point in the field of view. The weight of each feature map can be modulated to model the effect of top–down mechanisms competing with each other for the representation of the scene. This model was then augmented [16], by integrating principles of perceptual grouping of individual components that reflect 'Gestalt laws' as proximity, common fate, good continuity and closure [17].

These principles give perceptual saliency to regions of the visual field that can be perceived as 'proto-objects' [18, 19].

A proto-object describes regions of the visual field that may coincide to real objects in the physical world, referring to the human ability to organise part of the retina stimuli into structures [20]. The work of Russell *et al* [21] improved [16] by creating a filter capable of detecting partial contours. Recent studies added other

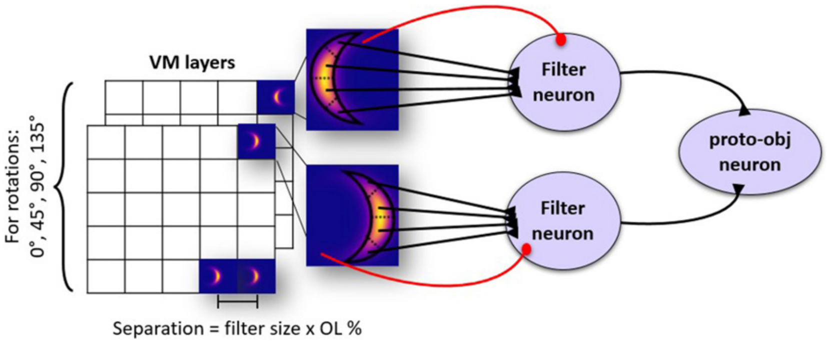

Figure 3. Representation of a VM layer and its connections. Each VM filter is split in 4 sections all connected to the same filter neuron. The area around the 'active' part of the neuron (moon shaped yellow region) is connected to the filter neuron with an Inhibitory connection (red lines). This stage of the model represents the border ownership pyramids detecting close contours. Two complementary VM filters with opposite orientation are then connected to the same proto-object neuron (grouping pyramid) to identify possible proto-objects. This structure is repeated for each layer with different orientations of the filter: 0°, 45°, 90° and 135°.

sources of information to the proto-object model such as motion [22], depth [23] and texture [24]. Further, a new line of research has started to develop these types of models using event-driven cameras as input. In these cameras, the contrast change in the scene is outputted asynchronously, with high temporal resolution, low latency, and most importantly, reducing data rate. For a real-time application in a robotics scenario this leads to a faster response given the low processing required [25, 26].

Adams *et al* [27] exploited the address-event representation and the neuromorphic platform SpiNNaker to allow the humanoid robot iCub [3] robot to perform real-world tasks fixating attention upon a selected stimulus. Rea *et al* [28] exploited visual attention for a bio-inspired pipeline using event-driven cameras (ATIS cameras) [29] mounted on iCub, the neuromorphic robot [30]. This implementation [28] exploits the low latency of the event cameras, further increasing the speed of the response towards online attention, but does not include the proto-object concept, that was later included by modifying a frame-based proto-object model [21] in a way that is suitable for event-based cameras [31]. The implementation proposed by Iacono *et al* [31] adapts the proto-object model based on RGB cameras to event-driven input, using the contrast feature maps naturally encoded by event-driven cameras. However in that work did not fully exploit the advantages given by the sensor. In fact events were accumulated over time generating frames that were then processed using a GPU. In an attempt to decrease latency and computational cost we implemented the model proposed in [31] on the SpiNNaker neuromorphic computing platform [32], that is able to properly exploit the asynchronous output of the event-based cameras. SpiNNaker is a dedicated neuromorphic computational device which provides a digital platform to model spiking neural networks at large scale in real time. Using an asynchronous and highly parallel architecture, large numbers of small data packets can be processed, which in most applications represents spikes being sent between biological neurons. This provides an ideal computational tool for event based processing.

The platform supports asynchronous spiking models that propagate events from the sensors in the network. Such models yield minimum processing latency, most of which depends on the propagation across layers and on the accumulation of sufficient information [33]. The contribution of this work is the validation of the model implemented on SpiNNaker (SNNevProto) through a direct comparison with the event-driven proto-object (PyTevProto) (i.e. its counterpart implemented on GPU using PyTorch). We compared the two models using the dataset from [31] (SalMapIROS) and benchmarked both against ground truth fixation maps [34]. We analyse the trade off between accuracy, number of neurons, computational cost and latency.

2. Event-based spiking neural network proto-object saliency model

This work takes inspiration from the bio-inspired saliency-based proto-object model for frame-based cameras initially proposed by Russell *et al* [21] and its event-camera adaptation [31]. The former is composed of three channels: intensity, colour opponency and orientation, competing with each other to represent the scene. Its core is composed of four layers: center surround pyramids (CSP), edge pyramids, border ownership and the grouping pyramid (see figure 1).

The CSP layer convolves the input image with a difference of Gaussians kernel to detect regions in the scene with either positive or negative contrast, emulating the center surround (or bipolar) cells present in the retina [35, 36]. In parallel, the system convolves the RGB image with Gabor filters, emulating the edge extraction

Table 1. Table showing the number of neurons and SpiNNaker boards required given a percentage of overlapping for the VM filters. The spalloc server was used to run these jobs which allocates boards in multiples of 3.

OL%	# of neurons	# of SpiNNaker boards
10%	10 428	3
20%	12 000	3
30%	15 801	3
40%	22 266	3
50%	30 306	6
60%	48 878	6
70%	82 084	12
80%	176 248	24

Table 2. The percentage firing thresholds for different population connections, input->filter is the only inhibitory connection. Percentage firing threshold is the percentage of the pre-synaptic population that need to fire to produce a spike in the post-synaptic population. Inhibitory connections do not induce a spike but are scaled in the same fashion. This metric is used to standardise weights across varying convolutional kernel sizes.

Input->segment	Segment->filter	Input->filter	Filter->proto-object
0.02%	0.8%	0.0013%	0.75%

done by the primary visual cortex [37]. The border qwnership and grouping pyramid implement the 'Gestalt laws' of continuity and figure-ground segmentation, mimicking the neurons in the Secondary visual cortex area, which are mostly selective to edges [38]. All the computation steps are performed at several scales to obtain object size invariance/tolerance. In the border ownership layer the output of the CSP is convolved with curved von Mises (VM) filter (see figure 2). The convolution with four different orientations of the filter detects partial contours of objects. All filters in the same location are connected via inhibitory connections to each other creating local competition for the dominant orientation. The output is then pooled by the grouping pyramid which combines oppositely rotated contours oriented to the same centre forming a partially closed contour. Closed contour activity is captured by the proto-object neurons whose combined activity creates the saliency map.

In [31] we have adapted this model to run using the output of event-driven cameras. Here we take a step further, implementing the model with spiking neurons on neuromorphic hardware.

