Bio-inspired Ganglion Cell Models for Detecting Horizontal and Vertical Movements

Pedro Machado, Andreas Oikonomou, Georgina Cosma, T.M. McGinnity
Computational Neurosciences and Cognitive Robotics Laboratory
School of Science and Technology
Nottingham Trent University
Nottingham, United Kingdom
{pedro.baptistamachado,andreas.oikonomou,georgina.cosma,martin.mcginnity}@ntu.ac.uk

***Abstract*—The retina performs the earlier stages of image processing in living beings and is composed of six different groups of cells, namely, the rods, cones, horizontal, bipolar, amacrine and ganglion cells. Each of those group of cells can be sub-divided into other types of cells that vary in shape, size, connectivity and functionality. Each cell is responsible for performing specific tasks in these early stages of biological image processing. Some of those cells are sensitive to horizontal and vertical movements. This paper proposes a multi-hierarchical spiking neural network architecture for detecting horizontal and vertical movements using a custom dataset which was generated in laboratory settings. The proposed architecture was designed to reflect the connectivity, behaviour and the number of layers found in the majority of vertebrates retinas, including humans. The architecture was trained using 2303 images and tested using 816 images. Simulation results revealed that each cell model is sensitive to vertical and horizontal movements with a detection error of 6.75 percent.**

***Index Terms*—SNN, retinal cells, bio-inspired retinal cells, movement sensitive cells, ReSuMe**

I. Introduction

The retina, which is considered an extension of the brain, has been widely studied since Cajal (1892) [1] and six different groups of cells have been identified in all vertebrates retinas, namely, rods and cones, bipolar, horizontal, amacrine and ganglion cells. From this general grouping of cells, 50 distinct types of cells have been identified [2]. The cell types differ widely in shape, size and connectivity. Recently, Gollish and Meinster [3] described a set of visual tasks relevant to all species. Object motion detection, light sensitive and looming detection are three of the visual tasks that are observed in all vertebrate retinas. These three types of cells are called object-motion-sensitive (OMS), looming and light sensitivity ganglion cells (GCs). The OMS GCs actively respond when a local patch on a receptive field centre moves with a trajectory different from the background, the light sensitive GCs respond to light intensity variation and the looming GCs respond to approach and recede motions [3]. More complex cells have also been identified including fast response and predictive GCs. The fast response cells respond to fast motion variations while the predictive cells trigger automatic responses to specific stimuli that were previously learnt [3].

Wu et al. prosed the use of hierarchical spiking networks for processing visual stimuli [4, 5, 6, 7, 8, 9]. In these works, *Wu et al.* describes different multi-hierarchical architectures for performing image segmentation and extracting different features (including straight lines, blobs and colour features). *Kerr et al.* [10] proposed a 4 layer hierarchical neural network for extraction complex features from natural images . *Kerr et al.* and the *Wu et al.* [4, 5, 6, 7, 8, 9] share the following similarities: (i) the use of leaky-integrate-and-fire (LIF) neuron models (see Section II-B), organized in an hierarchical network; (ii) the neural networks are composed of 3 or more layers which are interconnected via excitatory and/or inhibitory synapses, forming receptive fields; (iii) inclusion of final layer implementing spike time dependent plasticity (STDP). Neither study addressed emulation of the behaviour of OMS GCs.

Zylberberq et al. [11] used a biologically inspired sparse coding model using spiking neural networks for emulating the type of responses that are found in cortical neurons in the primate visual cortex (also known as Gabor functions). *Zylberberg et al.* use a population of inhibitory neurons and a second population of excitatory neurons. The inhibitory neurons are used for producing lateral inhibition which is required to generate the same patterns generated by Gabor functions. They also introduce a new unsupervised learning technique, an adaptation of Oja's learning rule [12] for training the weights of both populations of neurons. The work reported in [11] was extended by *Tavnaei et al.* [13] who proposed a biological-inspired convolutional spiking neural network (CSNN) composed of a convolution layer, followed by a pooling layer and a fully connected layer (feature discovery). In classification tasks on the MNIST digit dataset[1], the proposed architecture showed an accuracy of 98% for clean (without any additive noise) images. None of the architectures mentioned so far are able to detect movement. However, *Rueckert et al.* [14] proposed a recurrent spiking network (RSNN) for planning tasks which detect movements. The proposed architecture is composed of 2 layers of LIF neurons, one for saving the current state and another for keeping the context. The RSNN is inspired by hippocampal neurons found in rats; however, such recurrence is not found in vertebrate retinas. Neither the

[1]The MNIST database of handwritten digits which is widely used to measuring the accuracy of machine learning and artificial intelligence algorithms. http://yann.lecun.com/exdb/mnist/, last accessed on the 27/02/2018.

CSNN nor the RSNN mimic the functionalities of the retina described in [3, 15, 16].

The paper structure is as follows. Related works are presented in section II, the proposed multi-hierarchical SNN is described in Section III, the simulation methodology is described in section IV, the simulation results are presented in Section V and analysis and future work are discussed in Section VI.

II. Related Works

A. Biological Architecture of the Retina

Vertebrates retinas vary in type, shape, size, connectivity and number of cell types, but all are characterized by having 6 layers composed of cones and rods, bipolar, horizontal, amacrine and ganglion cells (Figure 1).