Event-driven camera's pixels asynchronously produce an event every time a local illumination change occurs providing the information of positive or negative change in contrast. As such, they perform an inherent operation of edge extraction that can functionally be equivalent to the edge extraction performed by center-surround (CS) cells in the frame-based model. A similar contrast change information is provided by the CS cells [43]. The event-driven camera does not obtain the local contrast change due to lateral inhibition as in the CS cells, but rather due to the relative motion between the camera and the scene. The two processes are different but the related outcome, the edge extraction and the contrast information, are similar. These inherent capabilities can be used as substitutes for the first two layers of processing in the event-based version of the saliency-based model [31]: CS filtering and edge extraction. In fact, assuming a dynamic scene where a dark object is moving over a white background the leading edge would produce negative events and the trailing edge positive events, therefore providing information about the object contrast with respect to the background. In the PyTevProto model implementation running on GPU, the output from the event-based cameras is used to create frames of events divided into positive and negative polarity. The frames of events are fed into the border ownership layer following the process explained above.

This work proposes a new fully spiking based pipeline, with dedicated neuromorphic hardware, aiming to improve the speed and reduce the latency of the model. The SpiNNaker neuromorphic platform [32] acts as a computation medium modelling the SNN in a feedforward architecture (see figure 1). The neural model mimics the cells as populations of current-based leaky integrate and fire neurons.

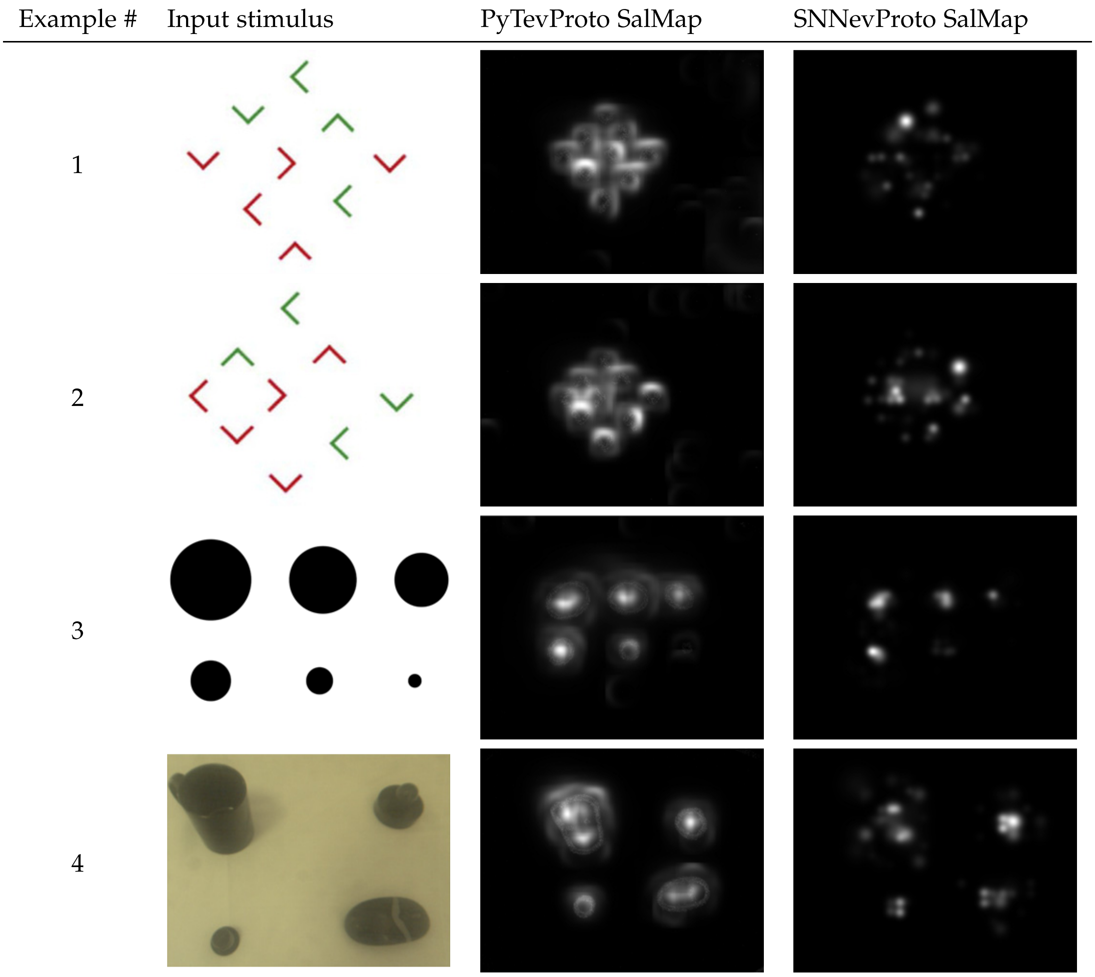

Figure 4. Qualitative comparison among the PyTevProto and the SNNevProto. From the left column to the right column: the example number, a RGB image representing the scene shown to iCub (the input stimulus), PyTevProto saliency map and SNNevProto saliency map. These examples are a selection from 13 scenarios of the SalMapIROS dataset. The events are recorded directly from the event-driven cameras mounted on iCub's eyes. The objects and the 2D printed patterns are placed on a desk in front of the robot. The RGB input images are only for a better visualisation of the input stimulus.

These neurons process the data coming from the ATIS cameras in form of events carrying the information of the position in the visual field, polarity (positive or negative contrast change) and the timestamp of the event. The VM filter, shown in figure 2, is a kernel designed to respond to curved edges that can potentially delimit a closed area. They are formalised as a curve (equation (1)) with the largest value at its midpoint providing the ideal shape to respond to closed contours:

$$\mathrm{VM}_{\theta}(x,y) = \frac{\exp\left(\rho \cdot R_0 \cdot \cos(a\ \tan\ 2(-y,x) - \theta)\right)}{I_0(\sqrt{x^2 + y^2 - R_0})} \tag{1}$$

where x and y are the kernel coordinates with origin in the centre of the filter, R_0 is the radius of the filter, ρ determines the arc length of active pixels in the kernel allowing to change the convexity of the kernel, θ its orientation and I_0 is the modified Bessel function of the first kind. The VM output is then thresholded to reduce sensitivity to localised activity:

$$e(x,y) = \begin{cases} 1 & \text{for } \mathrm{VM}_{\theta}(x,y) > 0.75 \\ -1 & \text{else} \end{cases} \tag{2}$$

where $e(x,y)$ describes whether the pixel at $(x,\ y)$ is connected to the filter neuron with excitatory synapses $(e(x,y) = 1)$ or inhibitory synapses $(e(x,y) = -1)$ (see figure 3). Connection weights, w, are determined using equation (3) where n is the size of the pre-synaptic population and p is the percentage firing threshold for that particular projection between populations. A value of $5\mu S$ is chosen as it is the minimum weight at which one excitatory input spike produces a spike in the post-synaptic neuron in this implementation of conductance based neurons. Inhibitory connections are scaled using the same method, but do not produce a post-synaptic

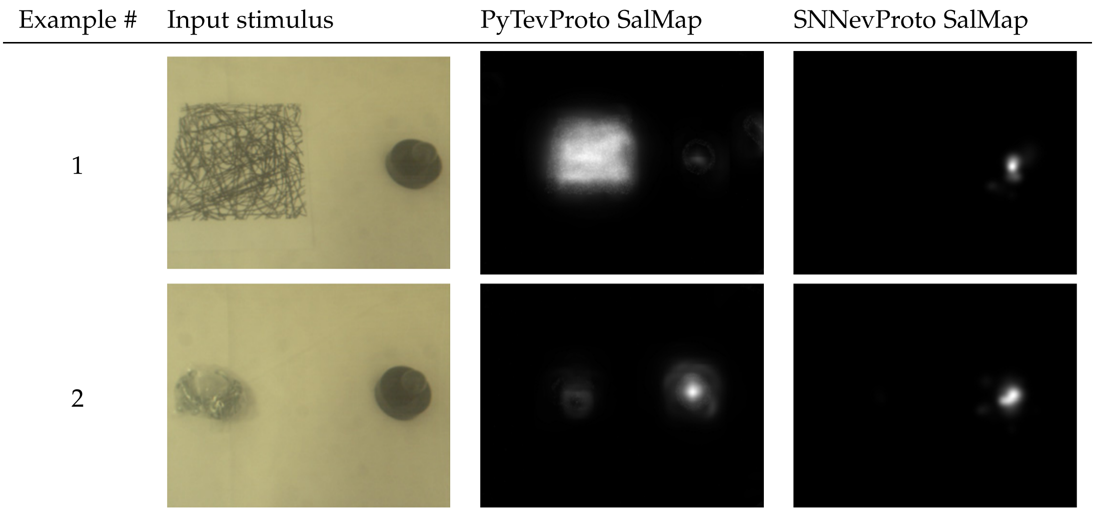

Figure 5. Qualitative comparison among the PyTevProto and the SNNevProto. From the left column to the right column: the example number, a RGB image representing the scene shown to iCub (the input stimulus), PyTevProto saliency map and SNNevProto saliency map. This table show only results from clutter experiments of the SalMapIROS dataset. The events are recorded directly from the event-driven cameras mounted on iCub's eyes. The objects and the 2D printed patterns are placed on a desk in front of the robot. The RGB input images are only for a better visualisation of the input stimulus.

spike. Values of the percentage firing thresholds of connection weights can be found in table 2. The filters are used as convolutional kernels which are tiled over the whole image.

$$w = \frac{5}{pn} \tag{3}$$

This implementation of the model is a spiking neural network where the first layer is covered with VM filters spaced with strides relative to their size. Consequently, each VM filter has its own receptive field onto the input layer. Therefore each incoming event triggers a specific pixel belonging to one filter. Each VM filter is composed of four rotationally distributed segments. As the inputs are discrete spikes generated by an event-based camera it is possible for noise and other artefacts to produce a high number of events in a small area unrelated to the visual scene. Splitting the VM filter into four sections helps to reduce the sensitivity to localised activity, aiding the filter to respond more selectively to input spikes arranged in the shape of the VM. As the strides of the convolutional kernels are relatively large, appropriate control of VM filter activity is important to reduce undesired spikes and, therefore, inaccurate saliency map generation. Each filter segment is connected to a neuron representing the entire VM filter. The refractory periods of the segment neurons and input weights to the filter neuron are balanced to require all segments to fire within a narrow temporal window to produce a spike. In addition, all spikes within the filter region that are not part of the VM kernel will have an inhibitory contribution to the combined filter neuron, effectively increasing the selectivity to the VM shape (see figure 2). The grouping cells, called proto-object neurons, pool the output of VM complementary cells that form a close contour representing proto-objects (see figure 3). The output of the convolution, and the subsequent output of the proto-objects which form the saliency map, are all represented as spikes emitted by a neuron. The filters exist in four rotation pairs with their complementary filters rotated 180°, evenly distributed from 0–135°, and in five spatial scales (104, 73, 51, 36, 25 pixels^2). Over each layer the VM filters are placed overlapped with each other. Overlap is related to stride used in the convolutional layers of neural networks. Instead of measuring how much the filter has shifted relative to the previous it measures how much it is overlapping with the previous. The overlap among the VM filters is important to define the robustness of the model. In biology, cell receptive fields are often overlapped for robustness, ensuring a response even if a cell no longer functions [44, 45]. Over time, cells overlapping have been used as a way to avoid the aliasing problem in bio-inspired models [46]. The overlapping percentage (OL) increases resolution and accuracy and it is directly linked to the number of neurons required in the implementation and, hence, its power and computational cost (see table 1). We therefore decided to use the OL as a parameter of the model to be explored. A percentage is used to ensure a uniform overlap at multiple spatial scales.

Each VM filter is connected with its mirrored one (VM in figure 3) of the opposite side creating a sub-population. All projections between sub-populations share a common weight as described in equation (3). This approach is analogous to tuning the percentage of the pre-synaptic neurons that must fire to produce a spike in the post-synaptic neuron of the next layer. A list of percentage firing thresholds for population

Table 3. Results of latency in milliseconds for different datasets of SalMapIROS. The test is done measuring the latency of two different samples for each dataset. Each row represents a dataset used to measure the latency in two separate samples. Each dataset represents static and dynamic objects placed in front of iCub (such as a paddle, a puck, calibration circles, proto-object patterns, a mouse, a cup and clutter (see figure 4).

Dataset #	First sample latency (ms)	Second sample latency (ms)
1	17	19
2	15	18
3	10	18
4	15	29
5	14	15
6	18	19
7	15	17
8	16	16
9	16	19
10	18	20
11	16	20
12	18	19
13	20	21
Average	16 ± 2.44	19.2 ± 3.37

projects can be found in table 2. This stage of the SNNevProto mimics the border ownership pyramid in [21]. A similar process to the border ownership in [21] pools the activity of mirrored VM filter orientations into a single neuron. The combined filter neuron has maximal activation at the presentation of a closed surface of the same size as the convolution filter size. Following the Gestalt principles [17] this represents detection of a proto-object. The proto-object spikes are added to a combined saliency map with their energy spread over the surrounding pixels using a 2D Gaussian distribution with standard deviation a third of the filter size in pixels. Therefore, a pooling stage mimicking the grouping pyramid is computed making the response size invariant. Values from all scales and the four pairs of rotations are pooled together to produce a combined saliency map.

3. Experiments and results

We validated the SpiNNaker implementation of the proto-object attention model, SNNevProto, by comparing its performance with the PyTorch GPU implementation, PyTevProto.

The system is further benchmarked using the ground truth 2D fixation maps of the NUS-3D dataset [34], obtained recording the eye movements of subjects observing the images of the dataset.