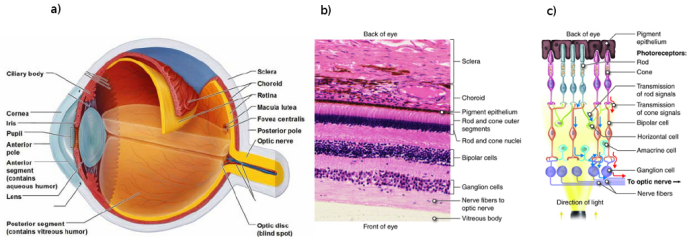

Fig. 1: (a) a cross section of the eye where the retina is the thin layer (about 1mm of thickness) of tissue located at the back of the eye ball and connected to the optic nerve; (b) layers of the retina in stained tissue where each type of retinal cells is shown (b) and (c) a drawing showing the interconnections between each type of retinal cell and the optic nerve[2].

Retinal rod and cone cells, also known as photoreceptors convert light into electrical signals that change the electrical potential of a cell's membrane allowing it to communicate information to other connected cells. The human retina consists of 120 million rod cells and 6 million cone cells. The input layer in the proposed architecture mimics those cells. Retinal bipolar cells in human retinas perform two sets of functions. They synapse either with rods or cones and accept synapses from horizontal cells. They transmit signals from the photoreceptors or the horizontal cells and pass them to ganglion cells either directly or indirectly (via amacrine cells). Communication is via graded potentials, which is unlike many other cell types which communicate via action potentials. Graded potentials describe a continuous change in membrane potential that can be either large or small and last either for short or long durations, unlike action potentials that have fixed amplitudes and timings[15, 16].

Amacrine cells are mostly inhibitory in operation and interact with ganglion cells. They are connected laterally to ganglion cells and affect the output of bipolar cells. There are at least 33 known types and each type releases one or several neurotransmitters. Among other functions they are responsible for detecting or contributing to the detection of directional motion. Retinal ganglion cells receive information from bipolar and amacrine cells and transmit action potentials to a number of regions in the brain including the thalamus, hypothalamus and mesencephalon where images are thought to be formed and classified [17, 18]. Therefore, the early stages of image pre-processing are done in an efficient way in the retina meaning that robust models of such cells will result in powerful and efficient computer vision algorithms.

B. Spiking Neural Networks

Spiking Neural Networks are used in this work because, unlike classical neurons, spiking neurons are bio-inspired and their dynamics seek to be equivalent to those observed in real neurons [19]. The Leaky-Integrate-and-Fire (LIF) neuron model was chosen as the LIF model has a good balance between processing cost and biological compatibility dynamics. More complex models are available (*e.g.* Izhkivich [20]), but they also increase the computation time and are not the best option for implementing in dedicated hardware (*e.g.* field-programmable gate arrays) which is one of the goals this research.

A RLC (Resistance, Inductance, Capacitance) electronic circuit can be used for emulating the dynamics of a LIF neuron as shown in picture Figure 2.

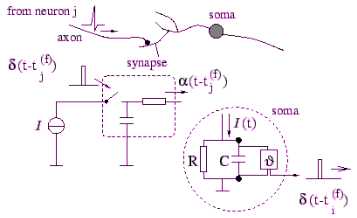

Fig. 2: Schematic diagram of the leaky-integrate-and-fire neuron model. The base circuit is the module inside the dashed circle on the right-hand side. A current I(t) charges the RC circuit. If the voltage u(t) across the capacitance reaches the threshold v then is a spike generated and u(t) is set to the reset voltage during a refractory period [19].

Therefore, the LIF neuron model action potential is governed by equation 1.

$$\frac{\tau \delta u}{\delta t} = -u(t) + RI(t) \tag{1}$$

where $\tau = RC$ is the time constant, R the membrane resistance, C the membrane capacitance, u(t) the membrane voltage at a given time t and I(t) is the current at time t.

III. Proposed Multi-Hierarchical Spiking Neural Networks for Detecting Horizontal and Vertical Movements

In this paper, we propose a four-layer architecture for mimicking the object movement sensitive ganglion cells which was described in [3]. The input layer converts the pixel graded values (0.0 up to 1.0) to spike events (where values above 0.85 are considered a spike event); layer 1 detects local edges,

layer 2 extracts movement features, layer 3 extracts movement features and layer 4 detects types of movement. The proposed architecture is shown in Figure 3.

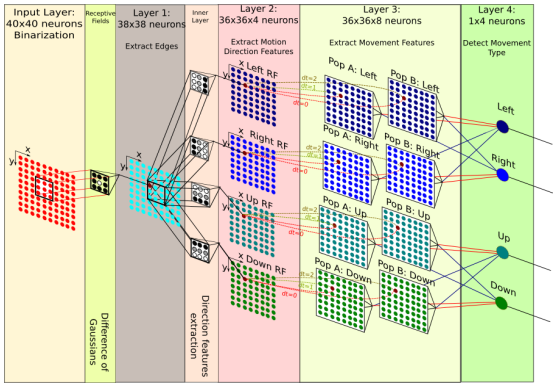

Fig. 3: Multi-hierarchical spiking neural network with (i) 40×40 image input followed by the four processing layers. Layer 1: Edge detection layer, Layer 2: Direction features extraction, Layer 3: movement extraction features and Layer 4: movement sensitive ganglion cells.