The characterisation compares the responses from the two models qualitatively, showing the strength and the weaknesses of each system. We then quantitatively compared the response between the SNNevProto and the PyTevProto using the latter model as the baseline. We searched for the best set of parameters, exploring different OL percentages of the VM filters on each layer and the best inhibition value.

To characterise the response, this analysis exploits the SalMapIROS dataset which contains patterns and robotic scenarios with objects and clutter in the scene. The SalMapIROS dataset is obtained recording the events coming from the event-driven cameras mounted on iCub looking at different scenes with real objects or 2D printed patterns. The robot performs small circular periodic stereotyped ocular movements to generate stimulus-dependent activity from event-driven cameras for static scenes. To estimate the selectivity to a range of sizes we used a pattern representing circles of different dimensions (see figure 4, third row). The other two patterns in figure 4 (first and second row) describe the definition of non proto-object and proto-object exploiting the design used by [21]. The proto-object is represented by the four corners facing each other forming close contours reminding of a square shape. The remaining pictures see objects of different sizes over a desk (fourth

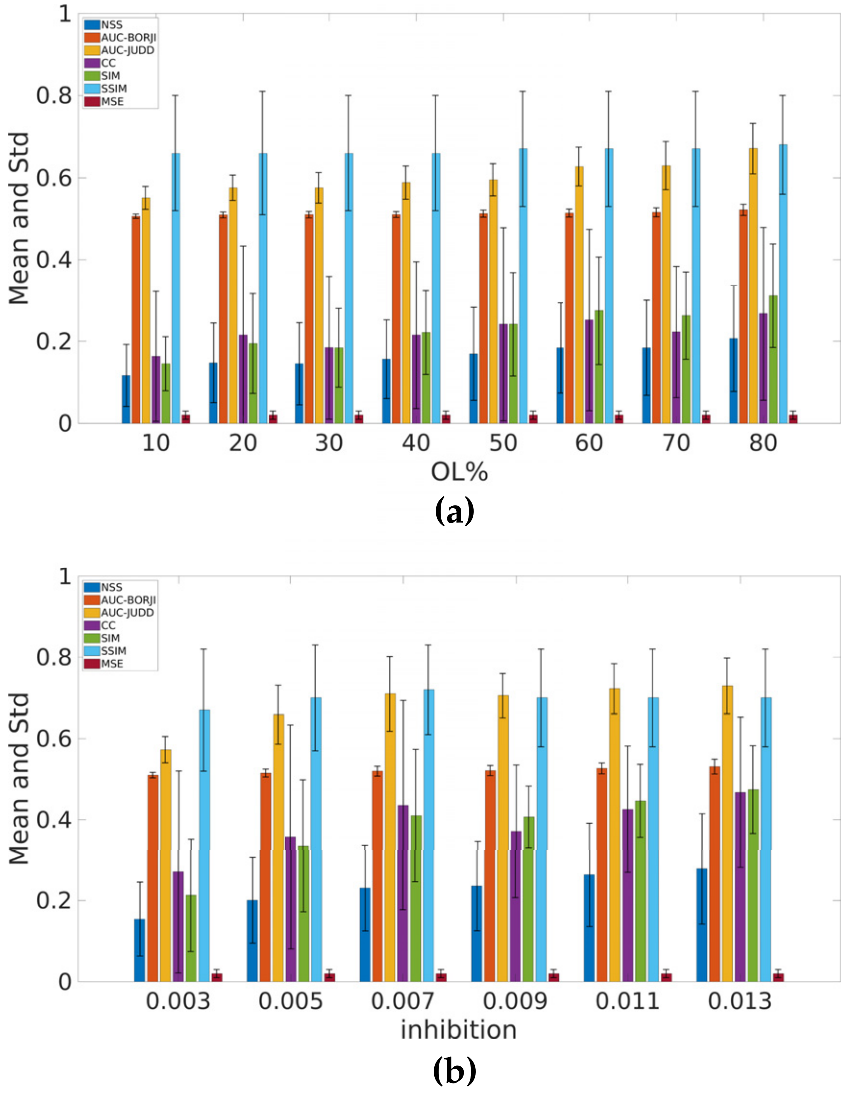

Figure 6. Comparison with different metrics evaluating the similarity between the SNNevProto saliency maps and the PyTevProto saliency maps [31] using the SalMapIROS figure 4) exploring different OL percentages (a) exploring a range of inhibitions (b) (μS conductances) with fixed OL percentage at 60%. The metrics used are: the normalized scanpath saliency (NSS), area under the ROC curve (AUC-Borji) & (AUC-Judd), Pearson's correlation coefficient (CC) and similarity (SIM) [39–42], structural similarity (SSIM) and mean square error (MSE). A higher score is better for all excluding the MSE where the lower score determines similarity.

row) to study the applicability of our system in a scenario where we want the robot to interact with items in the scene. Figure 5 shows two cases of simple clutter represented by a pattern and a bag of nails alongside with an object (a puck).

Figures 4 and 5 qualitatively show the saliency map from the two models on some samples of the SalMapIROS dataset. Overall, the response from the models is coherent and both implementations detect the objects in the scene. In figure 4 the response from the SNNevProto is less sparse and more localised over the targets which is helpful if a robot needs to locate and reach the object. The PyTevProto correctly gets rid of the clutter in figure 5 (first row) but not in figure 5 (second row). The SNNevProto instead successfully discards clutter in both cases. This results show robustness to clutter of the SNN model. This behaviour was achieved by tuning the level of inhibition. By balancing inhibition appropriately the filter can be made selective to the VM kernel shape without silencing the firing of the filter neurons. As the clutter did not contain the specific contours the VM filter is selective to, the inhibition effectively suppresses firing from the filter neurons.

The SalMapIROS dataset has been used also to obtain data related to the latency measurements. As the SpiNNaker simulation is run in real-time, latency is both walk-clock time and simulated time. The results in table 3 show the amount of time needed to obtain spikes from the proto-object neurons, which compose the saliency map, given an input. Each sample is obtained by waiting for the onset of input spikes following a quiescent period and measuring the time taken for activity to flow out of the model. This allows the delay of input spike to consequential output spike to be most clearly extracted. The average latency is 16 ms (2.44 ms