A. Input layer: Binarisation via conversion from pixel grade values to spike events

The input layer converts pixel graded values to spike events. Values equal or above 0.85 are considered a spike events similar to the functionality of rods [15]. A 1:1 connectivity is used between each pixel and the neurons in the input layer.

Figure 4 shows an example where three sequential image frames (from a synthetic dataset with objects performing horizontal and vertical movements) of 40×40 pixels are presented to the Input Layer, then processed by the neurons in Layers 1 to 4. The results are represented as vertical spikes in the output synapse coming from the right-to-left movement sensitive ganglion cell.

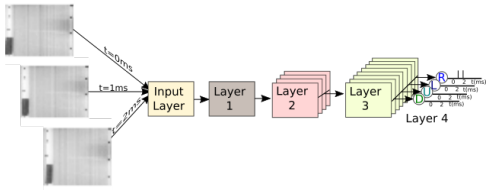

Fig. 4: Three image frames being processed by the proposed architecture. The images are presented to each layer in sequence (layer 1, 2, and 3) and finally the movement is detected in Layer 4 by the right-to-left (R), left-to-right (L), Up (U) and Down (D) ganglion cells.

In Figure 4 a black cylinder object is shown moving from the left to the right is shown. Each image is input to the proposed architecture for a period of 1ms.

B. Layer 1: Edge detection

The aim of layer 1 is to detect edges. 40×40 pixels images are exposed to Layer 1 for a period of 1 simulation time step. The neurons in Layer 1 receive spike events from a 3×3 patch where the central neuron is connected via an excitatory synapse (weight greater than 0) and the 8 neighbouring neurons are connected via inhibitory synapses (weight lower than 0), originating a receptive field (RF). The RF in the visual system comprises a 2D region in a specific visual space and may have different sizes and shapes. The latter are represented in Figure 3 by windows composed of 8 black circumferences (that act as inhibitory synapses) and a white central circumference (that acts as an excitatory synapse). The RFs have a stride of 1 for retaining the spatial information that is required for performing the edge detection of a RF weights distribution is given by the Difference of Gaussians (DoG) function. The DoG function in Layer is used for performing the edge detection and it is governed by equation 2.

$$DoG(x,y) = \frac{1}{2\pi\sigma_s^2} e^{-\frac{x^2+y^2}{2\sigma_s^2}} - \frac{1}{2\pi\sigma_c^2} e^{-\frac{x^2+y^2}{2\sigma_c^2}} \quad (2)$$

$$\frac{\sigma_c}{\sigma_s} = 1.6 \quad (3)$$

where the parameters σ_s and σ_c are the standard deviations of the neighbouring and central pixels of the DoG filter [10]. The number of neurons per pixels is reduced by 2 columns and 2 rows (*i.e.* required number of neurons is 38×38) because of the borders conditions. Zero or negative values are produced by the DoG filter when the patch is composed of similar pixels intensities (in this case the neuron will not spike); positive values are produced when the neighbouring pixels and the central pixels have a big variation (in this case the neuron will spike).

C. Layer 2: Horizontal and vertical features extraction

The aim of layer 2 is to extract horizontal and vertical features. Layer 2 is composed of 36 rows× 36 columns×4 groups of neurons (36×36×4) where each neuron is connected via a 3×3 RF and the connection between neurons is performed via inhibitory and excitatory synapses. The organisation of the excitatory synapses between the neurons of Layer 1 and Layer 2 varies accordingly to the type of filter used. The goal of each filter type is to generate unique spike patterns that are required for detecting the movement type while retaining spatial information. The filters, in the inner layer, are represented in Figure 3 with 9 blue circles overlapping 9 neurons in Layer 1.

The neurons in Layer 2 are used to extract features related to each type of movement (left-to-right, right-to-left, up and down). Again, 3×3 windows are used to produce features maps that capture the required details for a specific movement and distinct pattern from the homologous movement.

$$L_f(x,y) = \sum_{i=0}^{x}\sum_{j=0}^{y} \frac{x}{2} - i \quad (4)$$

$$R_g(x,y) = \sum_{i=0}^{x}\sum_{j=0}^{y} \frac{-x}{2} + i \quad (5)$$

$$U_p(x,y) = \sum_{j=0}^{y} \sum_{i=0}^{x} \frac{-y}{2} + j \tag{6}$$

$$D_w(x,y) = \sum_{j=0}^{y} \sum_{i=0}^{x} \frac{y}{2} - j \tag{7}$$

where the parameters L_f, R_g, U_p and D_w represents the left-to-right, right-to-left, up and down filters weight distribution. The number of neurons per patch of layer neurons is reduced by 2 columns and 2 rows (*i.e.* required number of neurons is 36×36) per filter type because of the borders conditions. Analogous to the DoG filters, the neuron will spike when the value given by the spikes, generated by the patch of neurons in Layer 1, multiplied by the weights of the respective synapses is positive and greater or equal to the threshold.

D. Layer 3: Extraction of movement features

The aim of layer 3 is to extract movement features. The neurons in Layer 3 are connected in a one-to-one (1:1) configuration to the Layer 2 neurons. There are two neuron populations, Population A and Population B. The Population A neurons are wired via 2 synapses, the excitatory synapse has no delay and the inhibitory synapse has a delay of 1 simulation time step. The use of synapses with different delays facilitates movement features extraction [21]. Delaying the spikes propagation a specific simulation time step produces a local memory (*e.g.* given a pre-neuron that spiked at the time step t and did not spiked in time step t-1 triggers a spike event in the post neuron because the result of the difference between spikes will be positive and greater of equal to the threshold.).

Population B differs from Population A only because the inhibitory synapse is delayed by two simulation time steps as opposed to one. The two populations are used for improving the accuracy of the movement features extraction by creating a bio-inspired buffer of 3 spike patterns. Each group of neurons is composed of groups of 36×36 neurons and in the overall configuration there are a total of eight groups of neurons (four per each population). The neurons in layer 3 are able to sense movement because they are inter-connected using different propagation delays. Such delays create a local residual memory for holding past neuron's events and compare them with current events (the simulation time-step was set to 1ms). The populations are used to improve the robustness of the motion detection, because fast movements are detected by Population A, slower movements by Population B and average speed movements detected by both populations of neurons. Therefore, a movement feature is extracted if a there is a spike in the actual frame which was not detected in the previous frame (population A with a propagation delay of one) or two frames before (population B with a propagation delay of 2). Therefore, it is possible to generate different spiking patterns that are used to detect differences between spike trains. The neurons in Layer 3 will spike if changes are detected.

E. Layer 4: Detection of movement type

The aim of layer 4 is to determine specific movements based on the movement features extracted in Layer 3. Each neuron in Layer 4 receives connections from all the neurons of population A and B, of its specific type of movement (*e.g.* the left-to-right movement cell is connected via excitatory cells to the left populations A and B) and inhibitory synapses from its type of movement (*e.g.* left-to-right movement cell is connected via inhibitory cells to the right populations A and B). The reason for having these connections is because the right-to-left cell must not spike when the left-to-right cell is spiking, or vice-versa. It is possible in certain circumstances to have different types of cells spiking at the same time (*e.g.* a simultaneous left-to-right and up). However a certain cell cannot spike at the same time as it pairing cell (*e.g.* left-to-right cell cannot spike at the same time as the right-to-left cell because that would mean that a given object was moving to the left-to-right and right-to-left at the same time). In Figure 3, the inhibitory synapses are represented with dashed red lines and the inhibitory synapses with blue dashed lines, and the brown and yellow dashed boxes represents that all the neurons are connected to the movement sensitive cells.

In this layer the movement sensitive neurons are found. Overall. there are four types of neurons, two of which respond to horizontal movements and the other two to vertical movements. These neurons receive connections from the neurons of both populations, A and B, of the same type and its pairing (*e.g.* left-to-right/right-to-left). The connections from the same type of movements are excitatory while the connections from the paring type are inhibitory. The weights of these connections were trained using the ReSuMe algorithm described in [22]. ReSuMe is a supervised learning method for training SNN by imposing on the neural network the desired input-output properties for producing the desired spike pattern to a stimulus. A teacher signal (desired spike pattern) is given to a neuron n_j^d that delivers a spike train $S^d(t)$ to a synaptic connection W_{ki} between an input n_k^{in} and an output n_i^{out} neurons. The learning occurs with modification of the weights.

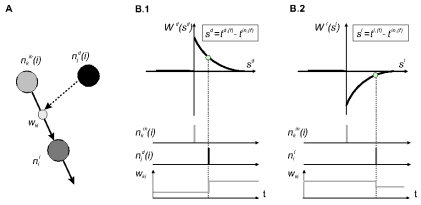

Fig. 5: ReSuMe learning: (A) Remote supervision. (B) Learning windows [22].

The ReSuMe equations are as follows:

$$W^d(s^d) = \begin{cases} +A^d e^{\left(\frac{-s^d}{\tau^d}\right)} & \text{if } s^d > 0, \\ 0 & \text{if } s^d \leq 0, \end{cases} \tag{8}$$

$$W^l(s^l) = \begin{cases} +A^l e^{\left(\frac{-s^l}{\tau^l}\right)} & \text{if } s^l > 0, \\ 0 & \text{if } s^l \leq 0, \end{cases} \tag{9}$$

where A^d, A^l and τ^d, tau^l are constants. A^d and A^l are positive in excitatory synapses and negative in inhibitory synapses. In both cases τ^d and tau^l are positive [22]. Teacher signals were used for training the Layer 4 neurons. The training was initially done to the horizontally sensitive cells and then to the vertically sensitive cells. In Figure 6 the two teacher signals used to train the synaptic weights of the horizontal cells during the simulation time window [2290, 2315]ms are shown. The period [2290, 2315]ms refers to a time period where a black cylinder object is moving from the left to right from [2290, 2303]ms and from the right to left from [2304, 2315]ms.