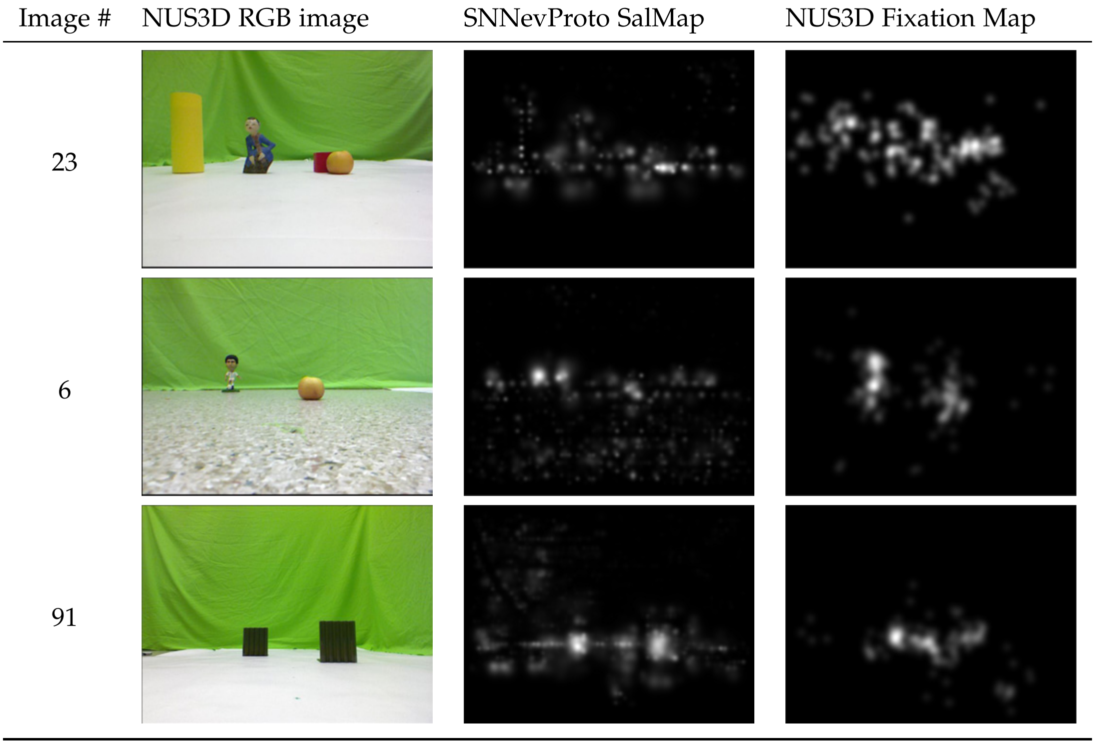

Figure 7. Representation of examples from the NUS3D (robot scenario) dataset. The three columns represent the input RGB image, the outcome from the SNNevProto and the related ground truth from the NUS3D dataset. These examples show how the model performs when the observer fixation maps focus on objects. The response from the model is with 60% OL and 0.013 inhibition.

standard deviation) and 19.2 ms (3.37 ms standard deviation) for the second set of samples, compared with the 170 ms needed in average for the PyTevProto model to obtain a saliency map of the scene.

Figure 6(a) shows the comparison between the SNNevProto and the PyTevProto saliency maps using the SalMapIROS dataset. We evaluated the similarity (SIM) among the outcomes using normalized scanpath saliency (NSS), area under the ROC curve (AUC-Borji) & (AUC-Judd), Pearson's correlation coefficient (CC) and SIM [39–42], structural similarity (SSIM) and mean square error (MSE). These metrics are computed to compare the saliency maps to the ground-truth, following standard analysis methods in the literature [39–42]. A single saliency map cannot perform well in all the metrics since they judge different aspects of the SIM between ground truth and predicted saliency map [47]. These metrics offer a way to determine how well a saliency-based model approximates human eye fixations. The properties of the chosen images for the benchmark, such as dataset bias (centre biasing, blur and scale), probabilistic input and spatial deviations, affect the result of the metrics [39]. Saliency based models can include such properties. In this work the robot needs to detect objects of different sizes to potentially interact with them. In fact, the SNNevProto only focuses on the scale of the objects rather than other properties. MSE and SSIM are metrics used in classical computer vision to explore the SIM among images. MSE estimates the error between two images and it is a global comparison, and the SSIM estimates the SIM between two images taking into account structural changes in the images.

There is not a significant difference over the OLs percentages comparing the saliency maps between the SNNevProto and the baseline (PyTevProto). Only AUC-JUDD and SIM slightly increased increasing the OL percentage. Although there is not a remarkable increment we chose 60% OL to explore the inhibition parameter (μS conductances). 60% OL represents a good compromise among the robustness of the model, ensuring enough overlap to cover the whole visual field without losing any area of the visual field, the number of SpiNNaker boards needed (see table 1) and the results obtained. Each significant increment of neurons causes an increment on the number of SpiNNaker boards required. Nevertheless, the number of neurons required does not affect the latency of the model because the pipeline remains unaltered. Figure 6(b) explores a range of different inhibitions showing again not a significant incremental or decremental trend. Only SIM and CC show a slight improvement increasing the inhibition parameter. The results exhibit a stable response exploring different parameters showing no need to create a complex network with a large number of neurons to get usable saliency maps. Overall SSIM and AUC-JUDD seem the best metrics to explain our saliency map results.

Along with the characterisation where we compared the response of our implementation with the PyTevProto, we evaluated the response from the model by benchmarking the saliency maps with the ground truth provided by the NUS-3D dataset [34]. The investigation includes the comparison between the saliency