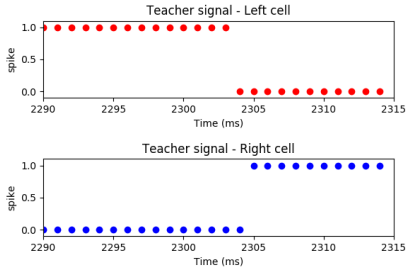

Fig. 6: Two teacher signals used to train the horizontal sensitive cells during the simulation time window [2290, 2315]ms.

IV. Simulation Methodology

This section explains how the dataset was generated and pre-processed, and the methodology that was followed for evaluating the performance of the proposed multi-hierarchical spiking neural network for detecting horizontal and vertical movements.

A. Dataset

A basic dataset was generated for testing the proposed architecture. A test scenario was prepared, where a black cylindrical object moved from left to right. Overall, one hundred repetitions were made. Every repetition has approximately 30 images (the number of images is proportional to the object speed). Overall 100 video-clips were generated. A total of 3120 images were generated from the one hundred video-clips. The dataset was augmented by performing rotations of $\frac{\pi}{2}$ rad, π rad and $\frac{2\pi}{3}$ rad. All the images were annotated. The original size of each figure is 640×480 pixels.

B. Image pre-processing

The natural images extracted from the video clips require pre-processing to reduce the dimension of the images and lower the number of neurons per layer. The image is first converted to grey scale and the number of channels reduced from 3 (red, green and blue) to one (grey scale). This is done by applying equation 10.

$$Grey = (0.299 \times R) + (0.587 \times G) + (0.114 \times B) \quad (10)$$

The second step is resizing the image to reduce the number of required neurons per layer; this was accomplished by using the OpenCV library built-in functions for Python 2.7. The images were scaled to 40×40 pixels while keeping the original aspect ratio. The third and final pre-processing steps were to compute Principal Component Analysis (PCA) followed by image whitening. Whitening transform is a conventional preprocessing step in statistical analysis for transformation random variables to orthogonality [23]. The PCA and whitening was done using the Python 2.7 Sklearn library. Figure 7 shows a sequence of 4 images extracted from one of the trials, where the black cylindrical object is moving from the left to the right.

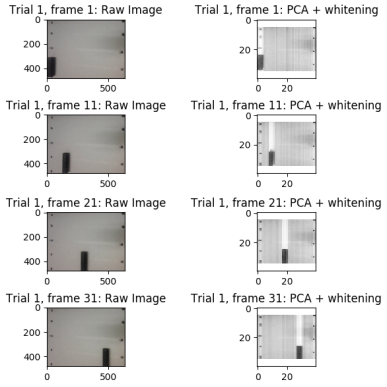

Fig. 7: Sequence of 4 raw images, where a black cylinder object is moving from the left to right (1^{st} column); image after pre-processing steps namely, conversion from RGB to greyscale, resizing, PCA and whitening (2^{nd} column).

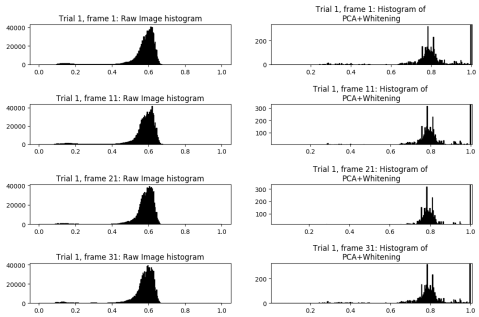

Fig. 8: Histograms of the sequence of the images shown in Figure 7. The histograms of pre-processed images are shown in the 1^{rd} column and histograms of the post-processed images are shown in the 2^{nd} column.

Figure 8 shows the histograms of the pre-/post-processed images. The histograms from the right, when compared with the ones from the left-to-right, show a dimensionality reduction as a consequence of applying the PCA, reducing the local correlation between pixels; These steps are required for speeding up the image processing.

C. Simulation Process

The simulation was performed using the Brian2 SNN simulator[3].

[3]The Brian2 SNN simulator. http://brian2.readthedocs.io/en/stable/index.html, last accessed: 31/01/2018

Step 1 - Preparing the simulation: The pre-processed images was loaded into memory and the proposed multi-hierarchical architecture was implemented in Brian2. The video-clips were randomly split into 4 datasets each comprising different training and testing data cases. In each dataset 75 video sequences (2303 images) were used for training and 25 video sequences (816 images) of the data was used for testing.

Step 2 - Setting the simulation parameters: The simulation was configured with a time step of $\tau = 1ms$. The following values of parameters are set all the neurons, $u_reset = -1mV$, $tau_refractory = 0$. The threshold for neurons in Layer 1, 2 and 3 was set to $u_th = 0.0mV$ and for layer 4 $u_th = 30.0mV$. The weights for layers 1, 2 and 3 were set constant. The weights of the neurons in Layer 4 were trained using the ReSuMe algorithm and parameters were set as follows, $\tau^d = 20ms$, $tau^l = 20ms$, $A^d = 0.01$, $A^l = 0.01$ (for excitatory synapses) and $A^d = -0.01$, $A^l = -0.01$ (for inhibitory synapses).

Step 3 - Simulation stages: The simulation can be divided into two stages, namely training and testing.