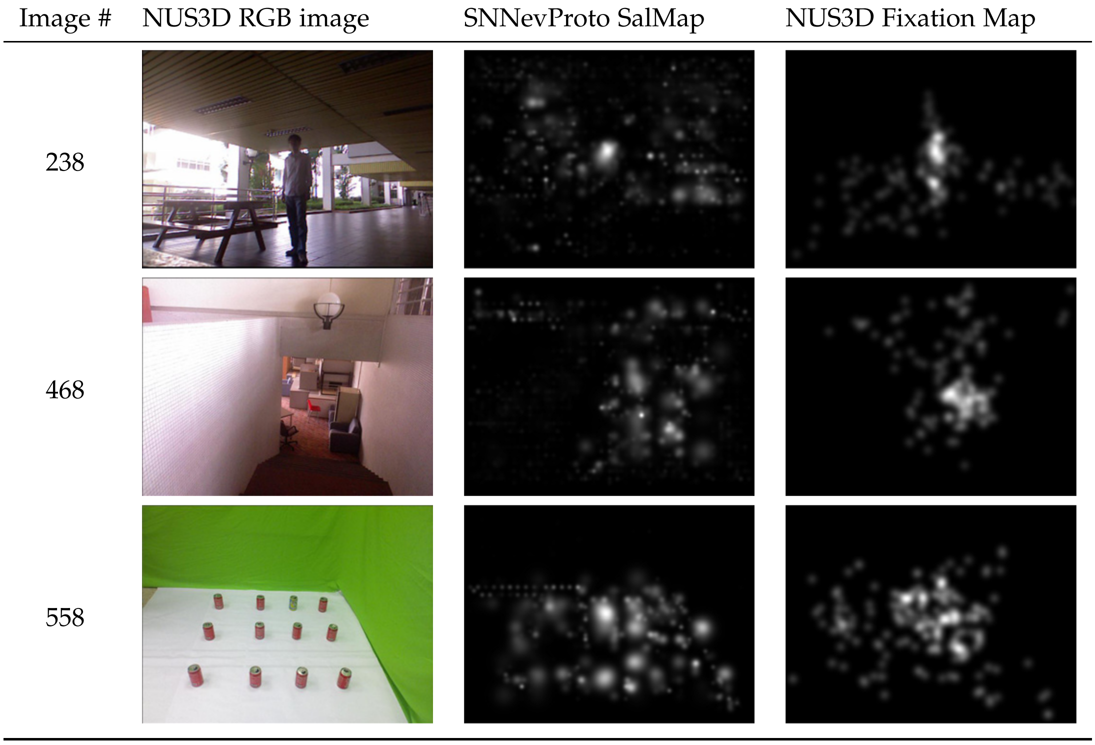

Figure 8. Representation of random chosen examples from the NUS3D (random subset) dataset. The three columns represent the input RGB image, the outcome from the SNNevProto and the related ground truth from the NUS3D dataset. These examples show how the model performs when the observer fixation maps are sparse and unclear. The response from the model is with 60% OL and 0.013 inhibition.

maps generated by the SNNevProto and the fixation maps qualitatively and quantitatively evaluating the SIM between the two maps. The 2D fixation maps of the NUS-3D were collected from subjects looking at images while recording eye movements. The ground truth obtained recording the response from the subjects includes different mechanisms of bottom–up and top–down processings, increasing the complexity of the observers' fixations. The observer response does not exclusively derive from a data-driven process but also a task-driven mechanism driving the gaze towards a particular region of the scene. Attention is a complex interplay between these two mechanisms combining bottom–up and top–down mechanisms to perceive the surrounding [10]. The model we propose is a bottom–up system that does not include top–down mechanisms, but 2D fixation maps can be used to evaluate the response of our system as they represent the only ground truth we can refer to.

To use the NUS-3D dataset within the event-driven proto-object model, we used the open event camera simulator [48] shaking the images to simulate small periodic circular eye movements.

We chose two subsets of data from the dataset: one is a selection of 50 images representative of a robotic scenario (robot scenario) and the second one is a collection of 50 random images (random subset). The first subset (see figure 7) represents a simple robotic scenario where objects are placed over a surface. The second subset (see figure 8) is a random selection among all the dataset images adding complexity and variety to the scenarios.

Qualitatively, the saliency maps from the model and the fixation maps are sparse and not easily understandable at a first glance (see figures 7 and 8). Figure 7 represents a scenario where the SNNevProto saliency map and the ground truth target select the same objects as interesting. The highest response (brightest) is located around the objects in the scene. Figure 8 shows a slightly sparse response from the model compared to the fixation maps, not allowing a clear understanding of the agent's attention.

Quantitatively, figure 9 shows good results for both datasets exploring different percentages of OL. Furthermore, all the metrics do not show a significant increment changing the OL%, validating the response of the model either for simple or for complex scenarios (figures 7 and 8 table 4).

Although we do not include the complex bottom–up top–down interplay [10] in our implementation, overall the results yield a good representation of the scene for our purposes. Moreover, the metrics used to quantify the SIM do not give equal results among them. All the metrics are used in literature to explain saliency-based model performances. They compare different aspects depending on the ground truth representation and the definition of the saliency map of the model. These metrics treat differently false negative and positives,

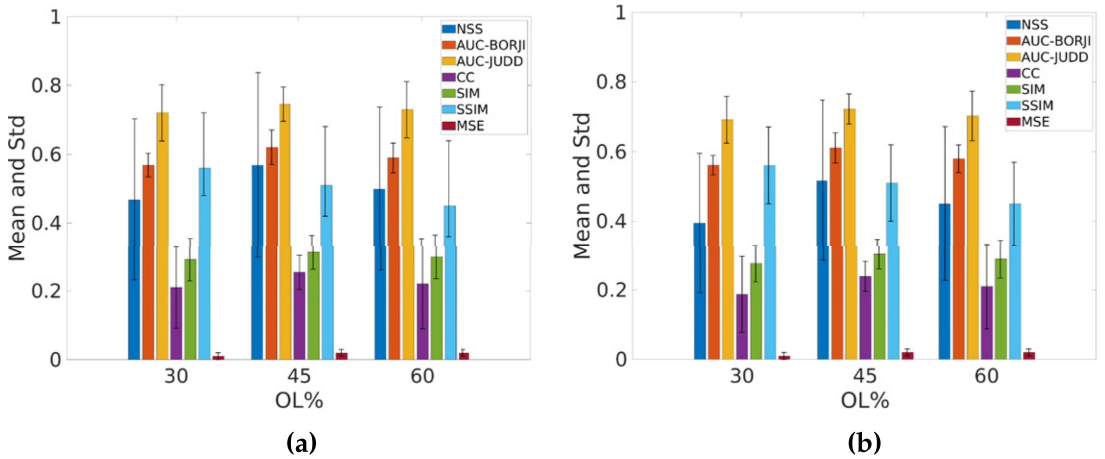

Figure 9. Comparison with different metrics evaluating the similarity of the SNNevProto saliency maps with the NUS3D fixation maps (ground truth) [34] in two different subsets (robot scenario (a) and random subset (b)) for different OL percentages. The metrics used are: the normalized scanpath saliency (NSS), area under the ROC curve (AUC-Borji) & (AUC-Judd), Pearson's correlation coefficient (CC) and similarity (SIM) [39–42], structural similarity (SSIM) and mean square error (MSE). A higher score is better for all excluding the MSE where the lower score determines similarity.

Table 4. Metrics summary. This table takes inspiration from [39].

Metrics	
NSS	CC approximation, good for saliency evaluation
Area under ROC curve (AUC)	Invariant to monotonic transformations, driven by high-valued predictions. Good for detection applications
Pearson's CC	Linear correlation between the prediction and ground truth distributions. Treats false positives and false negatives symmetrically
SIM	SIM computation between histograms, more sensitive to false negatives than false positives
SSIM	SIM among images, highly sensitive to structural changes
MSE	SIM among images, global comparison

viewing biases, spatial deviation and the pre-process of the saliency maps. We were initially interested in the location of the responses from the saliency maps rather then the value in that position, choosing the SSIM as the metric we could rely on. SSIM estimates the SSIM between two images comparing small sub-samples of the images with each other. This metric well describes our situation where we are more interested in having a response in the same location rather than having the same amount of response in terms of intensity. We further added other metrics used in literature for completeness [39]. The results seem to bare out our expectations. Overall, in our case SSIM seems a good metric to explain our saliency maps. Alongside with the SSIM, AUC-JUDD provides good results too, where each saliency pixel is treated as a classifier splitting them in 'fixation' and 'background'. This metric computes the ratio of true and false positives to the total number of fixations and saliency map pixels using a thresholded mechanism [49].