1) Training Stage: During the training stage of the simulation a teacher signal is used for training the weights of each cell. The weights of all synapses connected to all the layer 4 neurons were initialised with 1.0. The training is repeated 10000 times on the training batches while the weights are constantly updated.
2) Testing Stage: The testing mode is repeated 1 time on the testing batches while the weights are are kept constant. In the test mode, ReSuMe is not used and the outputs of the Layer 4 are scored against the expected results.

V. Simulation Results

The voltage threshold of the Layer 1 neuron was obtained by computing the average of all the images used for training. The Brian2 simulator has a structure called "TimedArray" that was used for exposing the images to the first layer neurons. Images are converted into 1D vectors and each image is a row of the "TimedArray".

The spatial information between image pixels and neurons was kept by creating a vector of indexes that was used for creating the desired connectivity between neurons. The Brian2 exposes each row (image) after the predefined interval which in this case was set to 1ms.

The spike monitors were configured for tracking all the spikes generated in each layer and were plotted at the end of the simulation. During each iteration in the training mode the synaptic weights were updated accordingly with the output generated ReSuMe training algorithm.

A. Horizontal movement test

The results obtained from the horizontal test sequences are shown in Figures 9, 10 and 11.

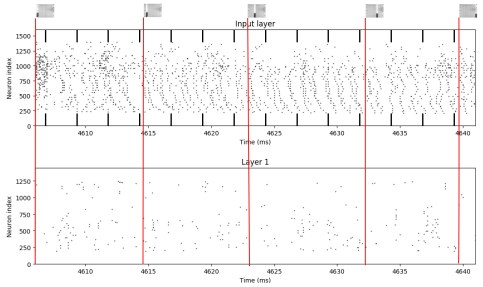

Fig. 9: Raster plot of the spiking pattern obtained during the period [4605,4640]ms (a black cylinder object was moving from the left to right) and generated by the input layer (after converting the graded values into spikes) and Layer 1 (edge extraction) neurons.

Referring to Figure 9, during the period [4605,4640]ms the image was performing a movement from the left to the right. The input layer illustrates the graded values after the conversion to spike events (values above 0.85 are considered spike events). Layer 1 shows the spike pattern generated from the edge extraction. The spiking pattern in Figure 9 was generated during in test mode (after training).

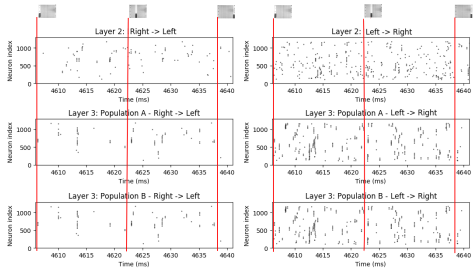

Fig. 10: Raster plot of the spikes obtained during the period [4605,4640]ms (a black cylinder object was moving from the left to right) and generated by the neurons in layers 2 and 3.

Referring to Figure 10, the spike activity from the left-to-right cells is more prominent than the right-to-left cells which is a consequence of the type of movement. The spike patterns obtained in population A (frame[t]-frame[t-1]) are very similar to the ones obtained from population B (frame[t]-frame[t-2]) which is a consequence of having slow movements which are detected by both populations of neurons.

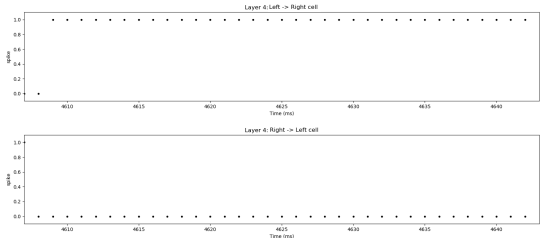

Fig. 11: Raster plot of the spikes pattern obtained during the period [4605,4640]ms (a black cylinder object was moving from the left to right) and generated by the horizontal sensitive cells.

Figure 11 shows that each cell is spiking in the correct time. The scoring algorithm compares the output spiking vector with

the expected output vector and adds a penalty of one unit every time that output and expected spikes do not match. The testing error given by the score algorithm was 7% for the horizontal cells. The error is associated with a sudden change of images sequences. This error occurs when the last image of a sequence is followed by the first image of a new sequence. This situation triggers a different spike pattern that will be compared with the previous spike patterns (Layer 3 populations A and B).

B. Vertical movement test

The results obtained from the vertical test sequences are shown in Figures 12, 13 and 14.

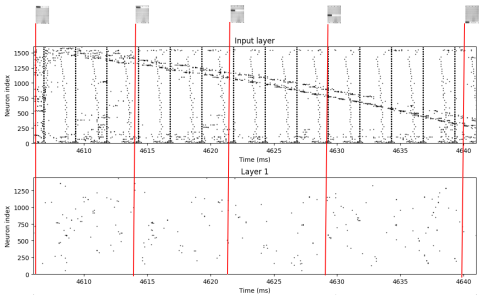

Fig. 12: Raster plot of the spikes obtained during the period [4605,4640]ms of the vertical test and generated by the input layer (after converting the graded values into spikes) and Layer 1 neurons.

Referring to Fig. 12, during the period [4605,4640]ms the object is moving down. In the input layer are shown the graded values after the conversion to spike events (values above 0.85 are considered spike events). In layer 1 the spike pattern generated from the edge extraction are shown. The spiking pattern in Figure 12 is clearly distinct from the spike pattern in Figure 9. It is possible to infer from the spike pattern of the input layer that the object is moving down.

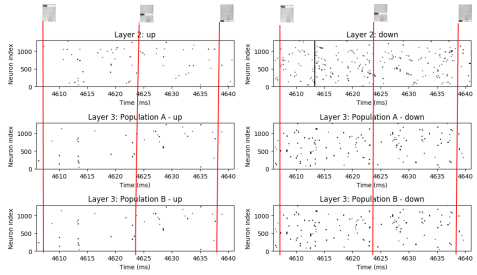

Fig. 13: Raster plot of the spiking pattern obtained during the period [4605,4640]ms of the vertical test and generated by the neurons in layers 2 and 3.

Referring to Fig. 13, the spike activity of the down cells is more prominent than the up cells which is a consequence of the type of the movement. Again, the spike patterns obtained from population A (frame[t]-frame[t-1]) are very similar to the ones obtained from population B (frame[t]-frame[t-2]) which is a consequence of having slow movements which are detected by both populations of neurons.

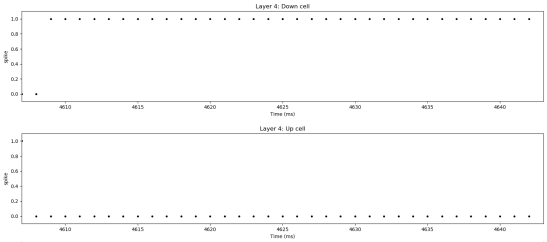

Fig. 14: Raster plot of the spiking pattern obtained during the period [4605,4640]ms of the vertical test and generated by the vertical sensitive cells.

Figure 14 shows that each cell is spiking in the correct time because the first set of images are up movements and then down movements. It is also clear from the image that on left-to-right cell spikes while the right-to-left cell does not spike and vice-versa. The scoring algorithm returned a testing error of 6.5% for the vertical cells. The errors occur when the last image of a sequence is followed by the first image of a new sequence (*i.e.* when a new sequence starts and the object moves from the end position to the start position of two sequential images).

VI. Conclusion and Future Work

Retinal ganglion cells perform the primary processing steps of natural images in an efficient way. Features such as object movement detection, looming, fast response to fast stimuli and even prediction are performed in the retina. This paper proposes a multi-hierarchical Spiking Neural Network (SNN) that was designed for simulating Object-Motion-Sensitive (OMS) Ganglion Cells (GC). The individual horizontal and vertical motion sensitive models were integrated and the outcome is an innovative multi-hierarchical bio-inspired SNN that reflects object motion detection in vertebrate retinas. Results suggest that the motion cells are sensitive to horizontal and vertical movements. The proposed architecture can be adapted for modelling many other retinal cell types (such as the texture motion[4] and/or omitted stimulus response[5] cells).

Several improvements are required in terms of increasing the robustness and performance of the horizontal and vertical motion cells. For example, the proposed architecture needs to be evaluated in different natural scenarios with different lighting conditions, different object shapes and multi-objects moving in the same scene. In addition, approaches are required for reducing the number of neurons per layer.

References

[1] R. Cajal. “The Structure of the Retina”. In: *hTorpe SA, Glickstein M, translators.* (1892).

[2] R. Masland. “The fundamental plan of the retina.” In: *Natual Neurosciences* 4 (2001), pp. 877–886.

[4]The texture motion cells are cells sensitive to light intensity variations caused when textured image patch moves across the retina [3].

[5]The omitted stimulus response cells are cells that react to periodic stimulus with a periodic response and keep generating such responses when some of those stimuli are dropped [3].

[3] Tim Gollisch and Markus Meister. "Eye Smarter than Scientists Believed: Neural Computations in Circuits of the Retina". In: *Neuron* 65.2 (2010), pp. 150–164. ISSN: 08966273. DOI: 10.1016/j.neuron.2009.12.009. arXiv: NIHMS150003.

[4] Q.X. Wu, T.M. McGinnity, L.P. Maguire, A. Belatreche, and B. Glackin. "Processing visual stimuli using hierarchical spiking neural networks". In: *Neurocomputing* 71.10-12 (June 2008), pp. 2055–2068. ISSN: 09252312. DOI: 10.1016/j.neucom.2007.10.020.

[5] QingXiang Wu, Rongtai Cai, T M McGinnity, Liam Maguire, and Jim Harkin. "Remembering Key Features of Visual Images Based on Spike Timing Dependent Plasticity of Spiking Neurons". In: *Image and Signal Processing, 2009. CISP '09. 2nd International Congress on* 12311117 (Oct. 2009), pp. 1–5. DOI: 10.1109/CISP.2009.5303978.

[6] QingXiang Wu, T Martin McGinnity, Liam Maguire, G.D. Valderrama-Gonzalez, and Jianyong Cai. "Detection of Straight Lines Using a Spiking Neural Network Model". In: *Fifth International Conference on Natural Computation* (2009), pp. 385–389. DOI: 10.1109/ICNC.2009.484.