4. Conclusion

Overall the response of the spiking implementation of the event-driven attention model on SpiNNaker (SNNevProto) is coherent with the PyTevProto, showing a significant improvement in removing the clutter with respect to the baseline GPU-based implementation (PyTevProto). This can be well explained by the nature of the model. The SNN model, as a result of the inhibitory connections, is far more selective to the shape of the VM filter, than in a classic convolution using a kernel with no negative weighting. The convolution will produce activity everywhere the filter overlaps with events, enabling clutter to evoke a response in the saliency map. The advantage of the resulting higher selectivity and localised activity in the saliency map is in the possibility to improve object localisation and segmentation and, hence, the interaction of the robot with the selected object.

For the same structural reason, the response from the SNN is less sparse and focused on the location where the detected objects are placed. Two VM filter of opposite side are connected together at every scale and with different rotations. Only when they both respond there is a response from the successive layer of the SNN. Therefore, this significantly helps in generating a preciser saliency map.

Given the parallel structure of SpiNNaker, increasing the number of neurons does not affect the latency performance. For this reason, we tested the model for increasing the OL percentage, and therefore increasing the density of the convolutions. This strategy appears to provide little benefit to model performance and requires the use of an additional number of SpiNNaker boards. Results for low values of OL percentage, equivalent to a large stride in CNNs, produce a similarly reasonable representation of the visual scene compared to high values, with significantly reduced network size. This displays the feasibility of fitting the SNNevProto model on a single SpiNNaker board and having it work in tandem with the iCub humanoid robot.

The SNN implementation provides a saliency map of the scene in around 17.5 ms. In comparison with the PyTevProto (120 ms), these results are a significant improvement, that enables the system to run online in dynamic environments, where the saliency map can be used to drive the gaze and actions of the robot in real-time. To this aim, the SNN implementation on SpiNNaker could easily include winner-take-all competition and inhibition of return [9] to dynamically select the location of the next saccade of the robot. Additionally, the saliency map allows the system to focus its attention towards a specific target, devoting computational resources to perform other tasks, such as object recognition, only in the area where they are needed.

Finally, attention and gaze of robots are extremely important in the interaction with humans [50], we therefore questioned how close the saliency map (used as proxy for the robot’s fixational eye movements) was close to humans’. We validated and characterised the system, but the quantitative results of the benchmark do not capture the true merit of the model. Quantitatively, the SIM among the benchmark results (robot scenario and random subset datasets) suggests another question; how do we define the complexity of a scenario? And which aspects should we take into consideration for attention? These results proved us that the random subset does not produce lower results, hence, it may not contain complex scenarios as we expected. Each metric captures a specific aspect of the saliency maps, our analysis is instrumental to give a quantitative comparison but mostly to study the effects of the different parameters on the model performance. Moreover, most of the metrics present a high variance due to the mismatch between the SNNevProto saliency maps and the ground truth. This should be investigated in depth creating several subsets from the 600 images of the NUS3D dataset investigating the responses variability. As expected, a pure bottom–up neuromorphic attention system taking into consideration only the intensity as a feature to determine the saliency map only partially predicts the fixational eye movements of humans. To this aim, the model can be enriched with other channels (such as motion, depth, texture, etc) and with top–down processing to focus the attention towards a specific task.

The model could benefit from the leveraging of learning dynamics in the fine tuning of network parameters. This could allow the model to adapt itself to particular data sets and reach a higher level of performance. This may improve the inference of the model given appropriate training and data as compared to handcrafted parameter selection.

Moreover, the spatial integration [51] and the lateral inhibition [52] could enrich the model following a detailed bioinspired pipeline and further reducing the amount of data to be processed. Finally, further experiments could be done emphasising the clutter removal capabilities exploring the potentiality of the model.

Acknowledgments

The development of SpiNNaker was supported by EPSRC Grants EP/D07908X/1 and EP/G015740/1 and ongoing software development is supported by the EU Human Brain Project (H2020 945539). This project has been possible thanks to the HiPEAC European network grants 2019. We also would like to show our gratitude to Jay Perrett for sharing his accurate review and support during the implementation of this work.

Data availability statement

The data that support the findings of this study are available upon reasonable request from the authors.

ORCID iDs

Giulia D’Angelo https://orcid.org/0000-0001-5529-7284
Steve Furber https://orcid.org/0000-0002-6524-3367

References

[1] Liu J, Xiao Y, Hao Q and Ghaboosi K 2009 Bio-inspired visual attention in agile sensing for target detection *Int. J. Sensor Netw.* **5** 98–111

[2] Tsotsos J and Rothenstein A 2011 Computational models of visual attention *Scholarpedia* **6** 6201

[3] Metta G, Sandini G, Vernon D, Natale L and Nori F 2008 The icub humanoid robot: an open platform for research in embodied cognition *Proc. 8th Workshop on Performance Metrics for Intelligent Systems* pp 50–6

[4] Mnih V, Heess N, Graves A and Kavukcuoglu K 2014 Recurrent models of visual attention *Advances in Neural Information Processing Systems* pp 2204–12

[5] Min X, Zhai G, Gu K and Yang X 2016 Fixation prediction through multimodal analysis *ACM Trans. Multimed. Comput. Commun. Appl.* **13** 1–23

[6] Minut S and Mahadevan S 2001 A reinforcement learning model of selective visual attention *Proc. 5th Int. Conf. Autonomous Agents* pp 457–64

[7] Rutishauser U, Walther D, Koch C and Perona P 2004 Is bottom-up attention useful for object recognition? *Proc. 2004 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)* vol 2 (IEEE) p 2

[8] Yarbus A L 2013 *Eye Movements and Vision* (Berlin: Springer)

[9] Itti L and Koch C 2001 Computational modelling of visual attention *Nat. Rev. Neurosci.* **2** 194–203

[10] Wykowska A and Schubö A 2010 On the temporal relation of top–down and bottom–up mechanisms during guidance of attention *J. Cognit. Neurosci.* **22** 640–54

[11] Eriksen C W and St. James J D 1986 Visual attention within and around the field of focal attention: a zoom lens model *Percept. Psychophys.* **40** 225–40

[12] Posner M I 1980 Orienting of attention *Q. J. Exp. Psychol.* **32** 3–25

[13] Treisman A M and Gelade G 1980 A feature-integration theory of attention *Cogn. Psychol.* **12** 97–136

[14] Koch C and Ullman S 1985 Shifts in selective visual attention: towards the underlying neural circuitry *Hum. Neurobiol.* **4** 219–27

[15] Mangun G R 1995 Neural mechanisms of visual selective attention *Psychophysiology* **32** 4–18

[16] Walther D and Koch C 2006 Modeling attention to salient proto-objects *Neural Netw.* **19** 1395–407

[17] Köhler W 1967 Gestalt psychology *Psychol. Forsch.* **31** XVIII–XXX

[18] Koch K, McLean J, Berry M, Sterling P, Balasubramanian V and Freed M A 2004 Efficiency of information transmission by retinal ganglion cells *Curr. Biol.* **14** 1523–30

[19] Strong S P, Koberle R, de Ruyter van Steveninck R R and Bialek W 1998 Entropy and information in neural spike trains *Phys. Rev. Lett.* **80** 197–200

[20] Lucas P 2007 Attention in cognitive systems. Theories and systems from an interdisciplinary viewpoint *4th Int. Workshop on Attention in Cognitive Systems (WAPCV 2007) Revised Selected Papers* (Hyderabad, India January 8 2007) vol 4840 (Springer)

[21] Russell A F, Mihalaş S, von der Heydt R, Niebur E and Etienne-Cummings R 2014 A model of proto-object based saliency *Vis. Res.* **94** 1–15

[22] Molin J L, Russell A F, Mihalas S, Niebur E and Etienne-Cummings R 2013 2013 Proto-object based visual saliency model with a motion-sensitive channel *IEEE Biomedical Circuits and Systems Conf. (BioCAS)* (IEEE) pp 25–8

[23] Hu B, Kane-Jackson R and Niebur E 2016 A proto-object based saliency model in three-dimensional space *Vis. Res.* **119** 42–9

[24] Uejima T, Niebur E and Etienne-Cummings R 2018 2018 Proto-object based saliency model with second-order texture feature *IEEE Biomedical Circuits and Systems Conf. (BioCAS)* (IEEE) pp 1–4

[25] Glover A and Bartolozzi C 2016 2016 Event-driven ball detection and gaze fixation in clutter *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)* (IEEE) pp 2203–8

[26] Rebecq H, Horstschäfer T, Gallego G and Scaramuzza D 2016 EVO: a geometric approach to event-based 6-DOF parallel tracking and mapping in real time *IEEE Robot. Autom. Lett.* **2** 593–600

[27] Adams S V, Rast A D, Patterson C, Galluppi F, Brohan K, Pérez-Carrasco J-A, Wennekers T, Furber S and Cangelosi A 2014 Towards real-world neurorobotics: integrated neuromorphic visual attention *Int. Conf. Neural Information Processing* (Springer) pp 563–70

[28] Rea F, Metta G and Bartolozzi C 2013 Event-driven visual attention for the humanoid robot iCub *Front. Neurosci.* **7** 234

[29] Posch C, Matolin D and Wohlgenannt R 2011 A QVGA 143 dB dynamic range frame-free PWM image sensor with lossless pixel-level video compression and time-domain CDS *IEEE J. Solid-State Circuits* **46** 259–75

[30] Bartolozzi C, Rea F, Clercq C, Fasnacht D B, Indiveri G, Hofstätter M and Metta G 2011 Embedded neuromorphic vision for humanoid robots *CVPR 2011 Workshops* (IEEE) pp 129–35

[31] Iacono M, D’Angelo G, Glover A, Tikhanoff V, Niebur E and Bartolozzi C 2019 Proto-object based saliency for event-driven cameras *IROS* pp 805–12

[32] Furber S and Bogdan P 2020 *SpiNNaker-A Spiking Neural Network Architecture* (Norwell, MA: Now Publishers)

[33] Camunas-Mesa L, Zamarreño-Ramos C, Linares-Barranco A, Acosta-Jimenez A J, Serrano-Gotarredona T and Linares-Barranco B 2011 An event-driven multi-kernel convolution processor module for event-driven vision sensors *IEEE J. Solid-State Circuits* **47** 504–17

[34] Lang C, Nguyen T V, Katti H, Yadati K, Kankanhalli M and Yan S 2012 Depth matters: influence of depth cues on visual saliency *European Conf. Computer Vision* (Springer) pp 101–15

[35] Burkhardt D A and Fahey P K 1998 Contrast enhancement and distributed encoding by bipolar cells in the retina *J. Neurophysiol.* **80** 1070–81

[36] Hubel D H and Wiesel T N 1962 Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex *J. Physiol.* **160** 106–54

[37] Kulikowski J J, Marčelja S and Bishop P O 1982 Theory of spatial position and spatial frequency relations in the receptive fields of simple cells in the visual cortex *Biol. Cybern.* **43** 187–98

[38] Zhou H, Friedman H S and von der Heydt R 2000 Coding of border ownership in monkey visual cortex *J. Neurosci.* **20** 6594–611

[39] Bylinskii Z, Judd T, Oliva A, Torralba A and Durand F 2019 What do different evaluation metrics tell us about saliency models? *IEEE Trans. Pattern Anal. Mach. Intell.* **41** 740–57

[40] Judd T, Durand F and Torralba A 2012 *A benchmark of computational models of saliency to predict human fixations MIT-CSAIL-TR-2012-001* MIT Technical Report http://hdl.handle.net/1721.1/68590

[41] Borji A, Sihite D N and Itti L 2013 Quantitative analysis of human-model agreement in visual saliency modeling: a comparative study *IEEE Trans. Image Process.* **22** 55–69

[42] Borji A and Itti L 2015 CAT2000: a large scale fixation dataset for boosting saliency research *CVPR 2015 Workshop on ‘Future of Datasets’*

[43] Shapley R and Hugh Perry V 1986 Cat and monkey retinal ganglion cells and their visual functional roles *Trends Neurosci.* **9** 229–35

[44] Sonoda T, Okabe Y and Schmidt T M 2020 Overlapping morphological and functional properties between M4 and M5 intrinsically photosensitive retinal ganglion cells *J. Comp. Neurol.* **528** 1028–40

[45] Fischer B 1973 Overlap of receptive field centers and representation of the visual field in the cat’s optic tract *Vis. Res.* **13** 2113–20

[46] Chessa M, Maiello G, Bex P J and Solari F 2016 A space-variant model for motion interpretation across the visual field *J. Vis.* **16** 12

[47] Kummerer M, Wallis T S A and Bethge M 2018 Saliency benchmarking made easy: separating models, maps and metrics *Proc. European Conf. Computer Vision (ECCV)* pp 770–87

[48] Rebecq H, Gehrig D and Scaramuzza D 2018 ESIM: an open event camera simulator *Conf. Robotics Learning (CoRL)*

[49] Riche N, Duvinage M, Mancas M, Gosselin B and Dutoit T 2013 Saliency and human fixations: state-of-the-art and study of comparison metrics *Proc. IEEE Int. Conf. Computer Vision* pp 1153–60

[50] Willemse C and Wykowska A 2019 In natural interaction with embodied robots, we prefer it when they follow our gaze: a gaze-contingent mobile eyetracking study *Phil. Trans. R. Soc.* B **374** 20180036

[51] D’Angelo G, Janotte E, Schoepe T, O’Keeffe J, Milde M B, Chicca E and Bartolozzi C 2020 Event-based eccentric motion detection exploiting time difference encoding *Front. Neurosci.* **14** 451

[52] Delbruck T, Li C, Graca R and Mcreynolds B 2022 Utility and feasibility of a center surround event camera (arXiv:2202.13076)