[7] QingXiang Wu, T. M. McGinnity, Liam Maguire, G. D. Valderrama-Gonzalez, and Patrick Dempster. "Colour Image Segmentation Based on a Spiking Neural Network Model Inspired by the Visual System". In: *Advanced Intelligent Computing Theories and Applications*. 2010, pp. 49–57. ISBN: 978-3-642-14921-4, 978-3-642-14922-1. DOI: 10.1007/978-3-642-14922-1_7.

[8] Qingxiang Wu, T. Martin McGinnity, Liam Maguire, Rongtai Cai, and Meigui Chen. "Simulation of Visual Attention Using Hierarchical Spiking Neural Networks". In: *International Conference on Intelligent Computing*. 2011, pp. 26–31. DOI: 10.1007/978-3-642-24553-4_5.

[9] QingXiang Wu, T.M. M. McGinnity, Liam Maguire, Rongtai Cai, and Meigui Chen. "A visual attention model based on hierarchical spiking neural networks". In: *Neurocomputing* 116 (Sept. 2013), pp. 3–12. ISSN: 09252312. DOI: 10.1016/j.neucom.2012.01.046.

[10] Dermot Kerr, T.M. M. McGinnity, Sonya Coleman, and Marine Clogenson. "A biologically inspired spiking model of visual processing for image feature detection". In: *Neurocomputing* 158 (2015), pp. 268–280. ISSN: 18728286. DOI: 10.1016/j.neucom.2015.01.011.

[11] Joel Zylberberg, Jason Timothy Murphy, and Michael Robert DeWeese. "A sparse coding model with synaptically local plasticity and spiking neurons can account for the diverse shapes of V1 simple cell receptive fields". In: *PLoS Computational Biology* 7.10 (2011). ISSN: 1553734X. DOI: 10.1371/journal.pcbi.1002250. arXiv: 1109.2239.

[12] Erkki Oja. "Simplified neuron model as a principal component analyzer". In: *Journal of Mathematical Biology* 15.3 (1982), pp. 267–273. ISSN: 0303-6812. DOI: 10.1007/BF00275687.

[13] Amirhossein Tavanaei and Anthony S. Maida. "Bio-Inspired Spiking Convolutional Neural Network using Layer-wise Sparse Coding and STDP Learning". In: *arXiv Pre-Print* (2016). arXiv: 1611.03000. URL: http://arxiv.org/abs/1611.03000.

[14] Elmar Rueckert, David Kappel, Daniel Tanneberg, Dejan Pecevski, and Jan Peters. "Recurrent Spiking Networks Solve Planning Tasks". In: *Scientific Reports* 6.February (2016), pp. 1–10. ISSN: 20452322. DOI: 10.1038/srep21142.

[15] Helga Kolb. "How the Retina Works". In: *American Scientist* 91 (2003), pp. 28 –34. URL: http://www.americanscientist.org/issues/feature/how-the-retina-works.

[16] H. Kolb, R. Nelson, E. Fernandez, and B. Jones. *The organization of the retina and visual system*. Ed. by H. Kolb, R. Nelson, E. Fernandez, and B. Jones. Webvision, 2018. URL: http://webvision.med.utah.edu/.

[17] R. Masland. "The Neuronal Organization of the Retina." In: *Neuron* 76(2).2 (2012), pp. 266–280. DOI: 0.1016/j.neuron.2012.10.002.

[18] Jonathan B. Demb and Joshua H. Singer. "Functional Circuitry of the Retina". In: *Annual Review of Vision Science* 1.1 (2015), pp. 263–289. DOI: 10.1146/annurev-vision-082114-035334. eprint: https://doi.org/10.1146/annurev-vision-082114-035334.

[19] Wulfram. Gerstner and Werner M. Kistler. *Spiking Neuron Models*. Cambridge: Cambridge University Press, 2002, p. 496. ISBN: 9780511815706. DOI: 10.1017/CBO9780511815706. URL: http://ebooks.cambridge.org/ref/id/CBO9780511815706.

[20] E.M. Izhikevich. "Which Model to Use for Cortical Spiking Neurons?" In: *IEEE Transactions on Neural Networks* 15.5 (Sept. 2004), pp. 1063–1070. ISSN: 1045-9227. DOI: 10.1109/TNN.2004.832719. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1333071.

[21] Aboozar Taherkhani, Ammar Belatreche, Yuhua Li, and Liam P. Maguire. "DL-ReSuMe: A delay learning-based remote supervised method for spiking neurons". In: *IEEE Transactions on Neural Networks and Learning Systems* 26.12 (2015), pp. 3137–3149. ISSN: 21622388. DOI: 10.1109/TNNLS.2015.2404938.

[22] Filip Ponulak and Andrzej Kasiski. "Supervised Learning in Spiking Neural Networks with ReSuMe: Sequence Learning, Classification, and Spike Shifting". In: *Neural Computation* 22.2 (2010). PMID: 19842989, pp. 467–510. DOI: 10.1162/neco.2009.11-08-901.

[23] Agnan Kessy, Alex Lewin, and Korbinian Strimmer. "Optimal whitening and decorrelation". In: *ArXiv Pre-Print* December 2015 (2015), pp. 1–14. ISSN: 0003-1305. DOI: 10.1080/00031305.2016.1277159. arXiv: 1512.00809.